# UM ESTUDO SOBRE A INFLUÊNCIA DA SELEÇÃO DE ATRIBUTOS NO AGRUPAMENTO DE VAGAS EM APLICAÇÕES DE E-RECRUITMENT

Joel Joaquim de Souza Junior e Leandro Nunes de Castro

**Apoio: PIBIC Mackenzie**

## RESUMO

Sistemas de recomendação, em geral, têm por finalidade recomendar itens ao usuário de forma eficiente e eficaz baseado em seu perfil. Um sistema de recrutamento online (*e-recruitment*) tem por finalidade recomendar vagas para um candidato de acordo com o seu perfil podendo também agir na via inversa, recomendando candidatos mais qualificados para determinada vaga. Definir quais variáveis serão utilizadas impacta diretamente na qualidade da recomendação de modo que, ao utilizar as variáveis mais importantes, temos uma melhor assertividade no processo. Tendo isso em foco, este trabalho destina-se a realizar a seleção das variáveis mais importantes de uma base de dados de recrutamento online utilizando técnicas de seleção de atributos (características). Foram utilizados os algoritmos de Mitra, SUD e ACA para realizar a seleção dos atributos aplicados no agrupamento da base. As bases utilizadas foram derivadas da base original considerando três cenários distintos: a base contendo os atributos com características das vagas; a base contendo o *bag of words* do atributo de descrição das vagas; e a base resultante da união das duas anteriores. Os subconjuntos de variáveis selecionadas em cada um dos cenários acima tiveram seu desempenho avaliado na tarefa de agrupamento. Os resultados obtidos em cada cenário mostram um ganho de desempenho do agrupamento quando é feita a seleção de variáveis em comparação com as bases originais.

**Palavras-chave:** Sistema de Recomendação. Recrutamento Online. Seleção de Atributos.

## ABSTRACT

Recommender systems aim to effectively recommend items to the user based on their profile. An online recruitment system recommends jobs for a candidate according to his profile and can also act in reverse, recommending more qualified candidates for a particular job. Defining which variables will be used impacts directly on the recommendation quality so that, when using the most important variables, we have a

better assertiveness in the process. With this in mind, the goal of this work is to select the most important features of an online recruitment database using feature selection techniques. More specifically, we used the algorithms of Mitra, SUD and ACA to perform feature selection. The datasets used were derived from the original dataset assuming three distinct scenarios: the dataset containing the attributes related with the jobs characteristics; the dataset containing the bag of words of the description attribute of the jobs; and the dataset resulting from the union of the two previous ones. The features subsets selected in each of the above scenarios had their performance evaluated in a clustering task. The results obtained in each scenario show a performance gain of the clustering process when feature selection is made over the original data.

**Keywords:** Recommender System. Online Recruitment. Feature Selection.

## 1. INTRODUÇÃO

O processo tradicional de recrutamento de candidatos seguido pelas empresas se baseia em anunciar as vagas disponíveis, analisar currículos, ligar para os amigos e referências dos candidatos, realizar diversas entrevistas, entre outras etapas. Todo este processo demanda muito tempo e dinheiro, além de limitar o grupo de candidatos para a vaga (Yoon Kin Tong, 2009; Capelli, 2001). Levando em conta essa limitação, a probabilidade de se contratar um candidato que melhor atenda a vaga é muito baixa.

Com o desenvolvimento tecnológico e o crescimento da internet, o acesso e a disponibilidade à informação cresceram de tal modo que atualmente recrutadores têm acesso a diversos currículos pela internet, podendo selecionar dentre eles o candidato melhor qualificado para determinada vaga (Capelli, 2001; Faliagka, Tsakalidis, & Tzimas, 2012). Diante disso, surgiram empresas de recrutamento online, denominado *e-recruitment* ou e-recrutamento, que possuem sistemas de recomendação especializados em atender a atual demanda de candidatos e vagas (Faliagka, Tsakalidis, & Tzimas, 2012; Yoon Kin Tong, 2009). De modo geral, o candidato envia seu currículo para esses sistemas em arquivos de texto ou através de formulários na plataforma, e recrutadores inserem as oportunidades de emprego especificando as características e exigências das vagas. O sistema então passa a calcular a melhor combinação entre candidato e vaga. A recomendação realizada pode ser feita em duas vias: recomenda-se ao candidato posições que mais se adequem ao seu perfil, ou seleciona-se para a vaga candidatos que mais atendam às suas especificações (Faliagka, Tsakalidis, & Tzimas, 2012).

Os sistemas de *e-recruitment*, entretanto, enfrentam o problema de lidar com os dados não-estruturados provenientes das descrições das vagas e também dos currículos dos candidatos. Técnicas são aplicadas a esses dados, como *stemming* e *filtering*, a fim de se estruturar esses dados (Faliagka, Tsakalidis, & Tzimas, 2012). Entretanto, após a estruturação enfrenta-se o problema da alta dimensionalidade que é gerada, pois são produzidas muitas variáveis (também denominadas atributos), tornando a recomendação mais custosa computacionalmente e menos assertiva. Selecionar as variáveis mais importantes levaria a um casamento candidato-vaga mais eficiente e eficaz.

Este projeto tem por objetivo investigar a influência da seleção de atributos no agrupamento de uma base de dados de vagas de emprego utilizada em aplicações

de recrutamento online de candidatos. Além de resolver o problema da alta dimensionalidade, e por consequência problemas relacionados como alto custo de processamento e de armazenamento, espera-se que a seleção das variáveis mais importantes resulte em um agrupamento de melhor qualidade, conforme métricas definidas adiante.

A base de dados utilizada nesse projeto foi obtida a partir de uma empresa real, é formada por atributos estruturados e não-estruturados e não possui uma classe alvo. Por não possuir classe alvo, a análise da base se concentrará em uma tarefa de agrupamento de dados. A solução proposta faz uso de métodos de seleção não supervisionada de atributos do tipo filtro, uma vez que não utilizamos classe alvo e nem classificadores no processo de geração do subconjunto. Mais especificamente, são utilizados três métodos de seleção de atributos: Mitra; SUD; e ACA. Esses métodos são comparados entre si e com o conhecido método de extração de variáveis baseado em Análise de Componentes Principais (PCA). Cada subconjunto de variáveis produzido por cada método será avaliado nos seguintes cenários: base de dados somente com atributos estruturados; base dados somente com atributos não estruturados (descrição da vaga); e base de dados completa, contendo ambos os tipos de atributos.

Este trabalho está organizado da seguinte forma. A Seção 2 descreve o processo de seleção de atributos bem como os algoritmos que serão analisados neste trabalho a Seção 3 descreve a metodologia aplicada e apresenta os resultados obtidos; e, por fim, a Seção 4 apresenta a conclusão e considerações finais.

## 2. SELEÇÃO DE ATRIBUTOS

Seleção de atributos é o processo que visa diminuir a quantidade de atributos (dimensão do espaço de busca) selecionando um subconjunto de atributos que melhor atenda a um critério de qualidade previamente estabelecido (Jain & Zongker, 1997; Liu & Yu, 2005; Molina, Belanche, & Nebot, 2002).

Formalmente, podemos definir a seleção de atributos da seguinte maneira: para um conjunto $X$, conjunto original de atributos, com cardinalidade $|X| = n$ temos um subconjunto $X'$ com cardinalidade $|X'| = d$, $X' \subseteq X$, que é selecionado de acordo com o critério de avaliação adotado, denotado pela função de avaliação $J(X')$, $J: X' \subseteq X \rightarrow$

$\mathbb{R}$. A tarefa de seleção de atributos consiste em encontrar $X'$, $X' \subseteq X$, de modo que $J(X')$ seja máximo, assumindo que um maior valor de $J(X')$ corresponde a uma melhor avaliação (Jain & Zongker, 1997; Molina, Belanche, & Nebot, 2002).

A seleção de atributos é uma etapa que pode ser encontrada em várias análises da Mineração de Dados, sendo utilizada tanto na tarefa de classificação, onde faz uso de dados rotulados e é conhecida como seleção supervisionada de atributos, como na tarefa de agrupamento, onde faz uso de dados não rotulados e é conhecida como seleção não supervisionada de atributos (Liu & Yu, 2005).

O processo de seleção de atributos pode ser dividido em quatro etapas (Liu & Yu, 2005):

1. **Geração do Subconjunto**: nessa etapa é produzido um subconjunto de atributos candidatos utilizando-se uma estratégia de busca. Esta etapa é, essencialmente, um processo de busca heurística no qual dois aspectos importantes devem ser definidos: um ou mais pontos iniciais, e a estratégia de busca. Um algoritmo de busca é responsável por guiar o processo de seleção segundo alguma estratégia. Neste sentido, as estratégias de busca que podem ser usadas para evitar a enumeração completa de todos os subconjuntos possíveis de atributos incluem a *seleção forward*, a *eliminação backward*, um *método composto* (seleção *forward* e eliminação *backward*) e um método aleatório (Liu & Yu, 2005; Molina, Belanche, & Nebot, 2002).
2. **Avaliação do subconjunto:** nessa etapa o subconjunto é avaliado de acordo com algum critério, comparando-o com o melhor subconjunto obtido até o momento, de modo que o melhor dos dois seja utilizado na próxima etapa (Liu & Yu, 2005; Molina, Belanche, & Nebot, 2002). Os critérios de avaliação são agrupados em (Liu & Yu, 2005): *dependentes*: a avaliação é feita com base no algoritmo de mineração de dados que será utilizado; *independentes*: não é utilizado o algoritmo de mineração de dados para se avaliar o subconjunto.
3. **Critério de parada:** determina quando a parte iterativa do processo deve encerrar. Se o critério adotado for alcançado então passa-se para a próxima etapa, caso contrário o processo volta para a primeira etapa. Em outras palavras, são gerados subconjuntos de forma iterativa até que o melhor deles cumpra este critério. Alguns critérios adotados são: o término da busca; alguma marca estabelecida, como número mínimo (ou máximo) de características;

após algumas subsequentes iterações sem melhora nos subconjuntos de atributos; um subconjunto suficientemente bom é obtido.

4. **Validação do resultado:** algum conhecimento prévio ou critério é usado para avaliar a qualidade do resultado.

Um Algoritmo para Seleção de Atributos, ou FSA, (do inglês *Feature Selection Algorithm*), (Molina, Belanche, & Nebot, 2002) é uma abordagem computacional para resolver o problema de seleção de atributos baseado em uma definição de relevância, ou critério de avaliação. A relevância adotada está fortemente ligada ao objetivo almejado no final do processo.

Baseado no critério de avaliação e no algoritmo de mineração de dados adotado, os algoritmos de seleção de atributos podem ser divididos em (Liu & Yu, 2005; Molina, Belanche, & Nebot, 2002):

- *Embedded*: quando o algoritmo de mineração de dados possui um FSA incorporado, de forma explícita ou não.
- *Filter*: quando o FSA se baseia em características gerais dos dados e é independente do algoritmo de mineração. Sua utilização geralmente se dá em uma etapa anterior, no pré-processamento, de forma que este age como um filtro das variáveis que serão utilizadas.
- *Wrapper*: quando o FSA possui seu próprio algoritmo de mineração, de forma a utilizá-lo como critério de avalição do subconjunto gerado. Embora este método aumente a performance do processo de mineração realizado, ele normalmente provoca um alto custo computacional.
- *Hybrid*: uma forma de se obter o melhor resultado utilizando os modelos *Filter* e *Wrapper* em diferentes estágios da geração do subconjunto.

Liu et al. (2005) propõem um framework tridimensional para categorização de algoritmos de seleção de atributos. Nesse framework, um algoritmo é categorizado segundo sua Estratégia de Busca (Completa, Sequencial ou Aleatória), Critério de Avaliação (*Filter, Wrapper* ou *Hybrid*) e a tarefa de Mineração de Dados à qual está associado (Classificação ou Agrupamento) (Liu & Yu, 2005).

**2.1**
## 2.1 ALGORITMOS UTILIZADOS

Para o escopo deste projeto procuramos trabalhar de forma independente de um classificador e utilizando uma base de dados não rotulada. Assim, as técnicas de seleção de atributos utilizadas são inerentemente não-supervisionadas, do tipo *Filtro*. Foram escolhidos três algoritmos para teste: o algoritmo de Mitra *et al* (Mitra, Murthy, & Pal, 2002), o algoritmo intitulado de SUD (Dash, Liu, & Yao, 1997) e o algoritmo ACA (Au, Chan, Wong, & Wang, 2005), todos com enfoque na seleção de atributos baseada na relação entre os próprios atributos.

### 2.1.1 Mitra

O algoritmo descrito em Mitra *et al* (Mitra, Murthy, & Pal, 2002) se baseia no agrupamento dos atributos de acordo com suas similaridades. O algoritmo trabalha em duas etapas: 1) dividir o conjunto inicial de atributos em grupos homogêneos; e 2) selecionar representantes de cada grupo. A similaridade dos atributos é calculada com base na dependência linear entre eles, utilizando o *Índice de Compressão Máxima de Informação* (Mitra, Murthy, & Pal, 2002; Ferreira & Figueired, 2012; Bharti & Singh, 2014):

$$2\lambda_2(x,y) = var(x) + var(y) - \sqrt{\left(var(x) + var(y)\right)^2 - 4var(x)var(y)(1-\rho(x,y)^2)} \quad (1)$$

onde $x$ e $y$ são dois atributos, $var(x)$ é a variância de $x$ e $\rho(x,y)$ é o coeficiente de correlação entre $x$ e $y$. Os valores obtidos estão contidos no intervalo $0 \leq \lambda_2(x,y) \leq 0.5(var(x) + var(y))$, sendo que duas variáveis são totalmente similares (linearmente dependentes) quando $\lambda_2$ é igual a zero e quanto mais dissimilares, maior é o valor de $\lambda_2$.

O algoritmo trabalha da seguinte forma (Mitra, Murthy, & Pal, 2002):

- Calcular os $n$ vizinhos mais próximos de cada atributo no conjunto original de atributos.
- Dentre os subconjuntos formados, selecionar o subconjunto com a menor distância entre o atributo principal e o vizinho mais distante do conjunto. Descartar os atributos pertencentes ao subconjunto ($n$ vizinhos) mantendo apenas o atributo principal.

- Na primeira iteração, guardar o valor calculado entre o atributo principal e o vizinho mais distante do subconjunto selecionado. Este valor será utilizado como um limiar de erro $\varepsilon$.
- Nas iterações seguintes, verificar o valor $\lambda_2$ do subconjunto, se for maior que $\varepsilon$ há um decréscimo no valor de $n$.

- Repetir o processo com os atributos restantes até que todos os atributos estejam no mesmo subconjunto.

Ao término do algoritmo obtemos como resultado uma lista de atributos, sendo que cada um deles é um representante de um subconjunto de atributos.

### 2.1.2 SUD

O algoritmo SUD (Dash, Liu, & Yao, 1997) é um algoritmo sequencial de seleção retroativa de atributos. O foco do SUD é determinar a importância relativa de cada atributo, sendo essa importância determinada pela entropia do conjunto de objetos sem o atributo, de modo que atributos mais importantes quando removidos resultam em uma entropia maior na base. A medida utilizada para se determinar a importância relativa dos atributos é a medida de entropia, definida como (Dash, Liu, & Yao, 1997):

$$E = -\sum_{i=1}^{N}\sum_{j=1}^{N}(S_{ij} \times \log S_{ij} + (1 - S_{ij}) \times \log(1 - S_{ij})) \quad (2)$$

onde $N$ é a quantidade de objetos sendo considerados e $S_{ij}$ é a medida similaridade entre dois objetos $x_i$ e $x_j$.

Quando todas as variáveis são numéricas ou ordinais, $S_{ij}$ é definida como (Dash, Liu, & Yao, 1997):

$$S_{ij} = e^{-\alpha \times D_{ij}} \quad (3)$$

onde $D_{ij}$ é a distância euclidiana normalizada definida como: $D_{ij} = \sqrt{\sum_{k=1}^{M}(\frac{x_{ik}-x_{jk}}{max_k - min_k})^2}$, sendo $max_k$ e $min_k$ o maior e o menor valor, respectivamente, da $k$-ésima variável; e $\alpha = \frac{-\ln 0.5}{\bar{D}}$, sendo $\bar{D}$ a distância média entre os objetos.

No caso de as variáveis serem nominais, $S_{ij}$ é definida como:

$$S_{ij} = \frac{\sum_{k=1}^{M} |x_{ik} = x_{jk}|}{M} \quad (4)$$

onde $|x_{ik} = x_{jk}|$ é igual a 1 caso $x_{ik} = x_{jk}$ e 0 caso contrário, e $M$ é a quantidade de variáveis sendo consideradas.

O algoritmo funciona da seguinte forma (Dash, Liu, & Yao, 1997), sendo $M$ a quantidade de variáveis:

- Seja $O$ o conjunto ordenado, inicialmente vazio, de variáveis de acordo com sua importância, sendo que os primeiros elementos são os menos importantes e os últimos os mais importantes.
- São realizadas $M - 1$ iterações.
    - Para cada iteração calcula-se a entropia após remover uma variável $v$ do conjunto das variáveis restantes $T$. Inicialmente $T$ contém todas as variáveis existentes. Para cada $v$ em $T$ calcula-se:
        - $T_v$ = o conjunto $T$ sem a variável $v$.
        - A entropia de $T_v$.
    - A variável $v$ que, ao ser removida faz com que a entropia de $T_v$ seja a menor de todos os $T_v$, é a variável de menor importância e é adicionada ao final de $O$.
    - $T = T_v$
- Retorna o conjunto $O$.

Para se obter uma lista das variáveis mais importantes basta reverter a ordem do conjunto resultante, de modo que as primeiras variáveis passam a ser as mais importantes e as últimas as menos importantes. Para redução da dimensionalidade da base basta selecionar as $d$ primeiras variáveis do conjunto resultante. Uma desvantagem aparente é que $d$ é dependente da base de dados em que o algoritmo é aplicado e é necessário determiná-lo de modo empírico, ou seja, achar a quantidade mínima que, se ultrapassada, não ocasiona melhora na performance e nos resultados finais do algoritmo (Dash, Liu, & Yao, 1997).

### 2.1.3 ACA

O algoritmo ACA foi proposto em Au *et al* (Au, Chan, Wong, & Wang, 2005). Sua estratégia se baseia em agrupar os atributos em conjuntos com base na similaridade entre eles. Para o cálculo da similaridade é utilizada a Medida de Redundância de Interdependência (Au, Chan, Wong, & Wang, 2005), capaz de mostrar correlações

negativa e positiva, bem como interdependência e proximidade dos valores, ao mesmo tempo em que é robusta para lidar com *outliers*.

A notação utilizada para descrever o algoritmo e as medidas associadas a ele será a seguinte: seja $O$ uma base de dados que contenha $p$ atributos, denotaremos por $A_i$, $i \in \{1, \dots, p\}$ cada atributo da base $O$ e por $v_{ik}$, $k \in \{1, \dots, m\}$, $i \in \{1, \dots, p\}$, o valor do atributo $A_i$ no objeto $k$.

A Medida de Redundância de Interdependência entre dois atributos $A_i$ e $A_j$, $i, j \in \{1, \dots, p\}$, $i \neq j$, é definida como (Au, Chan, Wong, & Wang, 2005):

$$R(A_i : A_j) = \frac{I(A_i : A_j)}{H(A_i, A_j)} \tag{5}$$

onde $I(A_i : A_j)$ é a medida de informação mútua entre $A_i$ e $A_j$, e é definida como:

$$I(A_i : A_j) = \sum_{k=1}^{m_i} \sum_{l=1}^{m_j} \Pr(A_i = v_{ik} \wedge A_j = v_{jl}) \times \log \frac{\Pr(A_i = v_{ik} \wedge A_j = v_{jl})}{\Pr(A_i = v_{ik}) \Pr(A_j = v_{jl})} \tag{6}$$

e $H(A_i, A_j)$ é a medida de entropia entre $A_i$ e $A_j$, definida como:

$$H(A_i, A_j) = -\sum_{k=1}^{m_i} \sum_{l=1}^{m_j} \Pr(A_i = v_{ik} \wedge A_j = v_{jl}) \times \log \Pr(A_i = v_{ik} \wedge A_j = v_{jl}) \tag{7}$$

Nas equações acima $Pr$ denota a função de probabilidade e é definida como: seja $\sigma$ a função SELECT da álgebra relacional, a probabilidade de um atributo $A_i$ qualquer ser igual a $v_{ik}$ é dada por:

$$\Pr(A_i = v_{ik}) = \frac{\left|\sigma_{A_i = v_{ik}}(R)\right|}{\left|\sigma_{A_i \neq NULL}(R)\right|} \tag{8}$$

e a probabilidade conjunta de dois atributos distintos $A_i$ e $A_j$ quaisquer possuírem valores iguais a $v_{ik}$ e $v_{jl}$, respectivamente, é dada por:

$$\Pr(A_i = v_{ik} \wedge A_j = v_{jl}) = \frac{\left|\sigma_{A_i = v_{ik} \wedge A_j = v_{jl}}(R)\right|}{\left|\sigma_{A_i \neq NULL \wedge A_j \neq NULL}(R)\right|} \tag{9}$$

com $i, j \in \{1, \dots, p\}$, $i \neq j$, e $k, l \in \{1, \dots, m\}$.

Atrelada à medida de redundância de interdependência o algoritmo também utilizada outra medida para calcular a interdependência de um atributo dentro de um grupo (ou subconjunto) $C = \{A_j | j = 1, \dots, p\}$, $C \subseteq O$, chamada de Medida de Redundância de Interdependência Múltipla (Au, Chan, Wong, & Wang, 2005), definida como:

$$MR(A_i) = \sum_{j=1}^{p} R(A_i : A_j) \tag{10}$$

O algoritmo (Au, Chan, Wong, & Wang, 2005) utiliza o conceito de moda para agrupar os atributos nos subconjuntos correspondentes. A moda de um subconjunto $C = \{A_j | j = 1, \dots, p\}$, denotado por $\eta(C)$, é o atributo $A_i \in C$, tal que $MR(A_i) > MR(A_j)$, para todo $A_j \in C$. O algoritmo segue os seguintes passos:

- O algoritmo recebe como argumento o número de grupos que deverão ser formados, $k$, que é um inteiro e $k \in \{2, \dots, p\}$. São selecionados, aleatoriamente, dentre os $p$ atributos, $k$ candidatos distintos a modas para os $k$ grupos, ou seja, para cada $C_r$, $C_r \subseteq O$, atribuímos $\eta_r = A_i$, $\eta_r \in C_r$, $r \in \{1, \dots, k\}$, $i \in \{1, \dots, p\}$.
- Cada atributo é adicionado ao subconjunto cuja Medida de Redundância de Interdependência entre a moda do subconjunto e o atributo é maior.
- Uma vez definidos os grupos, calcula-se para cada grupo um novo candidato a moda. O novo candidato a moda será o atributo do grupo que possuir a maior Medida de Redundância de Interdependência Múltipla, em outras palavras, para cada $C_r$, com $r \in \{1, \dots, k\}$, atribuímos $\eta_r = A_i$ se $MR(A_i) \geq MR(A_j)$, para todo $A_i$, $A_j \in C_r$, $i \neq j$.
- Os Passos 2 e 3 são repetidos até que não haja alteração das modas ou um limite pré-estabelecido de iterações seja alcançado.

O resultado da execução do algoritmo são grupos de atributos e dos quais, de cada grupo, pode-se escolher um atributo como representante, saindo de uma representação da base utilizando $p$ atributos, para uma representação da base utilizando-se $r$ atributos.

Diferente do algoritmo SUD, neste algoritmo temos que o valor de $k$ define a quantidade final de grupos que se deseja obter. Para se achar o valor ideal de $k$ é sugerida (Au, Chan, Wong, & Wang, 2005) uma abordagem empírica, na qual são testados valores até que se ache o que produz mais subconjuntos similares, $k = \arg max_{k \in \{2,\dots,p\}} \sum_{r=1}^{k} \sum_{A_i \in C_r} R(A_i : \eta_r)$.

## 3. AVALIAÇÃO DE DESEMPENHO

Os experimentos a serem realizados têm como objetivo avaliar o desempenho dos três algoritmos de seleção de atributos (ACA, SUD e Mitra) quando aplicados na seleção de características em uma tarefa de agrupamento de dados. Esses métodos serão comparados entre si e com o conhecido método de extração de variáveis baseado em Análise de Componentes Principais (PCA). O algoritmo de agrupamento que será utilizado é o algoritmo *k-means*, e o Coeficiente de Silhueta será empregado para avaliar o agrupamento. Como o algoritmo *k-means* é estocástico, ele será executado 10 vezes e o resultado apresentado será a média $\pm$ desvio padrão a partir das 10 simulações.

### 3.1 METODOLOGIA EXPERIMENTAL

A base de dados original utilizada é uma base real contendo 1.933 registros e constituída por informações das vagas, a saber: *nome da vaga*, *o idioma requisitado*, *período de trabalho*, *salário*, *área de atuação*, *modelo de contratação*, *nível de escolaridade*, *benefícios*, *experiência anterior necessária*, *localização*, *gestor*, *número de vagas*, e *descrição*. Todos os atributos das vagas são numéricos ou nominais, com exceção do atributo *descrição*, que corresponde a um campo de texto usado pela empresa para descrever a vaga.

Os atributos da base de dados original foram separados em três subconjuntos distintos, formando três cenários de teste. O primeiro subconjunto de atributos (**C1**) contém apenas aqueles atributos relacionados às vagas, desconsiderando o atributo *descrição*. O segundo subconjunto (**C2**) contém apenas a parte textual descritiva da base original, transformada em um *bag of words*. O terceiro subconjunto (**C3**) é a união dos dois anteriores. Para cada cenário serão considerados os seguintes valores de *k* do *k-means*: {2, 4, 6, 8}. Cada subconjunto e cada valor de *k* leva a um grupo de quatro experimentos:

1. O primeiro experimento servirá de *benchmarking* para comparar o desempenho dos algoritmos de seleção de características. Nele a base será avaliada contendo todos os atributos.
2. O segundo experimento aplicará o algoritmo de Mitra para selecionar os atributos da base. Esse algoritmo determina automaticamente o número *d* de variáveis a serem utilizadas.
3. O terceiro experimento aplicará o algoritmo SUD para selecionar os atributos da base. O valor de *d* a ser utilizado pelo SUD será o mesmo determinado por Mitra.
4. O quarto experimento aplicará o algoritmo ACA para selecionar os atributos da base. O valor de *d* a ser utilizado pelo ACA será o mesmo determinado por Mitra.

Cada experimento será avaliado utilizando o seguinte procedimento: aplicar o algoritmo de seleção de atributos, executar o algoritmo de agrupamento em cada cenário, e calcular a medida de silhueta do agrupamento formado.

## 3.2 ALGORITMO DE AGRUPAMENTO E AVALIAÇÃO DE DESEMPENHO

O algoritmo *k-means* (Arthur & Vassilvitskii, 2007) realiza o agrupamento dos objetos da base de dados em $k$ clusters de modo a minimizar a função de custo $\phi$:

$$\phi = \sum_{x \in \chi} min_{c \in C} \|x - c\|^2 \tag{11}$$

onde $\chi$ denota o conjunto de todos os objetos da base e $C$ o conjunto de todos os centroides dos clusters do agrupamento.

O algoritmo inicialmente escolhe de forma aleatória $k$ centroides e agrupa os demais objetos conforme o centroide mais próximo. Uma vez formados os clusters, um novo centroide é definido considerando o centro de massa do cluster, ou seja, a média de todos os objetos do cluster. Com os novos centroides calculados o algoritmo repete a fase de agrupamento dos objetos com base no centroide mais próximo. Esse processo é repetido até que não haja mais mudanças em $C$ (Arthur & Vassilvitskii, 2007).

O Coeficiente de Silhueta (Rousseeuw, 1987) é uma métrica utilizada para analisar um agrupamento de modo a avaliar a compactação dos grupos e a separação

entre eles. A métrica se baseia na distância que um objeto se encontra dos demais clusters, levando em consideração que a distância entre um objeto e um cluster se dá pela média da distância entre o objeto e cada elemento deste cluster. Formalmente, para cada objeto $i$ o coeficiente de silhueta $s(i)$ deste objeto é definido como:

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \quad (12)$$

onde $a(i)$ é a distância entre o objeto e os demais objetos do cluster a que pertence, e $b(i)$ é a distância média entre o objeto e o cluster mais próximo sem considerar seu próprio cluster. O método utilizado neste projeto para se determinar a distância dos objetos é o da distância Euclidiana. Os possíveis valores do coeficiente são $-1 \leq s(i) \leq 1$, de modo que valores próximos a $1$ indicam que a distância do objeto ao cluster ao qual pertence é menor que a distância dele para com qualquer outro cluster; valores próximos a $-1$ indicam o oposto, o objeto está mais próximo de outro cluster; e valores próximos a $0$ indicam que os clusters estão muito próximos, sobrepondo um ao outro, tornando mais difícil a tarefa de atribuir o objeto a um cluster específico. O Coeficiente de Silhueta do agrupamento todo é determinado pela média dos coeficientes de todos os objetos.

### 3.3 PROCESSAMENTO DE TEXTOS

Uma etapa de processamento de textos foi necessária para tratar o atributo *descritivo* da base, estruturando os dados para aplicação do algoritmo de agrupamento. Cada documento é, primariamente, submetido a um processo de *tokenização*, sendo cada palavra correspondente a um *token*. O documento é então representando por uma lista dos *tokens* que o compõe (Hotho, Nürnberger, & Paaß, 2005; Weiss, Indurkhya, Zhang, & Damerau, 2005). O processo para se obter os *tokens* fica sujeito ao idioma em questão, mas, de forma geral, eles são obtidos sabendo quais são os delimitadores de uma palavra. Por exemplo, caracteres em branco (e.g. espaço, nova linha, tabulação) são utilizados como delimitadores, bem como alguns caracteres não textuais e de pontuações (e.g. ( ) < > ! ?). Há caracteres que são ou não delimitadores dependendo do contexto (e.g. - : . , ‘) (Weiss, Indurkhya, Zhang, & Damerau, 2005). Os *tokens* também podem ser obtidos removendo os sinais de pontuação e substituindo caracteres não textuais por espaços em branco (Hotho, Nürnberger, & Paaß, 2005). O conjunto de todos os *tokens* (sem repetição) de todos os documentos da coleção é chamado de *dicionário* (Hotho, Nürnberger, & Paaß, 2005).

Como o número de palavras diferentes em um conjunto de documentos pode ser bastante elevado, ou seja, o dicionário pode conter muitas palavras, faz-se necessário selecionar os *tokens* mais relevantes a uma dada análise. Com o objetivo de reduzir a dimensionalidade do dicionário, alguns métodos para seleção, processamento e eliminação de palavras são aplicados, como *filtragem* e *stemming,* ou *lematização*, (Hotho, Nürnberger, & Paaß, 2005). O processo de *stemming* pode ser útil ou não dependendo da aplicação e seu uso pode acarretar em um custo computacional maior (Weiss, Indurkhya, Zhang, & Damerau, 2005). A filtragem está associada à remoção de palavras do dicionário com base em algum critério. O método mais comum é o de filtragem de *stopwords*, que são palavras que contêm pouca ou nenhuma informação, tais como preposições e artigos (Hotho, Nürnberger, & Paaß, 2005; Weiss, Indurkhya, Zhang, & Damerau, 2005). A lematização visa reduzir os tokens a uma forma padrão, desconsiderando, por exemplo, gênero e plurais. Para a lematização são aplicados métodos para remoção de certos afixos das palavras, propiciando a remoção de plural em substantivo e gerúndio em verbos. Também métodos mais complexos, que levam em consideração o valor morfológico da palavra, são aplicados, reduzindo a palavra a seu radical (Hotho, Nürnberger, & Paaß, 2005; Weiss, Indurkhya, Zhang, & Damerau, 2005).

Outra forma de se reduzir o tamanho do dicionário é através da criação de um dicionário local, que será composto por *tokens*, também chamados de termos, escolhidos com base em algum critério (Weiss, Indurkhya, Zhang, & Damerau, 2005). Algumas formas de fazer a seleção dos termos é utilizando algumas medidas, tal como o *ganho de informação*, a *entropia*, TF-IDF, ou outras medidas baseadas na frequência do termo no documento e/ou no dicionário como um todo, e basear-se em um ranking dos termos que levarem a um melhor desempenho na tarefa de análise (Hotho, Nürnberger, & Paaß, 2005; Weiss, Indurkhya, Zhang, & Damerau, 2005).

Concluída a etapa de processamento textual tem-se início a fase de construção da estrutura de dados que será utilizada na fase de análise, completando o processo de estruturação da base. No caso desse estudo especificamente, a estrutura utilizada será aquela conhecida como *bag of words* (ZHANG, JIN e ZHOU, 2010). Essa estrutura consiste em representar cada documento como um vetor numérico, de modo que cada componente do vetor está associado a um termo do dicionário (Hotho, Nürnberger, & Paaß, 2005).

Existem diversas abordagens para se determinar os valores numéricos, conhecidos como *pesos*, do vetor. Pode-se utilizar uma abordagem binária, associando o número 1 aos termos que existem no documento, e 0 aos que não existem (Hotho, Nürnberger, & Paaß, 2005); uma abordagem ternária, onde se associa 0 ao termo se ele não ocorre no documento, 1 se ele ocorre uma única vez e 2 se ele ocorre duas ou mais vezes (Weiss, Indurkhya, Zhang, & Damerau, 2005). Outra abordagem que associa peso aos termos com base na sua frequência no documento e na coleção de documentos é conhecida como TFIDF, que é o produto da frequência do termo ($tf$), ou seja, a quantidade de vezes que um termo ocorre no documento, pela frequência inversa do documento ($idf$), que é o logaritmo da razão da quantidade de documentos na coleção pela quantidade de documentos em que o termo ocorre (Hotho, Nürnberger, & Paaß, 2005; Weiss, Indurkhya, Zhang, & Damerau, 2005):

$$tf(d,t) = f_{t,d} \tag{13}$$

$$tfidf(D,d,t) = tf(d,t) * idf(D,t) \tag{14}$$

$$idf(D,t) = \log \frac{|D|}{|\{d \in D : t \in d\}|} \tag{15}$$

onde $t$ representa o termo (token) que está sendo considerado, $d$ o documento, $f_{t,d}$ a quantidade de vezes que $t$ aparece em $d$, e $D$ é o conjunto que representa a coleção de documentos.

Essa medida garante que uma palavra que ocorre em muitos documentos receba um peso baixo, e uma palavra que acontece em poucos documentos receba um peso mais alto (Weiss, Indurkhya, Zhang, & Damerau, 2005). Uma vez realizados os cálculos para toda a coleção teremos uma matriz na qual as linhas representarão os documentos e as colunas os termos do dicionário.

### 3.4 RESULTADOS E DISCUSSÃO

Os resultados obtidos estão sumarizados na *Tabela* 1. No primeiro cenário a base de dados é constituída apenas pelos atributos estruturados (*nome da vaga*, *idioma requisitado*, *período de trabalho*, *salário*, *área de atuação*, *modelo de contratação*, *nível de escolaridade*, *benefícios*, *experiência anterior necessária*, *localização*, *gestor*, *número de vagas*), desconsiderando o atributo *descrição*. Observa-se que para os quatro valores de *k* do *k-means* para agrupar a base ({2,4,6,8}), o agrupamento da

base sem seleção de características (**SF**) resulta em um valor baixo de silhueta. A aplicação dos métodos de seleção de características promoveu um aumento significativo no desempenho do *k-means*, cabendo destacar que o algoritmo SUD foi o que obteve uma silhueta mais próxima de 1 em todos os casos. O PCA, no entanto, não apresentou melhoria significativa de desempenho do *k-means* nas situações analisadas neste cenário.

No segundo cenário (**C2**), base de dados considerando apenas o atributo *descrição* estruturado usando um *bag of words*, notou-se que a seleção de características praticamente não influencia o resultado do algoritmo de agrupamento, embora ela reduza a quantidade de atributos de 1774 para 886, reduzindo o custo computacional do processo. Além do desempenho de todos os métodos ter sido similar, ele foi significativamente inferior ao melhor desempenho do cenário 1 (**C1**). O desempenho dos algoritmos foi afetado neste cenário principalmente pelo fato de, dentre os 1933 registros de vagas, apenas 272 possuírem descrição da vaga.

No terceiro e último cenário (**C3**), correspondente a união dos dois subconjuntos de atributos, observa-se que a base de dados sem seleção atributos possui silhuetas baixas nas quatro situações, contidas no intervalo de 0,30 a 0,38. O algoritmo de Mitra apresentou pouca melhoria, de modo que o desempenho do algoritmo de agrupamento não apresentou melhoria com a seleção de atributos. O algoritmo SUD apresentou um desempenho um pouco melhor que o Mitra nas quatro situações, porém foi o algoritmo ACA que obteve o melhor desempenho dentre todos os algoritmos, com Silhuetas entre 0,51 e 0,64. O PCA também não resultou em um conjunto de características que promove melhoria de desempenho quando comparado à ausência de seleção. Como no cenário 2 (**C2**), tivemos o desempenho dos algoritmos afetados neste cenário, principalmente pelo fato de quase 97% das variáveis neste cenário serem originárias do *bag of words* utilizado no cenário 2 (**C2**).

Em resumo, nota-se que o conjunto de atributos sem a *descrição* selecionados pelo método SUD obteve o melhor desempenho global. Entretanto, o uso do atributo *descrição* estruturado com o *bag of words* resultou em desempenho mais estável do *k-means* para todos os métodos de seleção de características, incluindo quando comparado com o uso da base sem a seleção. Observou-se que a utilização de todos os algoritmos combinados deteriorou o desempenho do *k-means*.

*Tabela 1: Desempenho dos métodos de seleção de características aplicados a base de vagas.* **C1***: todos os atributos da base com exceção da descrição;* **C2***: atributo descrição estruturado usando o bag of words;* **C3***: união de todos os atributos;* **SF***: resultados obtidos sem seleção de atributos; D: quantidade original de atributos; d: quantidade de atributos dos subconjuntos selecionados por cada método; k = quantidade de k-médias utilizada pelo k-means para agrupar as bases.*

| | ***k*** | **2** | **4** | **6** | **8** |
|---|---|---|---|---|---|
| **C1 D = 48 d = 23** | **SF** | 0.40 ± 0.00 | 0.33 ± 0.00 | 0.36 ± 0.00 | 0.37 ± 0.00 |
| | **Mitra** | 0.73 ± 0.00 | 0.78 ± 0.00 | 0.84 ± 0.00 | 0.88 ± 0.00 |
| | **SUD** | 0.94 ± 0.00 | 0.96 ± 0.00 | 0.96 ± 0.00 | 0.98 ± 0.00 |
| | **ACA** | 0.45 ± 0.00 | 0.54 ± 0.00 | 0.64 ± 0.00 | 0.72 ± 0.00 |
| | **PCA** | 0.41 ± 0.00 | 0.34 ± 0.00 | 0.36 ± 0.00 | 0.40 ± 0.00 |
| **C2 D = 1774 d = 886** | **SF** | 0.81 ± 0.00 | 0.84 ± 0.00 | 0.84 ± 0.00 | 0.85 ± 0.00 |
| | **Mitra** | 0.82 ± 0.00 | 0.82 ± 0.00 | 0.82 ± 0.00 | 0.82 ± 0.00 |
| | **SUD** | 0.82 ± 0.00 | 0.85 ± 0.00 | 0.86 ± 0.00 | 0.85 ± 0.00 |
| | **ACA** | 0.81 ± 0.00 | 0.81 ± 0.00 | 0.82 ± 0.00 | 0.83 ± 0.00 |
| | **PCA** | 0.81 ± 0.00 | 0.84 ± 0.00 | 0.84 ± 0.00 | 0.85 ± 0.00 |
| **C3 D = 1822 d = 910** | **SF** | 0.38 ± 0.00 | 0.30 ± 0.00 | 0.33 ± 0.00 | 0.35 ± 0.00 |
| | **Mitra** | 0.39 ± 0.00 | 0.32 ± 0.00 | 0.34 ± 0.00 | 0.37 ± 0.00 |
| | **SUD** | 0.41 ± 0.00 | 0.36 ± 0.00 | 0.38 ± 0.00 | 0.42 ± 0.00 |
| | **ACA** | 0.51 ± 0.00 | 0.55 ± 0.00 | 0.60 ± 0.00 | 0.63 ± 0.00 |
| | **PCA** | 0.38 ± 0.00 | 0.30 ± 0.00 | 0.33 ± 0.00 | 0.35 ± 0.00 |

## 4. CONSIDERAÇÕES FINAIS

Este trabalho teve por objetivo investigar a influência da seleção de características na tarefa de agrupamento de dados de uma base de dados de recrutamento online. Foram utilizados os algoritmos de Mitra, SUD e ACA para realizar a seleção dos atributos e esses foram analisados e comparados entre si e com o PCA. Os algoritmos de seleção de características foram avaliados em três cenários distintos: considerando todo os atributos menos a parte descritiva textual da base; considerando apenas o atributo *descrição* da base; considerando ambos subconjuntos de atributos.

Em nenhum dos cenários o PCA se mostrou superior aos algoritmos de seleção testados, tendo, pelo contrário, apresentado pouco ou nenhuma melhoria em cada cenário analisado. Dentre os algoritmos de seleção foi possível observar que os três apresentaram melhor desempenho no cenário de atributos estruturados, em contrapartida apresentaram pouca ou nenhuma melhoria no cenário não estruturado. No cenário misto o desempenho também foi afetado, sendo este superior ao do segundo cenário e inferior ao do primeiro.

## 5. AGRADECIMENTOS

Os autores agradecem ao CNPq, Capes, Fapesp e MackPesquisa pelo apoio financeiro. Também agradecemos à Intel pelo apoio ao LCoN como um Centro de Excelência em Inteligência Artificial.

## 6. REFERÊNCIAS

Arthur, D., & Vassilvitskii, S. (2007). k-means++: The advantages of careful seeding. *Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms* (pp. 1027-1035). Society for Industrial and Applied Mathematics.

Au, W. H., Chan, K. C., Wong, A. K., & Wang, Y. (2005). Attribute Clustering for Grouping, Selection, and. *IEEE/ACM transactions on computational biology and bioinformatics, 2*(2), pp. 83-101.

Bharti, K. K., & Singh, P. k. (2014). A Survey on Filter Techniques for Feature Selection in Text Mining. *Proceedings of the Second International Conference on Soft Computing for Problem Solving (SocProS 2012), December 28-30, 2012* (pp. 1545-1559). New Delhi: Springer.

Capelli, P. (2001). Making the most of on-line recruiting. *Harvard business review, 79*(3), pp. 139-148.

Dash, M., Liu, H., & Yao, J. (1997). Dimensionality Reduction of Unsupervised Data. *Tools with Artificial Intelligence, 1997. Proceedings., Ninth IEEE International Conference on* (pp. 532-539). IEEE.

Divya, P., & Kumar, G. S. (January de 2015). Study on Feature Selection Methods for Text Mining. *International Journal of Advanced Research Trends in Engineering and Technology, 2*(1).

Faliagka, E., Tsakalidis, A., & Tzimas, G. (2012). An integrated e-recruitment system for automated personality mining and applicant ranking. *Internet Research, 22*(5), pp. 551-568.

Ferreira, A. J., & Figueired, M. A. (2012). Efficient feature selection filters for high-dimensional data. *Pattern Recognition Letters, 33*(13), pp. 1794–1804.

Gupta, V., & Lehal, G. (2009). A survey of text mining techniques and applications. *Journal of emerging technologies in web intelligence, 1*(1), pp. 60-76.

Hotho, A., Nürnberger, A., & Paaß, G. (2005). A brief survey of text mining. *Ldv Forum, 20*(1), pp. 19-62.

Jain, A., & Zongker, D. (1997). Feature Selection: Evaluation, Application, and Small Sample Performance. *IEEE transactions on pattern analysis and machine intelligence, 19*(2), pp. 153-158.

Liu, H., & Yu, L. (2005). Toward integrating feature selection algorithms for classification and clustering. *IEEE Transactions on knowledge and data engineering, 17*(4), pp. 491-502.

Liu, L., Kang, J., Yu, J., & Wang, Z. (2005). A comparative study on unsupervised feature selection methods for text clustering. *Proceedings of Natural Language Processing and Knowledge Engineering*, (pp. 597–601).

Liu, T., Liu, S., Chen, Z., & Ma, W.-Y. (2003). An Evaluation on Feature Selection for Text Clustering. *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, (pp. 488-495).

Mitra, P., Murthy, C. A., & Pal, S. K. (2002). Unsupervised feature selection using feature similarity. *IEEE transactions on pattern analysis and machine intelligence, 24*(3), pp. 301-312.

Molina, L., Belanche, L., & Nebot, À. (2002). Feature Selection Algorithms: A Survey and Experimental Evaluation. (pp. 306-313). IEEE.

Rousseeuw, P. J. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics, 20*, pp. 53-65.

Weiss, S. M., Indurkhya, N., Zhang, T., & Damerau, F. J. (2005). *Text Mining: Predictive Methods for Analyzing Unstructured Information.* New York: Springer Science & Business Media, Inc.

Yoon Kin Tong, D. (2009). A study of e-recruitment technology adoption in Malaysia. *Industrial Management & Data Systems, 109*(2), pp. 281-300.

**Contato**s: joel.junior.95@gmail.com e lnunes@mackenzie.br