SELEÇÃO DE CARACTERÍSTICAS GENÉTICA COM MUTAÇÃO INDIVIDUAL POR BIT BASEADA EM PEARSON E CLUSTERIZAÇÃO DE VARIÁVEIS UTILIZANDO MEDIDAS DE DISSIMILARIDADE

Adriano Gomes Sabino de Araujo

Tese de Doutorado apresentada ao Programa de Pós-graduação em Engenharia de Sistemas e Computação, COPPE, da Universidade Federal do Rio de Janeiro, como parte dos requisitos necessários à obtenção do título de Doutor em Engenharia de Sistemas e Computação.

Orientador: Geraldo Zimbrão da Silva

Rio de Janeiro

Agosto de 2019

SELEÇÃO DE CARACTERÍSTICAS GENÉTICA COM MUTAÇÃO INDIVIDUAL POR BIT BASEADA EM PEARSON E CLUSTERIZAÇÃO DE VARIÁVEIS UTILIZANDO MEDIDAS DE DISSIMILARIDADE

Adriano Gomes Sabino de Araujo

TESE SUBMETIDA AO CORPO DOCENTE DO INSTITUTO ALBERTO LUIZ COIMBRA DE PÓS-GRADUAÇÃO E PESQUISA DE ENGENHARIA (COPPE) DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE DOUTOR EM CIÊNCIAS EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

Examinada por:

_____

Prof. Geraldo Zimbrão da Silva, D.Sc.

_____

Prof. Ruy Luiz Milidiú, Ph.D.

_____

Prof. Leandro Guimarães Marques Alvim, D.Sc.

_____

Prof. Geraldo Bonorino Xexéo, D.Sc.

_____

Prof. Luiz Pereira Calôba, Ph.D.

RIO DE JANEIRO, RJ - BRASIL
AGOSTO DE 2019

Araujo, Adriano Gomes Sabino de

Seleção de características genética com mutação individual por bit baseada em Pearson e clusterização de variáveis utilizando medidas de dissimilaridade/Adriano Gomes Sabino de Araújo. – Rio de Janeiro: UFRJ/COPPE, 2019.

XI, 97 p.: il.; 29,7 cm.

Orientador: Geraldo Zimbrão da Silva

Tese (doutorado) – UFRJ/ COPPE/ Programa de Engenharia de Sistemas e Computação, 2019.

Referências Bibliográficas: p. 84-97.

1. Seleção de Características. 2. Agrupamento de Características. 3. Algoritmos Genéticos. 4. Coeficiente de correlação de Pearson. I. Silva, Geraldo Zimbrão da. II. Universidade Federal do Rio de Janeiro, COPPE, Programa de Engenharia de Sistemas e Computação. III. Título.

Agradecimentos

Agradeço primeiramente a Deus, aquele que está comigo em todos os momentos e dá-me sabedoria para alcançar todos os meus objetivos.

Agradeço à minha família, a qual sempre me apoiou e contribuiu de todas as formas para que eu chegasse aonde eu cheguei.

Agradeço a todos que contribuíram de alguma forma para a conclusão desta tese.

Resumo da Tese apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Doutor em Ciências (D.Sc.)

SELEÇÃO DE CARACTERÍSTICAS GENÉTICA COM MUTAÇÃO INDIVIDUAL POR BIT BASEADA EM PEARSON E CLUSTERIZAÇÃO DE VARIÁVEIS UTILIZANDO MEDIDAS DE DISSIMILARIDADE

Adriano Gomes Sabino de Araújo

Agosto/2019

Orientador: Geraldo Zimbrão da Silva

Programa: Engenharia de Sistemas e Computação

Reduzir a quantidade de dimensões de um problema possibilita não só reduzir o tempo de processamento da técnica de aprendizado utilizada como também melhorar o desempenho da mesma. Seleção de Características e Agrupamento de Variáveis são duas importantes formas de realizar tal redução. A primeira consiste na busca do conjunto ideal de características para solucionar determinado problema, ou seja, aquele que possibilita alcançar o melhor resultado quando utilizando um preditor. A segunda tem como intuito agrupar dimensões a fim de usar os agrupamentos para gerar o novo conjunto de entrada do problema. Este trabalho introduz um algoritmo genético para seleção de características que se diferencia de outros nos seguintes aspectos: (1) pela taxa de mutação individual por bit e proporcional ao coeficiente de correlação de Pearson e (2) pela geração da população inicial baseada no mesmo coeficiente. Além disso, apresenta um algoritmo de agrupamento de características que, diferente de outros trabalhos da literatura, uni dimensões mais dissimilares quanto possível. Experimentos foram executados com ambos os algoritmos e os resultados obtidos foram promissores. Executados individualmente tiveram bons resultados e, quando executados um após o outro, resultaram em melhores desempenhos. Os experimentos foram realizados sobre diferentes bases de dados, destacando-se como principal a base de classificação de textos Reuters 21.578. O melhor resultado obtido em tal base foi com Precision (P) de 0,9890, Recall (R) de 0,9815 e F1 de 0,9852. O mesmo foi comparado com três outros trabalhos e foi superior ao melhor deles ([UĞUZ, 2011]).

Abstract of Thesis presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Doctor of Science (D.Sc.)

GENETIC FEATURE SELECTION WITH PEARSON INDIVIDUAL MUTATION RATE AND FEATURE CLUSTERING BASED ON DISSIMILARITY MEASURES

Adriano Gomes Sabino de Araujo

August/2019

Advisor: Geraldo Zimbrão da Silva

Department: System Engineering and Computer Science

Reducing the number of dimensions of a problem allows not only to reduce the processing time of the used learning technique but also to improve its performance. Feature Selection and Feature Clustering are two important ways to accomplish such a reduction. The first one is the search for the ideal feature set to solve a problem, that is, the one that makes it possible to reach the best result when using a predictor. The second one is intended to group dimensions in order to use the clusters to generate the new problem input set. This work introduces a genetic algorithm for feature selection and differs from others in the following aspects: (1) individual mutation rate per bit and proportional to the Pearson correlation coefficient and (2) initial population generation based on the same coefficient. In addition, it presents a feature clustering algorithm that, unlike other works in the literature, merge more dissimilar dimensions. Experiments were performed with both algorithms and the results obtained were promising. Individually performed well and, when performed one after another, resulted in better performances. The experiments were carried out on different databases, highlighting as main the text classification database Reuters 21,578. The best result was with Precision (P) of 0.9890, Recall (R) of 0.9815 and F1 of 0.9852. On Reuters, the result was compared with three other papers and was superior to the best of them ([UĞUZ, 2011]).

Sumário

Figuras

Tabelas

Capítulo 1. Introdução

1.1 Motivação

O excesso de variáveis ou a presença de características irrelevantes em um determinado problema pode reduzir o desempenho da predição. Com base nisso, técnicas que contribuem para eliminá-las ou diminuí-las levam a uma melhora do resultado.

Duas importantes formas de alcançar os objetivos citados são através da seleção de características e da clusterização de variáveis. O objetivo da primeira é selecionar um subconjunto de dimensões que melhor descrevem a saída do problema ([FORMAN, 2003], [GUYON & ELISSEEFF, 2003], [CHANDRASHEKAR & SAHIN, 2014]). Já o objetivo da segunda é utilizar técnicas de clusterização para agrupar dimensões e, a partir disto, gerar um novo conjunto de entrada com menor dimensionalidade ([BAKER & MCCALLUM, 1998], [SLONIM & TISHBY, 2001], [DHILLON et al., 2003], [KRIER et al., 2007]).

A seleção de características é uma forma interessante de diminuir o número de variáveis do problema, uma vez que o objetivo é tentar alcançar o conjunto de características ótimo, ou seja, aquele que consegue prever com maior precisão as saídas do problema ([PUDIL et al., 1994]). A única questão é que é inviável testar todas as combinações possíveis, uma vez que, para dados com muitas dimensões, este número torna-se muito grande. De uma forma geral, as abordagens focam em tentar testar as combinações corretas, ou seja, aquelas que são mais prováveis de preverem bem as saídas do problema ([PUDIL et al., 1994]).

Nas técnicas de clusterização de variáveis, a questão é outra. Em vez de apenas selecionar as dimensões, são criados grupos de características, onde cada um é relativo a uma variável do novo conjunto da entrada ([BAKER & MCCALLUM, 1998], [SLONIM & TISHBY, 2001], [DHILLON et al., 2003], [KRIER et al., 2007]). De forma geral, o problema consiste em como agrupar as características e, portanto, criar estes grupos. É interessante notar que, nestas técnicas, dados não são perdidos, pois nenhuma dimensão é descartada e sim utilizada, de alguma forma, para dar origem a uma nova dimensão.

1.2 Objetivo

O objetivo deste trabalho é introduzir dois novos algoritmos de seleção de características e agrupamento de variáveis. O primeiro é um algoritmo genético com geração da população inicial e taxa de mutação baseados no coeficiente de correlação de Pearson. Ele visa selecionar apenas as características relevantes do conjunto de entrada. Este foi proposto por dois motivos: (1) os algoritmos genéticos são muito utilizados em problemas de seleção de características ([OH et al., 2004], [FROHLICH et al., 2003], [LEARDI & GONZALEZ, 1998], [YANG & HONAVAR, 1998], [SIEDLECKI & SKLANSKY, 1989]) e facilitam a busca por ótimos locais e globais do problema sem a necessidade de testar todas as soluções possíveis; (2) o coeficiente de correlação de Pearson mede a associação entre variáveis de forma simples e é utilizado em diversos trabalhos na literatura ([HUANG, 2008], [MONEDERO et al., 2012]).

O segundo algoritmo, o qual é aplicado logo após o primeiro, tem como finalidade clusterizar variáveis utilizando medidas de dissimilaridade e o coeficiente de correlação de Pearson. Ele atua através da formação de grupos de características, os quais são usados como as novas dimensões do conjunto de entrada no lugar das características originais. O algoritmo de clusterização foi proposto, pois uma série de trabalhos tem demonstrado que clusterizar variáveis tem levado a bons resultados ([ZHAI et al., 2011], [BONDELL & REICH, 2008], [BUTTERWORYH et al., 2005], [DETTLING & B¨UHLMANN, 2004], [HASTIE et al., 2001]). Além disso, o algoritmo minimiza a perda de informação relevante, uma vez que as novas dimensões resultantes do algoritmo são nada mais do que uma simples combinação das variáveis originais.

1.3 Contribuição para a Literatura

Este trabalho contribui com a literatura ao introduzir dois novos algoritmos nas áreas de seleção de características e agrupamento de variáveis. Contribuições resultantes destes algoritmos são:

(1) Introdução de uma taxa de mutação para o algoritmo genético de seleção de características apresentando como diferencial o fato de ser individual por bit e proporcional ao coeficiente de correlação de Pearson (seção 3.3);
(2) Introdução de uma nova forma de geração da população inicial baseada também no coeficiente citado (seção 3.2);

(3) Algoritmo de Clusterização de Variáveis utilizando-se de medidas de similaridade existentes na literatura e permitindo a união de uma ou mais dimensões em grupos (capítulo 4);
(4) Diminuição da quantidade de variáveis de entrada do problema associado a uma melhora no desempenho da predição ao usar os algoritmos separadamente ou em conjunto (seção 6.5).
(5) Grande redução de tempo de treinamento dos classificadores, uma vez que a quantidade de características finais do processo (após a execução dos algoritmos introduzidos) é uma porcentagem muito pequena da quantidade inicial (seção 6.5).

1.4 Resultados Obtidos

Os resultados obtidos com as aplicações individuais e em conjunto dos dois algoritmos propostos foram promissores. A aplicação do algoritmo de seleção de características desenvolvido, por si só, já trouxe melhoras quando aplicado em conjunto com os algoritmos de aprendizado. A inclusão do algoritmo de clusterização de variáveis após a seleção contribuiu para aumentar ainda mais o desempenho obtido.

Os resultados foram computados sobre a base Reuters 21.578 (seção 6.5.3.2) e sobre uma base de avaliação de filmes usada por PANG & LEE (2004) - seção 6.5.3.1. Em relação à primeira, este trabalho foi comparado com outros três presentes na literatura. O desempenho obtido foi superior aos dos trabalhos comparados tanto em desempenho como na capacidade de diminuir a quantidade de características do problema. Os melhores resultados obtidos foram com precision de 0.9890, recall de 0.9815 e F1 de 0.9852 contra os do melhor trabalho comparado ([UĞUZ, 2011]) com precision de 0.9817, recall de 0.9752 e F1 de 0.9784. Em relação à redução, esta foi de 8.256 (100%) para 31 (0,38%) características contra uma redução de UĞUZ (2011) de 7.542 (100%) para 169 (2,24%). Em relação à base de filmes, o melhor desempenho foi ligeiramente, mas não significantemente, superior (87,39% contra 87,20%). Destacou-se como principais contribuições da aplicação dos dois algoritmos em conjunto a grande capacidade de redução do número de variáveis do problema, melhora no desempenho da predição e redução do tempo de execução.

1.5 Organização do Trabalho

Este trabalho está organizado da seguinte forma: no capítulo 2 é apresentada a revisão literária realizada durante este trabalho. Essa revisão fornece a base necessária para que o leitor tenha uma visão geral sobre as técnicas utilizadas e o contexto dos problemas propostos. No capítulo 3 é apresentado um algoritmo de seleção de características utilizando algoritmos genéticos e o coeficiente de correlação de Pearson. No capítulo 4, um algoritmo de clusterização de variáveis que utiliza medidas de dissimilaridade e o coeficiente de correlação de Pearson é introduzido. No capítulo 5 são apresentadas todas as etapas propostas neste trabalho, incluindo os dois algoritmos descritos nos capítulos 3 e 4. No capítulo 6, é descrito o estudo de caso utilizando as etapas propostas no capítulo 5. Nele, uma base de dados de avaliação de filmes, uma de classificação de textos (Reuters 21.578) e outras notícias relacionadas à ações na bolsa de valores de São Paulo (Bovespa) são utilizadas. Além disso, os algoritmos são comparados com a Análise de Componentes Principais (PCA) para avaliar o potencial de redução da dimensionalidade e o desempenho. No capítulo 7, o algoritmo de Clusterização de Variáveis é generalizado a fim de possibilitar com que o mesmo seja utilizado com dados não textuais. E, por fim, no capítulo 8, são apresentadas as conclusões.

Capítulo 2. Revisão Literária

Este capítulo visa descrever os principais conceitos necessários ao entendimento deste trabalho. A seção 2.1 trata do problema de seleção de características, a seção 2.2 descreve o problema de clusterização de variáveis e a seção 2.3 introduz os algoritmos genéticos.

2.1 Seleção de Características

Esta seção está organizada da seguinte maneira: na subseção 2.1.1 é realizada a definição do conceito de seleção de características, na seção 2.1.2 é apresentada uma formalização matemática para o problema e, na seção 2.1.3, são apresentados os principais métodos encontrados na literatura.

2.1.1 Definição

KIRA et al. (1992) dizem que a representação de dados em estado natural frequentemente usa muitas características e somente algumas são relevantes para prever o alvo. Além disso, os autores afirmam que as características irrelevantes degradam o desempenho dos preditores em velocidade e acurácia. Sobre o conceito de seleção de características, afirmam que este é o problema de escolher um pequeno subconjunto de características que idealmente é necessário e suficiente para descrever o conceito alvo. Sobre o mesmo assunto, GUYON & ELISSEEFF (2003) afirmam que existem muitos benefícios da seleção de características: facilitar a visualização e entendimento dos dados, reduzir os requisitos de medição e armazenamento, diminuir o tempo de treinamento e utilização e combater a maldição da dimensionalidade para melhorar a performance da predição.

A partir dos benefícios citados por GUYON & ISABELLE (2003) e da definição de KIRA et al. (1992), pode-se retratar a seleção de características como o problema de escolher um subconjunto das variáveis de entrada que melhor descrevem as variáveis alvo a fim de atingir um ou mais dos seguintes objetivos: melhorar o desempenho, tempo de treinamento e de operação de um preditor, facilitar a visualização dos dados e diminuir o tamanho dos dados a serem armazenados e manipulados. É importante deixar claro que nem todas as variáveis que possuem relação com o conceito alvo (a saída) estarão presentes no subconjunto selecionado,

uma vez que poderão existir características redundantes (aquelas em que toda ou grande parte da sua informação estão embutidas em outras).

2.1.2 Formalização Matemática

Com base na definição apresentada na seção anterior, podemos formalizar o problema. Para isso, considere:

- S como o conjunto de treinamento com n elementos.
- $F = \{f_1,\ f_2, \dots, f_p\}$ como o conjunto inicial de características.
- $X = \{x_1,\ x_2, \dots, x_p\}$ como um elemento do conjunto de treinamento, onde x_i denota o valor da característica f_i.

Considerando as informações acima, técnicas típicas de seleção de características precisam de uma função $J(E,\ S)$ que avalia o subconjunto E de F. Dados dois subconjuntos E_1 e E_2 de F, diz-se que E_1 é melhor que E_2 se $J(E_1,\ S) > J(E_2,\ S)$. Então, de forma geral, as técnicas se resumem a definir quais os subconjuntos serão testados e qual será a função de avaliação $J(E,\ S)$. Repare que, para um p (número de características) grande, como é o caso de muitos problemas do mundo real, não é viável computacionalmente testar todos os subconjuntos dois a dois e, por isso, precisa-se definir uma estratégia para decidir quais serão os subconjuntos testados.

2.1.3 Principais Trabalhos

Diversos trabalhos retratam a questão da seleção de características. CHANDRASHEKAR & SAHIN (2014) dividiram os métodos em três classes: filter, wrapper e embedded. O primeiro refere-se àqueles que ranqueiam as características e selecionam as top k a fim de considerá-las na tarefa de predição. Os métodos wrapper consideram como critério da seleção o desempenho do preditor, ou seja, o conjunto de características que permitem ao método de previsão obter o melhor desempenho visa ser selecionado. Por fim, os métodos embedded incluem o processo de seleção de características como parte do processo de treinamento.

No âmbito das abordagens filter, métricas de relevância são utilizadas para ordenar e ranquear as características. As técnicas que seguem esta abordagem diferem apenas na métrica utilizada e são muitas as que são utilizadas na literatura. Uma delas é o coeficiente de correlação de Pearson ([BATTITI, 1994], [HALL, 2000], [GUYON & ELISSEEFF, 2003], [BIESIADA

& DUCH, 2007], [LABANI et al., 2018], [RANJBAR & JAMALI, 2019]), o qual é calculado com base na equação 1, varia entre -1 e 1 e mede a correlação entre duas variáveis aleatórias (-1 indica correlação linear negativa perfeita, 1 indica correlação linear positiva perfeita e 0 indica que as variáveis são descorrelacionadas linearmente).

$$\rho(X,Y) = \frac{cov(X,Y)}{\sqrt{var(X) \cdot var(Y)}}$$

Equação 1: Normalização dos coeficientes de correlações de Pearson

Outra medida muito utilizada é a informação mútua – MI ([BATTITI, 1994], [LAZAR et al., 2012], ([GUYON & ELISSEEFF, 2003], [BENNASAR et al., 2015], [LIN et al., 2016], [VINH et al., 2016], [LI et al., 2017], [ALJAWARNEH et al, 2018], [RAHMANINIA & MORADI, 2018], [GAO et al., 2018]), a qual é baseada na entropia e na entropia condicional. Se o valor da MI for zero, as variáveis X e Y são consideradas independentes e, se for maior que zero, são consideradas dependentes. A medida de informação mútua é apresentada na equação 2, a entropia na equação 3 a entropia condicional na equação 4.

$$MI(Y,X) = H(Y) - H(Y|X)$$

Equação 2: Informação Mútua (MI)

$$H(Y) = -\sum_{y} p(y)\log(p(y))$$

Equação 3: Entropia

$$H(Y|X) = -\sum_{x}\sum_{y} p(x,y)\log(p(y|x))$$

Equação 4: Entropia Condicional

Diversas outras métricas são utilizadas na literatura. FREEMAN et al. (2015) faz uma avaliação de diferentes medidas usando como técnicas de aprendizado o SVM e KNN. Entre as métricas testadas estão: informação mútua, informação mútua condicional, medidas de correlação e Fisher. FORMAN (2003) também faz uma comparação entre diferentes métricas:

ganho de informação, qui-quadrado, razão de chances, razão de probabilidades, frequência de documentos, medida F1, numerador da razão de chances, duas medidas de acurácia, poder e separação bi-normal.

Ainda nas abordagens filters, LI et al. (2018) mencionam sobre os critérios para ranquear as variáveis e citam alguns: a habilidade discriminativa das características para separar amostras ([Kira and Rendell 1992; Robnik-Sikonja and Kononenko 2003; Yang et al. 2011; ˇ Du et al. 2013; Tang et al. 2014b]); a correlação ([Koller and Sahami 1995; Guyon and Elisseeff 2003]), a informação mútua ([Yu and Liu 2003; Peng et al. 2005; Nguyen et al. 2014; Shishkin et al. 2016; Gao et al. 2016]), a habilidade de preservar a estrutura principal dos dados ([He et al. 2005; Zhao and Liu 2007; Gu et al. 2011b; Jiang and Ren 2011]) e a habilidade de reconstruir os dados originais ([Masaeli et al. 2010; Farahat et al. 2011; Li et al. 2017b]).

Métodos wrapper visam encontrar o melhor subconjunto de características utilizando como função objetivo o desempenho do preditor. De acordo com XUE et al. (2016), Algoritmo Genético é a primeira técnica amplamente aplicada para seleção de características e a maioria dos trabalhos que utilizam AG's tem as suas abordagens classificadas como wrapper. Nesta linha, diversos trabalhos utilizam AG's para seleção de características wrapper ([SIEDLECKI & SKLANSKY, 1989], [LEARDI et al., 1992], [VAFAIE & JONG, 1998], [YANG & HONAVAR, 1998], [Demirekler & Haydar, 1999], [SMITH & BULL, 2005], [Umamaheswari et al., 2006], [GHEYAS & SMITH, 2010], [JEONG et al., 2014], [KAMATH et al., 2014], [ORESKI & ORESKI, 2014], [GHAMISI & BENEDIKTSSON, 2015], [GHAREB et al., 2016]. A próxima seção trata os algoritmos genéticos de forma particular, uma vez que é utilizado neste trabalho a fim de selecionar características.

Ainda sobre os métodos wrapper, as técnicas Particle Swarm Optimization (PSO) e Ant Colony Optimization (ACO) são muito utilizadas ([GHAMISI & BENEDIKTSSON, 2015], [KASHEF & NEZAMABADI-POUR, 2015], [MORADI & ROSTAMI, 2015], [FONG et al., 2016], [XUE et al., 2016], [MISTRY et al., 2017], [ZANG et al., 2017]). Segundo XUE et al. (2016), assim como os AG's, estas são técnicas de computação evolucionária. De acordo com MARINI e WALCZAC (2015), o PSO é inspirado no comportamento de animais sociais. O conjunto de soluções candidatas é definido como um enxame de partículas que podem fluir através do espaço de parâmetros. As trajetórias são baseadas no seu próprio desempenho ou de seus melhores vizinhos. Sobre o ACO, DORIGO e STÜTZLE (2019) afirmam que esta é uma meta-heurística que é inspirada no feromônio deixado nas trilhas das formigas e que segue o comportamento de algumas espécies.

Ainda no contexto dos métodos wrapper, algoritmos sequenciais são abordados na literatura ([PUDIL et al., 1994], [PUDIL et al., 1999], [REUNANEN, 2003], [SUN et al., 2006], [NAKARIYAKUL & CASASENT, 2009], [PARK & KIM, 2015], [MAYER et al., 2017], [HOMSAPAYA & SORNIL, 2017]). Seguindo esta linha, o algoritmo "Sequential Feature Selection" (SFS) inicia com um conjunto de características vazio e, a cada passo, adiciona o elemento que fornece o maior valor para a função objetivo ([PUDIL et al., 1994], [REUNANEN, 2003]). Para entender o SFS, considere a notação adotada na formalização matemática da seção anterior (2.1.2). Além disso, seja F'_i o conjunto de características selecionadas na iteração i. O algoritmo segue os passos abaixo:

***Passo 1**: Inicialização.*

$F'_0 = \{\}$, *i = 0*

***Passo 2**: Selecionar a melhor característica a ser adicionada à F'_i para formar F'_{i+1}.*

$$F'_{i+1} = F'_i + \{f_j\}, \text{ tal que } \begin{cases} f_j \in (F - F'_i) \\ J(F'_i + \{f_j\}) \geq J(F'_i + \{f_k\}) \; \forall f_k \in (F - F'_i) \end{cases}$$

***Passo 3**: Parada ou próxima iteração.*

Se $n(F'_{i+1}) = n(F)$, *retorne* $F^* = \{F'_j | \forall F'_k [J(F'_j) \geq J(F'_k)]\}$.

Caso contrário, faça $i = i + 1$ *e retorne ao passo 2.*

Algoritmo 1: Algoritmo Sequential Feature Selection (SFS)

No passo 1, o conjunto de características selecionadas na iteração de número 0 é inicializado com o conjunto vazio. No passo 2, o conjunto de características selecionadas para a próxima iteração (F'_{i+1}) é criado. Ele é formado pela adição no conjunto anterior (F'_i) de uma característica que ainda não tinha sido inserida ($f_j \in (F - F'_i)$). A escolha desta é feita de forma que a característica escolhida forneça o maior valor de avaliação para F'_{i+1}, ou seja, $J(F'_i + \{f_j\}) \geq J(F'_i + \{f_k\}) \; \forall f_k \in (F - F'_i)$.

O algoritmo "Sequential Backward Selection" (SBS) é similar ao SFS, mas começa com o conjunto cheio e, a cada passo, remove o elemento que provoca a menor redução na função objetivo. Outro algoritmo mais flexível é o "Sequential Floating Foward Selection" (SFFS). Semelhantemente ao SFS, ele começa com o conjunto vazio. Em cada iteração são executados

dois passos. O primeiro adiciona um elemento ao conjunto e o segundo remove um se a exclusão melhorar o valor da função de avaliação. Outros algoritmos sequenciais são o "Sequential random k-nearest neighbor feature selection" ([PARK & KIM, 2015]), "Adaptive Sequential Forward Floating Selection" ([PUDIL et al., 1999], [SUN et al., 2006]) e o método de busca "Plus-L-Minus-r" ([PUDIL et al., 1999], [NAKARIYAKUL & CASASENT, 2009]).

Passando agora para os métodos embedded, eles incorporam a seleção de características como parte do processo de treinamento ([CHANDRASHEKAR & SAHIN, 2014], [LI et al., 2018]). BATTITI (1994) criou uma algoritmo de busca gulosa que considera a informação mútua da variável candidata a ser selecionada com a saída e a informação mútua com as variáveis anteriormente selecionadas. Dessa forma, o algoritmo considera não só a iteração com a saída, mas como também com as outras variáveis. KWAK & CHOI (2002) melhoraram esse algoritmo com uma estimativa da informação mútua usando o método "Parzen Window". PENG et al. (2005) também utilizaram a informação mútua em uma abordagem em dois estágios.

Ainda nas abordagens embedded, alguns estudos se baseiam na remoção de características e utilizam pesos associados a um classificador para ranquear características. GUYON et al. (2002) usam pesos e o SVM para construir um algoritmo de eliminação de características recursiva (SVM RFE). Na mesma linha, SETIONO (1997) e ROMERO & SOPENA (2008) usam pesos e uma rede neural perceptron multicamadas. TIBSHIRANI (1996), ZOU & HASTIE (2005), TIBSHIRANI et al. (2005) e BONDELL & REICH (2008) utilizam os pesos da regressão linear para eliminar características.

No contexto da regressão, o algoritmo mais conhecido é o Lasso ([TIBSHIRANI, 1996]), o qual transforma o problema de seleção de características em um problema de minimização da soma do quadrado do erro residual sujeito a uma determinada restrição, conforme equação 5.

$$(\hat{\alpha}, \hat{\beta}) = \arg\min\left\{ \sum_{i=1}^{N} \left(y_i - \alpha - \sum_j \beta_j x_{ij} \right)^2 \right\} \qquad \text{subject to} \sum_j |\beta_j| \leqslant t.$$

Equação 5: Função a ser minimizada no Lasso

Na equação, y_i representa a i-ésima resposta, x_{ij} é a j-ésima variável preditora da i-ésima entrada, t é o parâmetro de afinação e $\hat{\alpha}$ e $\hat{\beta}$ são as estimativas.

O Lasso estimula a esparsidade, ou seja, que alguns coeficientes assumam valor zero e os outros assumam valores próximos de zero (dimensões associadas a coeficientes zerados podem ser eliminadas).

Outros algoritmos foram evoluções do Lasso, como são os casos: (1) do Elastic Net ([ZOU & HASTIE, 2005]), o qual introduz um regulador que consiste no quadrado da norma l2; (2) do Fused Lasso ([TIBSHIRANI et al., 2005]), o qual introduz um regulador que é o somatório do módulo da diferença entre dois coeficientes consecutivos; (3) do OSCAR ([BONDELL & REICH, 2008]), que introduz um regulador que é a soma do máximo entre dois coeficientes dois a dois.

Além dos diversos trabalhos citados, é importante mencionar aqueles que utilizam do Deep Learning para seleção de características. De acordo com [GOODFELLOW et. al, 2016], esta área de conhecimento permite a modelos computacionais aprender representações dos dados com múltiplos níveis de abstração e, além disso, tem melhorado o estado da arte de diversos segmentos, tais como reconhecimento de fala, reconhecimento de objetos visuais, detecção de objetos e muitos outros domínios. Além disso, os autores afirmam que Deep Learning permite descobrir a estrutura intríseca de grandes base de dados usando o algoritmo backpropagation. Sobre o assunto, mais especificamente sobre Rede Neurais Convolucionais, [ZHANG et al., 2015] afirmam que são úteis para extrair informações de sinais brutos, indo desde aplicações de visão computacional até reconhecimento de fala e outros.

Nesta área, [SEMWAL et. al, 2017] utilizam uma Rede Neural Profunda para seleção de características a fim de solucionar o problema de recuperação de impulsos em robôs, uma vez que tal tarefa é facilmente realizada por humanos, mas é uma tarefa difícil para os humanóides. [PORIA et al, 2015] aplicaram uma Rede Neural Convolucional Profunda (RNCP) para extrair características de textos curtos a fim de realizar análise de sentimento. O autor cita que uma desvantagem da RNCP é que ela converge para mínimos locais, dado que utiliza o algoritmo Backpropagation. Por este motivo, ela é utilizada apenas a fim de extrair características e outro algoritmo de aprendizado (SVM) é usado para realizar a análise de sentimento.

Em [ZOU et al., 2015], os autores usaram uma Rede de Crença Profunda (do inglês Deep Belief Network) a fim de selecionar características para a tarefa de classificação de imagens de sensoriamento remoto. Como no RCP cada nó é uma variável randômica binária, eles associaram cada um a uma característica do conjunto de entrada. Eles usaram um banco de dados com 2800 imagens e obtiveram um desempenho médio de 77%.

Em [SUK et al., 2016], os autores utilizam um aprendizado multitarefa esparso profundo para seleção de características. Tal método utiliza de regressão e de regularizadores a fim de realizar o aprendizado. Eles consideram o algoritmo como uma seleção hierárquica. Os autores tratam o problema de diagnóstico de Alzheimer e afirmam que o algoritmo desenvolvido por eles supera os três outros métodos analisados nas tarefas de classificação de três ou quatro classes.

2.1.3.1 Principais Trabalhos envolvendo Algoritmos Genéticos

De acordo com BING et a. (2016), técnicas de computação evolucionária (EC) têm ganhado muita atenção recentemente e têm se mostrado efetivas na resolução de problemas de seleção de características. Dentre os métodos, segundo os autores, se destacam os algoritmos genéticos, a programação genética, otimização de enxame de partículas (PSO) e otimização de colônia de formigas (ACO). Ainda de acordo com eles, o AG foi a primeira técnica de computação evolucionária amplamente utilizada para seleção de características. Por estes motivos, o AG é utilizado neste trabalho para este fim e esta seção descreve os principais trabalhos que os abordam com este intuito. Na seção 3.8 é apresentado um quadro comparativo dos AGs revisados nesta seção com o aqui desenvolvido (apresentado no capítulo 3).

GHAMISI & BENEDIKTSSON (2015) criaram uma abordagem híbrida utilizando AG e PSO com função fitness baseada na acurácia do SVM (HGAPSO + SVM). De acordo com eles, a hibridização é obtida ao integrar a velocidade padrão e regras de atualização com a seleção, crossover e mutação do GA.

A respeito das partículas do PSO, a dimensão das mesmas (tanto dos vetores velocidade como dos de posição) foi igual ao número de características do conjunto de entrada a fim de representar cada característica. Além disso, o vetor posição foi representado por valores binários a fim de indicar a seleção ou não da dimensão.

Inicialmente, uma população randômica é gerada. A cada geração, a função fitness é calculada para cada indivíduo e, em seguida, metade dos mais aptos são selecionados (elite) e melhorados pelo PSO. Eles utilizam a equação 6 para aplicar na elite, a equação 7 para definir a variação da velocidade e atualizam a posição utilizando a equação 8.

A população da próxima geração é formada pelos indivíduos melhorados mais os gerados pelo crossover aplicado sobre eles. O operador de cruzamento é o de dois pontos onde cada pai é selecionado ao escolher dois indivíduos aleatoriamente e, após isso, optar pelo mais apto.

A mutação é realizada juntamente com o crossover e consiste de uma mutação uniforme com probabilidade fixa de 0,01.

$$V_i^{k+1} = WV_i^k + C_1 r_1 (pb_i^k - X_i^k) + C_2 r_2 (gb_d^k - X_i^k),$$

Equação 6: Velocidade do PSO em GHAMISI e BENEDIKTSSON (2015)

$$\Delta X_i^{k+1} = \frac{1}{1 + \exp(-V_i^{k+1})}.$$

Equação 7: Variação da posição do PSO em GHAMISI e BENEDIKTSSON (2015)

$$X_i^{k+1} = \begin{cases} 1, & \Delta X_i^{k+1} \geq r_x. \\ 0, & \Delta X_i^{k+1} < r_x \end{cases}$$

Equação 8: Posição do PSO em GHAMISI e BENEDIKTSSON (2015)

GHAREB et al. (2016) propõem uma abordagem híbrida que combina as vantagens de um método filter com um algoritmo genético wrapper. O algoritmo genético possui como função fitness a média ponderada entre a média macro da F-measure do classificador Naive Bayes para o subconjunto de características selecionadas S_i e o inverso do tamanho de S_i, conforme a equação 9 (na equação, Z é uma constante definida por eles como 80%).

A função de seleção utilizada por eles é a roleta, a qual seleciona um indivíduo com determinada probabilidade. No caso, consiste da função de aptidão dividida pelo somatório das avaliações dos subconjuntos obtidos até a iteração atual. Além disso, o operador de cruzamento consiste em dividir os cromossomos pais em duas partes de mesmo tamanho e gerar dois filhos a partir delas. O primeiro é composto pelas duas melhores partes e o segundo pelas restantes. As melhores partes são definidas pelo peso cumulativo das suas características (definido por um dos seis métodos filter utilizados por eles: CDM, IG, OR, FM, GSS e TF-IDF). Cada método destes dá origem a um algoritmo diferente (CDM-EGA, IG-EGA, OR-EGA, FM-EGA, GSS-EGA e TF-IDF-EGA).

Por fim, o operador de mutação consiste em selecionar randomicamente as características e, após isso, modificar os seus valores. Além disso, eles utilizam o melhor subconjunto da geração anterior para substituir as características com menores pesos do conjunto que sofreu a mutação.

A conclusão, segundo eles, é que as abordagens híbridas criadas são mais efetivas que os métodos filters puros, pois produzem uma maior redução de dimensionalidade sem perda da precisão na maioria das situações.

$$Fit - Fun\ (S_i) = Z * C\ (S_i)\ +\ (1\ - - Z)\ (1\ /\ Size\ (S_i)$$

Equação 9: Função de aptidão de GHAREB et al. (2016)

ORESKI e ORESKI (2014) abordaram o problema de risco de crédito e modificaram pouco o AG padrão para realizar a seleção de características (o chamaram de HGA-NN). Como em GHAREB et al. (2016), os autores utilizam técnicas filters existentes para melhorar o AG proposto. Como método de seleção de sobreviventes eles realizaram experimentos com a roleta, torneio, estocástico, Boltzmann, corte e um esquema de seleção próprio deles (chamaram-no de único). Realizaram testes com dois tipos de crossover, o uniforme e o de um ponto. Fixaram o número máximo de gerações em 50, o tamanho da população também em 50 e utilizaram como função de aptidão uma rede neural feed forward com uma única camada escondida.

O que diferencia o trabalho dos autores é o uso de métricas de abordagens filters. Diferente de GHAREB et al. (2016), que utilizam tais abordagens nos operadores de crossover e mutação, ORESKI e ORESKI (2014) geram a população inicial com a utilização destas métricas e de soluções conhecidas a priori, partindo, portanto, de indivíduos melhores.

Como conclusão, os autores afirmaram que os resultados foram promissores para seleção de características e classificação em problemas de risco de crédito. Indicaram também que o HGA-NN é uma adição promissora nas técnicas existentes de mineração de dados.

Tsai et al. (2013) utilizaram algoritmos genéticos para seleção de características e de instâncias. O algoritmo genético utilizado foi o AG disponibilizado pelo WEKA ([IAN & EIBE, 1999]). Eles precisaram ajustar três parâmetros: o tamanho da população, a taxa de crossover e a taxa de mutação (diferentes valores foram testados). A função de aptidão utilizada foi a validação cruzada usando como método de aprendizado ou o SVM ou o Naive Bayes.

A intenção dos autores foi avaliar o uso das técnicas de seleção de instâncias e características isoladamente e em conjunto. Como conclusão, eles concluíram que o uso das mesmas em conjunto geralmente diminui ligeiramente o desempenho, mas esta redução não é negativa, uma vez que é pequena e outros benefícios, como o aumento da velocidade no

treinamento da técnica de aprendizado, compensam. Além disso, afirmam que os melhores resultados são obtidos quando primeiro é realizada a seleção de características e depois a de instâncias.

UGUZ (2011) se utiliza de uma etapa anterior à aplicação do AG para seleção de características. Ele faz uso do método filter de ganho da informação para ranquear as variáveis pela sua importância e manter apenas as mais importantes. Em seguida, o AG (IG-GA) ou o PCA (IG-PCA) é executado a fim de selecionar ou extrair características.

As características do AG proposto por ele são: (1) função de aptidão é a validação cruzada usando o KNN ou árvore de decisão C4.5; (2) a seleção de sobreviventes é o método da roleta; (3) crossover de dois pontos; (4) representação binária do cromossomo e (5) mutação de 0 para 1 ou de 1 para 0 com probabilidade fixa.

O autor executou experimentos com duas bases de dados (Reuters-21,578 e Classic3) de categorização de textos e mostrou que o modelo proposto é apto a alcançar alta efetividade na resolução do problema proposto.

Gheyas e Smith (2010) seleciona características em três fases. A primeira é através do algoritmo de Recozimento Simulado que faz uma busca inicial para utilização das soluções encontradas na próxima etapa. A segunda utiliza as características selecionadas para compor a população inicial de um AG. E, por fim, na terceira fase, o algoritmo Hill Climbing é executado a fim de refinar ainda mais as soluções encontradas.

O algoritmo de aprendizado utilizado por eles é a Rede Neural de Regressão Generalizada (GRNN) que, segundo os autores, não requer muito tempo de treinamento, é robusto contra mínimos locais, overfitting e outliers. Uma validação cruzada 10-fold é utilizada a fim de obter o desempenho do GRNN.

Sobre o AG, um indivíduo é representado por uma string de bits (representando a seleção ou não das características), eles utilizam uma população de 100 indivíduos (vindos das melhores soluções do algoritmo de recozimento simulado) e 50 pares são selecionados usando seleção baseada em rank. O operador de cruzamento é o metade uniforme, onde exatamente metade dos bits diferentes são trocados entre os pais. Após o cruzamento, os filhos sofrem mutação com probabilidade 0,001.

Os autores testaram o algoritmo proposto (SAGA) em 30 base de dados e mostraram que, para diferentes tempos de execução, o SAGA supera os demais algoritmos analisados (ACO,

FW, GA, PSO, SA, SBS, SFBS, SFFS e SFS). Eles concluíram também que não existe algoritmo inteiramente satisfatório por si só, mas que a combinação deles pode superar as fraquezas de cada um.

DERRAC et al. (2009) e LI et al. (2009) utilizam múltiplas populações no AG. DERRAC et al. (2009) utilizou três, onde a primeira foi responsável pela seleção de características, a segunda pela seleção de instâncias e a terceira focou em ambas. LI et al. (2009) permitiu o compartilhamento de indivíduos entre populações vizinhas a fim de possibilitar a troca de informações entre elas.

Algumas abordagens utilizam métodos de regressão juntamente com AGs. Exemplos são o GA-PLS ([HASEGAWA et al., 1997], [HASEGAWA et al., 1999]) e GA-IPLS ([BALABIN et al., 2011]), aos quais utilizam a regressão parcial dos mínimos quadrados e regressão parcial dos mínimos quadrados por intervalo juntamente com um AG para seleção de características.

TAN et al. (2008) criou uma abordagem híbrida onde um conjunto de algoritmos de seleção de características filter é utilizado para gerar um pool de soluções que, depois, é utilizado pelo algoritmo genético utilizado por eles. Usaram como função de aptidão a média ponderada entre a acurácia do SVM c(x) e o inverso da quantidade de características selecionadas s(x), conforme equação 10.

$$F = w * c(x) + (1 - w) * (1/s(x))$$

Equação 10: Função de aptidão de TAN et al. (2008)

ABBASI et al. (2008) desenvolveram um algoritmo genético para realizar a seleção de características. Eles o chamaram de Algoritmo Genético Pesado pela Entropia (em inglês, Entropy Weighted Genetic Algorithm - EWGA). Nele, o SVM foi utilizado como função de avaliação. Além disso, uma representação binária com *p* (número de características do problema) bits, onde cada bit representava se a característica estava selecionada (valor 1) ou não (valor 0), foi utilizada. Para geração da população inicial, definição do ponto de crossover e da taxa de mutação, a equação 10 foi utilizada. Em relação ao crossover, os autores utilizam o operador de um ponto (definido mais tarde na seção 2.3.1.3) modificado. O que muda em relação ao operador padrão é a forma de escolha do ponto que será utilizado. Em vez de ser aleatória, a escolha é feita de forma a gerar o primeiro filho com as características com maiores valores de MI selecionadas e o segundo filho com aquelas com menores valores de MI. Para isso, a equação 11 é utilizada.

$$\arg max_x \left| \sum_{A=1}^{x} MI(C,A)(S_A - T_A) + \sum_{A=x}^{p} MI(C,A)(T_A - S_A) \right|$$

Equação 11: Definição do ponto de crossover ([ABBASI et al., 2008])

Onde C é a classe que representa a saída do problema, A é o atributo/característica em questão, x é o ponto de crossover, S é o primeiro pai selecionado para gerar filhos, T é o segundo pai selecionado para gerar filhos, S_A é o bit associado à característica A do pai S, T_A é o bit associado à característica A do pai T e p é o número total de características.

Em relação ao operador de mutação, os autores o consideram como individual por bit com uma determinada probabilidade. A forma simples adotada na literatura considera a probabilidade com um valor fixo. Já os autores atribuem uma probabilidade de mutação de 0 para 1 diferente da de 1 para 0 utilizando a equação 12 abaixo. Nela, como no operador de crossover, a informação mútua é usada.

$$P_m(A) = \begin{cases} B[MI(C,A)], & se\ S_A = 0 \\ B[1 - MI(C,A)], & se\ S_A = 1 \end{cases}$$

Equação 12: Probabilidade de mutação ([ABBASI et al., 2008])

Onde $P_m(A)$ é a probabilidade de mutação da característica A e B é uma constante no intervalo [0,1].

2.2 Clusterização de Variáveis

Esta seção é dividida em duas subseções, as quais objetivam: (1) apresentar o conceito de clusterização de variáveis (feature clustering), (2) apresentar a formalização matemática para o problema e (3) apresentar os principais trabalhos de clusterização de variáveis que lidam com informação textual.

2.2.1 Definição

A fim de definir o conceito de clusterização de variáveis, primeiramente, é necessário explicitar o que é clusterização (clustering). A respeito deste, JAIN & DUBES (1988) afirmam que a análise de cluster é o estudo formal de algoritmos e métodos para agrupar ou classificar objetos. Ainda a respeito deste tema, os autores levantam os seguintes pontos:

- A análise de cluster é não supervisionada e, portanto, rótulos definidos a priori não são utilizados.
- O objetivo é, simplesmente, encontrar uma conveniente e válida organização dos dados. Não se trata de estabelecer regras para separar dados futuros em categorias.

Baseado nos pontos levantados pelos autores, podemos definir que a clusterização (ou análise de cluster) visa agrupar objetos de forma não supervisionada e com o simples intuito de organizar os dados.

Agora que já foi introduzido aquele conceito, já é possível definir a clusterização de variáveis. Esta consiste na utilização das técnicas de clusterização para agrupar as variáveis de entrada de um determinado problema e, após isso, utilizá-los para construir os novos vetores de entrada. Com relação a este conceito, KRIER et al. (2007) citam duas informações importantes:

- Métodos de clusterização clássicos são fáceis de transferir para a clusterização de variáveis. Basta apenas a definição da similaridade entre elas.
- A clusterização de variáveis contribui para a redução da dimensionalidade dos dados.

A primeira destaca o que já foi mencionado anteriormente, ou seja, a utilização dos algoritmos de clusterização para agrupar as variáveis. A única questão a ser definida para possibilitar tal uso é a definição da similaridade entre elas.

A segunda fala sobre a redução da dimensionalidade dos dados. O objetivo principal da clusterização de variáveis é justamente esta, ou seja, reduzir a dimensão dos vetores de entrada do problema. Tal redução é necessária, pois, quando temos dados com muitas dimensões e desejamos utilizar uma técnica de reconhecimento de padrões, o desempenho obtido é prejudicado. Esse problema é conhecido como maldição da dimensionalidade ([KEOGH & MUEEN, 2011]).

2.2.2 Formalização Matemática

O intuito desta seção é apresentar uma formulação matemática para o problema de clusterização de variáveis. Não que esta formalização será utilizada ao decorrer deste trabalho, uma vez que cada autor adota a sua própria e não existe uma unificada. O objetivo desta seção é unicamente facilitar o entendimento da técnica.

Com base na definição apresentada na seção anterior, podemos formalizar matematicamente o problema. Para isso, considere:

- S como o conjunto de treinamento com n elementos.
- $F = \{f_1,\ f_2, \dots, f_p\}$ como o conjunto inicial de características.
- $X = \{x_1,\ x_2, \dots, x_p\}$ como um elemento do conjunto de treinamento, onde x_i denota o valor da característica f_i.

Dadas as definições acima, as técnicas de clusterização de variáveis consistem em criar grupos de características $g_i = \{f'_1,\ f'_2, \dots, f'_m\}$ onde $f'_1,\ f'_2, \dots, f'_m \in F$ e $m \leq p$. A cada iteração j do algoritmo são formados diversos grupos dando origem a um conjunto $G_j = \{g_1, g_2, \dots, g_k\}$, onde todas as características de F estarão presentes em pelo menos um grupo g_i de G_j. Cada conjunto G_j é avaliado usando uma função de avaliação $J(G, S)$. No final do algoritmo, o conjunto G^* (G_j com maior valor de J(G, S)) será retornado. Por fim, G^* será utilizado para criar o novo conjunto de entrada do problema, o qual conterá uma dimensão para cada elemento de G^*.

Em outras palavras, para criar um algoritmo de clusterização de variáveis é necessário definir a forma como os conjuntos G_j serão formados e qual será a função de avaliação $J(G, S)$.

2.2.3 Principais Trabalhos

Antes de começar a apresentar os trabalhos relacionados ao assunto em questão, é importante deixar claro que o intuito aqui não é apresentar os principais trabalhos relacionados à clusterização e sim àqueles associados com a clusterização de variáveis para classificação de textos. Os principais trabalhos serão apresentados em um grau de detalhamento maior e, após isso, outros serão apresentados, mas com um grau menor. Desta forma, o leitor fica ciente da existência de outros trabalhos e, ao mesmo tempo, se desejar maiores informações, conseguirá saber onde buscá-las.

Na literatura, diversos trabalhos tratam da clusterização de variáveis para classificação de textos ([BAKER & MCCALLUM, 1998], [BAKERYZ & MCCALLUMYZ, 1998], [BEKKERMAN et al., 2001], [SLONIM & TISHBY, 2001], [Bekkerman et al, 2003], [DHILLON et al., 2003], [BEKKERMAN et al., 2005], [JIANG & LEE, 2007], [KRIER et al., 2007], [JIANG et al., 2011], [MYTHILY et al., 2015], [MEKALA et al., 2016], [SEDOC et al., 2017]).

A clusterização distribucional ([BKER & MCCALLUM, 1998], [DHILLON et al., 2003], [JIANG & LEE, 2007], [JIANG et al., 2011]), criação de agrupamentos de palavras baseados na distribuição dos rótulos associados a cada uma, é muito utilizada na literatura. Neste sentido, para realizar a criação de grupos de palavras, BKER & MCCALLUM (1998) utilizaram como medida de similaridade uma variação da divergência Kullback-Leibler (equação 13) conhecida como a divergência Kullback-Leibler para a média (equação 14).

$$KL(P(C|w_t), P(C|w_s)) = \sum_{j=1}^{|C|} P(c_j|w_t)\log(\frac{P(c_j|w_t)}{P(c_j|w_s)})$$

Equação 13: Divergência Kullback-Leibler

$$P(w_t)KL(P(C|w_t), P(C|w_t \vee w_s)) + P(w_s)KL(P(C|w_s), P(C|w_t \vee w_s))$$

Equação 14: Divergência Kullback-Leibler para a média

Nas equações:

- $C = \{c_1, c_2, \dots, c_m\}$ é o conjunto de classes do problema.
- w_t e w_s são duas palavras distintas.
- $P(c_j|w_t)$ é a probabilidade do documento pertencer à classe c_j dada a palavra w_t.
- $KL(P(C|w_t), P(C|w_t \vee w_s))$ é a divergência de Kullback-Leibler entre duas distribuições de probabilidade. A primeira está associada às probabilidades da palavra w_t pertencer a cada uma das classes. Já a segunda está relacionada à probabilidade de $w_t \vee w_s$ pertencerem a cada uma das classes.

Para calcular $P(C|w_t \vee w_s)$, os autores fazem uma média ponderada considerando as probabilidades de w_t e w_s, conforme expresso pela equação 15.

$$P(C|w_t \vee w_s) = \frac{P(w_t)}{P(w_t) + P(w_s)}P(C|w_t) + \frac{P(w_s)}{P(w_t) + P(w_s)}P(C|w_s)$$

Equação 15: Forma de cálculo da $P(C|w_t \vee w_s)$

Para formar os clusters, uma clusterização hierárquica aglomerativa ([JAIN & DUBES, 1988], [JAIN et al., 1999]) foi utilizada pelos autores.

Na mesma linha, DHILLON et al. (2003) utilizam uma clusterização distribucional baseada na medida de divergência de *Kullback-Leibler*. Em relação ao algoritmo utilizado, diferem de BKER & MCCALLUM (1998), uma vez que utilizam um algoritmo hierárquico divisivo. Para calcular as distribuições de probabilidade de um cluster $W_t = \{w_1, \dots, w_k\}$ sobre as classes $C = \{c_1, \dots, c_k\}$, eles utilizam a equação 16, a qual consiste em um somatório ponderado das distribuições de probabilidades de cada palavra pertencente ao cluster.

$$P(C|W_t) = \sum_{w_j \in W_t} \frac{P(w_j)}{\sum_{w_j \in W_t} P(w_j)} P\left(C\middle|w_j\right)$$

Equação 16: Cálculo da $P(C|W_t)$ realizado por DHILLON et al. (2003)

O objetivo do algoritmo é minimizar a função objetivo representada pela equação 17.

$$\sum_{t=1}^{k} \sum_{w_j \in W_t} P(w_j) KL(P\left(C\middle|w_j\right), P(C|W_t))$$

Equação 17: Função minimizada por DHILLON et al. (2003)

Na equação, a função KL é a divergência de Kullback-Leibler e, conforme mencionado anteriormente, mede a divergência entre duas distribuições de probabilidades.

Outros trabalhos, o qual utilizam uma clusterização distribucional são: (1) o de SLONIM & TISHBY (2001) que fazem uso de uma abordagem hierárquica aglomerativa e, no lugar da divergência de Kullback-Leibler (KL), é utilizada a divergência de Jensen-Shannon (JS). Com isso, introduzem o Information Bottleneck Framework; (2) Bekkerman et al (2003) utilizaram o IB framework para representar os documentos e o usaram para gerar as entradas para o SVM; (3) [DALMAU et al., 2007] tratam o problema através de abordagens típicas de processamento de sinais. Eles associam os termos a um sinal com determinada distribuição de probabilidade e medem a similaridade entre os sinais usando um coeficiente de correlação. Além disso, utilizam uma clusterização aglomerativa e sinais flat (com baixa variância) são considerados ruidosos; (4) BEKKERMAN et al. (2005) simultaneamente clusterizam variáveis de muitos tipos (documentos, palavras, autores, etc) baseado na iteração par a par entre eles. O algoritmo é

dividido em duas partes. A primeira é uma extensão do que foi feito por [DHILLON et al., 2003] e a segunda é uma clusterização usando procedimentos aglomerativos e divisivos; (5) MEKALA et al. (2016) obtém uma representação compacta dos documentos através de três passos: o primeiro consiste na representação de palavras por vetores de probabilidades e a clusterização das mesmas, o segundo utiliza os clusters para criar a representação a nível de documento e, por fim, o último passo normaliza os vetores de documentos e, através da utilização de um limiar, anula os valores próximos de zero, criando assim uma representação esparsa (vetores de documentos esparsos).

Sem utilizar uma clusterização distribucional, JIANG et al. (2011) tratam do assunto de clusterização de variáveis. Para entender o que foi feito por eles, considere:

- $D = \{d_1,\ d_2, \dots,\ d_n\}$ – o conjunto de n documentos.
- $W = \{w_1,\ w_2, \dots,\ w_m\}$ – o vetor de características representando m palavras.
- $C = \{c_1,\ c_2, \dots,\ c_p\}$ – o conjunto de p classes.

A partir dessas definições, os autores representam cada palavra w_i como um vetor x_i. Cada elemento j deste vetor é a probabilidade de o documento ser da classe c_j dado que este contém a palavra w_i ($P(c_j|w_i)$). O vetor x_i é definido formalmente pela equação 18 e a forma de calcular a $P(c_j|w_i)$ é apresentada na equação 19.

$$x_i = < x_{i1}, x_{i2}, \dots, x_{ip} > = < P(c_1|w_i), P(c_2|w_i), \dots, P(c_p|w_i) >$$

Equação 18: Vetor representativo de uma palavra w_i ([JIANG et al., 2011])

$$P(c_j|w_i) = \frac{\sum_{q=1}^{n} d_{qi} \times \delta_{qj}}{\sum_{q=1}^{n} d_{qi}}$$

Equação 19: Cálculo da $P(c_j|w_i)$ - JIANG et al.(2011)

Na equação, d_{qi} indica o número de ocorrências de w_i no documento d_q e δ_{qj} assume valor 1 (se d_q pertence a c_j) ou 0 (caso contrário).

O vetor x_i é então utilizado no algoritmo de agrupamento para representar cada elemento i a ser agrupado. Para calcular a similaridade entre um elemento x e um cluster G, a função de similaridade fuzzy definida pela equação 20 é utilizada.

$$\mu_G(x) = \prod_{i=1}^{p} \exp\left[-\left(\frac{x_i - m_i}{\sigma_i}\right)^2\right]$$

Equação 20: Cálculo da similaridade $\mu_G(x)$ entre o vetor x e o cluster G - JIANG et al.(2011)

Na fórmula, m_i é a média e σ_i é o desvio da dimensão i dos vetores x pertencentes ao cluster G. Estes estão representados pelas equações 21 e 22.

$$m_i = \frac{\sum_{j=1}^{q} x_{ji}}{|G|}$$

Equação 21: Cálculo da média m_i da dimensão i dos vetores x pertencentes ao cluster G - JIANG et al.(2011)

$$\sigma_i = \frac{\sum_{j=1}^{q} x_{ji}}{|G|}$$

Equação 22: Cálculo do desvio σ_i da dimensão i dos vetores x pertencentes ao cluster G - JIANG et al.(2011)

Repare que $0 \leq \mu_G(x) \leq 1$. Além disso, se $\mu_G(x) \approx 1$, o elemento x é muito semelhante ao cluster. Se $\mu_G(x) \approx 0$, o elemento x é muito diferente dele. Com base nisto, JIANG et al.(2011) introduzem um teste de similaridade, o qual consiste em verificar se $\mu_G(x) \geq \rho$.

Dado o exposto, o algoritmo de clusterização de JIANG et al. (2011) parte da situação em que todos os elementos não pertencem a nenhum cluster. A cada iteração, um elemento x não pertencente ainda a algum cluster é avaliado. É feito o teste de similaridade com cada cluster existente. Se nenhum passar no teste, um novo cluster com o elemento é criado. Caso contrário, o elemento é adicionado ao cluster de maior similaridade $\mu_G(x)$.

Apesar de não ter sido explicitamente citado como uma algoritmo de clusterização de variáveis, o trabalho de ZHAI et al. (2011) pode ser considerado assim, uma vez que eles agruparam características de produtos. No trabalho deles, duas etapas foram realizadas. Na primeira, componentes desconexas foram criadas ligando expressões referentes a características dos produtos que tinham alguma palavra em comum. Depois, a similaridade léxica do WordNet (um dicionário léxico em inglês) foi utilizada a fim de construir um grafo menos desconexo. Apenas as k maiores componentes foram mantidas e as expressões foram rotuladas como pertencente ao seu respectivo grupo. Na segunda etapa, um algoritmo supervisionado foi utilizado para agrupar as características não rotuladas.

2.3 Algoritmos Genéticos

De acordo com GOLDBERG & HOLLAND (1988), algoritmos genéticos são procedimentos de busca probabilísticos desenhados para trabalhar com espaços largos envolvendo estados que podem ser representados por strings. Ainda de acordo com eles, estes métodos são inerentemente paralelos e usam um conjunto de amostras do espaço (uma população de strings) para gerar um novo conjunto de amostras.

Usando as afirmações de GOLDBERG & HOLLAND (1988), podemos introduzir alguns conceitos em relação aos algoritmos genéticos. Primeiro, os "estados que podem ser representados por strings" são denominados indivíduos e, além disso, são utilizados para representar uma possível solução do problema. O conjunto de indivíduos (chamados de "amostras do espaço" por eles) é denominado de população.

EIBEN & SMITH (2003) afirmam que algoritmo genético é o tipo de algoritmo evolucionário mais comumente utilizado. Além disso, dizem que os algoritmos evolucionários possuem diversas variantes e a ideia por trás delas é a mesma: dada uma população de indivíduos dentro de algum ambiente que tem recursos limitados, a competição por estes recursos causa uma seleção natural (sobrevivência do mais apto).

Ainda sobre algoritmos evolucionários, EIBEN & SMITH (2003) falam acerca da função de avaliação, que mede a aptidão de um indivíduo no ambiente, dos operadores de mutação (aplicado sobre um indivíduo) e crossover (aplicado a dois ou mais indivíduos denominados pais) para geração de filhos e da escolha dos indivíduos para a próxima geração. Para entender melhor o funcionamento de um algoritmo evolucionário e, consequentemente, de um algoritmo genético, considere o pseudocódigo abaixo, o qual representa o esquema geral de um algoritmo evolucionário.

***Passo 1**: INÍCIO*

***Passo 2**:* *INICIALIZE a população com soluções candidatas randômicas*

***Passo 3**:* *AVALIE cada candidato*

***Passo 4**:* *REPITA ATÉ (CONDIÇÃO DE TERMINO seja satisfeita) FAÇA*

***Passo 5**:* *SELECIONE pais*

***Passo 6**:* *RECOMBINE pares de pais*

***Passo 7**:* *MUTE a cria resultante*

***Passo 8**:* *AVALIE os novos candidatos*

***Passo 9**:* *SELECIONE os indivíduos para a próxima geração*

***Passo 10**:* *FIM REPETIÇÃO*

***Passo 11**: FIM*

Algoritmo 2: Esquema geral de um algoritmo evolucionário

No pseudocódigo acima, são mostradas as etapas do algoritmo evolucionário. Como pode-se ver, a população inicial é iniciada, logo após, os candidatos são avaliados usando a função de avaliação e, em seguida, as iterações do algoritmo são executadas até que a condição de término seja satisfeita. Ao longo das iterações, os pais são selecionados, em seguida, os operadores de recombinação e mutação são aplicados, os indivíduos gerados são avaliados e indivíduos são selecionados para compor a próxima geração.

EIBEN & SMITH (2003) mencionam que não existe um algoritmo genético único e que diversas representações e operadores de seleção podem ser utilizados. Eles mencionam ainda o algoritmo genético clássico, o qual é conhecido como algoritmo genético simples (AGS) ou canônico. Na subseção seguinte, ele e seus componentes serão descritos.

2.3.1 Algoritmo Genético Simples (AGS)

A configuração do AGS é a seguinte. Os indivíduos são representados como strings binárias, o operador de recombinação é o de um ponto, o operador de mutação consiste na troca de bits (bit flip), a seleção dos pais é proporcional à aptidão e a seleção de sobreviventes é geracional ([EIBEN & SMITH, 2003]). Nas subseções seguintes cada componente deste será detalhado.

2.3.1.1 Representação Binária

Nesta representação, um indivíduo consiste de uma sequência de números binários ([EIBEN & SMITH, 2003]). Na figura 1 está retratado um exemplo desta representação.

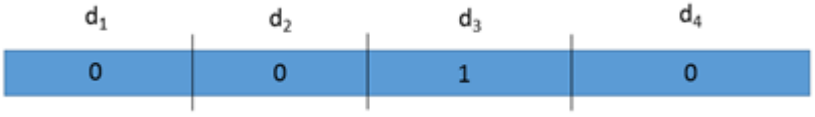

Figura 1: Exemplo de representação binária do algoritmo genético

2.3.1.2 Mutação por troca de bit (bit flip)

De acordo com EIBEN & SMITH (2003), mutação é um nome genérico dado a estes operadores de variação que usam somente um pai e criam um filho aplicando algum tipo de mudança randomizada na representação do indivíduo.

O operador de mutação por troca de bit permite com que cada bit da representação do indivíduo seja trocado (de 0 para 1 ou de 1 para 0) com uma determinada probabilidade p_m ([EIBEN & SMITH, 2003]). Um exemplo da aplicação deste operador está representado na figura 2. Nela, os bits das dimensões d_2 e d_3 são trocados, uma vez que eles foram os únicos que passaram no teste de probabilidade.

Figura 2: Exemplo de mutação por troca de bit

2.3.1.3 Recombinação de 1 ponto

A recombinação consiste no processo pelo qual um novo indivíduo é criado a partir da informação contida em dois ou mais pais ([EIBEN & SMITH, 2003]). Além disso, normalmente, este operador também é aplicado probabilisticamente com taxa p_c.

A recombinação de 1 ponto visa escolher um número aleatório entre 1 e L-1 (com L sendo o comprimento da representação do indivíduo), dividir ambos os pais nestes pontos e, por fim, trocar as caldas dos dois ([EIBEN & SMITH, 2003]). Este operador está ilustrado na figura 3. Nela, os indivíduos 1 e 2 dão origem aos 3 e 4.

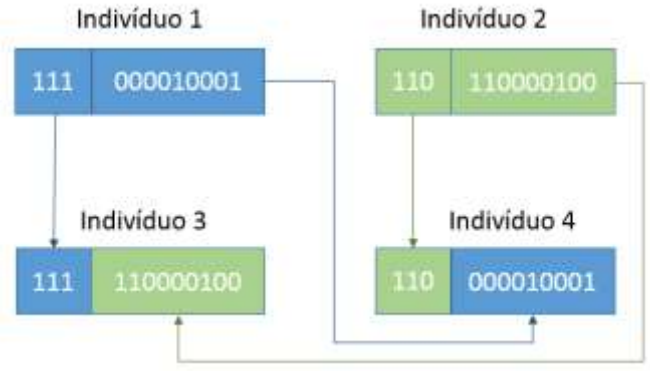

Figura 3: Exemplo de recombinação de 1 ponto

2.3.1.4 Seleção de Pais proporcional à aptidão

Nesta forma de seleção de pais, a probabilidade de um indivíduo *i* ser selecionado como pai é igual a $\frac{f_i}{\sum f_j}$, onde f_i é a aptidão absoluta do indivíduo *i* e $\sum f_j$ é o somatório da aptidão absoluta de toda a população ([EIBEN & SMITH, 2003]).

2.3.1.5 Seleção de Sobreviventes Geracional

A seleção de sobreviventes é responsável por selecionar, para a próxima geração, μ indivíduos dentre μ pais e λ filhos ([EIBEN & SMITH, 2003]).

Considerando que o número de filhos gerados é igual ao número de pais ($\mu=\lambda$), a seleção de sobreviventes geracional é bem simples e consiste em substituir os pais pelos filhos, ou seja, os indivíduos gerados durante a iteração darão origem à nova geração.

Capítulo 3. Algoritmo Genético para Seleção de Características

Este capítulo visa especificar os detalhes do algoritmo genético adotado neste trabalho a fim de selecionar as características relevantes e, portanto, contribuir para o aumento do desempenho do algoritmo de aprendizado a ser utilizado. A principal contribuição deste algoritmo é a probabilidade de mutação individual para cada bit e proporcional ao coeficiente de correlação de Pearson. Outra inovação introduzida é a geração da população inicial baseada também no coeficiente de correlação citado. O restante do algoritmo genético se baseia no que já existe na literatura: cromossomo binário, recombinação (crossover) de 1 ponto, elitismo e seleção natural por classificação ([EIBEN & SMITH, 2003]), função de avaliação SVM usando validação cruzada ([FROHLICH et al., 2003]).

Este capítulo está disposto da seguinte forma: na seção 3.1 é descrito como o cromossomo é representado; na seção 3.2 a metodologia para geração da população inicial é especificada; na seção 3.3 o operador de mutação é descrito; na seção 3.4 o operador de crossover adotado é apresentado, na seção 3.5 a função de avaliação é especificada, na seção 3.6 é introduzido o conceito de elitismo e como ele é utilizado e, por fim, na seção 3.7, o procedimento adotado para seleção natural é descrito.

3.1 Cromossomo

Nos algoritmos genéticos (AG's), o cromossomo é a representação computacional de uma possível solução do problema. A forma escolhida para representa-lo é a codificação binária ([EIBEN & SMITH, 2003]). Portanto, o cromossomo é representado como uma string de bits, onde cada posição da string está relacionada a uma dimensão e o seu valor indica se a dimensão é (valor 1) ou não (valor 0) selecionada. Por exemplo, considere a figura 4. O cromossomo representado pela string "0010" embute a informação de que o conjunto de dados utilizado é composto apenas por quatro dimensões e que a terceira característica (relacionada à terceira dimensão) está sendo selecionada e as outras não.

Figura 4: Exemplo da Representação do Cromossomo

3.2 Seleção da População Inicial

De acordo com EIBEN & SMITH (2003), a inicialização da população é mantida simples nos algoritmos genéticos: os indivíduos são gerados randomicamente. Dizem ainda que heurísticas específicas podem ser utilizadas para gerar uma população inicial com maior aptidão.

Neste trabalho, para realizar tal geração, é utilizada uma medida de correlação entre duas variáveis aleatórias: o coeficiente de correlação de Pearson. Este é utilizado para definir a importância de uma determinada característica (representada por um determinado gene do cromossomo) do conjunto de entrada. Para cada característica do problema, a sua importância é definida como o módulo do coeficiente de correlação de Pearson dela com a variável que representa a saída, representando, portanto, o quanto a variável de entrada em questão é capaz de traduzir a saída.

Com base no exposto, para gerar uma população inicial de tamanho **K**, o coeficiente de correlação de Pearson de cada dimensão do conjunto de entrada com a saída foi calculado. A seguir, as **N** dimensões foram ordenadas de forma decrescente pelo módulo do coeficiente. Após isso, o vetor foi dividido em **K** segmentos de igual tamanho. Os segmentos foram utilizados para gerar os indivíduos de forma que o primeiro indivíduo foi criado definindo os bits das dimensões pertencentes ao primeiro segmento (aquele que contém as variáveis mais importantes) como **1** e das dimensões restantes como **0**. O segundo indivíduo teve os bits dos dois primeiros segmentos definidos como **1** e, assim por diante, até que o último indivíduo teve os bits dos **K** segmentos (todos os segmentos) definidos como **1**. Portanto, através do processo, foram gerados **K** (tamanho da população e também número de segmentos) indivíduos. Este processo de geração é representado pelo algoritmo abaixo:

***Passo 1**: Calcule o coeficiente de correlação de Pearson com a saída para cada dimensão.*

***Passo 2**: Ordene, de forma decrescente, as dimensões pelo coeficiente de correlação de Pearson.*

***Passo 3**: Divida as dimensões ordenadas em **K** segmentos de igual tamanho.*

***Passo 4**: Para **i** de **1** até **K** faça*

***Passo 5**:* *Crie um indivíduo **R**.*

***Passo 6**:* *Defina os bits dos **i** primeiros segmentos do indivíduo **R** como **1**.*

***Passo 7**:* *Defina os bits dos segmentos restantes do indivíduo **R** como **0**.*

***Passo 8**:* *Adicione **R** ao conjunto **IND** de indivíduos gerados*

***Passo 9**: Retorne o conjunto **IND**.*

Algoritmo 3: Algoritmo de geração da população inicial

A título de ilustração, considere o caso com N=4 dimensões onde queremos gerar uma população de K=2 indivíduos. Este caso é ilustrado na figura 5. Repare que a primeira etapa corresponde ao cálculo do módulo do coeficiente de correlação de Pearson para cada dimensão. Após isso (etapa 2), as dimensões são ordenadas considerando os módulos dos coeficientes. Na última etapa, os K=2 indivíduos foram gerados. Repare que o vetor foi cortado em K=2 segmentos compreendendo 2 dimensões cada. O primeiro indivíduo foi gerado definindo as dimensões do primeiro segmento como 1 e as outras dimensões como 0. Já o segundo indivíduo teve as dimensões dos dois primeiros segmentos (neste caso todos os segmentos) definidas com o valor 1.

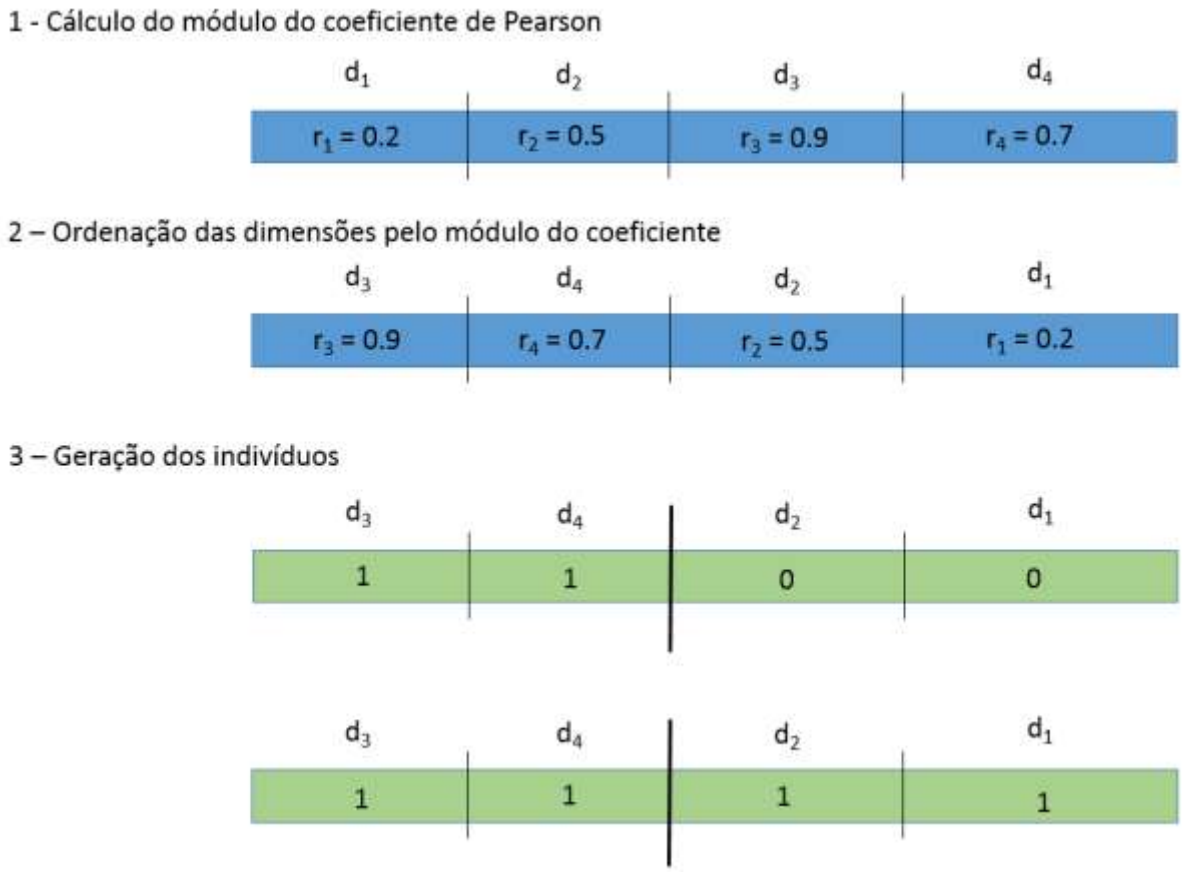

Figura 5: Exemplo de geração da população inicial

3.3 Mutação

Como já dito, a mutação desenvolvida neste trabalho é um diferencial quanto aos outros trabalhos presentes na literatura. Ela é semelhante à mutação por troca de bits (descrita na seção **2.3.1.2**), uma vez que consiste em alterar os bits de 0 para 1 ou 1 para 0. O que este trabalho introduz é a atribuição de probabilidades diferentes de mutação para cada bit (alelo) do cromossomo. Tais probabilidades são baseadas no módulo do coeficiente de correlação de Pearson, ou seja, quanto mais correlacionada for a dimensão com a saída, maior será a probabilidade de modificar um bit de 0 para 1 e menor será a de mudar de 1 para 0. O algoritmo a seguir ilustra o processo de mutação:

***Passo 1**: Calcule o coeficiente de correlação de Pearson com a saída para a dimensão i.*

***Passo 2**: Calcule as probabilidade de mutação p_i (0->1) e p_i (1->0) para a dimensão i usando as equações 6 e 7, respectivamente.*

***Passo 3**: Para **i** de **1** até **d** (número de dimensões) faça // percorrendo as dimensões*

***Passo 4**: Sorteie um número aleatório **a** entre 0 e 1.*

***Passo 5**: Se (valor da dimensão i)=0 e [a < p_i (0->1)] faça*

***Passo 6**: (valor da dimensão i)=1 // a mutação do bit de 0 para 1*

***Passo 7**: Se (valor da dimensão i)=1 e [a < p_i (1->0)] faça*

***Passo 8**: (valor da dimensão i)=0 // a mutação do bit de 1 para 0*

Algoritmo 4: Algoritmo de mutação de um indivíduo

Como se desejava que, em média, apenas 1 bit fosse trocado por cromossomo, foi necessário normalizar os módulos dos coeficientes de correlação de Pearson de cada dimensão com a saída para que tivessem média ***1/N*** (onde ***N*** é o número de dimensões). Além disso, para evitar que houvesse muitos módulos dos coeficientes negativos ou muito próximos de 0 após a normalização, além de levar a média para ***1/N***, levou-se também o desvio para ***1/(2N)***, garantindo assim que aproximadamente ***96%*** dos valores estivessem compreendidos no intervalo ***[0, 2/N]***. O coeficiente normalizado é representado pela equação 23.

$$\rho_i' = Max[0, \sigma_\rho' \cdot \left(\frac{|\rho_i| - \mu_{|\rho|}}{\sigma_{|\rho|}}\right) + \mu_\rho']$$

Equação 23: Normalização dos coeficientes de correlações de Pearson

Na fórmula, ρ_i' representa o coeficiente da dimensão i após a normalização, $|\rho_i|$ representa o módulo do coeficiente de correlação de Pearson da dimensão i com a saída, $\mu_{|\rho|}$ é a média dos módulos dos coeficientes de correlação, $\sigma_{|\rho|}$ é o desvio dos módulos dos coeficientes, σ_ρ' é o novo desvio (***1/2N***) e $\mu_\rho{}'$ é a nova média (***1/N***).

A partir da equação 23 é possível definir as probabilidades de trocar um bit para 1 e para 0 (equações 24 e 25, respectivamente).

$$P_i(0 \rightarrow 1) = \rho_i'$$

Equação 24: Probabilidade de mutar o bit i de zero para um

$$P_i(1 \rightarrow 0) = Max[0, \frac{2}{N} - \rho_i']$$

Equação 25: Probabilidade de mutar o bit i de um para zero

Após testes, percebeu-se que trocar em média apenas 1 bit por cromossomo não seria o ideal. Com relação à taxa de mutação, BEASLEY et al. (1996) diz que é necessário uma taxa de mutação alta quando o AG converge. Além disso, diz que é benéfico ter uma taxa de mutação variável. Considerando o exposto, uma taxa de mutação variável está sendo proposta neste trabalho e está representada pela equação 26. Ela foi desenvolvida de forma a manter a faixa (pm_{inf} – pm_{sup}) para a taxa de mutação. Ela começa com o limite inferior e, de acordo com o número de iterações em que uma nova solução não é encontrada, ela aumenta até chegar ao limite superior. A quantidade de iterações necessárias para chegar do limite inferior ao superior é representada pela variável ni_{max}(foi considerado o valor desta variável como 10% do número de iterações do AG) e o número de iterações sem melhora no desempenho é representado pela variável ni. Uma vez construída a taxa de mutação variável, as probabilidades de mutação podem ser ajustadas para terem em média $(t_{mut} \cdot N)$ bits, conforme expresso pelas equações 27 e 28.

$$t_{mut} = \left[\frac{(pm_{sup} - pm_{inf}) \cdot MIN(ni, ni_{max})}{ni_{max}} + pm_{inf} \right]$$

Equação 26: Taxa de mutação variável

$$P_i'(0 \rightarrow 1) = P_i(0 \rightarrow 1) \cdot (t_{mut} \cdot N)$$

Equação 27: Probabilidade ajustada de mutar o bit i de zero para um

$$P_i'(1 \rightarrow 0) = P_i(1 \rightarrow 0) \cdot (t_{mut} \cdot N)$$

Equação 28: Probabilidade ajustada de mutar o bit i de um para zero

3.4 Recombinação (crossover)

O operador de recombinação escolhido foi o operador de um ponto, onde com determinada probabilidade $\boldsymbol{P_c}$ um ponto na String de bits é escolhido e as Strings após o ponto são trocados entre os indivíduos ([EIBEN & SMITH, 2003]). Tal operador foi explicado na seção 2.3.1.3 e ilustrado na figura 4.

3.5 Função de Avaliação

Como função de avaliação, neste trabalho será utilizado o próprio algoritmo de aprendizado (no caso o SVM) utilizando a validação cruzada (como em FROHLICH et al. (2003)). Por exemplo, dado um cromossomo "0010", apenas a dimensão relacionada ao terceiro bit será utilizada para treinar e testar o SVM utilizando a validação cruzada. A taxa de acerto resultante será considerada como o retorno da função de avaliação.

O único problema de utilizar essa função de avaliação é o custo computacional, uma vez que, para um número grande de dimensões, o SVM demora um tempo razoável para ser executado e ainda existe a validação cruzada, a qual faz com que o mesmo seja executado diversas vezes. Apesar disto, ainda é viável utilizar esta função de avaliação, pois o processo de seleção de características só é realizado uma única vez (uma espécie de pré-processamento dos dados). Além disso, diversos trabalhos da literatura utilizam o SVM em conjunto com algoritmos genéticos para seleção de características ([MIN et al., 2006], [SAMANTA, 2004], [FROHLICH et al., 2003], [JACK & NANDI, 2002]).

3.6 Elitismo

O elitismo é um mecanismo que mantém os melhores indivíduos na população a fim de evitar com que eles sejam perdidos ao decorrer das iterações do AG ([EIBEN & SMITH, 2003]). Neste trabalho, uma porcentagem da população (taxa de elitismo - t_{elit}) é copiada de geração em geração (os mais aptos). Dessa forma, $t_{elit} \cdot tam_{pop}$ indivíduos são mantidos sem que os operadores genéticos sejam aplicados.

3.7 Seleção Natural (seleção de sobrevivente)

Como mecanismo de seleção natural, o método de seleção por classificação foi escolhido. De acordo com EIBEN & SMITH (2003), ele preserva uma pressão de seleção constante ordenando a população pela aptidão e alocando probabilidades de seleção a indivíduos de acordo com seu ranking. Neste trabalho, a probabilidade de seleção de um indivíduo da posição *i* é igual à razão entre o seu índice (começando em um) e o somatório dos índices de todos os outros indivíduos. A ordenação é feita pela aptidão, mas, quando ocorre empate, o número de características selecionadas (quantidade de bits 1 no cromossomo) é utilizada como critério.

3.8 Quadro comparativo com outros trabalhos

Esta seção visa comparar os algoritmos genéticos encontrados na literatura e especificados na seção 2.1.3.1 com o AG aqui desenvolvido. A tabela 1 possui este intuito. Nela, existem 6 colunas, a primeira para identificar o trabalho (o AG aqui desenvolvido é identificado pela primeira linha de dados da tabela), a segunda, terceira, quarta, quinta e sexta colunas especificam, respectivamente, a geração da população inicial, a mutação, o crossover, a aptidão e os diferenciais do trabalho.

Trabalho	População Inicial	Mutação	Crossover	Aptidão	Diferenciais
Este	Geração baseada em Pearson	Individual por bit com probabilidades diferentes baseadas em Pearson	Crossover de 1 ponto	Validação cruzada – SVM	Operador de mutação e geração da população inicial consideram a informação da correlação da variável com a saída, permitindo melhores resultados
GHAMISI & BENEDIKTSSON (2015)	Randômica	Mutação uniforme com probabilidade 0,001	Crossover de 2 pontos	Validação cruzada – SVM	PSO utilizado para melhorar os indivíduos mais aptos encontrado em cada rodada do AG
GHAREB et al. (2016)	Randômica	Mutação uniforme	Pais são divididos em duas partes do mesmo tamanho. Primeiro filho fica com as melhores partes e segundo com as piores. As melhores partes são definidas pelo peso cumulativo das suas características (definido por um dos seis métodos filter utilizados por eles: CDM, IG, OR, FM, GSS e TF-IDF).	Média ponderada entre a média macro da F-measure do classificador Naive Bayes para o subconjunto de características selecionadas Si e o inverso do tamanho de Si	Métodos filters são utilizados para gerar filhos melhores (cruzamento) e para substituttuir as piores características nos filhos gerados após a mutação.
ORESKI e ORESKI (2014)	Utilizam métodos filters e soluções conhecidas a priori para geração da população inicial	Mutação uniforme	Uniforme e de 1 ponto	Rede neural feed foward com uma única camada escondida	Métodos filters são utilizados para gerar a população inicial e partir de boas soluções.
TSAI et al. (2013)	Não diz	Não diz	Não diz	SVM ou Naive Bayes	Utiliza o AG para selecionar características e instâncias
UGUZ (2011)	Mantém apenas as características mais importantes ao manter uma etapa anterior de ranqueamento usando o método filte de Ganho da Informação	Mutação uniforme	Crossover de dois pontos	Validação cruzada usnado KNN ou árvore de decisão C4.5	Reduz a dimensão através de uma abordagem filter (ganho da informação) antes de rodar o algoritmo genético.
Ghevas e Smith (2010)	Melhores soluções encontradas pelo algoritmo de recozimento simulado são utilizadas como população inicial	Mutação uniforme	Crossover metade uniforme	Validação cruzada usando uma Rede Neural de Regressão Generalizada	Utiliza três diferentes algoritmos para se aproveitar dos benefícios dos três. A aplicação ocorre na seguinte ordem: recozimento simulado, algoritmo genético e hill
TAN et al. (2008)	Melhores soluções encontradas por um conjunto de métodos filter compõem a população inicial	N bits são escolhidos aleatoriamente para serem trocados (de 0 para 1 ou de 1 para 0).	Crossover de 1 ponto	Média ponderada entre a acurácia do SVM c(x) e o inverso da quantidade de características selecionadas s(x)	Utilizam uma população inicial composta por soluções de algoritmos filter, levando o AG a partir de soluções melhores

Tabela 1: Quadro comparativo entre o AG aqui desenvolvido e aqueles propostos por outros trabalhos

Capítulo 4. Algoritmo de Clusterização de Variáveis

Com o intuito de reduzir ainda mais a dimensionalidade dos dados, uma nova técnica de redução de dimensionalidade está sendo proposta neste trabalho. Baseado no fato de que a representação textual "bag of words" produz uma matriz de entrada muito esparsa ([LE & MIKOLOV, 2014]), a técnica proposta consiste na união de palavras através de um operador "ou" a fim de unir termos que não aparecem nas mesmas sentenças e, portanto, tornar tal matriz menos esparsa. Note que tal metodologia possibilita que os termos individuais sejam "descartados" e apenas os conjuntos (união de palavras pelo operador "ou") sejam utilizados. Tal técnica difere da metodologia tradicional dos n-gramas que utiliza um operador "e" e, na grande maioria dos casos, exige a utilização dos termos individuais como entradas distintas (aumentando assim a dimensionalidade dos dados). Repare também que essa união de palavras nada mais é do que uma forma de clusterizar variáveis.

A fim de possibilitar a identificação dos termos candidatos a serem unidos, três diferentes medidas presentes na literatura e já utilizadas para formar n-gramas foram utilizadas e estão representadas pelas equações 8 (afinidade), 9 (informação mútua) e 10 (log likelihood). Note que as três equações tendem a unir termos correlacionados de alguma forma. O propósito do método desenvolvido é contrário, ou seja, unir termos descorrelacionados, por isso todas as equações foram multiplicadas por -1, dando origem a uma medida de dissimilaridade **d**. Além das equações listadas, o coeficiente de correlação de Pearson foi também utilizado a fim de evitar com que termos correlacionados positivamente com a saída fossem agrupados com termos negativamente correlacionados com ela.

Note que as técnicas existentes na literatura tendem a unir termos muito correlacionados ou sinônimos ou, no caso de problemas de regressão, unir as dimensões associadas a coeficientes de regressão iguais. O algoritmo aqui proposto difere destas técnicas, pois visa unir termos o máximo descorrelacionados ou que aparecem menos vezes juntos. A ideia é aproveitar uma mesma dimensão para duas ou mais características que nunca ou quase nunca aparecem juntas.

Nas equações 29 e 30, w_1 e w_2 representam palavras (ou tokens) e a função f é a frequência. A equação 31 precisa da tabela de contingência (tabela 2) para ser entendida. Nela, a variável

i pode assumir os valores $i = \{a, b, c, d\}$ e a variável j pode assumir os valores $j = \{c_1, c_2, r_1, r_2\}$.

$$Aff(w_1, w_2) = \frac{f(w1, w2)}{\min[f(w1), f(w2)]}$$

Equação 29: Medida de afinidade

$$MI(w1, w2) = log_2 \frac{P(w1, w2)}{\mathrm{P(w1)} \cdot \mathrm{P(w2)}}$$

Equação 30: Informação Mútua

$$\sum_i i \cdot \ln(i) + N \cdot \ln(N) - \sum_i j \cdot \ln(j)$$

Equação 31: Log Likelihood Ratio

	W_2	**Não W_2**	
W_1	A	B	r1 = a + b
Não W_1	C	D	r2 = c + d
	c1 = a + c	c2 = b + d	N = a + b + c + d

Tabela 2: Tabela de Contingência

As equações descritas são utilizadas a fim de construir conjuntos de tokens que darão origem ao novo conjunto de entrada. O mesmo será formado considerando que os elementos internos aos conjuntos criados serão unidos por um "ou".

A fim de explicar o algoritmo proposto, considere os exemplos presentes nas figuras 6 e 7. Note, que o processo detalhado a seguir pode ser considerado um algoritmo hierárquico aglomerativo, uma vez que os conjuntos começam vazios e os tokens vão sendo alocados a eles ao decorrer da execução do mesmo. Para facilitar o entendimento, o algoritmo foi dividido em duas etapas: a etapa 1 de construção, eliminação e ordenação de combinações e a etapa 2 de

elaboração dos conjuntos. Elas são representadas nos exemplos pelas figuras 9 e 10, respectivamente.

Em relação à primeira etapa, ela começa no passo 1 (linha representada pelo número em romanos I na figura 6), onde os tokens são associados a seus respectivos coeficientes de correlação de Pearson com a saída. Após isso (II), os tokens que possuem o mesmo sinal do coeficiente de correlação são combinados. Repare que, no exemplo, somente o token 5 possui coeficiente com sinal negativo e, portanto, ele não pôde ser combinado com nenhum dos outros tokens. Ainda na linha II, os valores de dissimilaridade ***d*** entre os tokens são calculados para cada combinação gerada. Concluindo a primeira etapa, na linha III as combinações são ordenadas em ordem decrescente considerando os valores ***d***.

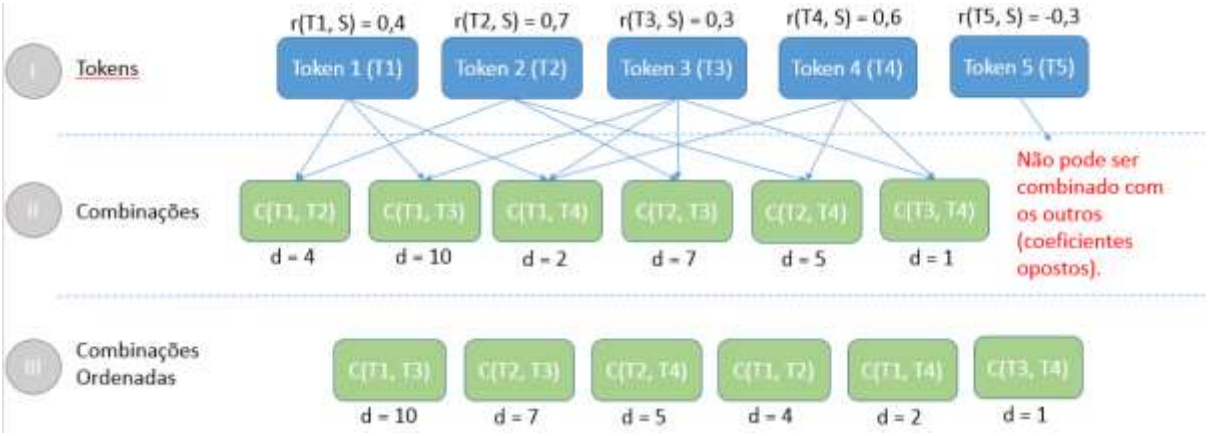

Figura 6: Exemplo do algoritmo de clusterização de variáveis – Etapa 1

Passando para a segunda etapa (representada pelo exemplo da figura 7), começamos com as combinações ordenadas geradas na primeira etapa (figura 6). O elemento presente no topo da lista de combinações (destacado na linha 1) resulta na união do termo 1 e do termo 3 gerando assim o conjunto representado na linha 2.

Após isso, ainda na etapa 2, passamos para o próximo elemento da lista, o qual consiste na combinação dos tokens 2 e 3. A combinação dos dois tokens resultaria em acrescentar T2 no único conjunto existente (que contém T1 e T3). Mas, não pode-se adicionar T2 ao conjunto somente considerando a sua união com T3, ou seja, considerar apenas d(T2, T3). Devemos considerar também se T2 deve ser unido com os outros elementos do conjunto (no caso, T1). Para isso, recalculamos o valor de ***d*** da combinação C(T2, T3) como o mínimo dentre as

combinações de T2 com os elementos do grupo. Esse recálculo é expresso no lado esquerdo da linha 3 e acima do retângulo arredondado que representa a combinação de T2 com T3 (item selecionado na linha).

O próximo passo (expresso na linha 4) consiste em reordenar as combinações após a atualização do valor de ***d***. Importante notar que antes da atualização, as combinações já estavam ordenadas. Dessa forma, para reordená-las, basta remover a combinação atualizada C(T2, T3) e inseri-la na posição correta. Na verdade, este procedimento de reordenação é desnecessário, uma vez que a combinação atualizada irá parar na posição anterior à combinação de T2 com o elemento do conjunto que fornece o menor valor de ***d*** (no caso do exemplo, com a combinação de T1 com T2). As duas combinações adjacentes (C(T2, T3) e C(T1, T2)) são iguais, pois ambas sugerem a inclusão de T2 no conjunto que contém T1 e T3. Desse modo, o procedimento necessário (representado pela figura 8) seria apenas incluir o elemento no conjunto, caso o valor de d não tivesse sido atualizado ou, caso contrário, apenas passar para o próximo elemento da lista.

Após o passo de reordenação, no exemplo, o próximo passo consiste em considerar a próxima combinação. No caso, a de T2 com T4. Como nem T2 e nem T4 pertenciam a algum conjunto criado anteriormente, um novo conjunto foi criado e os dois tokens foram adicionados a ele (passo 5).

Analisando as próximas combinações, todos os tokens presentes nelas já fazem parte de algum conjunto. Dessa forma, nada precisou ser feito (passos 6 e 7). Por fim, no passo 8, todos os tokens que não foram adicionados a algum conjunto dão origem a novos (no caso do exemplo, o token 5 dá origem ao terceiro conjunto).

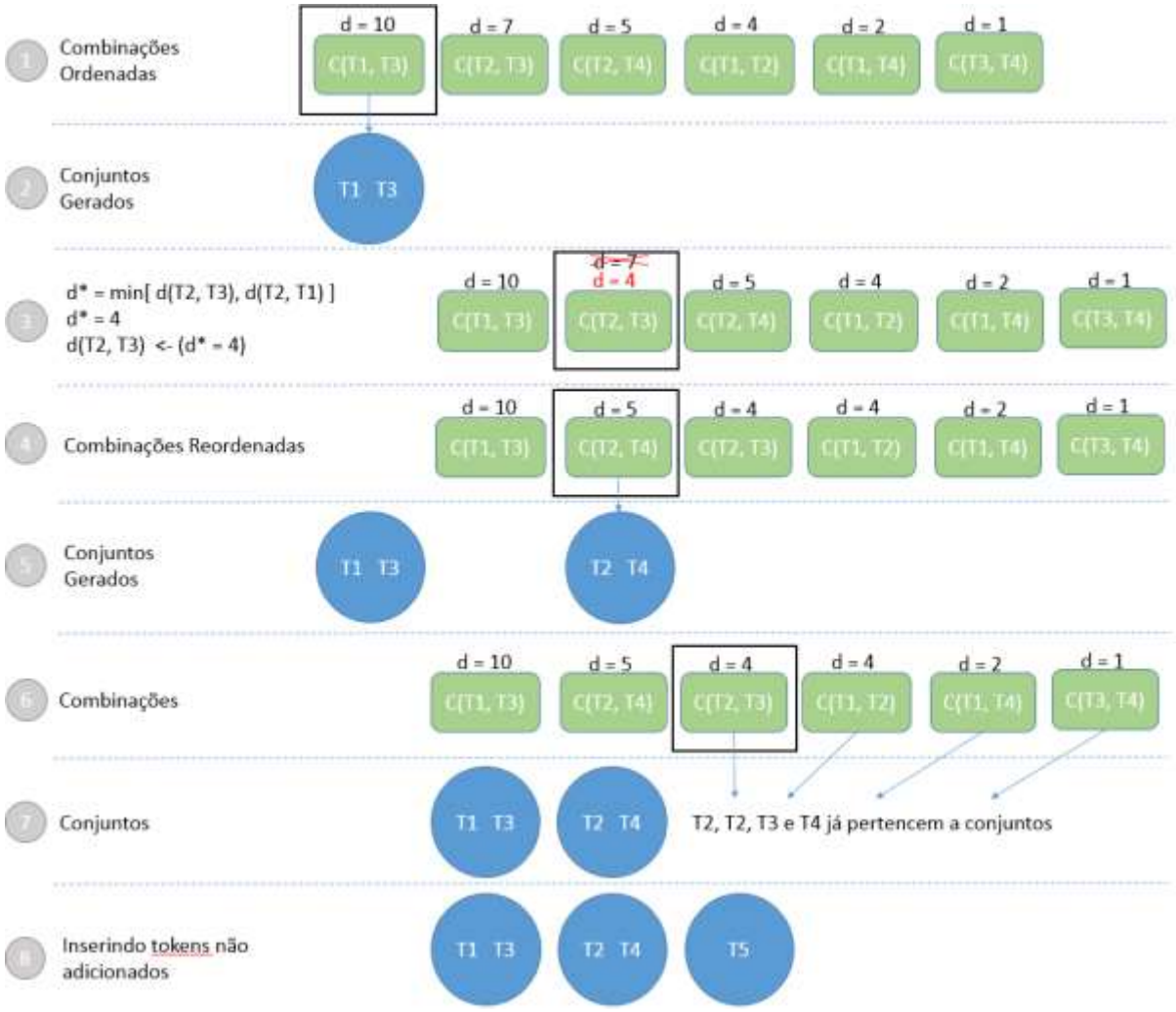

Figura 7: Exemplo do algoritmo de clusterização de variáveis – Etapa 2 com ordenação desnecessária

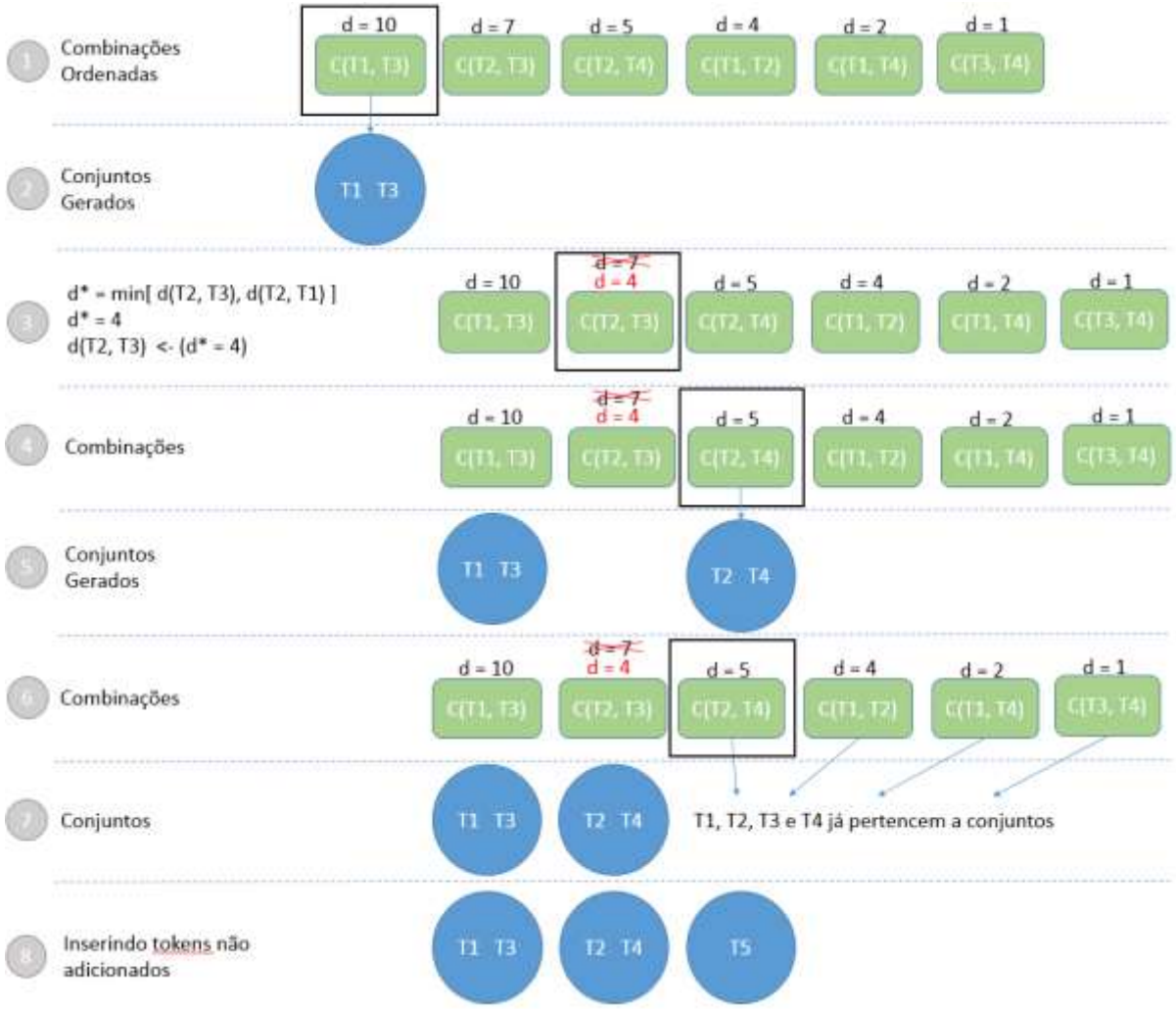

Figura 8: Exemplo do algoritmo de clusterização de variáveis – Etapa 2 com melhor desempenho

Analisando o exemplo apresentado, apenas duas etapas do algoritmo não foram apresentadas nele. A primeira consiste na etapa necessária à limitação da quantidade de elementos por conjunto. Esta é necessária para evitar que o algoritmo crie conjuntos muito grandes, o que poderia acabar prejudicando o desempenho com os conjuntos criados. A segunda consiste na limitação do número máximo de junções permitidas. Esta é necessária para evitar que junções prejudiciais sejam feitas, uma vez que as últimas junções são as piores (menores valores de ***d***).

O pseudo-codigo do algoritmo completo, incluindo as etapas mencionadas, pode ser conferido abaixo. Ele começa percorrendo todas as combinações de palavras/tokens a fim de calcular os valores de dissimilaridade ***d*** e adicionar as combinações que possuem mesmo sinal de correlação com a saída ao conjunto de combinações ***C*** (passo 1). No passo 2, as combinações

são ordenadas. No passo 3, os conjuntos são criados e adicionados à lista ***K***. Por fim, nos passos 4 e 5, os tokens que não foram adicionados a nenhum conjunto dão origem a conjuntos unitários que são adicionados à ***K*** e esta é retornada pelo algoritmo.

1 - Para toda combinação de palavras/tokens w1 e w2 faça

*1.1 – Calcule d = -1 * (valor calculado usando uma das três equações – 8, 9 ou 10).*

1.2 – Se o sinal do coeficiente de correlação de Pearson de w1 com a saída S for igual ao sinal de w2 com S, adicione a combinação c={w1, w2, d} à lista de combinações C.

2 – Ordene a lista de combinações C em ordem decrescente usando o valor d.

3 – Para cada combinação c={w1, w2, d} pertencente a C faça

3.1 – Verifique se o número máximo de junções foi alcançado. Se sim, pare de percorrer as combinações e passe para o passo 4.

3.2 – Verifique se existe algum conjunto k que contenha w1 ou w2.

3.3 – Se não houver, crie um novo conjunto k e adicione à lista de conjuntos K.

3.4 – Se os dois já fizerem parte de outros conjuntos, não faça nada.

3.5 – Se apenas um fizer parte de k (w = w1 ou w2), faça

3.5.1- Se k já tiver o seu número máximo de elementos, ignore a combinação atual e passe para a próxima (voltando ao passo 3.1).

3.5.2 – Caso contrário, recupere o menor valor d dentre todas combinações de w com os elementos do conjunto k.*

3.5.2.1 – Se d for diferente de d, simplesmente passe para a próxima combinação.*

3.5.2.2 – Caso contrário, adicione w à k.

4 – Crie novos conjuntos com um elemento caso algum token não tenha sido adicionado a algum conjunto. Adicione-os à lista de conjuntos K.

5 – Retorne a lista de conjuntos K.

Algoritmo 5: Algoritmo de clusterização de variáveis

Capítulo 5. Etapas Propostas

Este capítulo visa apresentar as etapas utilizadas neste trabalho. Estas serão utilizadas no capítulo 6 para realizar o estudo de caso e estão representadas pela figura 9. Nela, o lado esquerdo representa o fluxo principal e, no lado direito, estão os detalhes de cada uma das etapas.

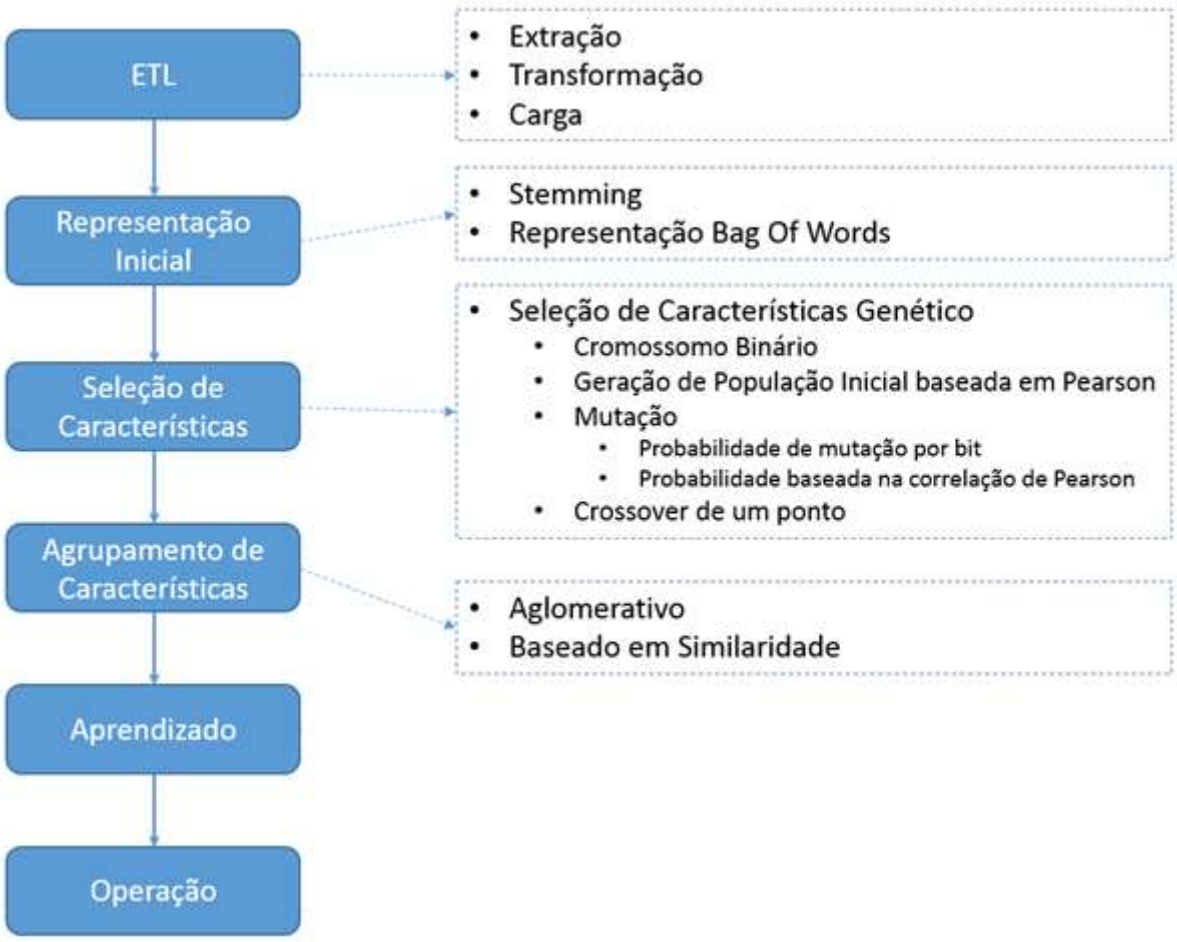

Figura 9: Etapas propostas para análise de sentimento

A primeira etapa consiste na extração, transformação e carga dos dados (extract, transform and load – ETL – [VASSILIADIS, 2009]). Segundo VASSILIADIS (2009), programas ETL periodicamente extraem dados do sistema origem, transformam os dados para dentro de um formato comum e então carregam os dados para dentro do banco de dados alvo. E é exatamente

pensando nesta metodologia que esta primeira etapa foi proposta. Na extração, os dados são captados da fonte. Na transformação, são realizadas operações sobre os dados extraídos de forma a deixá-los prontos para serem utilizados e, na carga, os dados são armazenados na base de dados.

A segunda etapa é aquela responsável por representar os dados. Nela, é realizado o processo de Stemização ou, no inglês, Stemming ([LOVINS, 1968]), onde as palavras flexionadas são reduzidas para o seu tronco (stem). Após isso, a notação bag of words é utilizada para representar o texto como um vetor. Existem alguns pontos que levaram à adoção da notação citada. O primeiro é que o trabalho com o qual serão comparados os resultados obtidos na base de avaliação de filmes ([PANG & LEE, 2004]) adota esta notação e deseja-se que as etapas iniciais sejam as mais semelhantes possíveis. O segundo ponto está relacionado à complexidade que seria adicionada ao se usar n-gramas com o algoritmo de clusterização de variáveis, uma vez que neste já são utilizados grupos de características. O terceiro tem como base a afirmação de CAMBRIA (2016), o qual diz que as abordagens de mineração de sentimento de texto ou discurso têm sido baseadas, principalmente, no modelo bag of words. O quarto e último ponto está relacionado ao principal intuito deste trabalho, o de apresentar dois algoritmos (para seleção de características e clusterização de variáveis) e mostrar que eles realmente são úteis no processo de redução de dimensionalidade de dados, ou seja, a forma de representação não é o foco deste trabalho.

A terceira etapa consiste na seleção de características, que é a aplicação do algoritmo genético apresentado no capítulo 3. Detalhes do algoritmo são especificados, tais como o tipo de cromossomo utilizado (binário), o operador de mutação (probabilidade individual para cada bit e baseada no coeficiente de correlação de Pearson), a geração da população inicial (indivíduos gerados definindo os bits das dimensões mais correlacionadas com a saída como um e o restante dos bits como zero) e o operador de crossover (operador de um ponto).

A quarta etapa é responsável por agrupar as características utilizando o algoritmo especificado no capítulo 4. Já a quinta é responsável por realizar o aprendizado. Neste trabalho são utilizados três algortimos de aprendizado SVM, k-vizinhos e Naive Bayes. O kernel utilizado no SVM foi o linear e foi escolhido por três motivos. O primeiro é devido o kernel linear superar ou ter desempenho semelhante à SVM's com outros kernels quando considerados no contexto de classificação de textos, como, por exemplo, constatado em MOHAMMAD et al. (2013), LUSS & D'ASPREMONT (2015), ZEHE et al. (2017), GOH & UBEYNARAYANA (2017). O segundo é por ainda ser amplamente utilizado atualmente

([ROUSSEAU et al., 2015], [YANG et al., 2015], [ZHANG & ZHONG, 2016], [ARRAS et al., 2017], [MAJUMDER et al., 2017] e [GOH & UBEYNARAYANA, 2017]). O terceiro motivo é que o propósito deste trabalho não é encontrar o melhor resultado possível e sim mostrar que os algoritmos desenvolvidos são úteis para o fim a que se destinam. Por isso, compara-se com um trabalho anterior ([PANG & LEE, 2004]) utilizando-se da mesma base e do mesmo algoritmo de aprendizado empregado pelos autores (SVM com kernel linear). O quarto motivo é que a execução de algoritmos genéticos com o SVM como função de aptidão é extremamente lenta e, como o kernel linear só necessita da refinação de um único parâmetro, a execução deste é mais rápida que com os outros kernels. O quinto e último motivo é que o experimento 1B (com kernel RBF) foi realizado na seção 6.5.3.1 e comparado com o 1A (com kernel linear), constatando que o linear é suficiente. Voltando às etapas, a sexta e última está associada à operação do classificador treinado na etapa anterior.

É importante destacar também que este framework foi expandido para os experimentos realizados sobre a base de dados Reuters 21.578 (seção 6.5.3.2). Neles foi acrescentada uma nova etapa após à execução do algortimo de agrupamento de características, a qual consiste na execução do algoritmo de redução de dimensionalidade PCA.

Capítulo 6. Estudo de Caso

O estudo de caso consistiu na utilização de três bases de dados distintas para teste das etapas propostas no capítulo anterior. O objetivo da utilização da primeira é tentar prever se uma determinada notícia sobre uma empresa com ações na bolsa de valores impacta positivamente ou negativamente as suas ações (a fim de testar os algoritmos desenvolvidos, apenas as ações e notícias sobre a empresa Petrobrás foram consideradas). A segunda e a terceira base é utilizada com o propósito de comparação com outros trabalhos presentes na literatura. Mais detalhes sobre as mesmas são dados na seção 6.1.

A fim de realizar a previsão, a técnicas adotadas foram o SVM com kernel linear (explicado no capítulo 5 o motivo da escolha deste kernel), Naive Bayes e K-vizinhos mais próximos. O texto foi representado usando a notação "bag of words" e os métodos aqui desenvolvidos para seleção de características (algoritmo genético) e redução de dimensionalidade (clusterização de variáveis) foram utilizados.

6.1 Descrição dos Dados

Três bases de dados distintas foram utilizadas nos experimentos deste trabalho. São elas: (1) uma base de notícias e cotações de empresas na bolsa de valores, (2) uma base de dados contendo avaliações, escritas em inglês, sobre filmes ([PANG & LEE, 2004]) e (3) a base de dados Reuters 21.578 que é comumente utilizada para medir o desempenho de técnicas de classificação de textos.

A primeira base pode ser dividida em duas partes: (1) as cotações das empresas listadas na bolsa e (2) as notícias sobre essas empresas. Para a obtenção das cotações, a API do Yahoo Finance ([YAHOO_FINANCE]) e a sua linguagem de consulta Yahoo Query Language ([YQL]) foram utilizadas. Já para a base de notícias, os dados foram extraídos da web utilizando técnicas de mineração de dados ([PIATETSKY-SHAPIRO, 1996], [BERRY & LINOFF. 1997]). Tanto as cotações como as notícias foram obtidas utilizando a linguagem *Java* ([JAVA], [GOSLING, 2000]).

Apesar de notícias e cotações de diversas empresas terem sido obtidas, apenas as referentes à Petrobrás foram utilizadas nos experimentos. Isso foi feito por dois motivos: (1) a quantidade

de notícias sobre a Petrobrás é maior do que sobre as outras empresas e (2) necessidade de limitar a quantidade de dados utilizada a fim de permitir com que os resultados dos experimentos fossem obtidos mais rapidamente.

O processo de extração de dados realizado para criação da primeira base é apresentado na figura 10. Inicialmente, o minerador de cotações se conecta à API do Yahoo Finance e obtém os dados e o minerador de notícias se conecta à internet e obtém os sites de interesse. Após isso, o minerador de cotações e o de notícias processam os dados e geram as cotações e notícias, respectivamente. Por fim, os dados gerados são armazenados na base de dados. Foram consideradas as notícias e cotações dos anos de 2013, 2014 e 2015. Inicialmente, foram obtidas 28.109 notícias que, depois da limpeza (eliminação de notícias com títulos com menos de três palavras, daquelas que claramente não eram notícias e eliminação de repetições), um total de 23.596 foram mantidas. Em relação às cotações, 720 foram extraídas.

A segunda base foi disponibilizada por PANG & LEE (2004) e já estava quase pronta para ser utilizada. Os únicos processamentos necessários para utilização dos dados foram: (1) a criação de um script para união das avaliações dos filmes, as quais estavam separadas por arquivos; (2) o embaralhamento dos dados e (3) a remoção de quebra de linhas existentes dentro de cada avaliação. Em relação à quantidade, a base em questão consiste de 2.000 avaliações positivas e 2.000 negativas.

A terceira e última base de dados utilizada, a Reuters 21.578, consiste de um conjunto de textos agrupados por tópicos. Nela, foi-se necessário, como em outros trabalhos, limitar os tópicos utilizados aos que continham maior número de amostras (neste trabalho, foram utilizados somente os seis maiores).

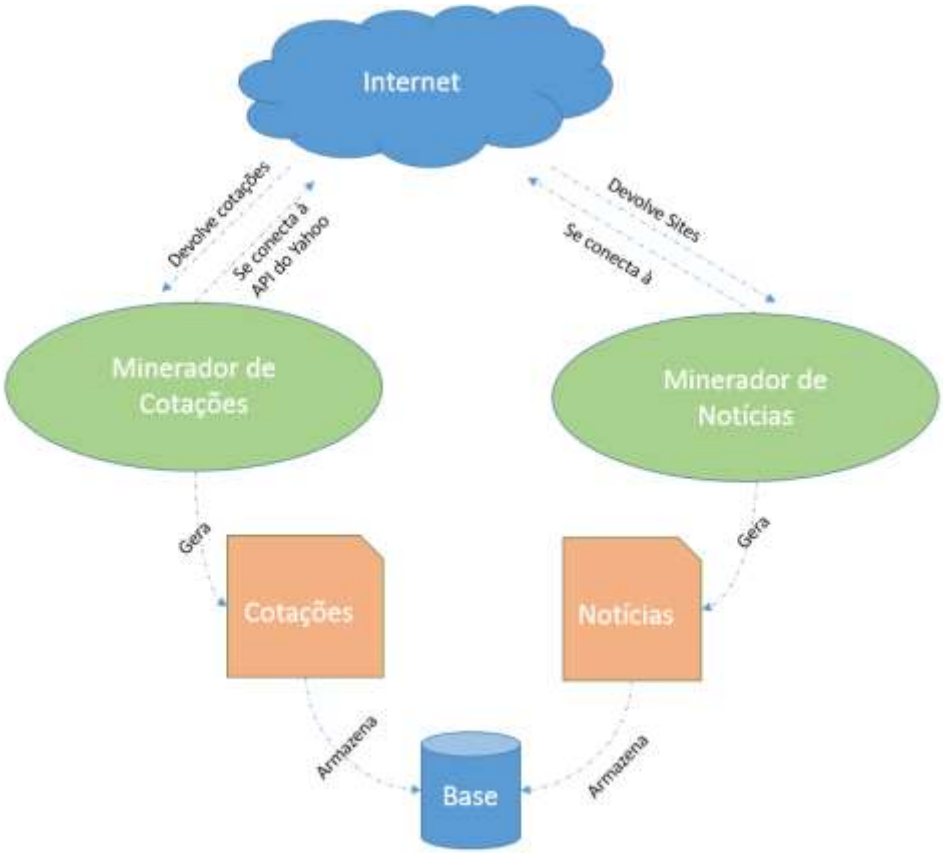

Figura 10: Processo de Extração dos Dados

6.2 Representação

Para construir as entradas do SVM, os dados foram representados usando a notação "bag of words". Na base de dados de notícias, todos os títulos (o texto não foi utilizado) de um mesmo dia foram agrupados para formar uma única entrada e, depois, foram representadas usando a notação citada. Em tal representação, o valor de cada dimensão dos vetores de entrada foi binário, ou seja, o bit 1 foi utilizado para indicar a presença de um token e o bit 0 para indicar a ausência. A saída foi representada por um único bit para cada entrada, representando se houve uma variação positiva ou negativa no preço da ação no dia posterior à publicação da notícia. Nas base de filmes, cada avaliação foi considerada como uma entrada e a representação foi feita da mesma forma (binária). A saída também foi codificada por um único bit e indicou se a avaliação foi positiva ou negativa. Já a base Reuters teve representação binária para os textos e suas saídas (seis ao total, sendo um para cada tópico) foram codificadas da mesma forma (indicando se a amostra pertence ou não ao tópico em questão).

A fim de evitar a construção de um conjunto de entrada com dimensão muito elevada, a técnica de stemização (stemming) foi utilizada. Tal técnica reduz as palavras para os seus

troncos (stem) a fim de evitar com que palavras flexionadas sejam consideradas como palavras distintas. Referimo-nos neste trabalho ao resultado deste processo como tokens.

A figura 11 ilustra o processo de representação das entradas para a base de notícias. Na primeira linha (marcada pelo círculo à esquerda e com o número 1 dentro), estão as notícias de um mesmo dia (por motivos didáticos apenas duas foram consideradas). Os títulos das notícias são concatenados e separados por pontos e, portanto, dão origem a uma única string (linha 2). Após isso, a string resultante é separada em tokens (linha 3). Cada um é referente a uma palavra da string após o processo de stemização. Por fim, um vetor binário é criado a fim de representar uma única entrada (linha 4). Na representação, bits 1 indicam a presença do token na string e bits 0 indicam a ausência. Repare que as stop words foram consideradas (não foram descartadas), uma vez que foi constatado, por meio de experimentos iniciais realizados neste trabalho, que a remoção delas acaba afetando ligeiramente o desempenho. Esta afirmação vai de encontro ao que diz MAAS et al. (2011), ou seja, que a remoção das mesmas não foi realizada devido existirem certas stop words que são indicativas de sentimento.

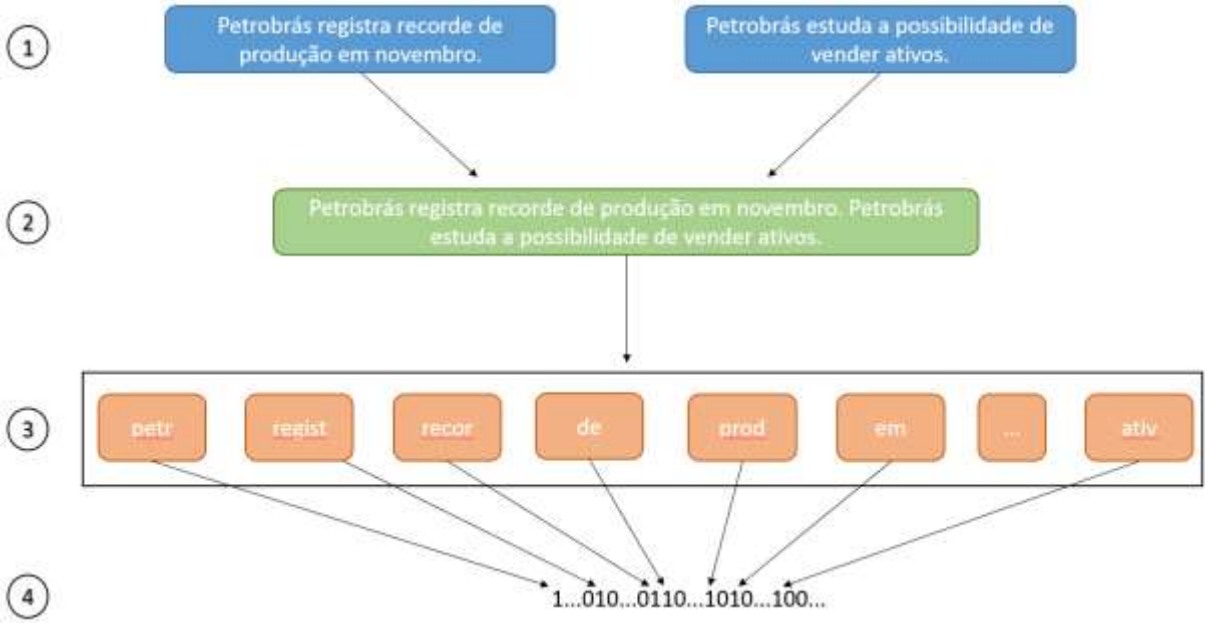

Figura 11: Ilustração de como as entradas foram representadas

6.3 Divisão dos dados

Para a execução destes experimentos, foi realizada a validação cruzada 10-fold executada 10 vezes. No fold de treinamento, os algoritmos de seleção de características e de clusterização de variáveis foram executados, uma vez que ambos são supervisionados e não poderiam ser executados no conjunto de dados por completo. Como já dito anteriormente, para o algoritmo de clusterização de variáveis, a função de avaliação do AG foi a repetição de 10 vezes a validação cruzada 10-fold usando o método de aprendizado SVM. Importante ressaltar novamente que esta validação foi feita apenas sobre o conjunto de treinamento, ou seja, o algoritmo não usou em momento algum o conjunto de testes para a seleção de características. Importante também deixar claro que as validações cruzadas executadas foram distintas, ou seja, uma validação mais interna foi utilizada para cálculo da função fitness usando o SVM e uma mais externa foi executada para calcular o desempenho geral dos algortimos de seleção de características e de agrupamento de variáveis aqui introduzidos.

6.4 Ambiente

Para a extração de notícias, coleta das cotações das ações, desenvolvimento dos algoritmos e etapas propostos e execução dos experimentos, a linguagem de programação *Java* ([JAVA]) foi utilizada. Para armazenamento das notícias e cotações, o sistema gerenciador de banco de dados *Postgresql* ([POSTGRESQL]) foi utilizado. Em relação à máquina para realização dos experimentos, a tabela 3 contém detalhes desta.

Processador	Intel Core i7 (4 núcleos e 8 simulados de 3.4 GHz)
Memória RAM	24 GB
Memória Secundária	HD de 500 GB
Sistema Operacional	Windows 7 (64 bits)

Tabela 3: Informações sobre a máquina utilizada

6.5 Experimentos

Os experimentos foram divididos em quatro seções a fim de facilitar o entendimento de tudo que foi feito neste trabalho. A seção 6.5.1 demonstra os experimentos que foram

realizados com o algoritmo genético de seleção de características a fim de possibilitar, através de visualização gráfica, a verificação dos benefícios introduzidos pelos componentes propostos. A seção 6.5.2 contém os experimentos para avaliação das medidas da clusterização de características. A seção 6.5.3 visa apresentar o desempenho das etapas desenvolvidas e descritas no capítulo 5 e a seção 6.5.4 apresenta o desempenho quando utilizadno apenas o PCA. Além destas, é apresentada uma quinta seção (6.5.5) a fim de resumir os resultados obtidos em todos os experimentos executados.

6.5.1 Experimentos para seleção de características

Esta seção visa mostrar a eficácia do algoritmo de seleção de características genético aqui proposto e foi dividida em duas outras seções. A primeira apresenta os experimentos realizados com o intuito de refinar os parâmetros utilizados no AG (taxa de elitismo e taxa de crossover). Já a segunda apresenta os experimentos necessários para testar e computar estatísticas sobre a mutação baseada na normalização do coeficiente de correlação de Pearson e da geração da população inicial proposta.

Como o intuito desta seção é apenas refinar os parâmetros do AG proposto e mostrar a sua eficácia, apenas a base de dados de filmes foi utilizada para este propósito. O único parâmetro fixado foi o número de indivíduos da população que foi de 20. Este foi escolhido por algumas razões: (1) os trabalhos de seleção de características que utilizam AGs ([FROHLICH et al., 2003], [LEARDI, 2000], [YANG & HONAVAR, 1998]) utilizam poucos indivíduos devido ao alto custo de computação das funções de avaliação de uma população (fixam este valor em 20 ou 30); (2) este foi baseado em trabalhos como estes, ou seja, que utilizam AGs para seleção de características.

6.5.1.1 Ajuste dos parâmetros do AG proposto

A fim de ajustar os parâmetros do AG, 10 execuções da validação cruzada 10-fold foram executadas. Os parâmetros ajustados foram cinco: a taxa de elitismo, a taxa de cruzamento, o número de iterações, o limite inferior do operador de mutação e o limite superior do mesmo. De acordo com CHAN & TANSRI (1994), típicas probabilidades de crossover estão compreendidas entre 0,5 e 1. Desta forma, foram testadas as taxas de cruzamento “0,5”, “0,6”, “0,7”, “0,8”, “0,85”, “0,9” e “0,95”, os limites inferiores da taxa de mutação “0,002”, “0,003”, “0,02”, “0,03”, “0,04” e “0,05” e os limites superiores da mesma taxa “0,04”, “0,05”, “0,06”, “0,07”, “0,08”, “0,1”, “0,2” e “0,3”.

Já para a probabilidade de elitismo, baseado na literatura ([BHASKAR & MAHESWARAPU, 2013], [KUMARI & SYDULU, 2009], [DEVI & KRISHNA, 2008] e [FARD et al., 2006]), valores no intervalo 0,1 a 0,5 foram testados, mais especificamente “0,1”, “0,2”, “0,3”, “0,4” e “0,5”.

Para cada parâmetro, um gráfico contendo as médias das 10 execuções da validação cruzada para cada um dos valores testados será apresentado. Por exemplo, para a taxa de cruzamento, 7 valores foram testados e, portanto, o gráfico associado a tal parâmetro contém 7 linhas (uma para cada valor), conforme pode ser constatado no gráfico da figura 12.

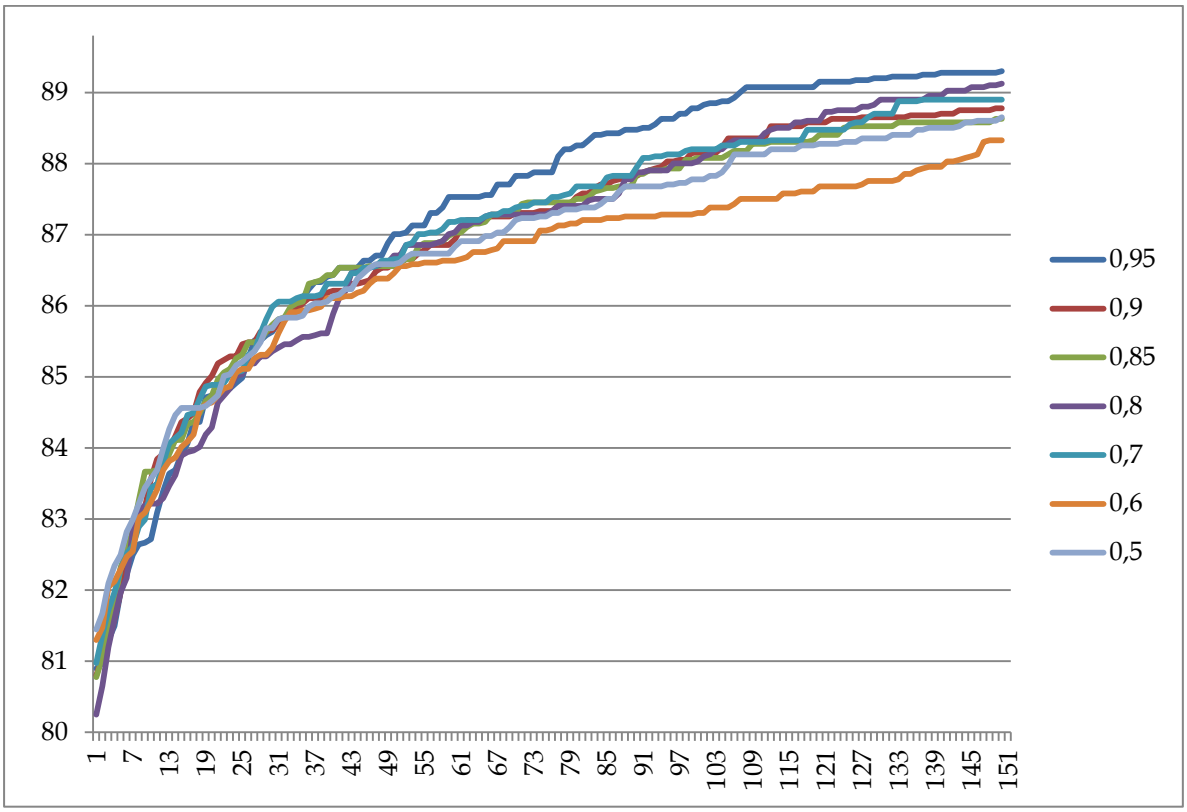

Figura 12: Gráfico com o desempenho do AG para diferentes taxas de cruzamento. Eixo das abscissas: iterações. Eixo das ordenadas: desempenho obtido com 10 execuções da validação cruzada 10-fold.

Repare no gráfico que a taxa de 0,95 foi a melhor, uma vez que teve desempenho superior às demais taxas por um número considerável de iterações e atingiu o maior valor com o número de iterações consideradas. Note também que o número de iterações foi pequeno e que isto não importa, já que é possível notar a superioridade da taxa de 0,95. O curto número de iterações foi utilizado neste experimento e em alguns outros, uma vez que isto reduz o tempo

necessário para executá-los e não atrapalha na identificação dos melhores valores de parâmetros.

Seguindo o mesmo raciocínio, o gráfico representado pela figura 13 tem como finalidade descobrir a melhor taxa de elitismo. Repare que a linha roxa, referente à taxa de 0,2, apresenta desempenho superior às demais na maior parte do tempo.

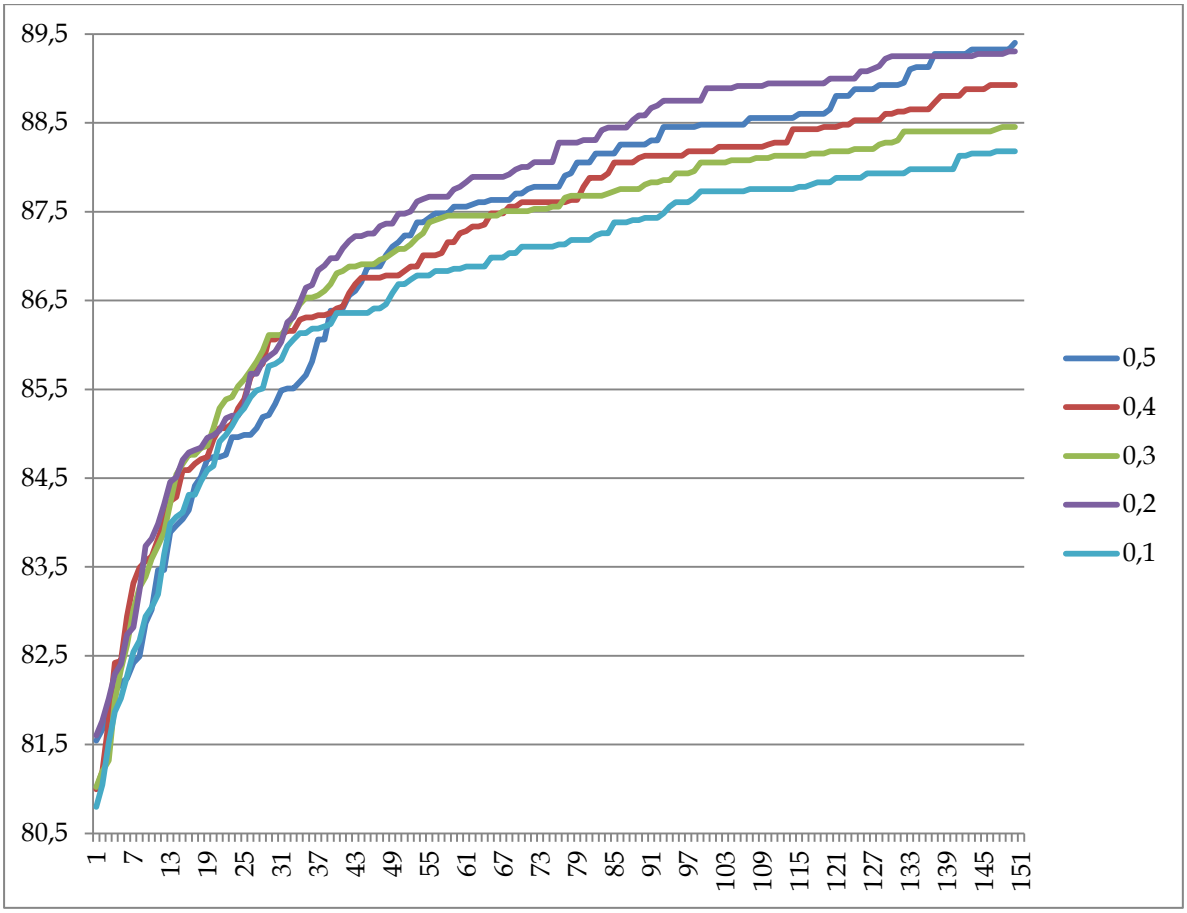

Figura 13: Gráfico com o desempenho do AG para diferentes taxas de elitismo. Eixo das abscissas: iterações. Eixo das ordenadas: desempenho obtido com 10 execuções da validação cruzada 10-fold.

Da mesma forma, os gráficos das figuras 14 e 15 têm como intenções permitir a escolha das melhores taxas inferior e superior do operador de mutação, respectivamente. Como visto anteriormente, o operador definido neste trabalho usa uma taxa dinâmica que varia do limite inferior ao superior. Da interpretação dos gráficos, é possível verificar que o melhor limite inferior é 0,04 e superior é 0,07.

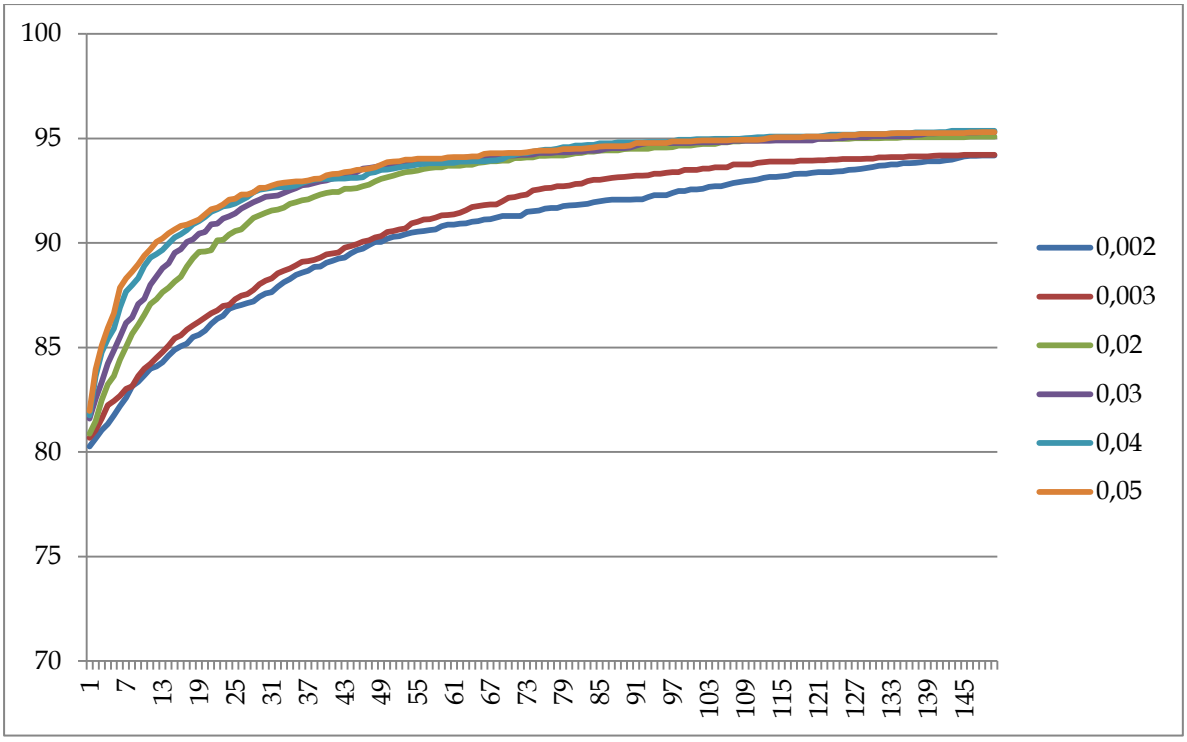

Figura 14: Gráfico com o desempenho do AG para diferentes limites inferiores do operador de mutação. Eixo das abscissas: iterações. Eixo das ordenadas: desempenho obtido com 10 execuções da validação cruzada 10-fold.

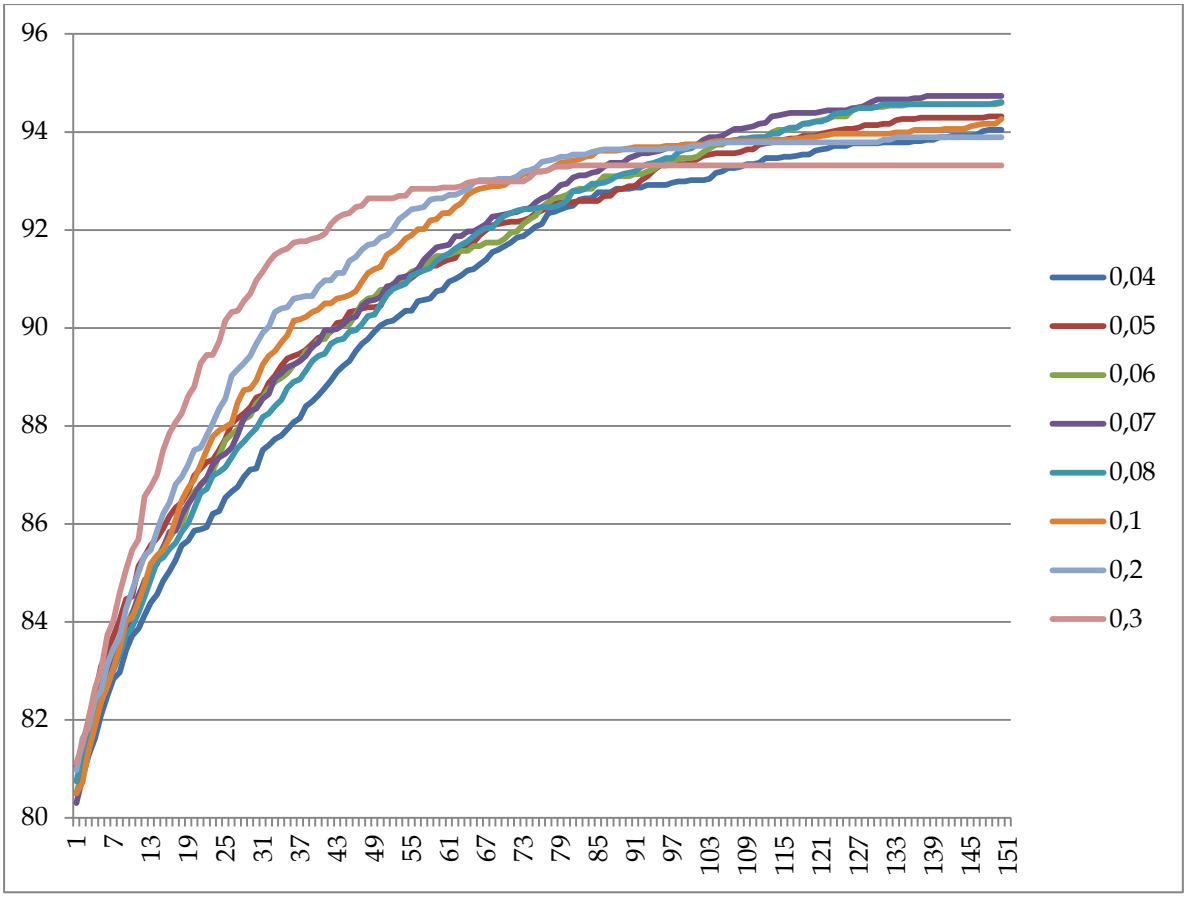

Figura 15: Gráfico com o desempenho do AG para diferentes limites superiores do operador de mutação. Eixo das abscissas: iterações. Eixo das ordenadas: desempenho obtido com 10 execuções da validação cruzada 10-fold.

O último parâmetro refinado foi o número de iterações do AG. Ele foi deixado por último com o intuito de considerar o número de iterações necessárias para o AG já otimizado, ou seja, com os demais parâmetros já escolhidos. O gráfico da figura 16 foi desenvolvido para permitir escolher o melhor valor para tal parâmetro. Nele, é possível visualizar que até a última iteração, soluções melhores são encontradas. Mas, apesar disso, é notável que, a partir, aproximadamente, da iteração 160, a taxa de descoberta de novas soluções diminui. Em outras palavras, a partir deste momento, mais iterações são necessárias para encontrar indivíduos melhores. Com base nisso, definiu-se como número máximo de gerações do AG o valor de 200 iterações (aproximação para cima do valor 160).

Também com o intuito de definir aquele parâmetro, o gráfico da figura 17 foi construído. Nele, as variações médias de desempenho entre iterações consecutivas são apresentadas. É possível confirmar que, após a iteração 160, a descoberta de melhores indivíduos se torna mais

lenta, reafirmando, portanto, que este pode ser um ponto a ser considerado como o número máximo de iterações.

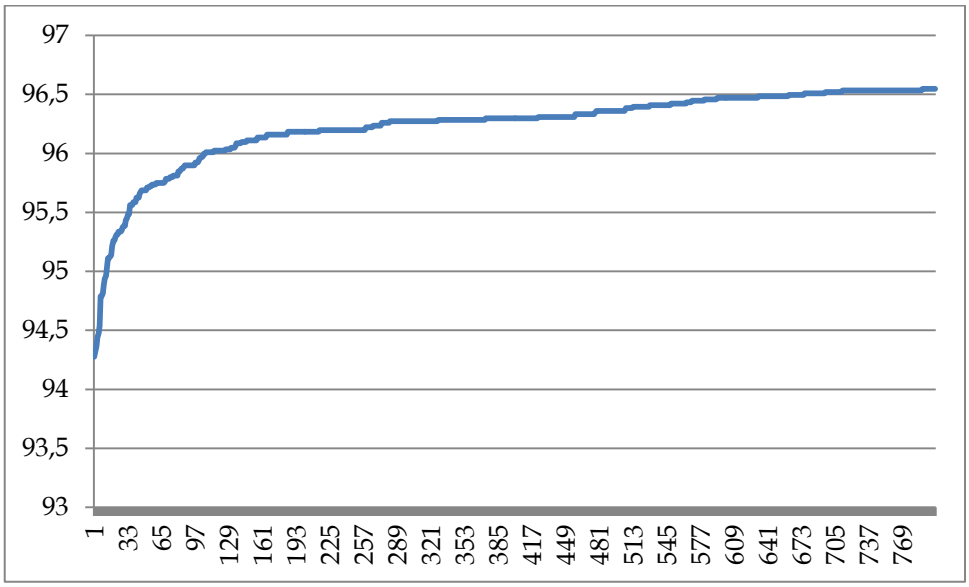

Figura 16: Gráfico com o desempenho do AG usado para definir o número de iterações a ser utilizado. Eixo das abscissas: iterações. Eixo das ordenadas: desempenho obtido com 10 execuções da validação cruzada 10-fold.

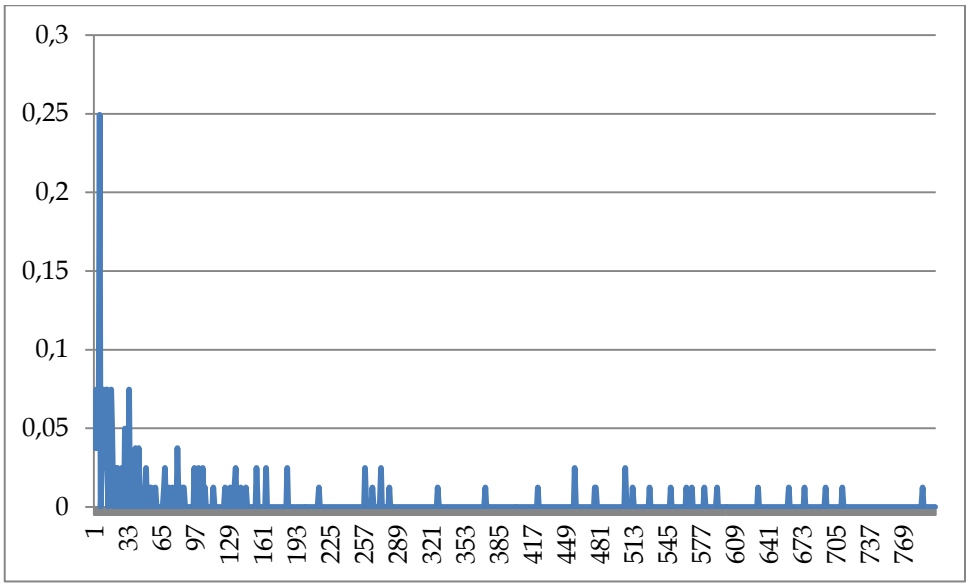

Figura 17: Gráfico com a variação média de desempenho em cada iteração do AG (10 execuções da validação cruzada 10-fold). Eixo das abscissas: iterações. Eixo das ordenadas: variação média do desempenho na iteração.

A fim de resumir os parâmetros refinados através dos experimentos acima, a tabela abaixo foi desenvolvida. Nela, os parâmetros com os seus respetivos valores foram especificados.

Parâmetro	**Valor**
Taxa de Elitismo	0,2
Taxa de mutação inferior	0,04
Taxa de mutação superior	0,07
Taxa de cruzamento	0,95
Número máximo de iterações	200

Tabela 4: Parâmetros dos AGs definidos após refinamento

6.5.1.2 Avaliação dos componentes do AG proposto

Nesta seção dois componentes do AG proposto foram testados: o operador de mutação e a geração da população inicial. Para o teste, foi elaborado um gráfico (figura 18) contendo o resultado de quatro experimentos. O primeiro está relacionado à execução do AG básico, ou seja, aquele que não utiliza as inovações trazidas neste trabalho (operador de mutação e geração da população inicial baseados em Pearson). O segundo acrescenta ao mesmo o operador de mutação proposto. O terceiro acrescenta ao AG básico a geração da população inicial e, por fim, o último acrescenta ambos os componentes.

No gráfico, a média das 10 execuções da validação cruzada 10-fold de cada experimento está relacionada a uma linha do gráfico. Como pode ser constatado, o acréscimo dos componentes propostos ao AG básico possibilita alcançar um desempenho superior. A geração da população inicial permite partir de desempenhos altos e a mutação proposta faz com que soluções melhores sejam exploradas mais rapidamente.

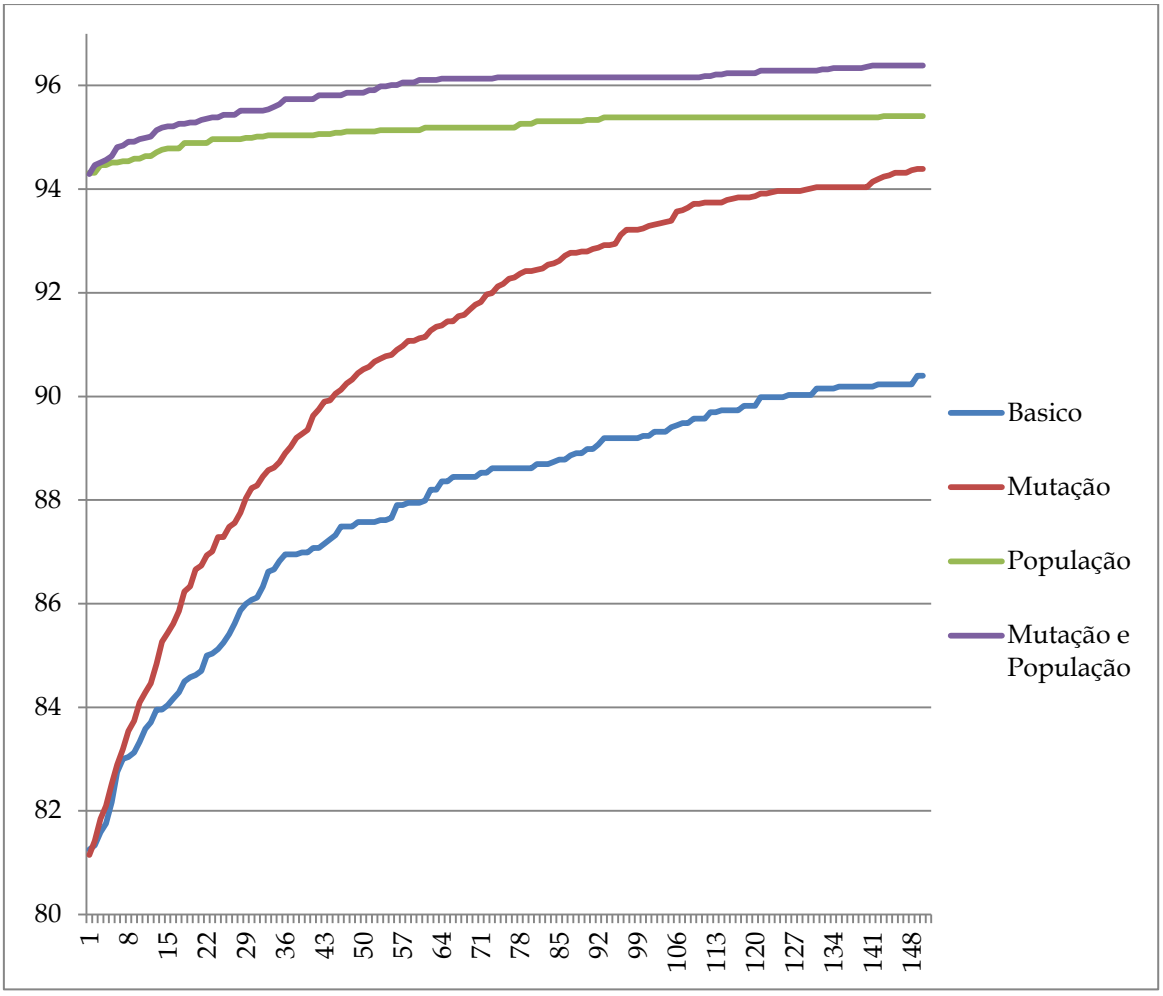

Figura 18: Gráfico comparativo entre o AG básico, o AG básico + mutação proposta, o AG básico + geração da população proposta e o AG básico + geração + mutação. Eixo das abscissas: iterações. Eixo das ordenadas: desempenho obtido com 10 execuções da validação cruzada 10-fold.

Ainda a partir do gráfico, é possível ver, ao comparar a linha azul (primeira de baixo para cima) com a linha vermelha (segunda de baixo para cima) e a verde (terceira de baixo para cima) com a roxa (quarta de baixo para cima), que a inclusão do operador de mutação proposto permite alcançar soluções melhores mais rapidamente.

Ao comparar as linhas azul e vermelha com as verde e roxa, nota-se que a geração da população inicial proposta permite partir de soluções melhores. Esse ponto de partida é tão bom que, nas 150 iterações consideradas, o desempenho inicial dos experimentos que utilizam a geração da população proposta (linhas verde e roxa) é melhor que o desempenho final daqueles que não a consideram (linhas azul e vermelha).

6.5.2 Experimentos para avaliação das medidas da clusterização de características

Algumas observações são necessárias para o correto entendimento desta seção. A primeira é que o objetivo da mesma é avaliar as medidas da clusterização de variáveis e avaliar o comportamento do algoritmo para diferentes valores de parâmetros. A segunda é que o conjunto de treinamento é utilizado para esta tarefa, dado que a utilização do conjunto de testes poderia tornar o resultado viesado. A terceira e última é que, como nas seções anteriores, a base de dados reduzida é usada.

Foram dois os parâmetros considerados nesta seção: (1) o número máximo de elementos por conjunto e (2) a porcentagem de características permitidas a serem unidas. Para isso, foram construídos os gráficos representados nas figuras 19, 20 e 21. Neles, cada curva representa um valor do parâmetro 2, o eixo das abscissas representa o valor do parâmetro 1 e o eixo das ordenadas representa o desempenho alcançado no conjunto de testes com a combinação dos dois parâmetros. Eles diferem entre si, pois a medida utilizada é diferente: a figura 22 é referente à afinidade, a figura 20 à informação mútua e a figura 21 ao log likelihood. Observe que os desempenhos apresentados nos gráficos são obtidos no conjunto de treinamento e não no de testes, uma vez que o intuito é refinar os parâmetros sem comprometer a confiabilidade dos resultados.

Nos gráficos citados, existe uma linha com o rótulo 0 (preta e tracejada), a qual serve como referência (linha de base), uma vez que ela representa o desempenho sem a realização da clusterização de variáveis. Desta forma, valores situados abaixo desta linha pioram o desempenho enquanto valores acima o melhoram. É possível ver, através da análise dos gráficos, que os melhores desempenhos obtidos no conjunto de treinamento foram 96,26% (669 de 3150 variáveis), 95,76% (499 de 3178 variáveis) e 98,50% (211 de 3140 variáveis) para as medidas de afinidade, informação mútua e log likelihood, respectivamente. Da análise dos gráficos, também é possível constatar que a medida de log likelihood permite uma maior redução de dimensionalidade (já que o desempenho melhora à medida que mais itens são permitidos por conjuntos) e ainda possibilita obter melhores desempenhos no conjunto de treinamento do problema. O resultado no conjunto de testes é disponibilizado na próxima seção.

É importante ressaltar os dois pontos que levam a indicar que a medida de log likehood é superior às demais. A primeira é referente ao comportamento do gráfico relativo a esta medida

(figura 21). O desempenho melhora conforme aumentamos os valores de ambos os parâmetros. O segundo ponto, o qual é consequência do primeiro, é que a redução de dimensionalidade é maior, o que pode ser visto ao notar que, para o melhor resultado, a redução foi de 3140 para 211 variáveis (aproximadamente 93%).

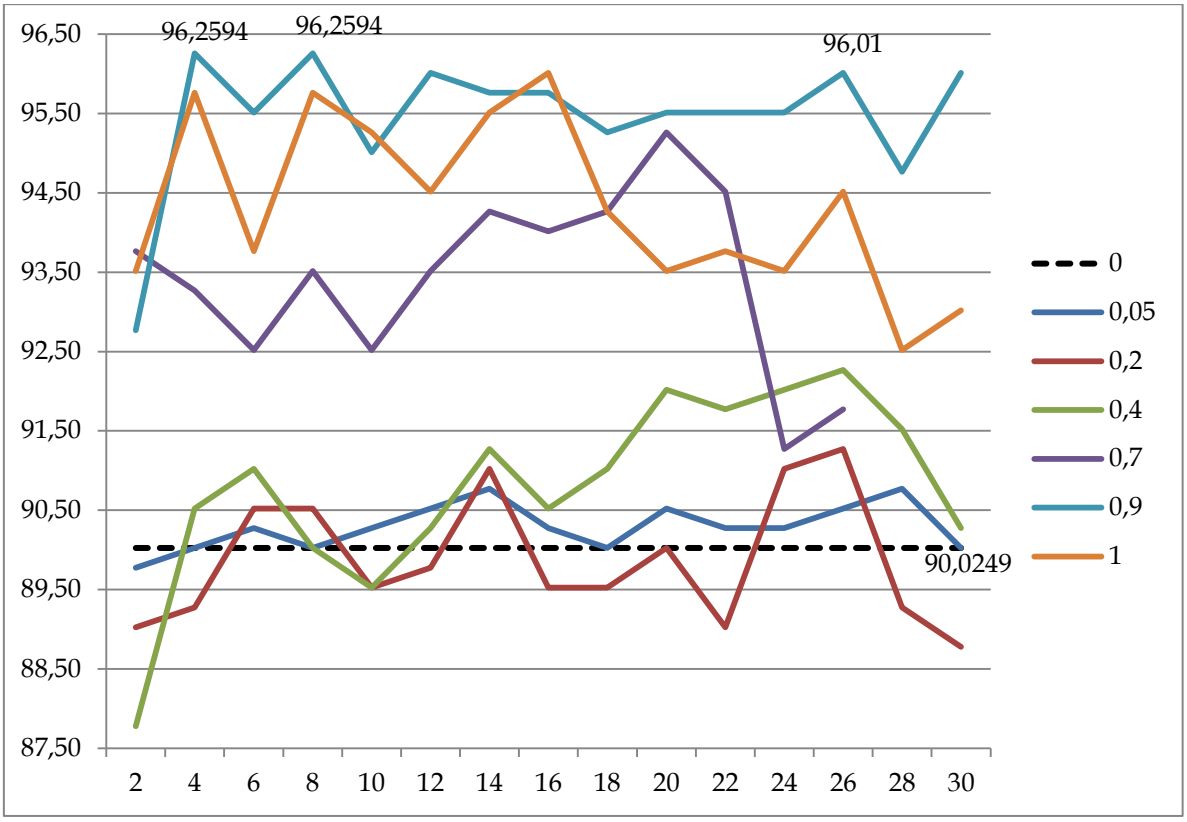

Figura 19: Evolução do desempenho para cada combinação de parâmetros do algoritmo de clusterização de variáveis usando a medida afinidade. Linhas: porcentagem de características permitidas a serem unidas; Eixo das abscissas: número máximo de elementos por conjunto; Eixo das ordenadas: desempenho.

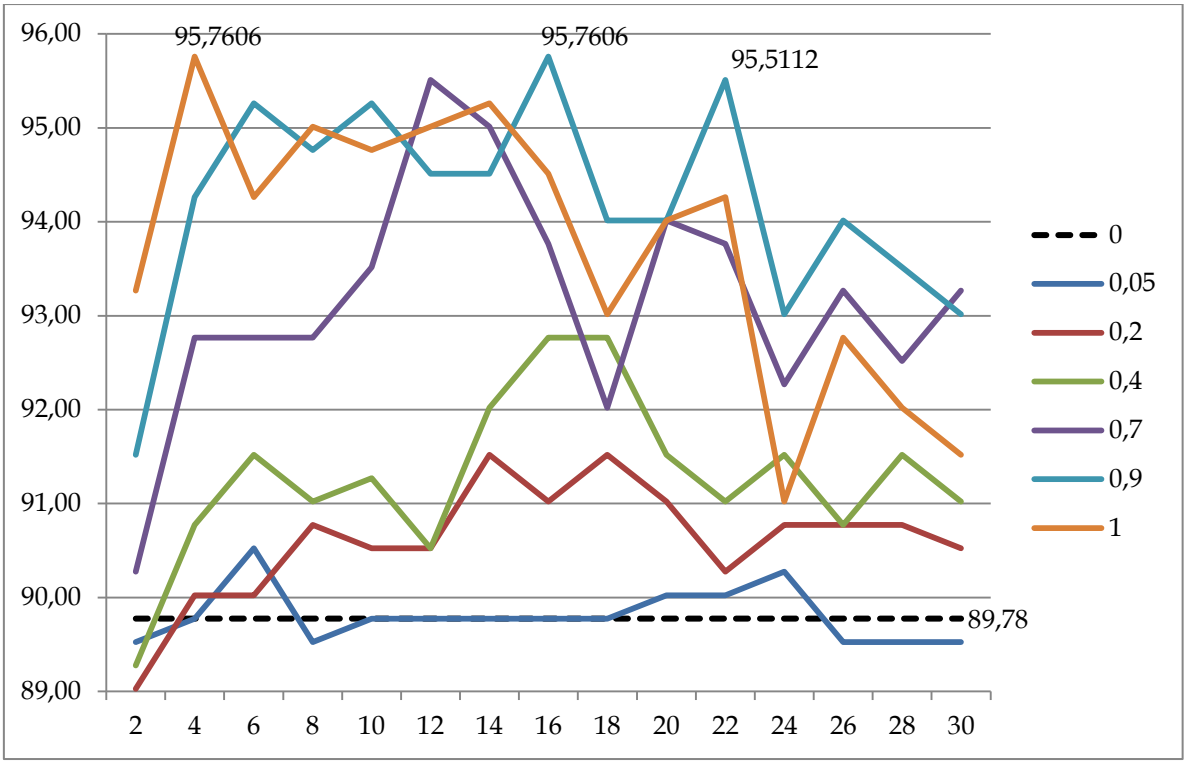

Figura 20: Evolução do desempenho para cada combinação de parâmetros do algoritmo de clusterização de variáveis usando a medida informação mútua. Linhas: porcentagem de características permitidas a serem unidas; Eixo das abscissas: número máximo de elementos por conjunto; Eixo das ordenadas: desempenho.

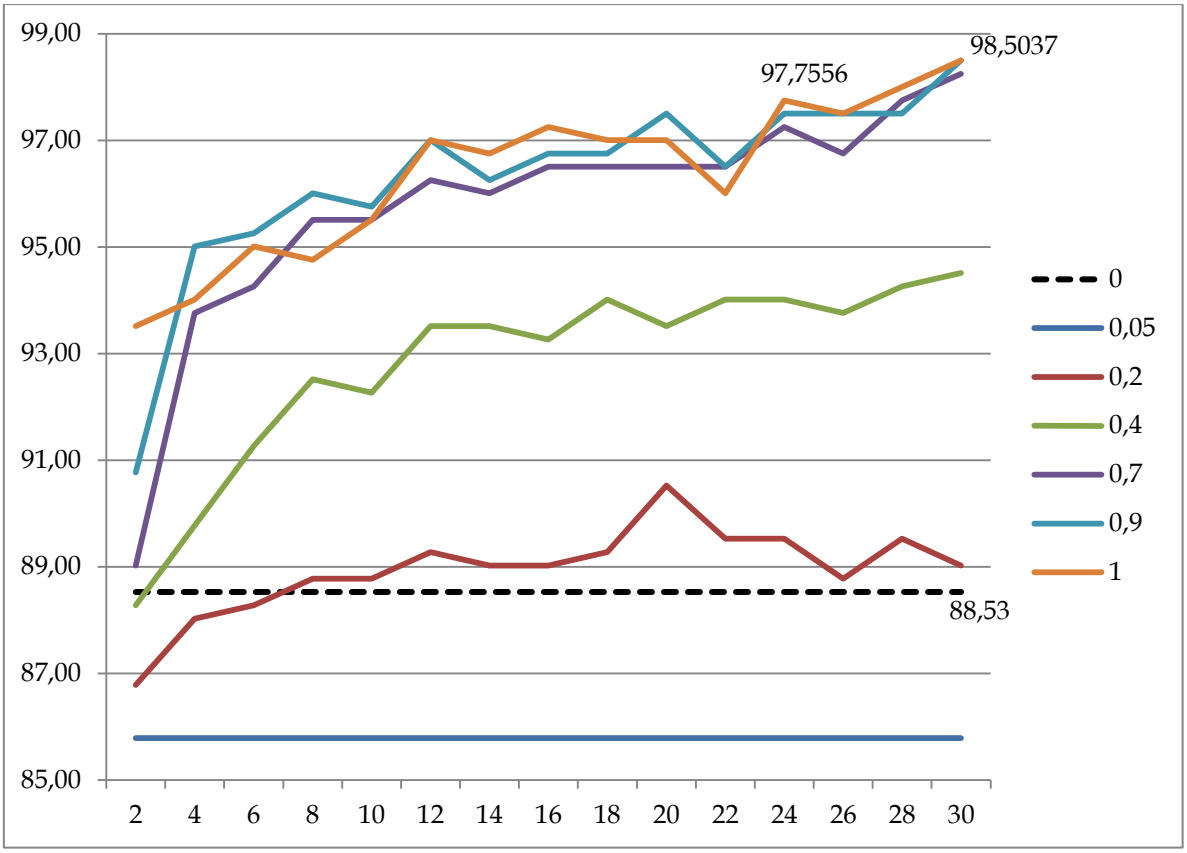

Figura 21: Evolução do desempenho para cada combinação de parâmetros do algoritmo de clusterização de variáveis usando a medida log likelihood. Linhas: porcentagem de características permitidas a serem unidas; Eixo das abscissas: número máximo de elementos por conjunto; Eixo das ordenadas: desempenho.

6.5.3 Experimentos para avaliação das etapas propostas

Esta seção foi dividida em quatro subseções. As três primeiras têm como finalidade separar os experimentos realizados sobre a base de dados de filmes daqueles realizados sobre base Reuters 21.578 e sobre a de notícias. A primeira (base de filmes) e a segunda (Reuters 21.578) possui como intuito possibilitar a comparação com outros trabalhos realizados na literatura, uma vez que as bases em questão são utilizadas por outros autores. Já a terceira (base de notícias) é uma aplicação dos algoritmos aqui desenvolvidos em notícias a respeito da empresa Petrobrás. O propósito é tentar prever a sua polaridade. A última seção (6.5.3.4) foi desenvolvida para comparar os resultados do framework aqui proposto com aqueles obtidos para o algoritmo de redução de dimensionalidade PCA.

6.5.3.1 Experimentos com a base de avaliação de filmes

Nesta seção, seis experimentos foram realizados na base de avaliação de filmes a fim de mostrar a evolução de desempenho das etapas propostas. O primeiro serve como linha de base para o segundo, o segundo para o terceiro e, assim por diante, até o último. Os experimentos começam com a aplicação dos algoritmos de aprendizado sobre a entrada bruta (este foi dividido em dois A e B a fim de comparar os kernels linear e RBF), ou seja, sem o procedimento de seleção e de clusterização de variáveis. O segundo inclui no primeiro o processo de seleção de características proposto neste trabalho.

Seguindo para os próximos experimentos, os terceiro, quarto e quinto incluem no segundo experimento a clusterização de variáveis e só diferem entre si na métrica de agrupamento utilizada. São as medidas de afinidade, informação mútua e log likelihood, respectivamente. Repare que estes três experimentos utilizam a seleção de características seguida da clusterização. Eles foram criados para decidir pela melhor métrica utilizada e verificar o desempenho das etapas propostas.

A base utilizada aqui é muito usada na literatura para a realização da análise de sentimento. O intuito dos experimentos é possibilitar a comparação com outros trabalhos que tratam do tema. A base utilizada nos experimentos foi a mesma utilizada e disponibilizada por Pang & Lee (2004). Eles criaram um detector de subjetividade a fim de eliminar das avaliações dos filmes as sentenças que não expressam a opinião do autor. Após a eliminação, eles utilizaram as avaliações modificadas para detectar polaridade. Eles utilizaram os métodos de aprendizado SVM e Naive Bayes e as entradas foram representadas com bits que indicam a presença ou ausência dos tokens encontrados na base de dados. O melhor resultado obtido por eles foi de 87,20% quando utilizando o SVM para prever a polaridade.

Os resultados dos experimentos realizados nesta seção estão representados na Tabela 5 abaixo. Na Tabela, constam os resultados de 18 casos. Cada coluna da tabela retrata o resultado para um determinado algoritmo de aprendizado dentre os considerados neste trabalho (SVM, Naive Bayes e K Vizinhos mais próximos). Cada linha é relativa a um experimento e visa apresentar os resultados sem ou com o uso dos algoritmos que foram implementados e estão sendo analisados neste trabalho (seleção de características usando algoritmos genéticos e clusterização de variáveis). Como exemplo, considere a primeira linha e primeira coluna, o qual representa o primeiro caso do primeiro experimento. Nele, só o algoritmo de aprendizado é aplicado, ou seja, sem a utilização de nenhuma das duas técnicas implementadas. Já os casos do experimento 2 utilizam o resultado do algoritmo de aprendizado após a utilização do

algoritmo genético para seleção de características. O mesmo raciocínio pode ser utilizado para os demais casos.

Exp	Algoritmo	Dimensões	Tempo	Teste (SVM)	Teste (NB)	Teste (k_vizinhos)
1A	Só Aprendizado (SVM - kernel linear)	6.875 (100%)	49s (100%)	85,14%	85,40%	53,56%
1B	(SVM - kernel RBF)	6.875 (100%)	52s (100%)	85,09%	85,53%	53,61%
2	Algoritmo Genético (AG)	2.823 (41,06%)	21s (42,85%)	85,20%	80,20%	58,84%
3	Clusterização de Variáveis	2.800 (40,72%)	20s (40,82%)	86,98%	80,17%	68,49%
3	AG + Clusterização de Variáveis usando Afinidade	1.324 (19,25%)	11s (22,45%)	87,39%	82,42%	72,13%
4	AG + Clusterização de Variáveis usando Informação Mútua	1012 (14,72%)	10s (20,41%)	86,98%	82,19%	72,01%
5	AG + Clusterização de Variáveis usando Log Likelihood	213 (3,1%)	6s (12,24%)	86,58%	82,26%	71,90%

Tabela 5: Tabela de resultados dos experimentos com a base de filmes. As acurácias são apresentadas nas três últimas colunas.

É importante destacar que o único experimento com kernel não linear foi o 1B, pois ele foi realizado apenas com o intuito de verificar a necessidade ou não da utilização de outro kernel. Apesar da literatura mostrar que o linear é suficiente (conforme destacado no capítulo 5), o experimento em questão foi realizado. Como os experimentos 1A e 1B tiveram desempenhos semelhantes, foi optado pelo linear para a execução dos demais (pois tem complexidade menor e roda mais rápido que outros kernels).

Analisando a tabela, o melhor resultado obtido é ligeiramente, mas não significantemente, superior ao obtido por Pang & Lee (2004) (87,39% contra 87,20%). Ele é referente à aplicação, em conjunto, dos dois algoritmos construídos neste trabalho (seleção de características e clusterização de variáveis) e, como dito anteriormente, foi atingida uma taxa de acerto de 87,39% usando a técnica de aprendizado SVM e a medida denominada Afinidade.

Alguns pontos que devem ser notados ao analisar os resultados dos experimentos:

a. O desempenho ao usar o SVM, como já esperado, foi superior ao Naive Bayes e K-Vizinhos mais próximos.
b. Os resultados obtidos com a clusterização de variáveis, quando utilizada a medida Log Likelihood, foram ligeiramente, mas não significantemente, inferiores aos resultados da medida de Afinidade e Informação Mútua. Apesar disso, a superioridade da medida, conforme constatado na seção 6.5.2, é verificada ao analisar o potencial de redução da dimensionalidade dos dados. Enquanto a medida Log Likelihood reduz a entrada para 213 variáveis (3,1%), a Afinidade e Informação Mútua reduzem para 1.324 (19,25%) e 1.012 (14,72%), respectivamente.
c. A técnica de aprendizado utilizando os k vizinhos mais próximos tornou-se mais eficaz ao utilizar as duas técnicas apresentadas neste trabalho (seleção de características e agrupamento de variáveis), levando a taxa de acerto de 53,56% a 72,13%.
d. Não houve melhora quando utilizada a técnica de aprendizado Naive Bayes, mas, isto não necessariamente invalida os experimentos ou impede que os algoritmos desenvolvidos sejam utilizados, uma vez que, dependendo do intuito, a boa redução da dimensionalidade pode compensar a perda de desempenho.

Destaca-se como contribuições principais deste trabalho: (1) introdução de dois métodos que podem ser utilizados em outros campos além da análise de sentimento (seleção de características e clusterização de variáveis); (2) grande redução no número de características (de 6.875, 100%, para 213, 3,1%), implicando em menor tempo necessário na operação do

método de aprendizado e (3) redução de dimensionalidade com perda mínima de informações, uma vez que a clusterização não elimina variáveis (apenas agrupa-as).

6.5.3.2 Experimentos com a base Reuters 21.578

Nesta seção, todos os experimentos descritos foram executados para a base Reuters. Foram considerados somente os 6 tópicos com mais amostras e a quantidade total e por tópicos de cada um são apresentados na tabela abaixo.

Tópico	**Quantidade**
Earn	3713
Acquision	2055
Crude	321
Trade	298
Money-Fx	245
Interest	197
Total	6.829

Tabela 6: Quantidade de amostras consideradas em cada tópico – base Reuters 21.578

Como nos demais experimentos e já informado anteriormente, as entradas foram codificadas usando a notação bag of words onde cada bit indica se determinado token está ou não presente na amostra (texto) considerada. As saídas foram codificadas usando seis variáveis onde o valor das mesmas é binário, representando se a amostra pertence ou não ao tópico em questão. Além disso, foram realizadas etapas de pré-processamento como a normalização dos dados, o steeming e a remoção de variáveis que apareceram menos que duas vezes no conjunto de treinamento. Foram utilizados os algortimos de aprendizado K-vizinhos mais próximos e o SVM.

Na tabela abaixo diferentes casos são considerados: (1) com as características originais; (2) com as obtidas após a execução do algortitmo de seleção de características, (3) com as obtidas com ambos os algoritmos e (4) com os dois algoritmos mais o PCA. Para todos esses

casos, a taxa de acerto (A), precision (P), recall (R), f1 (F1), quantidade de dimensões (D) e tempo de execução de uma validação cruzada (T) são apresentadas para cada algortimo de aprendizado considerado (SVM e k-vizinhos mais próximos).

Exp	**Algoritmo**	**Teste (SVM)**	**Teste (k_vizinhos)**
1	Só aprendizado	A: 0.983839 P: 0.958994 R: 0.942724 F1: 0.950435 D:8246 (100%) T: 52s (100%)	A: 0.934201 P: 0.937583 R: 0.930936 F1: 0.933249 D:8254 (100%) T: 28s (100%)
2	Seleção	A: 0.984006 P: 0.962934 R: 0.94432 F1: 0.953117 D: 6682 (81%) T: 33s (62%)	A: 0.959778 P: 0.959116 R: 0.959961 F1: 0.959507 D: 2384 (29%) T: 20s (69%)
3	Seleção + Agrupamento	A: 0.986780 P: 0.967363 R: 0.960490 F1: 0.96380 D: 1283 (16%) T: 10s (19%)	A: 0.95964 P: 0.958963 R: 0.959865 F1: 0.959379 D: 902 (11%) T: 17s (61%)
4	Seleção + Agrupamento + PCA	A: 0.982399 P: 0.989061 R: 0.981535 F1: 0.9852841	A: 0.979625 P: 0.978319 R: 0.978362 F1: 0.978340

		D: 151 (1,83%) T: 5s (9,6%)	D: 31 (0,38%) T: 2s (7,14%)

Tabela 7: Resultados obtidos – base Reuters 21.578 (A: acerto, P: precision, R: recall, D: dimensões e T: tempo).

Como propósito de comparação, três dos principais trabalhos presentes na literatura que envolvem seleção de características utilizando algoritmos genéticos foram tomados como base ([BHARTI & SINGH, 2015], [UĞUZ, 2011] e [AGHDAM, 2009]).

A tabela abaixo mostra uma comparação entre os melhores resultados encontrados neste trabalho com aqueles que estão servindo de comparação. Todos os três utilizam a mesma base de dados, ou seja, a Reuters 21.578. Logo em seguida, uma comparação restrita ao potencial de redução de dimensionalidade é apresentada. Nela, DI representa a quantidade de Dimensões Iniciais, ou seja, sem o uso de nenhuma técnica para reduzir a quantidade de características e DF de Dimesões Finais, ou seja, o número de características obtido quando o trabalho em questão obteve o melhor desempenho (em termos de precision, recall e F1).

Este trabalho	**[BHARTI & SINGH, 2015]**	**[UĞUZ, 2011]**	**[AGHDAM, 2009]**
P: 0.9890 R: 0.9815 F1: 0.9852	P: 0.6752 R: 0.3790 F1: 0.4855	P: 0.9817 R: 0.9752 F1: 0.9784	P: 0.7713 R: 0.7975 F1: 0.7842

Tabela 8: Resultados obtidos comparados a outros trabalhos que realizam a seleção de características (P: precision e R: recall).

Este trabalho	**[BHARTI & SINGH, 2015]**	**[UĞUZ, 2011]**	**[AGHDAM, 2009]**
DI: 8256 (100%) DF: 31 (0,38%)	DI: 7118 (100%) DF: 247 (3,47%)	DI: 7542 (100%) DF: 169 (2,24%)	DI: --- DF: ---

Tabela 9: Comparação do potencial de redução de dimensionalidade com outros trabalhos que realizam a seleção de características (DI: dimensões iniciais e DF: dimensões finais).

A partir da análise das tabelas é possível concluir que:

1) As técnicas aqui desenvolvidas são complementares e, portanto, quando executadas uma após a outra resultam em desempenhos muito bons quando comparados a outros trabalhos presentes na literatura.
2) A execução do PCA após o algoritmo de seleção e agrupamento de características contribuiu para uma maior redução de dimensionalidade e levou a um melhor desempenho.
3) A redução de dimensionalidade proporcionada pela execução dos três algoritmos (AG + Agrupamento + PCA) foi muito grande, levando a uma redução de 8254 (100%) para 31 características (0,38%). No total a dimensionalidade foi reduzida de 99,62%.
4) Além da redução de dimensionalidade, o tempo de processamento foi de 52s (100%) para 5s (9,6%) para realizar uma validação cruzada de 10-fold usando o SVM e de 28s (100%) para 2s (7,14%) com o k-vizinhos mais próximos.

A título de curiosidade, na tabela abaixo são mostrados os tokens selecionados (stem das palavras) pelo AG em uma das execuções para o tópico "Earn" quando usando o algoritmo k-vizinhos mais próximos (total de 1736 tokens). Repare que muitos com sinal de "<" e ">" foram selecionadas, indicando, talvez, que um tokenizador mais eficiente possa aumentar ainda mais o desempenho final obtido. Além disso, o tokenizador utilizado remove números e, portanto, eles não estiveram presentes nos tokens selecionados.

of ; it ; said ; board ; to ; sharehold ; the ; april ; stock ; mln ; share ; for ; in ; fourth ; quarter ; food ; earn ; also ; exce ; from ; dlr ; year ; an ; ct ; be ; prior ; dean ; should ; fiscal ; compani ; will ; co ; benefit ; acquisit ; inc ; he ; and ; told ; analyst ; record ; march ; v ; loss ; shr ; net ; profit ; rev ; after ; tax ; includ ; charg ; or ; unit ; that ; oper ; note ; qtr ; pct ; common ; feder ; on ; yr ; gain ; corp ; qtly ; div ; pai ; three ; exchang ; two-for-on ; split ; i ; approv ; at ; annual ; meet ; increas ; mth ; extraordinari ; sale ; avg ; expect ; fall ; end ; period ; yen ; thi ; ntt ; plan ; capit ; a ; total ; telecommun ; effect ; januari ; major ; wa ; ltd ; dividend ; chairman ; make ; ani ; not ; cent ; last ; adjust ; bonu ; ha ; rang ; termin ; earlier ; report ; against ; exclud ; post ; remain ; perform ; than ; with ; product ;

by ; nine ; but ; hong ; kong ; larg ; trade ; improv ; result ; hold ; off ; all ; which ; issu ; new ; stake ; power ; invest ; govern ; seven ; asset ; ar ; develop ; interest ; oil ; canada ; better ; ka-sh ; did ; give ; specif ; firm ; less ; one-for-four ; four-for-on ; equal ; market ; hotel ; <hkeh ; local ; associ ; anoth ; growth ; industri ; look ; british ; coloni ; propos ; declar ; won ; six ; payabl ; holder ; june ; two ; first ; month ; discontinu ; mthly ; incom ; presid ; we ; over ; next ; have ; had ; past ; averag ; export ; well ; drug ; test ; franklin ; insur ; tax-fre ; fund ; eight ; ga ; distribut ; announc ; group ; provid ; thei ; been ; rate ; statement ; about ; ad ; while ; lead ; japanes ; demand ; declin ; were ; mine ; amount ; foreign ; item ; writedown ; cash ; continu ; bui ; full ; busi ; other ; receiv ; under ; transpond ; purchas ; would ; right ; certain ; subject ; agreement ; own ; carryforward ; bank ; futur ; intern ; ago ; secur ; becaus ; spokesman ; later ; much ; gener ; between ; up ; into ; contract ; some ; decis ; financi ; help ; order ; us ; economi ; point ; deal ; sever ; deliveri ; if ; reason ; good ; outlook ; sai ; part ; turn ; nil ; system ; sold ; writeoff ; close ; proce ; complet ; stabil ; volum ; line ; attempt ; takeov ; combin ; out ; store ; both ; name ; payment ; partner ; outstand ; valu ; limit ; held ; plc ; l> ; sell ; success ; there ; now ; financ ; director ; price ; set ; edward ; west ; produc ; take ; yesterdai ; reuter ; expens ; sinc ; todai ; merril ; can ; more ; so ; commun ; american ; cut ; go ; commiss ; neg ; deregul ; hope ; further ; real ; dutch ; transport ; depend ; base ; determin ; sterl ; currenc ; manufactur ; no ; execut ; offic ; jame ; offer ; those ; comput ; standard ; through ; health ; who ; util ; without ; howev ; subsidiari ; file ; restat ; merger ; life ; quarterli ; continent ; retract ; each ; do ; made ; allow ; late ; until ; term ; detail ; around ; trust ; comment ; boost ; rais ; bought ; come ; administr ; debit ; like ; unchang ; when ; sheet ; lend ; possibl ; vote ; vi> ; despit ; could ; energi ; manag ; agre ; open ; consid ; farah ; move ; week ; avail ; involv ; corp> ; leav ; largest ; area ; world ; repres ; attribut ; schedul ; restrict ; houston ; talk ; deficit ; ltd> ; becom ; privat ; call ; seek ; proport ; inflat ; option ; consider ; these ; author ; seen ; negoti ; staff ; hi ; assum ; role ; land ; mawr ; convert ; immedi ; japan ; ministri ; accord ; poll ; offici ; survei ; stage ; revers ; water ; wast ; four-for-thre ; post-split ; video ; del ; work ; recent ; citi ; qtrly ; nation ; compon ; extend ; bottom ; small ; corpor ; hard ; non-manufactur ; such ; pearson ; polici ; entertain ; time ; newspap ; plant ; london ; china ; acquir ; east ; swiss ; regist ; nomin ; back ; warrant ; certif ; short ; confus ; block ; vice ; law ; prepar ; taxpay ; them ; side ; get ; peabodi ; bonanza ; buchbind ; larger ; season ; million ; worldwid ; act ; conced ; case ; low ; taken ; roll ; structur ; expans ; ir ; how ; want ; directli ; braatz ; their ; chapter ; date ; unsecur ;

committe ; creditor ; subordin ; debentur ; claim ; princip ; capitol ; ratio ; near ; delai ; origin ; tofutti ; then ; greater ; cambridg ; tradition ; veri ; optimist ; charl ; exlei ; ncr ; penetr ; introduct ; magnet ; dai ; whole ; clausen ; bankamerica ; arm ; problem ; begin ; onli ; transact ; joint ; <bank ; went ; appli ; br> ; particular ; societ ; general ; to> ; sharpli ; wall ; focus ; trader ; australian ; worth ; reynold ; australia ; familiar ; ask ; investor ; furman ; selz ; boddington ; bauxit ; adopt ; all-suit ; within ; ventur ; construct ; ub ; closur ; particip ; condit ; yet ; spend ; chain ; rank ; film ; commit ; quot ; gmt ; night ; latest ; philip ; jr ; review ; consult ; histori ; f> ; wiehn ; deutsch ; babcock ; absolut ; moratorium ; brazil ; press ; brief ; brazilian ; countri ; action ; after-tax ; chemic ; extens ; vista ; home ; institut ; fhlmc ; correct ; discount ; york ; customarili ; pharmaceut ; confid ; cpc ; program ; groceri ; lawsuit ; suit ; brother ; alleg ; perelman ; deni ; conduct ; defend ; franchis ; where ; schenker ; merchandis ; appear ; de ; groot ; excel ; ongo ; solid ; canadian ; decid ; whether ; feel ; stai ; exercis ; segment ; few ; alleghani ; auditor ; lift ; question ; matur ; wrote ; marbl ; octob ; emhart ; itself ; mani ; primari ; inform ; enterpris ; proportion ; laurentian ; opportun ; doe ; forseeabl ; suspend ; maker ; inc> ; size ; mitek ; top ; follow ; excess ; primarili ; entir ; iomega ; coven ; <un ; nv ; control ; western ; bad ; environment ; adequ ; preliminari ; pension ; hous ; reform ; per-shar ; center ; enter ; dresser ; alden ; brown ; non-recur ; bancorp ; too ; reclassifi ; medium- ; nonperform ; interst ; mean ; accru ; deduct ; qualifi ; opinion ; dealer ; brokerag ; ethanol ; pubco ; person ; entitl ; event ; respons ; effort ; trustco ; laboratori ; backlog ; advanc ; softwar ; workforc ; process ; steven ; tsai ; one-tim ; jefferi ; bear ; break ; respond ; far ; olson ; yearend ; forc ; kleinwort ; benson ; depositari ; nuclear ; revis ; salari ; lilco ; establish ; state ; explain ; spokeswoman ; minorco ; compli ; draw ; dinar ; rafidain ; branch ; instead ; sourc ; float ; difficulti ; broadli ; fridai ; surplu ; soon ; mcdonald ; focu ; simon ; unavail ; drexel ; burnham ; lambert ; wendi ; ti ; martin ; rather ; econom ; cycl ; digit ; support ; replac ; mid-rang ; togeth ; never ; machin ; aspect ; meant ; whose ; threat ; custom ; ship ; comparison ; militari ; space ; reorganis ; launch ; bae ; method ; distributor ; shipment ; semiconductor ; conting ; wilkinson ; throughout ; procedur ; free ; individu ; richard ; head ; citicorp ; divis ; pick ; earner ; braddock ; extinguish ; commerci ; transfer ; paper ; t> ; nippon ; crude ; below ; automobil ; programm ; fuel ; proceed ; percept ; recoveri ; might ; hovnanian ; still ; present ; short-term ; purpos ; combust ; oversea ; measur ; lummu ; mail ; depart ; lifo ; labatt ; three-year ; target ; strategi ; stem ; bring ; phase ; medar ; indiana ; pressur ; materi ; hoechst ; germani ; polystyren ; raw ; enough ; bulk ; celanes ; siemen ;

kask ; german ; jump ; abroad ; uncertainti ; instal ; union ; grew ; newli ; autom ; achiev ; technic ; full-year ; weston ; import ; ownership ; progress ; fabric ; disposit ; goal ; abandon ; examin ; intend ; took ; charge-off ; environ ; sign ; one-for-f ; reg ; petroleum ; pct-own ; mitsubishi ; recov ; sustain ; output ; aluminium ; passeng ; light ; suffer ; mn ; tokheim ; famili ; tc ; conserv ; explor ; galact ; concern ; remov ; summitvil ; amort ; ounc ; leach ; bid ; cistron ; three-for-on ; southwestern ; int ; accompani ; suppli ; extern ; found ; audit ; arthur ; almost ; natur ; retroact ; element ; proxi ; newmont ; imperi ; tomorrow ; slow ; put ; toronto-bas ; lot ; wood ; meredith ; michael ; identifi ; try ; ow ; wait ; walsh ; minim ; thu ; what ; fulli ; accept ; advantag ; caus ; field ; affili ; overfund ; weiss ; among ; central ; my ; alloc ; first-quart ; heat ; underwrit ; exceed ; morgan ; reach ; began ; banker ; prime ; treasuri ; california ; northern ; civil ; calif ; main ; realti ; expir ; morri ; retain ; alreadi ; murdoch ; gulf ; texa ; interim ; sought ; principl ; -base ; monei ; barrel ; peso ; francisco ; classifi ; merg ; here ; georg ; think ; spirit ; bill ; absorb ; senior ; spread ; aug ; centuri ; five-for-four ; led ; rowan ; ln ; coupon ; need ; kokan ; peter ; tender ; ropak ; definit ; satisfactori ; undisclos ; antibiotico ; osr ; parti ; disclos ; european ; letter ; intent ; maximum ; <bcom ; cooper ; attract ; amsterdam ; paris-bas ; request ; sweden ; specialist ; co> ; strateg ; step ; baker ; monetari ; austrian ; design ; member ; oppos ; kodak ; acm ; discuss ; must ; reject ; rule ; secretari ; landmark ; avoid ; resort ; mcdowel ; interpharm ; imatron ; lube ; studi ; advis ; mt ; roffman ; casino ; met ; wai ; broke ; deadlin ; mainten ; jim ; specul ; doubt ; buyer ; unidentifi ; arbitrageur ; effici ; commerc ; england ; seat ; plai ; pursu ; resolut ; goldman ; explan ; headquart ; district ; screen ; someon ; newport ; daili ; pass ; charter ; <san ; miguel ; washington ; approach ; bartlett ; former ; cancel ; prevent ; malcolm ; complaint ; <fairchild ; piec ; toward ; extent ; sa> ; sat-apt ; jeumont-schneid ; economist ; reagan ; drive ; enforc ; probe ; petrochem ; keep ; diversitech ; natali ; am ; bp ; slate ; alon ; rhode ; connecticut ; murrai ; express ; irvin ; bil ; st ; mountain ; possibli ; suspect ; confidenti ; incent ; pill ; feeder ; capabl ; beat ; woodbridg ; you ; signal ; <i ; minist ; legend ; walker ; wisconsin ; tell ; veget ; dedic ; tesco ; glass-mak ; santa ; burpe ; lindner ; oklahoma ; incras ; pilot ; aci ; manner ; <cawl ; shirt ; jerom ; <bld ; quebec ; liberti ; delta ; parliamentari ; infus ; memotec ; sent ; congress ; passiv ; firmer ; weigh ; parsow ; decad ; chesebrough ; stauffer ; wedgeston ; ye ; -dlr ; usual ; eddi ; arrow ; ericsson ; percent ; domin ; <elxa ; enjoin ; founder ; disk ; assist ; monier ; attornei ; illeg ; huron ; award ; okla ; upsid ; copi ; dayton-hudson ; craft ; rudi ; thwart ; whatev ; invit ; ssmc ; trailwai ; relev ; duplic ; cie ; crash ; maxtec ; riyal ; outlin ; foremost ; opec ; saudi ;

pact ; spot ; deliber ; warmer ; overcom ; residu ; user ; exempt ; categori ; sit ; bpd ; sea ; attack ; oecd ; terra ; foulk ; dead ; promot ; abdul ; scottish ; impos ; mobil ; santo ; weekend ; consumpt ; lawson ; huge ; remit ; pdvsa ; gasolin ; api ; warn ; bbl ; policymak ; stabl ; <oxy> ; climb ; risen ; mediterranean ; heavier ; cif ; costli ; uneconom ; reestablish ; egyptian ; lago ; border ; fast ; hodel ; diminish ; load ; second-half ; moscow ; jaim ; lack ; arturo ; top-level ; draft ; questionnair ; retali ; multilater ; conflict ; sight ; tariff ; microchip ; korea ; agricultur ; disturb ; diplomat ; rich ; frequent ; dump ; hata ; ho ; sophist ; allegedli ; eiaj ; saba ; -japan ; haven ; chines ; dram ; toshiba ; nevertheless ; slap ; stanc ; featur ; pari ; fitzwat ; tackl ; mulronei ; knowledg ; offend ; geneva ; devalu ; hover ; em ; belgium ; increasingli ; backdrop ; tripl ; take-up ; wage ; bow ; volcker ; sumita ; manila ; categor ; donor ; steer ; visa ; voluntarili ; spill ; overdraft ; lock ; homebuy ; asid ; bilion ; luckei ; bradlei ; disastr ; <morgan ; end-us ; svizzera ; impress ; faster ; ey ; atcor ; unprofit ; falloff ; seoul ; beta-format ; lord ; pittsburgh ; issuer ; nonrecur ; kloeckner ; krupp ; <hkld ; yochum ; vendor ; rugbi ; government-own ; diversif ; kg ; squeez ; reinvest ; therapeut ; boeski ; regardless ; mainfram ; leung ; browning-ferri ; surrend ; lane ; eckenfeld ; tycoon ; guido ; indebted ; greenwood ; ident ; mart ; hammer ; excit ; <bac> ; easi ; compromis ; triad ; leroi ; iacocca ; america> ; paralax ; armi ; erect ; cashin ; treati ; hut ; gate ; infortext ; magnat ; forum ; beverli ; cb ; steinhardt ; mobex ; <gener ; york-area ; lipper ; flush ; resal ; cvn ; regi ; undergon ; cityquest ; deltec ; kent ; father ; sceptr ; kirschner ; stagger ; <afg> ; <nova ; <pglo ; lomak ; mcintyr ; itj ; <smb ; <pioneer ; fund> ; siaf ; sportswear ; staf ; dynalectron ; baie ; newfoundland ; neptunia ; skandia ; skeptic ; atmospher ; atchison ; plainwel ; textron ; nl> ; architectur ; bashaw ; kick ; cooper-eromanga ; <jpi> ; protein ; allot ; bimp ; harcourt ; brace ; jovanovich ; convey ; suject ; co-op ; sybron ; samuel ; healthsouth ; maxwel ; <hbj> ; unwelcom ; monica ; joe ; undersecretari ; alpert ; endors ; brandi ; faith ; editor ; ali ; nigerian ; al-khalifa ; al-sabah ; self-impos ; pemex ; lukman ; all-tim ; drawn ; bin ; everyth ; rafael ; ambassador ; throughput ; vat ; kuwaiti ; peke ; conabl ; unsur ; ahm ; tonight ; hydrocrack ; dunham ; randol ; perhap ; rosemari ; disciplin ; calm ; re-evalu ; incid ; heaviest ; tass ; divert ; spoken ; crumbl ; arbitrari ; guidanc ; wheat ; costa ; korean ; trip ; frighten ; imped ; her ; hajim ; democraci ; indian ; american-mad ; unregul ; professor ; uchida ; underpin ; imf ; repeatedli ; wit ; non-tariff ; onto ; constitu ; bucharest ; kenya ; cross-bord ; three-dai ; framework ; suna ; khartoum ; herstatt ; dattel ; exchange-trad ; payrol ; host ; bonn ; medium-term ; suicid ; context ; fixed-r ; expiri ; optima ; grip ; rubio ; scrambl ; whippl ;

loyn ; ca ; batteri ; <ut> ; dalton ; christma ; <aluz ; reassur ; reuss ; taxabl ; disallow ; shopwel ; laenderbank ; wort ; lebanon ; peninsular ; carolian ; dr ; <crk> ; liedtk ; welltech ; nowak ; strenger ; fasb ; heineken ; nonetheless ; rwanda ; hotel/casino ; <kiena ; burmah ; financier ; pooling-of-interest ; <emr> ; ionic ; drove ; nugget ; diablo ; liang ; reconven ; <hunter ; contend ; <sterl ; entrant ; wilshir ; bishop ; purus ; multibank ; presum ; khj-tv ; wor-tv ; disqualifi ; incomplet ; predominantli ; cairo ; northeastern ; puppi ; therapi ; marietta ; crew ; pressuris ; opprobrium ; non ; gu ; fichtel ; chanceri ; avaq ; ventra ; hine ; <shearson ; scan-graph ; spermicid ; fc ; cineplex ; sayam ; fragranc ; sheldahl ; <psy> ; irwin ; tike ; nobuo ; <ge ; novatron ; alcan ; consortia ; jean ; smuggl ; spectrum ; chubb ; prei ; foodservic ; mhi ; logo ; trans-info ; grossman ; gammara ; aslambeck ; backer ; monopolis ; cote ; jupit ; needham ; bristol-my ; on-market ; wale ; foreclos ; tower ; credenc ; wendel ; vodka ; <c ; lvi ; division ; darbi ; kio ; sime ; sdn ; outfitt ; symtron ; agree ; vend ; augenthal ; <dtx ; balzac ; crane ; <hyo ; ot ; saur ; cha ; coronet ; courtauld ; nettleton ; rte ; capacitor ; <gs> ; <dh> ; atlanti ; woolowrth ; turkish ; athen ; protocol ; aguarico ; hook ; <imo ; pciac ; <dia> ; sore ; simpl ; ceremoni ; sacrific ; insuffici ; sen ; saf ; sur ; isam ; tract ; haltenbanken ; calend ; particulari ; metric ; <stat ; ratifi ; <cgp> ; deck ; chen ; bailei ; paraguai ; -ussr ; ussr ; ratif ; rough ; peterson ; mood ; profil ; honda ; inexpens ; supply-demand ; fashion ; incompat ; censu ; tropic ; praug ; last-ditch ; concepcion ; hard-press ; softwood ; shoichi ; tunisia ; mainstai ; exit ; -china ; re-export ; sdi ; emcf ; chivuno ; urban ; prudenc ; forgotten ; lanston ; shill ; spotlight ; kohl ; gazett ; cox ; stopout ; conneri ; sam ; ccc ; <anza ; disburs ;

Tabela 10: Tokens selecionados em uma execução do AG para a técnica de aprendizado k-vizinhos mais próximos (tópico Earn do Reuters 21.578)

6.5.3.3 Experimentos com a base de notícias

Nesta seção, todos os experimentos descritos foram executados para a base de notícias. Como já falado anteriormente, somente aquelas sobre a empresa Petrobrás foram consideradas e a polaridade foi definida de acordo com a cotação da ação no dia posterior à publicação. As notícias de um mesmo dia foram unidas e representaram uma única entrada. Sabe-se de antemão que o resultado obtido não será muito bom, uma vez que o correto seria definir a polaridade das seguintes formas:

(1) Verificar se a cotação já teve uma variação grande na abertura do pregão do dia D1 e considerar o sinal da variação como a polaridade das notícias publicadas entre o horário de fechamento do pregão do dia útil anterior (D2) e o horário de abertura do dia D1;
(2) Verificar se a cotação teve uma variação muito grande durante o pregão do dia D. Se houver, considerar o horário H da variação e definir a polaridade das notícias publicadas perto do horário H de acordo com o sinal da variação.

Como a base de dados utilizada continha apenas a data de publicação (sem horário), não foi possível adotar este mecanismo para a definição da polaridade. Desta forma, como já mencionado, a polaridade de uma notícia foi definida de forma mais simples, ou seja, através do sinal da variação da ação no dia posterior. O mecanismo adotado gera problemas nos casos em que a notícia relevante foi publicada antes do fechamento do pregão, pois, provavelmente, ela já impactou as ações no mesmo dia e, talvez, não no posterior. Adotou-se a estratégia simples, pois, apesar de introduzir erros, espera-se conseguir obter uma taxa de acerto superior ao aleatório (acima de 50%), o que já seria muito bom por se saber que existirão dias em que não existirão notícias relevantes e que erros foram introduzidos pela forma como a polaridade foi definida.

Exp	**Algoritmo**	**Dimensões**	**Tempo**	**Acerto**
1	Somente SVM	1.177 (100%)	9s (100%)	51,13%
2	AG	228 (19,37%)	7s (77, 8%)	52,11%
3	AG + Clusterização de Variáveis usando Afinidade	56 (4,75%)	3s (33,3%)	58,42%
4	AG + Clusterização de Variáveis usando Informação Mútua	23 (1,95%)	2s (22,2%)	57,85%
5	AGs + Clusterização de Variáveis usando Log Likelihood	61 (5,18%)	4s (44,4%)	56,42%

Tabela 11: Tabela de resultados dos experimentos com a base de notícias

Analisando a tabela, é possível concluir que:

1) O melhor resultado foi obtido usando a medida de clusterização afinidade.

2) Os desempenhos obtidos com a execução das técnicas desenvolvidas neste trabalho contribuem para uma melhora expressiva no desempenho das técnicas de aprendizado.
3) A dimensionalidade dos dados é grandemente reduzida, indo de 1177 (100%) dimensões para 23 (1,95%).

6.5.4 Experimentos envolvendo a Análise de Componentes Principais (PCA)

Esta seção tem como intuito utilizar o algoritmo PCA para reduzir a dimensionalidade dos dados e, consequentemente, permitir a comparação com os algoritmos desenvolvidos neste trabalho. Nos experimentos realizados nesta seção, o PCA é executado e, imediatamente depois, o SVM também o é. Com isso, é possível realizar a comparação proposta nesta seção. Com relação aos dados, apenas a base de avalição de filmes foi utilizada, uma vez que o único propósito desta seção é de comparação com o PCA.

A tabela abaixo apresenta o resultado obtido na execução do algoritmo de aprendizado para a base de avaliação de filmes com: (1) todas as dimensões, (2) dimensões após a execução do PCA e (3) dimensões após a execução do AG e da clusterização de variáveis (melhor resultado obtido na seção 6.5.3.1), ou seja, do experimento que utiliza o AG e a Clusterização de Variáveis usando a medida de afinidade (87,39%). Em relação ao algoritmo de aprendizado, apenas o SVM foi utilizado, uma vez que foi aquele que resultou em melhores resultados.

Exp	Algoritmo	Dimensões	Tempo	Teste (SVM)
1	Só Aprendizado	6.875	49s (100%)	85,14%
2	PCA	1.966 (28,6%)	14s (28,57%)	74,52%
3	AG + Clusterização de Variáveis usando Afinidade	1.324 (19,26%)	11s (22,45%)	87,39%

Tabela 12: Tabela de resultados comparativos com o PCA

Analisando os resultados expostos na Tabela, é visível que os dois algoritmos propostos resultaram em um número de dimensões menor e a resultados melhores do que o PCA. Isto pode ser explicado pelo fato de os algoritmos propostos considerarem informações específicas sobre o problema, como, por exemplo, o fato de que dimensões relacionadas a tokens que quase não aparecem juntos podem ser mescladas em uma só dimensão.

Um fato que, à primeira vista, pode parecer estranho é que o algoritmo de aprendizado executado com os dados obtidos após o PCA teve desempenho inferior ao da execução com os dados originais. Mas, ao recorrer à trabalhos que lidam com texto e com o PCA, é possível ver que não é incomum encontrar resultados em que essa redução também ocorre. Por exemplo, [TAKAMURA & MATSUMOTO, 2001] tratam do campo de categorização de textos e também utilizam o SVM como método de aprendizado. Eles concluem que os dados originais fornecem um bom desempenho somente quando existem muitos dados e que o PCA só alcança melhores resultados que os dados originais quando existem poucos dados disponíveis. Outro exemplo é [YONG & JAIN, 1998], onde os autores também abordam a classificação de textos e não obtêm melhoras no desempenho quando utilizam o método dos k-vizinhos mais próximos.

6.5.5 Resumo dos resultados obtidos

O intutito aqui é resumir todos os resultados obtidos ao longo da seção 6.5. Considere a tabela abaixo, a qual considera todos os melhores resultados dos experimentos realizados ao longo da seção citada.

Base de Dados	Resultado	Dimensões	Tempo
Reuters	A: 0.982399 P: 0.989061 R: 0.981535 F1: 0.9852841	151 (1,83%)	T: 5s (9,6%)
Filmes	A: 0,8739	1.324 (19,25%)	11s (22,45%)
Notícias	A: 0,5842	56 (4,75%)	3s (33,3%)
Filmes (PCA)	A: 0,7452	1966 (28,59%)	14s (28,57%)

Tabela 13: Resumo dos resultados obtidos em todos os experimentos

A partir da análise da tabela, podemos notar alguns pontos:

1) O número de variáveis é reduzido para, no pior caso, 28,59% e para, no melhor, 1,83%. Por isso, os resultados reafirmam a grande capacidade de diminuição de características do problema.
2) O tempo de processamento, assim como a dimensão dos dados, é reduzido grandemente (para 9,6% no melhor caso e 33,3% no pior).
3) Os resultados finais obtidos por cada experimento foram superiores aos trabalhos comparados e também quando comparados à execução dos experimentos com todas as características originais.
4) Somente para a base Reuters as medidas precision, recall e F1 foram coletadas. Isso não foi feito para as demais, pois a comparação com outros trabalhos não exigiam.

Capítulo 7. Generalização da Clusterização de Variáveis

O capítulo 4 introduz o algoritmo de clusterização de variáveis voltado especificamente para lidar com dado textual. O intuito deste capítulo é generalizar o algoritmo em questão de forma a possibilitar com que o mesmo seja utilizado para dados de qualquer natureza. Ao observar a descrição do algoritmo presente no capítulo 4, é possível verificar que os únicos entraves para a aplicação do algoritmo com dados de qualquer natureza é a medida de dissimilaridade e a função de agrupamento (forma como duas dimensões são unidas). As três medidas apresentadas no capítulo 4 são específicas para dados textuais. Em relação à função de agrupamento, ou seja, a forma como as dimensões são unidas, esta foi especificada anteriormente como uma função "ou" entre os tokens (como a representação era binária, se um dos tokens pertencentes ao conjunto estivesse presente no texto, o bit 1 seria atribuído à dimensão relativa a este conjunto).

Para generalizar o algoritmo, portanto, seria necessário definir duas funções genéricas, as quais seriam: a medida de dissimilaridade entre dimensões e não mais entre tokens $\boldsymbol{d}(d_1, d_2)$ e a função de agrupamento $\boldsymbol{a}(d_1, d_2, \ldots, d_k)$. Repare que outro componente do algoritmo é a medida de correlação de Pearson e, como esta já é aplicável para números reais, então, ela pode ser utilizada em qualquer situação e não precisamos nos preocupar com ela. O interessante desta generalização é que pode-se escolher a medida de dissimilaridade e a função de agrupamento de acordo com informações conhecidas sobre o problema. Por exemplo, no caso textual, sabe-se que existem palavras que nunca ou quase nunca aparecerão juntas e, portanto, a ideia de priorizar a união destas palavras é interessante, uma vez que ocorreria o reaproveitamento da dimensão.

Neste capítulo, além dessa generalização, que permite a criação de outros trabalhos ao modificar as funções $\boldsymbol{a}$ e $\boldsymbol{d}$, a seção 7.1 retrata o caso de dados binários e como estas funções poderiam ser escolhidas ao adotar tal notação. Desta forma, passaremos a ter um algoritmo não só para processamento de texto como para qualquer problema que envolva dados binários.

7.1 Clusterização de Variáveis para dados binários

Adaptar o algoritmo de Clusterização de Variáveis apresentado neste trabalho, o qual considerou dados textuais, é uma tarefa fácil, uma vez que a forma utilizada para representar o texto foi a binária. Vamos começar pela mais simples, a função de agrupamento, a qual foi representada por um "ou" entre os tokens pertencentes ao conjunto. Com dados binários, a função "ou" (∨) pode ser utilizada. Desta forma, considerando um conjunto formado pelas dimensões $d_1, d_2, \ldots, d_k$, a função de agrupamento $\boldsymbol{a}$, pode ser representada por:

$$\boldsymbol{a}(d_1, d_2, \ldots, d_k) = d_1 \vee d_2 \vee \ldots \vee d_k$$

Equação 32: Função de agrupamento para dados binários

Em relação à função de dissimilaridade, também podemos facilmente reaproveitar as relacionadas no capítulo 4. Por exemplo, a frequência com que w1 e w2 aparecem juntos $f(w1, w2)$ pode ser modificada para $\boldsymbol{d}(d_1, d_2) = f(d_1, d_2)$ e interpretada como o número de vezes em que existem bits 1 nas duas dimensões analisadas. Pensando de forma parecida, as três medidas presentes no capítulo 4 podem ser adaptadas para generalizar para o caso binário.

Capítulo 8. Conclusão

Neste trabalho foram implementados dois algoritmos que contribuem com a literatura nos seguintes aspectos:

(1) Seleção de Características, contribuindo para a reduzir a quantidade de características;
(2) Clusterização de Variáveis, permitindo a união de uma ou mais dimensões em grupos e, portanto, também resultando em um número menor de dimensões;

Em relação ao algoritmo de seleção de características, uma abordagem genética foi apresentada, onde a principal contribuição para a literatura se resume a uma taxa de mutação variável e individual por bit proporcional ao coeficiente de correlação de Pearson com a saída.

A clusterização de variáveis pode ser vista como um algoritmo hierárquico aglomerativo onde a decisão de quais as características unir são baseadas em uma medida entre as dimensões duas a duas. Três medidas foram apresentadas com base em trabalhos de análise de sentimento presentes na literatura e foram elas: afinidade, informação mútua pontual e log likelihood.

Com base nos algoritmos propostos, o capítulo 5 apresentou um conjunto de etapas para realizar a análise de sentimento. Nele, técnicas de pré-processamento como o stemming foram utilizadas e os algoritmos de seleção de características e clusterização de variáveis foram executados em sequência. Após eles, o SVM foi utilizado para aprender no conjunto de treinamento e para prever a polaridade de cada uma das entradas do conjunto de teste.

No capítulo 6, experimentos foram realizados com as etapas propostas e mostraram que os algoritmos contribuem para a melhora do desempenho e redução do número de variáveis. Os principais experimentos foram realizados sobre uma base de dados de classificação de textos (Reuters 21.578) e outra de avaliação de filmes disponibilizada por PANG & LEE (2004). Os experimentos realizados sobre a base Reuters mostraram que o framework aqui desenvolvido levou a resultados melhores do que os obtidos nos trabalhos comparados (tanto em performance como na capacidade de diminuição da quantidade de características). Houve uma redução de 8.256 (100%) para 31 (0,38%) características contra uma redução de UĞUZ (2011) de 7.542 (100%) para 169 (2,24%). Na base de filmes, os resultados obtidos foram ligeiramente, mas não significantemente, superiores aos realizados por Pang & Lee (2004). Notou-se que, apesar de os resultados não terem sido significantemente superiores, o número de características foi

grandemente reduzido, contribuindo para atestar a qualidade dos algoritmos propostos. Além disso, os algoritmos desenvolvidos neste trabalho foram comparados com a Análise de Componentes Principais (PCA) e concluiu-se que aqueles foram superiores a este no que diz respeito ao desempenho e também à diminuição do número de variáveis do problema.

Por fim, no capítulo 7, o algoritmo de clusterização de variáveis foi generalizado para possibilitar a aplicação deste quando utilizando dados não textuais. Com esta generalização, o escopo do algoritmo aumenta e, portanto, possibilita explorar um conjunto maior de problemas e permite com que mais pesquisadores possam utilizá-lo a fim de contestarem a sua eficácia e se aproveitarem da capacidade de redução do número de variáveis do problema.

Bibliografia

[ABBASI et al., 2008] ABBASI, Ahmed; CHEN, Hsinchun; SALEM, Arab. Sentiment analysis in multiple languages: Feature selection for opinion classification in Web forums. ACM Transactions on Information Systems (TOIS), v. 26, n. 3, p. 12, 2008.

[ANBARASI et al, 2010] ANBARASI, M.; ANUPRIYA, E.; IYENGAR, N. C. S. N. Enhanced prediction of heart disease with feature subset selection using genetic algorithm. International Journal of Engineering Science and Technology, v. 2, n. 10, p. 5370-5376, 2010.

[BAKER & MCCALLUM, 1998] L.D. Baker, A. McCallum, Distributional clustering of words for text classification, in: Proc. 21st Annual International ACM SIGIR, 1998, pp. 96–103.

[BALABIN et al., 2011] BALABIN, Roman M.; SMIRNOV, Sergey V. Variable selection in near-infrared spectroscopy: benchmarking of feature selection methods on biodiesel data. Analytica chimica acta, v. 692, n. 1-2, p. 63-72, 2011.

[BATTITI, 1994] Battiti R. Using mutual information for selecting features in supervised neural net learning. IEEE Trans Neural Networks 1994;5.

[BAUDAT & ANOUAR, 2001] BAUDAT, Gaston; ANOUAR, Fatiha. Kernel-based methods and function approximation. In: Neural Networks, 2001. Proceedings. IJCNN'01. International Joint Conference on. IEEE, 2001. p. 1244-1249.

[BEASLEY et al., 1996] BEASLEY, John E.; CHU, Paul C. A genetic algorithm for the set covering problem. European Journal of Operational Research, v. 94, n. 2, p. 392-404, 1996.

[BEKKERMAN et al, 2003] R. Bekkerman, R. El-Yaniv, N. Tishby, Y. Winter, Distributional word clusters vs. words for text categorization, J. Mach. Learn. Res. 3 (2003) 1183–1208.

[BEKKERMAN et al., 2005] BEKKERMAN, Ron; EL-YANIV, Ran; MCCALLUM, Andrew. Multi-way distributional clustering via pairwise interactions. In: Proceedings of the 22nd international conference on Machine learning. ACM, 2005. p. 41-48.

[BEKKERMAN et al., 2001] BEKKERMAN, Ron et al. On feature distributional clustering for text categorization. In: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 2001. p. 146-153.

[BERRY & LINOFF. 1997] BERRY, Michael J.; LINOFF, Gordon. Data mining techniques: for marketing, sales, and customer support. John Wiley & Sons, Inc., 1997.

[BKER & MCCALLUM, 1998] BAKER, L. Douglas; MCCALLUM, Andrew Kachites. Distributional clustering of words for text classification. In: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval. ACM, 1998. p. 96-103.

[BOGDAN et al., 2015] BOGDAN, Małgorzata et al. SLOPE—adaptive variable selection via convex optimization. The annals of applied statistics, v. 9, n. 3, p. 1103, 2015.

[BONDELL & REICH, 2008] Bondell, H.D. and Reich, B.J. Simultaneous regression shrinkage, variable selection and clustering of predictors with OSCAR. Biometrics, 64(1):115, 2008.

[BUTTERWORTH et al., 2005] BUTTERWORTH, Richard; PIATETSKY-SHAPIRO, Gregory; SIMOVICI, Dan A. On Feature Selection through Clustering. In: ICDM. 2005. p. 581-584.

[CAMBRIA, 2016] CAMBRIA, Erik. Affective computing and sentiment analysis. IEEE Intelligent Systems, v. 31, n. 2, p. 102-107, 2016.

[CARENINI et al., 2005] CARENINI, Giuseppe; NG, Raymond T.; ZWART, Ed. Extracting knowledge from evaluative text. In: Proceedings of the 3rd international conference on Knowledge capture. ACM, 2005. p. 11-18.

[CHAN & TANSRI, 1994] CHAN, K. C.; TANSRI, H. A study of genetic crossover operations on the facilities layout problem. Computers & Industrial Engineering, v. 26, n. 3, p. 537-550, 1994.

[CHANDRASHEKAR & SAHIN, 2014] CHANDRASHEKAR, Girish; SAHIN, Ferat. A survey on feature selection methods. Computers & Electrical Engineering, v. 40, n. 1, p. 16-28, 2014.

[CHEN et al., 2008] D. Chen, K. C. C. Chan, and X. Wu, “Gene expression analyses using genetic algorithm based hybrid approaches,” in Proc. IEEE Congr. Evol. Comput., Hong Kong, 2008, pp. 963–969

[DALMAU & FLÓREZ, 2007] M.C. Dalmau, O.W.M. Flórez, Experimental results of the signal processing approach to distributional clustering of terms on reuters-21578 collection, in: Proc. 29th European Conf. IR Research, 2007, pp. 678–681

[DAVE et al., 2003] Dave K, Lawrence S, Pennock D (2003) Mining the peanut gallery: opinion extraction and semantic classi- fication of product reviews. In: Proceedings of the 12th international conference on World Wide Web, ACM, New York, NY, USA, WWW’03, pp 519–528. doi:10.1145/775152.775226.

[DETTLING & B¨UHLMANN, 2004] Dettling, M. and B¨uhlmann, B. Finding predictive gene groups from microarray data. Journal of Multivariate Analysis, 90(1):106–131, 2004.

[DERRAC et al., 2009] J. Derrac, S. Garcia, and F. Herrera, “A first study on the use of coevolutionary algorithms for instance and feature selection,” in Hybrid Artificial Intelligence Systems (LNCS 5572). Berlin, Germany: Springer, 2009, pp. 557–564.

[DHILLON et al., 2003] DHILLON, Inderjit S.; MALLELA, Subramanyam; KUMAR, Rahul. A divisive information-theoretic feature clustering algorithm for text classification. Journal of machine learning research, v. 3, n. Mar, p. 1265-1287, 2003.

[DONG & DONG, 2006] Dong Z. and Dong Q. 2006. Hownet and the computation of meaning. World Scientific Publishing Co., Inc.

[EBERHART & SHI, 1998] EBERHART, Russell C.; SHI, Yuhui. Comparison between genetic algorithms and particle swarm optimization. In: International Conference on Evolutionary Programming. Springer Berlin Heidelberg, 1998. p. 611-616.

[FAN & LI, 2001] FAN, Jianqing; LI, Runze. Variable selection via nonconcave penalized likelihood and its oracle properties. Journal of the American statistical Association, v. 96, n. 456, p. 1348-1360, 2001.

[FIGUEIREDO & NOWAK, 2014] M. Figueiredo and R. Nowak. Sparse estimation with strongly correlated variables using ordered weighted `1 regularization. arXiv preprint arXiv:1409.4005, 2014.

[FONG et al., 2016] FONG, Simon; WONG, Raymond; VASILAKOS, Athanasios V. Accelerated PSO swarm search feature selection for data stream mining big data. IEEE transactions on services computing, v. 9, n. 1, p. 33-45, 2016.

[FORMAN, 2003] FORMAN, George. An extensive empirical study of feature selection metrics for text classification. Journal of machine learning research, v. 3, n. Mar, p. 1289-1305, 2003

[FRIEDMAN et al., 2010] J. Friedman, T. Hastie, and R. Tibshirani. A note on the group lasso and a sparse group lasso. Arxiv preprint arXiv:1001.0736, 2010.

[FROHLICH et al., 2003] FROHLICH, Holger; CHAPELLE, Olivier; SCHOLKOPF, Bernhard. Feature selection for support vector machines by means of genetic algorithm. In: Tools with Artificial Intelligence, 2003. Proceedings. 15th IEEE International Conference on. IEEE, 2003. p. 142-148.

[FÜRNKRANZ, 1998] FÜRNKRANZ, Johannes. A study using n-gram features for text categorization. Austrian Research Institute for Artifical Intelligence, v. 3, n. 1998, p. 1-10, 1998.

[GAMON, 2004] Gamon, M. Sentiment classification on customer feedback data: Noisy data, large feature vectors, and the role of linguistic analysis. In Proceeding of the 20th intl. conference on computational linguistics (p. 84).

[GHAMISI & BENEDIKTSSON, 2015] GHAMISI, Pedram; BENEDIKTSSON, Jon Atli. Feature selection based on hybridization of genetic algorithm and particle swarm optimization. IEEE Geoscience and remote sensing letters, v. 12, n. 2, p. 309-313, 2015.

[GHAREB et al., 2016] GHAREB, Abdullah Saeed; BAKAR, Azuraliza Abu; HAMDAN, Abdul Razak. Hybrid feature selection based on enhanced genetic algorithm for text categorization. Expert Systems with Applications, v. 49, p. 31-47, 2016.

[GHIASSI et al., 2013] GHIASSI, M.; SKINNER, J.; ZIMBRA, D. Twitter brand sentiment analysis: A hybrid system using n-gram analysis and dynamic artificial neural network.Expert Systems with applications, v. 40, n. 16, p. 6266-6282, 2013.

[GOODFELLOW et. al, 2016] GOODFELLOW, Ian; BENGIO, Yoshua; COURVILLE, Aaron. Deep learning. MIT press, 2016.

[GOSLING, 2000] GOSLING, James. The Java language specification. Addison-Wesley Professional, 2000.

[GUYON et al., 2002] GUYON, Isabelle et al. Gene selection for cancer classification using support vector machines. Machine learning, v. 46, n. 1-3, p. 389-422, 2002.

[GUYON & ELISSEEFF, 2003] Guyon I, Elisseeff A. An introduction to variable and feature selection. J Mach Learn Res 2003;3:1157–82

[HALL, 2000] HALL, Mark A. Correlation-based feature selection of discrete and numeric class machine learning. 2000.

[HARISH & MANJUNATH, 2010] HARISH, Bhat S.; GURU, Devanur S.; MANJUNATH, Shantharamu. Representation and classification of text documents: A brief review. IJCA, Special Issue on RTIPPR (2), p. 110-119, 2010.

[HASEGAWA et al., 1997] HASEGAWA, Kiyoshi; MIYASHITA, Yoshikatsu; FUNATSU, Kimito. GA strategy for variable selection in QSAR studies: GA-based PLS analysis of calcium channel antagonists. Journal of Chemical Information and Computer Sciences, v. 37, n. 2, p. 306-310, 1997.

[HASEGAWA et al., 1999] HASEGAWA, Kiyoshi; KIMURA, Toshiro; FUNATSU, Kimito. GA strategy for variable selection in QSAR studies: enhancement of comparative molecular binding energy analysis by GA-based PLS method. Quantitative Structure-Activity Relationships, v. 18, n. 3, p. 262-272, 1999.

[HASTIE et al., 2001] Hastie, T., Tibshirani, R., Botstein, D., and Brown, P. Supervised harvesting of expression trees. Genome Biology, 2(1), 2001.

[HAUPT, 2000] HAUPT, Randy L. Optimum population size and mutation rate for a simple real genetic algorithm that optimizes array factors. In: Antennas and Propagation Society International Symposium, 2000. IEEE. IEEE, 2000. p. 1034-1037.

[HOMSAPAYA & SORNIL, 2017] HOMSAPAYA, Kanyanut; SORNIL, Ohm. Improving Floating Search Feature Selection using Genetic Algorithm. Journal of ICT Research and Applications, v. 11, n. 3, p. 299-317, 2017.

[HSU et al., 2003] Hsu, Chih-Wei, Chih-Chung Chang, and Chih-Jen Lin. "A practical guide to support vector classification." (2003): 1-16.

[HUANG, 2008] HUANG, Anna. Similarity measures for text document clustering. In: Proceedings of the sixth new zealand computer science research student conference (NZCSRSC2008), Christchurch, New Zealand. 2008. p. 49-56.

[JACK & NANDI, 2002] JACK, L. B.; NANDI, A. K. Fault detection using support vector machines and artificial neural networks, augmented by genetic algorithms. Mechanical systems and signal processing, v. 16, n. 2-3, p. 373-390, 2002.

[JAIN & DUBES, 1988] JAIN, Anil K.; DUBES, Richard C. Algorithms for clustering data. Prentice-Hall, Inc., 1988.

[JAIN et al., 1999] JAIN, Anil K.; MURTY, M. Narasimha; FLYNN, Patrick J. Data clustering: a review. ACM computing surveys (CSUR), v. 31, n. 3, p. 264-323, 1999.

[JAVA] Disponível em: https://www.oracle.com/br/java/index.html. Acessado em 14/11/2017.

[JIANG et al., 2011] JIANG, Jung-Yi; LIOU, Ren-Jia; LEE, Shie-Jue. A fuzzy self-constructing feature clustering algorithm for text classification. IEEE transactions on knowledge and data engineering, v. 23, n. 3, p. 335-349, 2011.

[JIANG & LEE, 2007] JIANG, Jung-Yi; LEE, Shie-Jue. A weight-based feature extraction approach for text classification. In: Second International Conference on Innovative Computing, Informatio and Control (ICICIC 2007). IEEE, 2007. p. 164-164.

[KASHEF & NEZAMABADI-POUR, 2015] KASHEF, Shima; NEZAMABADI-POUR, Hossein. An advanced ACO algorithm for feature subset selection. Neurocomputing, v. 147, p. 271-279, 2015.

[KEOGH & MUEEN, 2011] KEOGH, Eamonn; MUEEN, Abdullah. Curse of dimensionality. In: Encyclopedia of Machine Learning. Springer US, 2011. p. 257-258.

[KIRA et al., 1992] Kira, Kenji, and Larry A. Rendell. "The feature selection problem: Traditional methods and a new algorithm." AAAI. Vol. 2. 1992.

[KOHAVI & JOHN, 1997] KOHAVI, Ron; JOHN, George H. Wrappers for feature subset selection.Artificial intelligence, v. 97, n. 1, p. 273-324, 1997.

[KRIER et al., 2007] KRIER, Catherine et al. Feature clustering and mutual information for the selection of variables in spectral data. In: ESANN. 2007. p. 157-162.

[KWAK & CHOI, 2002] Kwak N, Choi C-H. Input feature selection for classification problems. IEEE Trans Neural Networks 2002;13:143–59.

[LAZAR et al., 2012] LAZAR, Cosmin et al. A survey on filter techniques for feature selection in gene expression microarray analysis. IEEE/ACM Transactions on Computational Biology and Bioinformatics, v. 9, n. 4, p. 1106-1119, 2012.

[LE & MIKOLOV, 2014] LE, Quoc; MIKOLOV, Tomas. Distributed representations of sentences and documents. In: Proceedings of the 31st International Conference on Machine Learning (ICML-14). 2014. p. 1188-1196.

[LEARDI & GONZALEZ, 1998] LEARDI, Riccardo; GONZALEZ, Amparo Lupianez. Genetic algorithms applied to feature selection in PLS regression: how and when to use them. Chemometrics and intelligent laboratory systems, v. 41, n. 2, p. 195-207, 1998.

[LEARDI, 2000] LEARDI, Riccardo. Application of genetic algorithm–PLS for feature selection in spectral data sets. Journal of Chemometrics, v. 14, n. 5-6, p. 643-655, 2000.

[LEUNG et al., 2006] Leung CWK, Chan SCF, Chung FL (2006) Integrating collaborative filtering and sentiment analysis: a rating inference approach. In: ECAI 2006 workshop on recommender systems, pp 62–66

[LI et al., 2009] Y. Li, S. Zhang, and X. Zeng, “Research of multi-population agent genetic algorithm for feature selection,” Expert Syst. Appl., vol. 36, no. 9, pp. 11570–11581, 2009.

[LIN et al., 2014] F. Lin, D. Liang, C.-C. Yeh, and J.-C. Huang, “Novel feature selection methods to financial distress prediction,” Expert Syst. Appl., vol. 41, no. 5, pp. 2472–2483, 2014.

[LIU et al., 2012] B. Liu, Sentiment Analysis and Opinion Mining, ser. Synthesis Lectures on Human Language Technologies. Morgan & Claypool, 2012, vol. 16.

[LOVINS, 1968] LOVINS, Julie Beth. Development of a stemming algorithm. Mech. Translat. & Comp. Linguistics, v. 11, n. 1-2, p. 22-31, 1968.

[MAAS et al., 2011] MAAS, Andrew L. et al. Learning word vectors for sentiment analysis. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1. Association for Computational Linguistics, 2011. p. 142-150.

[MAYER et al., 2017] MAYER, Joshua et al. Sequential feature selection and inference using multi-variate random forests. Bioinformatics, v. 34, n. 8, p. 1336-1344, 2017.

[MEKALA et al., 2016] MEKALA, Dheeraj et al. SCDV: Sparse Composite Document Vectors using soft clustering over distributional representations. arXiv preprint arXiv:1612.06778, 2016.

[MIN et al., 2006] MIN, Sung-Hwan; LEE, Jumin; HAN, Ingoo. Hybrid genetic algorithms and support vector machines for bankruptcy prediction. Expert systems with applications, v. 31, n. 3, p. 652-660, 2006.

[MOHAMMAD et al., 2013] MOHAMMAD, Saif M.; KIRITCHENKO, Svetlana; ZHU, Xiaodan. NRC-Canada: Building the state-of-the-art in sentiment analysis of tweets. arXiv preprint arXiv:1308.6242, 2013.

[MONEDERO et al., 2012] MONEDERO, Iñigo et al. Detection of frauds and other non-technical losses in a power utility using Pearson coefficient, Bayesian networks and decision trees. International Journal of Electrical Power & Energy Systems, v. 34, n. 1, p. 90-98, 2012.

[MORADI & ROSTAMI, 2015] MORADI, Parham; ROSTAMI, Mehrdad. Integration of graph clustering with ant colony optimization for feature selection. Knowledge-Based Systems, v. 84, p. 144-161, 2015.

[MYTHILY et al., 2015] MYTHILY, R.; BANU, Aisha; RAGHUNATHAN, Shriram. Clustering models for data stream mining. Procedia Computer Science, v. 46, p. 619-626, 2015.

[NIKHIL et al., 2015] Nikhil, R., Tikoo, N., Kurle, S., Pisupati, H. S., & Prasad, G. R. (2015, April). A survey on text mining and sentiment analysis for unstructured web data. InJournal of Emerging Technologies and Innovative Research (Vol. 2, No. 4 (April-2015)). JETIR.

[OH et al., 2004] OH, Il-Seok; LEE, Jin-Seon; MOON, Byung-Ro. Hybrid genetic algorithms for feature selection. IEEE Transactions on pattern analysis and machine intelligence, v. 26, n. 11, p. 1424-1437, 2004.

[ORESKI & ORESKI, 2014] ORESKI, Stjepan; ORESKI, Goran. Genetic algorithm-based heuristic for feature selection in credit risk assessment. Expert systems with applications, v. 41, n. 4, p. 2052-2064, 2014.

[PAK & PAROUBEK, 2010] PAK, Alexander; PAROUBEK, Patrick. Twitter as a Corpus for Sentiment Analysis and Opinion Mining. In: LREc. 2010.

[PANG et al., 2002] Pang B, Lee L, Vaithyanathan S (2002) Thumbs up? Sentiment classification using machine learning techniques. In: EMNLP 2002, pp 79–86

[PANG & LEE, 2004] PANG, Bo; LEE, Lillian. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In: Proceedings of the 42nd annual meeting on Association for Computational Linguistics. Association for Computational Linguistics, 2004. p. 271.

[PANG et al., 2008] PANG, Bo et al. Opinion mining and sentiment analysis. Foundations and Trends® in Information Retrieval, v. 2, n. 1–2, p. 1-135, 2008.

[PARK & KIM, 2015] PARK, Chan Hee; KIM, Seoung Bum. Sequential random k-nearest neighbor feature selection for high-dimensional data. Expert Systems with Applications, v. 42, n. 5, p. 2336-2342, 2015.

[PENG et al., 2005] Peng H, Long F, Ding C. Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy. IEEE Trans Pattern Anal Mach Intell 2005;27.

[PIATETSKY-SHAPIRO, 1996] PIATETSKY-SHAPIRO, Gregory. Advances in knowledge discovery and data mining. Menlo Park: AAAI press, 1996.

[POGGIO & CAUWENBERGHS, 2001] POGGIO, T.; CAUWENBERGHS, G. Incremental and decremental support vector machine learning. Advances in neural information processing systems, v. 13, p. 409, 2001.

[PORIA et al, 2015] PORIA, Soujanya; CAMBRIA, Erik; GELBUKH, Alexander. Deep convolutional neural network textual features and multiple kernel learning for utterance-level multimodal sentiment analysis. In: Proceedings of the 2015 conference on empirical methods in natural language processing. 2015. p. 2539-2544.

[POSTGRESQL] Disponível em: https://www.postgresql.org/about/. Acessado em 16/11/2017.

[PUDIL et al., 1994] Pudil P, Novovicova J, Kittler J. Floating search methods in feature selection. Pattern Recog Lett 1994;15:1119–25.

[PUDIL et al., 1999] Pudil P, Novovicova J, Kittler J, Paclik P. Adaptive floating search methods in feature selection. Pattern Recog Lett 1999;20:1157–63.

[REBELO, 2008] Rebelo, L. D. T., Avaliação automática do resultado estético do tratamento conservador do cancro de mama. Faculdade de Engenharia da Universidade do Porto.

[REUNANEN, 2003] Reunanen J. Overfitting in making comparisons between variable selection methods. J Mach Learn Res 2003;3:1371–82

[ROMERO & SOPENA, 2008] ROMERO, Enrique; SOPENA, Josep María. Performing feature selection with multilayer perceptrons. IEEE Transactions on Neural Networks, v. 19, n. 3, p. 431-441, 2008.

[ROWEIS & SAUL, 2000] ROWEIS, Sam T.; SAUL, Lawrence K. Nonlinear dimensionality reduction by locally linear embedding. science, v. 290, n. 5500, p. 2323-2326, 2000.

[SAMANTA, 2004] SAMANTA, B. Gear fault detection using artificial neural networks and support vector machines with genetic algorithms. Mechanical Systems and Signal Processing, v. 18, n. 3, p. 625-644, 2004.

[SCHOUTEN & FRASINCAR, 2016] SCHOUTEN, Kim; FRASINCAR, Flavius. Survey on aspect-level sentiment analysis. IEEE Transactions on Knowledge and Data Engineering, v. 28, n. 3, p. 813-830, 2016.

[SEDOC et al., 2017] SEDOC, Joao; PREOTIUC-PIETRO, Daniel; UNGAR, Lyle. Predicting emotional word ratings using distributional representations and signed clustering. In: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers. 2017. p. 564-571.

[SEMWAL et. al, 2017] SEMWAL, Vijay Bhaskar; MONDAL, Kaushik; NANDI, Gora Chand. Robust and accurate feature selection for humanoid push recovery and classification: deep learning approach. Neural Computing and Applications, v. 28, n. 3, p. 565-574, 2017.

[SETIONO, 1997] SETIONO, Rudy; LIU, Huan. Neural-network feature selector. IEEE transactions on neural networks, v. 8, n. 3, p. 654-662, 1997.

[SHEN & HUANG, 2010] Shen, X. and Huang, H.C. Grouping pursuit through a regularization solution surface. Journal of the American Statistical Association, 105(490):727–739, 2010

[SIEDLECKI & SKLANSKY, 1989] SIEDLECKI, Wojciech; SKLANSKY, Jack. A note on genetic algorithms for large-scale feature selection. Pattern recognition letters, v. 10, n. 5, p. 335-347, 1989.

[SINDHWANI & MELVILLE, 2008] SINDHWANI, Vikas; MELVILLE, Prem. Document-word co-regularization for semi-supervised sentiment analysis. In: Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on. IEEE, 2008. p. 1025-1030.

[SINGH et al., 2016] SINGH, Bharat; KUSHWAH, Saroj; DAS, Sanjoy. Multi-Feature Segmentation and Cluster based Approach for Product Feature Categorization.International Journal of Information Technology and Computer Science (IJITCS), v. 8, n. 3, p. 33, 2016.

[SLONIM & TISHBY, 2001] SLONIM, Noam; TISHBY, Naftali. The power of word clusters for text classification. In: 23rd European Colloquium on Information Retrieval Research. 2001. p. 200.

[SUDHAKAR et al., 2016] CH, Sudhakar et al. Cluster Based Feature Subset Selection (CFSS) for High-Dimensional Data. International Journal of Applied Engineering Research, v. 11, n. 2, p. 1369-1372, 2016.

[SUK et al., 2016] SUK, Heung-Il et al. Deep sparse multi-task learning for feature selection in Alzheimer's disease diagnosis. Brain Structure and Function, v. 221, n. 5, p. 2569-2587, 2016.

[SUN et al., 2006] SUN, Y.; BABBS, C. F.; DELP, E. J. A comparison of feature selection methods for the detection of breast cancers in mammograms: adaptive sequential floating search vs. genetic algorithm. In: 2005 IEEE Engineering in Medicine and Biology 27th Annual Conference. IEEE, 2006. p. 6532-6535.

[SUYKENS & VANDEWALLE, 1999] SUYKENS, Johan AK; VANDEWALLE, Joos. Least squares support vector machine classifiers. Neural processing letters, v. 9, n. 3, p. 293-300, 1999.

[TAKAMURA & MATSUMOTO, 2001] TAKAMURA, Hiroya; MATSUMOTO, Yuji. Feature space restructuring for SVMs with application to text categorization. In: Proceedings of the 2001 Conference on Empirical Methods in Natural Language Processing. 2001.

[TAN et al., 2008] F. Tan, X. Z. Fu, Y. Q. Zhang, and A. G. Bourgeois, "A genetic algorithm-based method for feature subset selection," Soft Comput., vol. 12, no. 2, pp. 111–120, 2008.

[TENENBAUM et al., 2000] TENENBAUM, Joshua B.; DE SILVA, Vin; LANGFORD, John C. A global geometric framework for nonlinear dimensionality reduction. science, v. 290, n. 5500, p. 2319-2323, 2000.

[TIBSHIRANI, 1996] TIBSHIRANI, Robert. Regression shrinkage and selection via the lasso.Journal of the Royal Statistical Society. Series B (Methodological), p. 267-288, 1996.

[TIBSHIRANI et al., 2005] Tibshirani, R., Saunders, M., Rosset, S., Zhu, J., and Knight, K. Sparsity and smoothness via the fused lasso. Journal of the Royal Statistical Society, Series B, 67(1):91–108, 2005.

[TSYTSARAU & PALPANAS, 2012] M. Tsytsarau and T. Palpanas, "Survey on Mining Subjective Data on the web," Data Mining and Knowledge Discovery, vol. 24, no. 3, pp. 478–514, 2012.

[UGUZ, 2011] UGUZ, Harun. A two-stage feature selection method for text categorization by using information gain, principal component analysis and genetic algorithm. Knowledge-Based Systems, v. 24, n. 7, p. 1024-1032, 2011.

[VASSILIADIS, 2009] VASSILIADIS, Panos. A survey of Extract–transform–Load technology. International Journal of Data Warehousing and Mining (IJDWM), v. 5, n. 3, p. 1-27, 2009.

[WAN, 2008] Wan X. 2008. Using bilingual knowledge and ensemble techniques for unsupervised Chinese sentiment analysis. In Proceedings of EMNLP08, pp553-561

[XIANG et al., 2013] XIANG, Shuo; TONG, Xiaoshen; YE, Jieping. Efficient Sparse Group Feature Selection via Nonconvex Optimization. In: ICML (1). 2013. p. 284-292.

[XUE et al., 2016] XUE, Bing et al. A survey on evolutionary computation approaches to feature selection. IEEE Transactions on Evolutionary Computation, v. 20, n. 4, p. 606-626, 2016.

[YAHOO_FINANCE] Disponível em: https://finance.yahoo.com/. Acessado em 14/11/2017.

[YANG & HONAVAR, 1998] YANG, Jihoon; HONAVAR, Vasant. Feature subset selection using a genetic algorithm. In: Feature extraction, construction and selection. Springer US, 1998. p. 117-136.

[YI et al., 2003] YI, J., NASUKAWA, T., BUNESCU, R. AND NIBLACK, W. 2003. Sentiment analyzer: Extracting sentiments about a given topic using natural language processing techniques, In Proceedings of the 3rd IEEE International Conference on Data Mining, 427-434.

[YONG & JAIN, 1998] LI, Yong H.; JAIN, Anil K. Classification of text documents. The Computer Journal, v. 41, n. 8, p. 537-546, 1998.

[YUN & LIN, 2006] Yuan, M. and Lin, Y. Model selection and estimation in regression with grouped variables. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 68(1):49–67, 2006.

[ZABIN & JEFFERIES, 2008] J. Zabin and A. Jefferies, “Social media monitoring and analysis: Generating consumer insights from online conversation,” Aberdeen Group Benchmark Report, January 2008.

[ZANG et al., 2017] ZHANG, Yong et al. A PSO-based multi-objective multi-label feature selection method in classification. Scientific reports, v. 7, n. 1, p. 376, 2017.

[ZEHE et al., 2017] ZEHE, Albin et al. Towards Sentiment Analysis on German Literature. In: Joint German/Austrian Conference on Artificial Intelligence (Künstliche Intelligenz). Springer, Cham, 2017. p. 387-394.

[ZHAI et al., 2011] Zhai, Z., Liu, B., Xu, H., & Jia, P. (2011, February). Clustering product features for opinion mining. In Proceedings of the fourth ACM international conference on Web search and data mining (pp. 347-354). ACM.

[ZHANG et al., 2015] ZHANG, Xiang; ZHAO, Junbo; LECUN, Yann. Character-level convolutional networks for text classification. In: Advances in neural information processing systems. 2015. p. 649-657.

[ZHONG & KWOK, 2012] ZHONG, Leon Wenliang; KWOK, James T. Efficient sparse modeling with automatic feature grouping. IEEE transactions on neural networks and learning systems, v. 23, n. 9, p. 1436-1447, 2012.

[ZHU et al., 2009] Zhu J, Zhu M, Wang H, Tsou BK (2009) Aspect-based sentence segmentation for sentiment summarization. In: Proceeding of the international CIKM workshop on topic-sentiment analysis for mass opinion measurement. ACM, New York, NY, USA, TSA'09, pp 65–72. doi:10.1145/1651461.1651474.

[ZOU et al., 2015] ZOU, Qin et al. Deep learning based feature selection for remote sensing scene classification. IEEE Geoscience and Remote Sensing Letters, v. 12, n. 11, p. 2321-2325, 2015.

[ZOU & HASTIE, 2005] Zou, H. and Hastie, T. Regularization and variable selection via the elastic net. Journal of the Royal Statistical Society. Series B, 67(2):301, 2005.