Estratégia Distribuída para Análise de Assuntos Abordados no Twitter Via Evolução de Clusters

Priscila Rocha Ferreira Rodrigues[1], Ticiana L. Coelho da Silva[2], José Maria da Silva M. Filho[1], José Antônio F. de Macêdo[1]

Universidade Federal do Ceará (UFC)

[1]Fortaleza – CE – Brasil
[2]Quixadá – CE – Brasil

priscila.rfr@gmail.com,

{ticianalc, monteiro, jose.macedo}@dc.ufc.br

Abstract. *Recent techniques have applied algorithms of clusters evolution to analyze transitions of subjects in social networks and present themselves effective in the monitoring of these. However, the high rate of data production in social networks creates the need to process an increasing amount of data. This paper proposes a more scalable strategy for analyzing the evolution of subjects in social networks, through the use of a distributed solution in the data clustering stage. The experiments were performed using data obtained from Twitter and demonstrate that the proposed solution is promising, presenting considerable gains in performance.*

Resumo. *Recentes técnicas têm aplicado algoritmos de evolução de clusters para analisar transições de assuntos em redes sociais e apresentam-se eficazes no monitoramento destes. No entanto, a elevada taxa de produção de dados nas redes sociais cria a necessidade de processamento de uma quantidade de dados cada vez maior. Este trabalho propõe uma estratégia mais escalável para análise da evolução de assuntos em redes sociais, por meio do emprego de uma solução distribuída na etapa de clustering dos dados. Os experimentos foram realizados utilizando dados obtidos do Twitter e demonstram que a solução proposta é promissora, apresentando ganhos consideráveis de desempenho.*

1. Introdução

Considerando a relevância do monitoramento de assuntos e comunidades em redes sociais ao longo do tempo, referenciados em [Kim e Han 2009], [Lee et al 2014], [Tajeuna et al 2015] e [Rodrigues et al 2016], algumas técnicas têm sido propostas para automatizar esse tipo de análise. Dentre essas técnicas, destaca-se a evolução de *clusters,* que consiste na investigação das alterações que um *cluster* pode sofrer ao longo do tempo em virtude da constante evolução e produção dos dados. O acompanhamento da repercussão de determinado evento, *feedback* sobre algum produto, elaboração de melhores estratégias de marketing e melhoria de serviços, são algumas das finalidades que podem ser alcançadas por meio do emprego desse método em dados produzidos nas redes sociais.

[Rodrigues et al 2016] demonstra a aplicabilidade da utilização da estratégia de evolução de *clusters* em bases de dados altamente dinâmicas como as de redes sociais.

Utilizando o algoritmo DBSCAN [Ester et al 1996], [Rodrigues et al 2016] aplicou *clustering* em uma base de dados composta por *tweets*, detectando os assuntos repercutidos no *dataset* analisado. Por meio do monitoramento da evolução dos *clusters*, verificou-se a dinâmica e transição dos assuntos detectados, apresentando uma visão panorâmica que permitiu compreender as motivações de tais evoluções. Através do algoritmo de evolução de *clusters* empregado em [Rodrigues et al 2016] foi possível detectar as seguintes transições: Criação, Sobrevivência, Desaparecimento, União, Divisão, Expansão e Retração de *clusters.*

Apesar dos bons resultados apresentados no trabalho supramencionado, ressalta-se que a elevada taxa de produção de conteúdo em redes sociais cria a necessidade de processamento de uma quantidade de dados cada vez maior. Logo, a utilização de algoritmos de alto custo computacional, como o DBSCAN, pode agravar ainda mais o desempenho da análise de grandes volumes de dados [Gan e Tao 2015], e este é um problema em aberto na proposta de [Rodrigues et al 2016]. Problemas que fazem uso intensivo de CPU em suas resoluções ou que processam grandes volumes de dados são essencialmente resolvidos por meio da distribuição de tarefas entre vários núcleos de processamento.

Motivados pelos benefícios que o monitoramento de assuntos numa janela temporal propicia a diversas áreas, e com o intuito de estender a proposta de [Rodrigues et al 2016], este trabalho propõe uma solução mais escalável para o monitoramento dos assuntos em redes sociais via evolução de *clusters*. Na etapa de *clustering* dos dados, é proposta uma abordagem distribuída, aplicando o algoritmo DBSCAN através do paradigma MapReduce, com o intuito de obter melhor desempenho na análise da evolução de assuntos ao longo do tempo. As observações obtidas até o momento, por meio de experimentos utilizando *tweets*, apontam que a solução proposta é adequada, apresentando uma melhoria considerável no tempo de execução quando comparada com a abordagem centralizada.

Este artigo está organizado da seguinte forma: Na Seção 2, é descrita a abordagem distribuída que está sendo proposta para acompanhamento da evolução dos assuntos nas redes sociais. Na Seção 3 é apresentada a análise dos resultados preliminares. Os trabalhos relacionados estão descritos na seção 4. Finalmente, a Seção 5 apresenta as conclusões obtidas até o momento e as direções futuras desta pesquisa.

2. Detecção Distribuída de *Clusters* e suas Evoluções

A estratégia para o monitoramento da evolução de *clusters* proposta por [Rodrigues et al 2016], consiste em 3 etapas principais: coleta e pré-processamento dos dados, *clustering* e evolução de *clusters*. Na primeira etapa, os dados são coletados da rede social *r* e em seguida, são aplicadas técnicas de processamento de linguagem natural para limpeza dos dados. Na etapa seguinte, para o emprego da técnica de *clustering*, o algoritmo DBSCAN foi implementado e aplicado utilizando como base a medida de similaridade *Fading* [Lee et al 2014]. Tal medida faz uso do atributo de tempo para o cálculo da similaridade, considerando que publicações mais próximas no tempo têm maior possibilidade de versarem sobre o mesmo assunto. O DBSCAN recebe como entrada um conjunto de dados X, o tamanho da vizinhança denominado *eps* e o número mínimo de pontos de uma vizinhança, denotado *minPts*. O algoritmo classifica os dados em *core*, *border* ou *outlier*. Um ponto é *core* se possuir uma vizinhança maior ou igual do que o estabelecido em *minPts*. É considerado *border* se pertence à vizinhança de um ponto *core*, ou seja, se a distância entre eles é menor que *eps*. Pontos *outliers* são pontos atípicos ou inconsistentes que, por não pertencerem a *cluster* nenhum, são descartados [Ester et al 1996].

Para detecção das evoluções sofridas pelos *clusters* ao longo dos dias, foi implementado e executado o algoritmo de evolução proposto por [Silva et al 2014]. Através de comparações entre os *clusters*, o algoritmo detecta criação, sobrevivência, desaparecimento, união, divisão, expansão e retração de *clusters.* Considerando que das 3 etapas da estratégia de Evolução de *Clusters* apresentadas, a etapa de *clustering* é a que apresenta uma maior necessidade de processamento, fazendo uso de um considerável custo computacional, aplicamos o paradigma MapReduce [Dean e Ghemawat 2004] na etapa de *clustering* dos dados. Tal paradigma destaca-se como uma das abordagens mais utilizadas para solucionar problemas de processamento de grandes conjuntos de dados, uma vez que foi projetado com a finalidade de tornar simples e eficiente o processamento distribuído através de um conjunto de computadores comuns. Dentre as implementações mais conhecidas do paradigma MapReduce destaca-se o framework de código aberto Hadoop [White 2009], desenvolvido pela Apache Software Foundation, e referenciado recentemente na literatura em trabalhos que têm aplicado soluções distribuídas do DBSCAN para domínios diversos [He et al 2011], [Dai et al 2012] e [Neto et al 2015]. A execução de tarefas nos nós de processamento e a tolerância a falhas são de total responsabilidade do framework, restando ao desenvolvedor apenas a programação das funções Map e Reduce, além da configuração da infraestrutura.

A função Map recebe todos os dados gerando um conjunto intermediário de dados, no formato <chave, valor>. A função Reduce é executada para cada chave intermediária gerada, com todos os conjuntos de valores intermediários associados aquela chave combinados. Em geral, a tarefa de mapeamento é usada para selecionar um conjunto de valores, e a tarefa de redução é usada para fazer a sumarização do resultado. Além disso, em processamentos distribuídos é importante que as partições de dados a serem processadas tenham tamanhos aproximadamente iguais, uma vez que o tempo total de processamento é delimitado pelo tempo que o nó com uma maior quantidade de dados leva para finalizar a computação dos dados a ele atribuídos. Logo, a base de dados a ser analisada é particionada de forma proporcional com base na quantidade de *tweets* nela contidos.

As funções Map e Reduce definidas para tornar a proposta de [Rodrigues et al 2016] escalável são explicadas a seguir:

1. **Map–** O algoritmo DBSCAN é aplicado ao conjunto de dados. Em seguida, cada *cluster* é descrito como um par <chave, valor>, onde a chave corresponde ao *tweet core* e o valor corresponde aos *tweets borders* do *cluster*.

2. **Reduce–** Essa função recebe como entrada o conjunto de pares de *clusters* gerados anteriormente e aplica a medida de similaridade *Fading* nos pontos *cores* desses *clusters*, unindo possíveis *clusters* do mesmo assunto que foram detectados em partições diferentes. Os pares de *clusters* gerados nessa etapa serão utilizados na etapa posterior pelo algoritmo que detecta as transições dos *clusters*.

Os Algoritmos 1e 2 mostram como é feita a implementação das funções Map e Reduce descritas acima, respectivamente.

Algoritmo 1: Map

```
Entrada: Conjunto de dados D,minPoints,eps
Saída: Conjunto PC de pares de clusters
1Início
2 |  DBSCAN (P, eps, minPoints)
3 |  retorna PC;
4fim
```

Algoritmo 2: Reduce

```
Entrada: Conjunto PC de pares de clusters <c, b>, eps
Saída: Conjunto PC de pares de clusters após uniões
1Início
2 |  para c1∈ PC faça
3 |       para c2∈ PC e c2 ≠ c1 faça
4 |            se SimilaridadeFading (c1,c2) ≥ eps então
5 |            |  PC <- cx<- União (c1,c2)
6 |            |  PC-> Remove (c1,c2)
7 |            fim
8 |       fim
9 |  fim
10|  retorna PC;
11fim
```

3. Análise dos Resultados Preliminares

Os dados do Twitter foram coletados entre os dias 17/05/2017 e 23/05/2017 por meio de um *crawler* que recebeu como parâmetro a palavra "Temer" e retornou *tweets* que continham essa palavra. Foram coletados 2.254.628 *tweets*, o que resultou em aproximadamente 2,4 Gb de dados. Para estratégia distribuída, os experimentos foram realizados em uma infra-estrutura de nuvem privada com 8 máquinas virtuais, 4 núcleos (vCPU), 8G de RAM e 80GB de disco cada. No experimento centralizado, apenas uma dessas máquinas foi utilizada. Os parâmetros utilizados nos algoritmos, são os mesmos empregados em [Rodrigues et al 2016]. Logo, isso corresponde a *eps*: 0.3 e *minPts*: 0.25 * (quantidade de *tweets* do snapshot). Cada snapshot corresponde a um dia de coleta.

A Figura 1 apresenta os assuntos que foram detectados e a maneira como esses assuntos repercutiram e se relacionaram ao longo dos dias. Foram detectados os mesmos *clusters* e transições para ambas as estratégias aplicadas: com *clustering* centralizado e distribuído. No dia 17/05 foi identificado um assunto que se tratava sobre a divulgação de uma gravação que o Dono da JBS, Joesley Batista, havia feito em uma conversa com Michel Temer que se tratava do pagamento pelo silêncio do ex-deputado Eduardo Cunha[1]. No dia seguinte, 18/05, esse tópico sofreu uma **divisão**, entre o assunto que se tratava das manifestações contrárias ao presidente Temer ocorridas no país[2] e o assunto que repercutia o afastamento de Aécio Neves do cargo de senador[3]. No dia 19/05 ambos os assuntos passaram a repercutir juntos (**união**) sobre um possível impeachment de Michel Temer. Em 20/05 esse assunto **expandiu,** porém nos dias subsequentes passou a sofrer pequenas **retrações**. No dia 22/05 surgiu também um pequeno *cluster* que se tratava do atentado ocorrido em Londres no *show* da cantora Ariana Grande[4]. Usuários ironizavam falando que o atentado deveria ter ocorrido no Congresso Nacional.

Apesar de nesse estudo de caso, os *clusters* gerados utilizando as abordagens centralizada e distribuída terem produzido mesmos resultados, é possível que em outros experimentos possamos obter resultados distintos. Isto ocorrerá quando determinado *tweet* classificado *border* em uma partição, poderia se tornar um *tweet core* se fosse comparado a outra partição de dados. Apesar dessa possibilidade, observa-se que a estratégia distribuída é, em média, 10,3 vezes mais rápida que a abordagem centralizada. A Figura 2 apresenta um gráfico com o tempo de processamento do *clustering* para cada *dataset* analisado, comparando os resultados entre as abordagens empregadas.

[1] https://oglobo.globo.com/brasil/dono-da-jbs-grava-temer-dando-aval-para-compra-de-silencio-de-cunha-21353935

[2] http://www.opovo.com.br/noticias/politica/2017/05/protestos-acontecem-em-cidades-apos-divulgacao-de-denuncia-contra-teme.html

[3] http://g1.globo.com/jornal-nacional/noticia/2017/05/aecio-neves-e-afastado-da-funcao-de-senador-e-deixa-presidencia-do-psdb.html

[4] http://g1.globo.com/jornal-nacional/noticia/2017/05/explosao-mata-19-pessoas-em-show-de-ariana-grande-na-inglaterra.html

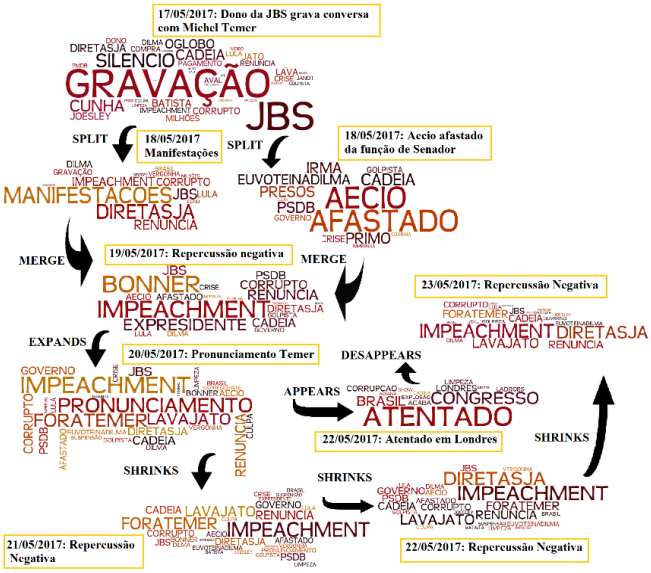

Figura 1- Gráfico de transições dos assuntos sobre "Temer" - 17/05/2017 à 23/05/2017

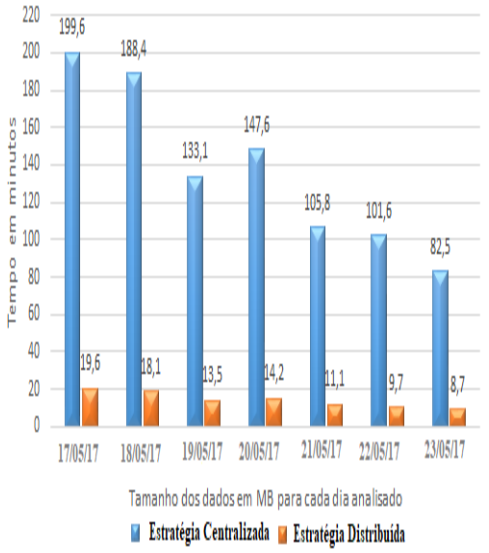

Figura 2- Tempo de execução dos algoritmos na etapa de clustering

4. Trabalhos Relacionados

Em [Tajeunaet al 2015], técnicas de evolução de *clusters* são aplicadas em dados de redes sociais com a finalidade de acompanhar a evolução de grupos de usuários ao longo do tempo. Além disso, o trabalho propõe uma medida de similaridade com o intuito de capturar eventos de transição significativos para evolução de comunidades. No entanto, além de [Tajeuna et al 2015] aplicar a estratégia de evolução de *clusters* para o contexto de comunidades, e não de assuntos como no trabalho em questão, sua abordagem é centralizada. Já [Neto et al 2015] estuda o problema de processar o algoritmo de *clustering* DBSCAN de maneira distribuída através do paradigma MapReduce. Sua estratégia propõe também uma abordagem de particionamento de dados, que usa estruturas de grade e grafo para auxiliar na divisão do espaço em regiões de baixa densidade. No entanto, [Neto et al 2015] direciona sua estratégia para o contexto de dados espaciais, o que torna sua abordagem, em muitos pontos, diferente da proposta neste trabalho, tendo em vista as características específicas de ambos os contextos. [Dai e Lin 2012] e [He et al 2011], também abordaram o problema de distribuição do DBSCAN, no entanto, diferente deste trabalho, as técnicas propostas requerem que parte dos dados sejam replicados nas partições.

Por fim, em [Rodrigues et al 2016] são apresentados resultados sobre a efetividade do uso de técnicas de evolução de *clusters* para monitorar a transição de assuntos em redes sociais. [Rodrigues et al 2016] apresentou a evolução de temas que giraram em torno da ex-presidenta Dilma Rousseff em 2015. No entanto, sua estratégia é centralizada e não aborda questões de desempenho. Além disso, os autores destacam que é importante que outros estudos de caso sejam realizados para validar a estratégia proposta. Sendo assim, o presente trabalho visa estender a pesquisa iniciada por [Rodrigues et al 2016], propondo uma abordagem distribuída para a *clustering* dos dados e realizando um novo estudo de caso em um *dataset* com *tweets* coletados em 2017.

5. Conclusões e Direções Futuras

Os resultados apresentados mostram um ganho considerável de desempenho na estratégia de evolução de *clusters*, ao aplicar uma abordagem distribuída na etapa de *clustering* dos dados. É ainda reforçado que a aplicabilidade das técnicas de evolução de *cluster* para

monitorar as transições de assuntos em redes sociais é válida e promissora. Ressalta-se que a aplicação dessas estratégias é recente no contexto de redes sociais, e a apresentação de uma abordagem distribuída para este cenário dá abertura para o surgimento de novas estratégias de distribuição de processamento, que podem tirar proveito desse modelo para abordagens mais eficientes. Posteriormente, pretende-se realizar um estudo mais profundo da estratégia distribuída em comparação com a centralizada, a fim de garantir que ambas gerem os mesmos resultados. Adiante, objetiva-se também analisar a variação dos parâmetros utilizados no DBSCAN para este contexto de *clustering* em redes sociais, além de realizar experimentos que verifiquem o desempenho da estratégia distribuída com diferentes quantidades de máquinas.

Referências

Dai, B. R., & Lin, I. C. (2012). Efficient map/reduce-based dbscan algorithm with optimized data partition. In Cloud Computing (CLOUD), 2012 IEEE 5th International Conference on (pp. 59-66). IEEE.

Dean, J., & Ghemawat, S. (2008). MapReduce: simplified data processing on large clusters. Communications of the ACM, 51(1), 107-113.

Ester, M., Kriegel, H. P., Sander, J., & Xu, X. (1996). A density-based algorithm for discovering clusters in large spatial databases with noise. In Kdd(Vol. 96, No. 34, pp. 226-231).

Gan, J., & Tao, Y. (2015). DBSCAN revisited: mis-claim, un-fixability, and approximation. In Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data (pp. 519-530). ACM.

He, Yaobin, et al. (2011) "Mr-dbscan: an efficient parallel density-based clustering algorithm using mapreduce." Parallel and Distributed Systems (ICPADS), 2011 IEEE 17th International Conference on. IEEE.

Kim, M. S., & Han, J. (2009). A particle-and-density based evolutionary clustering method for dynamic networks. Proceedings of the VLDB Endowment, 2(1), 622-633.

Lee, P., Lakshmanan, L. V., & Milios, E. E. (2014). Incremental cluster evolution tracking from highly dynamic network data. In Data Engineering (ICDE), 2014 IEEE 30th International Conference on (pp. 3-14). IEEE.

Neto, Antonio Cavalcante Araujo, et al. (2015) "G2P: A partitioning approach for processing dbscan with mapreduce." International Symposium on Web and Wireless Geographical Information Systems. Springer, Cham.

Rodrigues, Priscila R.F. et al. (2016)“Dinâmica de Temas Abordados no Twitter Via Evolução de Clusters.” Proceedings of the 31thSBBD, p. 151-157.

Silva, Ticiana L. Coelho, José AF de Macêdo, and Marco A. Casanova. (2014) "Discovering frequent mobility patterns on moving object data." Proceedings of the Third ACM SIGSPATIAL International Workshop on Mobile Geographic Information Systems.ACM.

Tajeuna, Etienne Gael, Mohamed Bouguessa, and ShengruiWang. (2015) "Tracking the evolution of community structures in time-evolving social networks." DSAA, 2015. 36678 2015.IEEE International Conference on.IEEE,.

White, T. (2012). Hadoop: The definitive guide. " O'Reilly Media, Inc.".