

Determinação da sensibilidade aos parâmetros do método Kmeans da biblioteca scikit-learn por meio de ANOVA

Paulo Ellery Alves de Oliveira (UFRN) pauloellery@hotmail.com
Pedro Osvaldo de Alencar Regis (UFRN) pedroalencarregis@hotmail.com
Arthur Arcelino de Brito (UFRN) arthurbritoe@gmail.com
Rafael de Azevedo Palhares (UFRN) rafaelpalhareseng@hotmail.com

Resumo:

Atualmente, faz-se necessário o uso de técnicas capazes de extrair informações concisas de um expressivo conjunto de dados. Um problema comumente observado nos diversos âmbitos é a classificação de objetos com base em seus atributos. Para isto, o algoritmo K-means pode ser utilizado. Este algoritmo é um famoso problema da classe NP-difícil, isto é, sua solução utiliza métodos não determinísticos, como heurísticas e metaheurísticas. Uma de suas implementações está contida na biblioteca scikit-learn da linguagem de programação Python, vale salientar que esta linguagem vem sendo cada vez mais difundida na comunidade científica. Posto o crescente interesse nesta ferramenta e dado que o algoritmo é não determinístico, o objetivo deste trabalho é determinar a influência de diferentes escolhas dos parâmetros de inicialização do método Kmeans da biblioteca *scikit-learn* da linguagem de programação Python. Para isto, aplicou-se o algoritmo ao famoso conjunto de dados Iris, que é composto por dados de três tipos de flores, formando uma tabela de 150 linhas e 5 colunas. Para calcular o quanto o algoritmo errou/acertou foi utilizado o ARI de modo que um experimento computacional que simulasse a utilização do método por um usuário que escolhesse diferentes combinações dos valores padrão dos parâmetros do método da biblioteca fosse analisado. Após a obtenção dos dados foi feita uma ANOVA. A partir dos resultados encontrados, não se tem evidência suficiente para rejeitar a hipótese nula, esta diz que uma diferente escolha dos parâmetros da biblioteca influencia no resultado do algoritmo.

Palavras chave: Python, Índice ajustado de Randi, ARI, Iris.

Determination of sensitivity to the Kmeans method parameters of the scikit-learn library using ANOVA

Abstract

Currently, it is necessary to use techniques capable of extracting concise information from an expressive set of data. A problem commonly observed in the various scopes is the classification of objects based on their attributes. For this, the K-means algorithm can be used. This algorithm is a famous problem of the NP-difficult class, that is, its solution uses non-deterministic methods, such as heuristics and metaheuristics, one of its implementations is contained in the scikit-learn library of the programming language python, language has been becoming scientific community. Given the increasing interest in this tool and given that the algorithm is non-deterministic, the purpose of this work is to determine the influence of different choices of initialization parameters of the Kmeans method of the scikit-learn library of the python programming language. For this, the algorithm was applied to the famous data set Iris, which is composed of data of three types of flowers, forming a table of 150 rows and

5 columns. In order to calculate how much the algorithm was wrong, the ARI was used so that a computational experiment that simulated the use of the method by a user that chose different combinations of the standard values of the method parameters of the library was analyzed. After obtaining the data an ANOVA was made. From the results found, there is not sufficient evidence to reject the null hypothesis, this says that a different choice of the parameters of the library influences the result of the algorithm.

Key-words: Python, Adjusted Randi Index, ARI, Iris.

1. Introdução

As pessoas vêm buscando padrões nos dados desde que a vida humana começou, e não há nada de novo nisso. Os caçadores buscam padrões no comportamento de migração de animais, os agricultores buscam padrões no crescimento das plantações, os políticos buscam padrões na opinião dos eleitores e os amantes buscam padrões nas respostas de seus parceiros, (WITTEN et al., 2016). Avanços na tecnologia de sensoriamento e armazenamento e o dramático crescimento de aplicações na internet, imagens digitais, e vigilância por vídeo criaram muitos conjuntos de dispositivos com alto volume de dados. Estima-se que o universo digital tenha usado aproximadamente 281 exabytes em 2007 (GANTZ et al., 2008).

Witten et al. (2016) diz que estamos o tempo todo sobrecarregados de dados e que sua quantidade no mundo e em nossas vidas aumentam demasiadamente. Todos podem comprovar a lacuna entre o crescimento da quantidade de dados produzida e armazenada com benefício que extraímos desta (WITTEN et al., 2016). Tem-se muitos dados sendo gerados e pouco proveito sendo absorvidos destes. No entanto, é notório o aumento do uso de técnicas para processá-los.

Na área de aprendizado de padrões é comum que se adote uma divisão entre os tipos de aprendizado, por exemplo Duda et al. (2001) divide em dois tipos (i) supervisionados (classificação) ou (ii) não supervisionados (agrupamento). De acordo com Jain (2010), o primeiro envolve apenas dados rotulados (padrões de treinamento com rótulos de categorias conhecidas, que se sabe previamente a resposta), enquanto o segundo envolve apenas dados. Já Chapelle e Scholkopf e Eds. (2009) trata de um terceiro tipo, o aprendizado semi supervisionado. Neste caso, apenas uma pequena parte do conjunto de dados que se deseja classificar é fornecida ao algoritmo de forma que este possa “aprender” a reconhecer o padrão desejado.

O objetivo do agrupamento de dados, também conhecido como análise de cluster, é que se descubra o agrupamento natural de um conjunto de padrões, pontos ou objetos (JAIN, 2010). O interesse na utilização deste tipo de técnica, segundo Witten et al. 2016, é devido a possibilidade de dar sentido aos dados, descobrir os padrões que governam o funcionamento do mundo físico e encapsula-los em teorias que possam ser usadas para prever o que acontecerá em novas situações. Um dos métodos que lida com a clusterização de dados é o K-means.

Dentro de um espectro muito amplo de partes de códigos, pacotes, bibliotecas, e até programas especialistas neste tipo de análise foi escolhida a biblioteca Scikit-learn Pedregosa et al. (2011) da linguagem de programação Python criada por van Rossum (1995) como opção para a realização deste trabalho. Alguns dos motivos para a escolha desta biblioteca podem ser encontrados em Buitinck et al. (2013), o scikit-learn é uma biblioteca de aprendizado de

máquina cada vez mais popular. Escrita em Python, ele é projetado para ser simples e eficiente, acessível a não especialistas e reutilizável em vários contextos. A biblioteca scikit-learn possui um método para a realização do cálculo do problema de clusterização, o então chamado kmeans, objeto de estudo deste artigo.

Assim, o objetivo do trabalho foi determinar a influência de diferentes escolhas dos parâmetros do método kmeans da biblioteca Scikit-learn da linguagem de programação python em sua resposta final utilizando um conjunto de dados previamente conhecido da literatura, o conjunto iris. As hipóteses de pesquisa foram identificadas da seguinte forma: a hipótese nula, H_0 , é que não existe influência significativa de diferentes escolhas dos parâmetros do método kmeans e a hipótese alternativa H_1 deste trabalho foi que existe influência significativa de diferentes escolhas dos parâmetros do método kmeans.

Esse trabalho é estruturado em 5 seções, conforme descrito a seguir. Seção 1 tem-se a introdução com contextualização do problema tratado no estudo. Na Seção 2 o referencial teórico para base de entendimento do estudo. A seção 3 com os procedimentos metodológicos realizados. A Seção 4 as discussões e análises realizados e a Seção 5 com as conclusões obtidas com trabalho.

2. Referencial Teórico

2.1. Problema de clusterização

O problema de clusterização é apresentado de forma conceitual por Jain (2010) como uma representação de n objetos que encontre grupos K baseados em uma medida de similaridade tal que as semelhanças entre os objetos no mesmo grupo são elevados, enquanto as semelhanças entre objetos em grupos diferentes são baixos. A formulação matemática clássica pode ser encontrada em Singh e Goel (2015) e em Dasgupta (2008).

Consideremos $x_1, \dots, x_n \in \mathbb{R}^d$, estes são os valores (vetores) de entrada e $k \in \mathbb{N}$, k é o número de clusters. A resposta esperada é uma partição dos pontos x_i em clusters $C_1, \dots, C_k$, com centro μ_j , de forma a se minimizar a equação que segue abaixo:

$$\sum_{i=1}^k \sum_{x_i \in C_j} \|x_i - \mu_j\|^2 \quad (1)$$

Apesar de parecer um problema simples, por suas formulações matemáticas “simples” de ser entendida, este foi provado ser um problema NP-difícil por Drineas et al. (2004). Singh e Goel (2015), Dasgupta (2008) e Drineas et al. (2004) provaram que mesmo para o caso de apenas dois clusters o problema é, ainda assim, NP-difícil. Por ser um problema NP-difícil este acaba por ser resolvido por heurísticas ou meta heurísticas. De forma simplificada, um problema ser NP-difícil significa que não existe um algoritmo em tempo polinomial determinístico que o resolva de forma exata (CORMEN et al., 2012). Isso quer dizer que computacionalmente o problema é bastante difícil de ser resolvido de forma ótima.

Para resolver esse tipo de problema ao longo da história, foram criadas heurísticas e metaheurísticas. De acordo com Raidl (2006), o termo metaheurística foi introduzido pela primeira vez por Glover (1986). Uma metaheurística é um processo principal iterativo que orienta e modifica as operações de heurísticas subordinadas para produzir com eficiência soluções de alta qualidade.

2.2. O algoritmo do método kmeans

O termo K-means foi empregado primeiramente por MacQueen et al. (1967). O “Standard algorithm” foi proposto primeiramente por Stuart Lloyd em 1957, como uma técnica para modulação por código de pulso, mas não foi publicada fora dos laboratórios de *Bell* com a publicação Lloyd (1982). O algoritmo K-means é de longe o mais amplamente método usado para descobrir *clusters* em dados (ELKAN, 2003).

Segundo Bussab et al. (1990) o método Kmeans propicia indicações mais precisas sobre a quantidade de clusters a serem formados. É comumente utilizado e adequado a casos em que se exige a análise de um extenso conjunto de objetos com poucas variações entre si a serem aglomerados. A homogeneidade entre os clusters é obtida a partir da soma dos quadrados residual com base na análise de variância.

Ainda de acordo com Bussab et al. (1990) os passos envolvidos no algoritmo k-means, onde *k* é o número de grupos desejado e informado inicialmente, seguem da seguinte forma:

- Step (1) As posições iniciais dos *k* centróides dos clusters são determinadas;
- Step (2) Cada elemento é designado ao cluster do centróide mais próximo;
- Step (3) Os centros de cada clusters são recalculados a partir dos elementos alocados;
- Step (4) Repete-se os steps 2 a 4 até que algum critério de convergência ocorra.

O algoritmo do método k means é não determinístico, ou seja, ao ser executado duas vezes este, em geral, não resulta em uma mesma solução. Sabendo desta informação, da sensibilidade do algoritmo em encontrar uma boa solução sempre, foi assim levantada a questão de se mensurar com um conjunto de dados teste o quão sensível é o algoritmo ao diferentes execuções com mudanças em seus parâmetros de inicialização (CORMEN et al., 2012).

2.3. ANOVA

A análise de variância (ANOVA) é o método paramétrico mais eficiente disponível para a análise de dados de experimentos. Foi originalmente concebido para testar as diferenças entre vários grupos diferentes de tratamentos, contornando assim o problema de fazer comparações múltiplas entre as médias do grupo usando *t*-testes (ARMSTRONG et al., 2002).

O nome "análise de variância" baseia-se na abordagem em que o procedimento utiliza variâncias para determinar se as médias são diferentes. O procedimento funciona através da comparação da variância entre as médias de grupos versus a variância dentro dos grupos como uma maneira de determinar se os grupos são todos parte de uma população maior ou populações distintas com características diferentes(MINITAB,2018).

3. Procedimentos metodológicos

3.1. Setup computacional

Para a realização dos experimentos computacionais foi utilizado um notebook de uso comum, com o sistema operacional Ubuntu versão 16.04LTS. Com o intuito de evitar problemas com configuração de ambiente e instalação de programas, além de facilitar a reprodutibilidade do trabalho aqui desenvolvido foi utilizado o Docker versão 17.05.0-ce como virtualizador de ambientes. O Docker vem ganhando destaque também com essa utilização, ver por exemplo: Boettiger (2014), Chamberlain e Schommer (2014), Cito, Ferme e Gall (2016), Boettiger (2015). Boettiger (2015) é um dos trabalhos que trata exatamente de reprodutibilidade em pesquisa, e é desenvolvido de forma bastante didática.

A versão do python van Rossum (1995) utilizada esta foi a 3, a versão da biblioteca scikit-learn referida em Pedregosa et al. (2011) foi 0.19.1. para o processamento dos dados tabulares foi utilizada a biblioteca pandas versão 0.22.0 McKinney (2011), para visualização plotagem de figuras foi usada a biblioteca Matplotlib versão 2.2.2 Hunter (2007), para facilitar as operações com arrays de um modo geral foi utilizada a biblioteca NumPy versão 1.14.3 Oliphant (2006), para o cálculo da ANOVA foi usada a biblioteca SciPy versão 1.1.0 Jones et al. (2001). E finalmente para o desenvolver dos códigos de forma iterativa foi utilizado o jupyter notebook Kluyver et al. (2016).

3.2 Escolha dos parâmetros do método kmeans a serem variados

Após análise da documentação oficial do método foram escolhidos cinco parâmetros para serem variados e realizados os testes, por supostamente serem de grande influência no resultado do algoritmo. Estes são descritos em sequência e seguidos de uma breve explicação do seu significado adaptado da documentação oficial da biblioteca scikit-learn Pedregosa et al. (2011).

init: este parâmetro seleciona os centros de clusters iniciais para clusters de k-means de uma maneira inteligente para acelerar a convergência, podendo ser k-means++, random, ou um array fornecido pelo usuário, este array representa um chute inicial do centro dos clusters. O padrão é k-means++.

Bahmani et al. (2012) diz que como é bem conhecido, uma inicialização apropriada para o k-means é crucial para obter uma boa solução final. Este autor explica que o algoritmo de inicialização k-means++, alcança isso, obtendo um conjunto inicial de centros que é comprovadamente próximo da solução ótima.

n_init: a quantidade de vezes que o algoritmo será executado com diferentes sementes aleatórias de centróides. O padrão é 10.

max_iter: número máximo de iterações do algoritmo k-means para uma única execução. O padrão é 300.

precompute_distances: forma de acelerar a execução do algoritmo, mas que gasta mais memória. Possui as seguintes opções: auto, True, False, a opção auto só pré calcula as distâncias se o número de clusters vezes o número de amostras for menos que 12 milhões. O padrão é auto.

algorithm: algoritmo K-means para ser usado. O algoritmo clássico de é “full”. A variação “elkan” é mais eficiente usando a desigualdade triangular, mas atualmente não suporta dados esparsos. “Auto” escolhe “elkan” para dados densos e “full” para dados esparsos.

Elkan (2003) criador do método implementado na biblioteca, mostra em seu trabalho como foi possível acelerar drasticamente o algoritmo usando a desigualdade triangular, mantendo exatamente o mesmo resultado do algoritmo padrão.

3.3 Dados usado para a realização do experimento

Para a realização do experimento foi utilizado o conjunto de dados iris, Fisher (1936). Este conjunto trata de dados sobre três espécies de flores diferentes. Estes dados são: comprimento da sépala, largura da sépala, comprimento da pétala, largura da pétala; organizados em uma tabela com 4 colunas por 150 linhas. A rigor são 5 colunas, pois existe uma coluna que é a coluna que contém as respostas que informam a que tipo de flor os dados se referem, algumas linhas da tabela podem ser vistas na tabela.

Este conjunto foi escolhido por ser de tamanho pequeno se comparados com outros conjuntos presentes na literatura, pois por seu tamanho facilitou o seu manuseio e processamento realizado neste trabalho, e por ser um conjunto que possui as respostas, permitindo assim que se compare o desempenho do algoritmo.

3.4 Configuração do experimento e análise estatística

Foram escolhidos 5 parâmetros como fonte de variação da qualidade da resposta do método, estes foram, com os respectivos níveis: init: k-means++, random; n_init: 5, 10, 15; max_iter: 1, 20, 50; precompute_distances: True, False; e algorithm: full, elkan. O conjunto de dados iris foi processado de forma a que cada um dos tipos de flores fosse atribuído um número representando o cluster a que aquele tipo de flor pertence.



Fonte: <http://dataaspirant.com/2017/01/25/svm-classifier-implementacion-python-scikit-learn/>
 Figura 1 – Exemplos reais das flores: Setosa, Versicolor e Virginica

index	sepal_length	sepal_width	petal_length	petal_width	espécies
0	5.1	3.5	1.4	0.2	setosa
1	4.9	3.0	1.4	0.2	setosa
2	4.7	3.2	1.3	0.2	setosa
3	4.6	3.1	1.5	0.2	setosa
4	5.0	3.6	1.4	0.2	setosa
50	7.0	3.2	4.7	1.4	versicolor
51	6.4	3.2	4.5	1.5	versicolor
52	6.9	3.1	4.9	1.5	versicolor
53	5.5	2.3	4.0	1.3	versicolor
54	6.5	2.8	4.6	1.5	versicolor
100	6.3	3.3	6.0	2.5	virginica
101	5.8	2.7	5.1	1.9	virginica
102	7.1	3.0	5.9	2.1	virginica
103	6.3	2.9	5.6	1.8	virginica
104	6.5	3.0	5.8	2.2	virginica

Fonte: Autoria própria

Tabela 1 – Algumas linhas da tabela de dados do conjunto iris

Após isso a coluna dos labels dos tipos de flores foram retirados para que fossem feitas as simulações.

Para verificar o quão correto o algoritmo foi em comparação com as respostas classicamente conhecidas da literatura foi utilizado o índice ajustado de Rand (ARI) Rand (1971), muitos trabalhos o discutem, por exemplo: Yeung e Ruzzo (2001) e Santos e Embrechts (2009).

Para cada umas dessas configurações possíveis foram feitas réplicas da configuração e desta forma foi montada uma tabela em que cada configuração possuísse um número fixo de réplicas. Foi adotado um total de 20 réplicas para a realização do experimento.

Cada linha desta tabela, que representa uma configuração da simulação feita, foi usada como um grupo para a realização de uma ANOVA.

Para a realização de ANOVA foi usado o método da própria biblioteca scipy Jones et al. (2001), o método é encontrado em `scipy.stats.f_oneway`.

4. Resultados

Como resultado deste trabalho obteve-se um código em python que realiza todo o experimento e realiza sua análise estatística. Este facilita a reprodutibilidade do presente trabalho.

Os resultado numéricos do trabalho, aqui suprimidos, foram uma tabela com os resultado do ARI para um total de 20 réplicas em suas colunas e um total de 48 linhas.

Após a obtenção dos dados foi feita uma ANOVA. Os resultados do valor p e do f foram respectivamente, 0.78 e 0.73. Ou seja não foi significativa a variação dos dados. E o valor mínimo do ARI foi 0.41, o máximo foi 0.73, e o valor médio foi 0.72. O que quer dizer que na média ao a variação foi muito próxima do valor máximo.

Estes valores não nos fornecem evidência suficiente para que se rejeite a hipótese nula, H_0 , do trabalho que foi que não existe influência significativa de diferentes escolhas dos parâmetros do método Kmeans da biblioteca scikit-learn da linguagem python. ao menos não para o conjunto de dados iris, que foi o que foram feitas as simulações.

5. Conclusões

Um problema recorrente desde o começo da história da humanidade é a busca de padrões em algum conjunto de dados. Atualmente com os desenfreado aumento na geração de dados que a internet tem nos proporcionado a quantidade de dados sendo gerados é enorme (chegamos a era do big data). Uma preocupação que surge é como processar e analisar essa quantidade de dados de forma que se possa extrair significado construtivos para algum fim.

Dentro desse contexto da criação de grandes quantidades de dados e da necessidade de extração de informação destes por meio de alguma técnica de classificação, o aprendizado de máquina tem crescido e se mostrado cada vez mais eficaz.

Dentro desse campo da ciência o algoritmo K-means vem sendo utilizado por mais de 50 anos desde sua criação. Para que técnicas como essa possam ser utilizadas ao longo dos anos foram criados pacotes computacionais que implementam tais algoritmos, e o caos do K-means não é diferente.

A linguagem de programação python possui uma implementação deste algoritmo, esta se encontra na biblioteca scikit-learn, e é nomeada como Kmeans. Este método foi o objeto de estudo deste artigo.

Por conta do algoritmo k-means ser um problema NP-difícil, os algoritmos que o implementam são não determinísticos. Este fato, em geral, leva a soluções diferentes para execuções diferentes do mesmo algoritmo, além de um mal ajuste nos parâmetros do método computacional. Isso pode desencadear um mal aproveitamento do potencial do algoritmo implementado.

Por conta deste problema este trabalho propôs que fosse realizado um experimento computacional de forma a verificar o quão sensível a variações dos parâmetros do método computacional método Kmeans da biblioteca scikit-learn são sensíveis.

Para a determinação desta sensibilidade foram escolhidos como fatores de fontes de variação, e seus respectivos níveis: init: k-means++, random; n_init: 5 10 15; max_iter: 1, 20, 50; precompute_distances: True, False; algorithm: full, elkan. Os dados de teste utilizados foram os do famoso conjunto de dados iris. Este é composto de dados de três tipos de flores, formando uma tabela de 150 linhas por 5 colunas, sendo uma dessas informando a que tipo de flor cada linha pertence.

Para calcular o quanto o algoritmo errou/acertou foi utilizado o ARI. Este foi utilizado para que com se fosse montado um experimento computacional que simulasse a utilização do método por um usuário que escolhesse diferentes combinações dos valores padrão dos parâmetros do método da biblioteca.

A conclusão da ANOVA feita com base nos dados da simulação forneceu o valor $p = 0.78$ e o valor $f = 0.73$. Estes valores indicam que não se tem evidência suficiente para rejeitar a hipótese nula do trabalho, que foi a de que uma diferente escolha dos parâmetros da biblioteca influenciaram no resultado do algoritmo

Referências

ARMSTRONG, R. A; EPERJESI, F.; GILMARTIN, B. *The application of analysis of variance (ANOVA) to different experimental designs in optometry.* Ophthalmic & Physiological Optics, v.5, n. 3, p.248-256

BAHMANI, B. et al. *Scalable k-means++.* Proceedings of the VLDB Endowment, VLDB Endowment, v. 5, n. 7, p. 622–633, 2012.

BOETTIGER, C. *An introduction to docker for reproducible research, with examples from the R environment.* CoRR, abs/1410.0846, 2014.

BOETTIGER, C. *An introduction to docker for reproducible research.* SIGOPS Oper. Syst. Rev., ACM, New York, NY, USA, v. 49, n. 1, p. 71–79, jan. 2015. ISSN 0163-5980.

BUITINCK, L. et al. *API design for machine learning software: experiences from the scikit-learn project.* CoRR, abs/1309.0238, 2013.

CHAMBERLAIN, R.; SCHOMMER, J. *Using docker to support reproducible research.* DOI: <https://doi.org/10.6084/m9.figshare.v.1101910>, p. 44, 2014..

CHAPELLE, O.; SCHOLKOPF, B.; EDS., A. Z. *Semi-supervised learning* (chappelle, o. et al., eds.; 2006) [book reviews]. IEEE Transactions on Neural Networks, v. 20, n. 3, p. 542–542, March 2009. ISSN 1045-9227.

CITO, J.; FERME, V.; GALL, H. C. *Using docker containers to improve reproducibility in software and web engineering research.* In: SPRINGER. International Conference on Web Engineering. 2016. p. 609–612. ISBN 978-3-319-38791-8..

CORMEN, T. H. et al. *Algoritmos Teoria e Prática.* 3. ed. Cambridge, Massachusetts London, England: ELSEVIER, 2012. 1313

DASGUPTA, S. *The hardness of k-means clustering*. Department of Computer Science and Engineering, University of California, San Diego, 2008. Disponível em: .

DRINEAS, P. et al. *Clustering large graphs via the singular value decomposition*. Machine learning, Kluwer Academic Publishers-Plenum Publishers, v. 56, n. 1-3, p. 9–33, 2004. ISSN 1573-0565.

DUDA, R. O. et al. *Pattern classification*. 2nd. Edition. New York, p. 55, 2001..

ELKAN, C. *Using the triangle inequality to accelerate k-means*. In: Proceedings of the 20th International Conference on Machine Learning (ICML-03). [s.n.], 2003. p. 147–153.

FISHER, R. A. *The use of multiple measurements in taxonomic problems*. Annals of human genetics, Wiley Online Library, v. 7, n. 2, p. 179–188, 1936.

GANTZ, J. F. et al. *The diverse and exploding digital universe*. White Paper, Citeseer, 2008.

GLOVER, F. *Future paths for integer programming and links to artificial intelligence*. Computers & Operations Research, v. 13, n. 5, p. 533 – 549, 1986. ISSN 0305-0548. Applications of Integer Programming.

HUNTER, J. D. *Matplotlib: A 2d graphics environment*. Computing in Science & Engineering, v. 9, n. 3, p. 90–95, 2007.

JAIN, A. K. *Data clustering: 50 years beyond k-means*. Pattern Recognition Letters, v. 31, n. 8, p. 651 – 666, 2010. ISSN 0167-8655. Award winning papers from the 19th International Conference on Pattern Recognition (ICPR).

JONES, E. et al. *SciPy: Open source scientific tools for Python*. 2001. .

KLUYVER, T. et al. *Jupyter notebooks – a publishing format for reproducible computational workflows*. In: LOIZIDES, F.; SCHMIDT, B. (Ed.). Positioning and Power in Academic Publishing: Players, Agents and Agendas. 2016. p. 87 – 90.

LLOYD, S. *Least squares quantization in pcm*. IEEE transactions on information theory, IEEE, v. 28, n. 2, p. 129–137, 1982.

MACQUEEN, J. et al. *Some methods for classification and analysis of multivariate observations*. In: OAKLAND, CA, USA. Proceedings of the fifth Berkeley symposium on mathematical statistics and probability. [S.l.], 1967. v. 1, n. 14, p. 281–297.

MCKINNEY, W. *pandas: a foundational python library for data analysis and statistics*. Python for High Performance and Scientific Computing, p. 1–9, 2011.

MINITAB. *O que é ANOVA?* Disponível em :
<https://support.minitab.com/pt-br/minitab/18/help-and-how-to/modeling-statistics/anova/supporting-topics/basics/what-is-anova/> Acessado em 26 de Set 2018

OLIPHANT, T. E. *A guide to NumPy*. USA: Trelgol Publishing, 2006. v. 1.

PEDREGOSA, F. et al. *Scikit-learn: Machine learning in Python*. Journal of Machine Learning Research, v. 12, p. 2825–2830, 2011.

RAIDL, G. R. *A unified view on hybrid metaheuristics*. In: Proceedings of the Third International Conference on Hybrid Metaheuristics. Berlin, Heidelberg: Springer-Verlag, 2006. (HM'06), p. 1–12. ISBN 3-540-46384-4, 978-3-540-46384-9.

RAND, W. M. *Objective criteria for the evaluation of clustering methods*. Journal of the American Statistical association, Taylor & Francis Group, v. 66, n. 336, p. 846–850, 1971.

SANTOS, J. M.; EMBRECHTS, M. *On the use of the adjusted rand index as a metric for evaluating supervised classification.* In: SPRINGER. International Conference on Artificial Neural Networks. Berlin, Heidelberg, 2009. p. 175–184. ISBN 978-3-642-04277-5.

SINGH, I. P.; GOEL, S. Col870: *Clustering algorithms hardness of k-means clustering.* 2015.

Van ROSSUM, G. *Python tutorial.* Amsterdam, 1995.

Voß, S. et al. *Meta-Heuristics: Advances and trends in local search paradigms for optimization.* 1. ed. New York: Springer US, 1999. 523 p. ISBN 978-1-4613-7646-0.

WITTEN, I. H. et al. *Data Mining: Practical machine learning tools and techniques.* [S.l.]: Morgan Kaufmann, 2016.

YEUNG, K. Y.; RUZZO, W. L. *Details of the adjusted rand index and clustering algorithms, supplement to the paper an empirical study on principal component analysis for clustering gene expression data.* Bioinformatics, v. 17, n. 9, p. 763–774, 2001.