

Agrupamento por interseção de conjuntos para identificação de *spam* em *emails*

Luyza Ellen D. do Nascimento, David P. Galvão Júnior, Gilberto F. de Sousa Filho, Lucidio dos Anjos Formiga Cabral

Programa de Pós-Graduação em Informática - Centro de Informática - UFPB
Rua dos Escoteiros – Mangabeira, 58058-600, João Pessoa, PB
l.uyzadn09@gmail.com, galvao2010@icloud.com,
{gilberto, lucidio}@ci.ufpb.br

RESUMO

Emails desempenham um papel muito importante no mundo empresarial e social. Ao mesmo tempo, há uma grande quantidade de *emails* não solicitados, ou *spam*, que contêm conteúdo malicioso, como *malware* e *phishing*. O aumento no volume de mensagens e a maior complexidade nas técnicas de *spam* tornam necessário adotar métodos eficientes para identificá-lo. Nesse sentido, para a classificação de *emails* como *spam*, este trabalho propõe um modelo intitulado Intersect Bayes. O modelo proposto combina o Teorema de Bayes com o Problema de Agrupamento por Interseção de Conjuntos (PAIC) para explorar as dependências entre palavras em uma base de *emails* rotulados. Para isso, introduzimos uma nova formulação matemática para o PAIC e uma heurística polinomial para obter soluções aproximadas. Os resultados demonstraram que o modelo proposto apresenta melhores métricas na classificação em comparação com modelos clássicos da literatura baseados no Teorema de Bayes, com um ganho de até 6% na acurácia.

PALAVRAS CHAVE. Naive Bayes, Spam, Agrupamento.

Tópicos: Aprendizagem de Máquina, Naive Bayes e Interseção de Conjuntos

ABSTRACT

Emails play a very important role in the business and social world. At the same time, there is a large amount of unsolicited *emails*, or *spam*, which contain malicious content such as *malware* and *phishing*. The increase in the volume of messages and the greater complexity in *spam* techniques make it necessary to adopt efficient methods to identify it. In this regard, for classifying *emails* as *spam*, this work proposes a model entitled Intersect Bayes. The proposed model combines Bayes' Theorem with the Set Intersection Clustering Problem (SICP) to explore the dependencies between words in a labeled *emails* database. To this end, we introduce a new mathematical formulation for SICP and a polynomial heuristic to obtain approximate solutions. The results demonstrated that the proposed model presents better classification metrics compared to classical models in the literature based on Bayes' Theorem, with a gain of up to 6% in accuracy.

KEYWORDS. Naive Bayes, Spam, Clustering.

Paper topics: Machine Learning, Naive Bayes, Intersection of Sets

1. Introdução

Atualmente, a comunicação baseada em texto está se tornando cada vez mais difusa, seja por *email*, aplicativos de mensagens, rede sociais ou notícias. O *email* é uma forma de comunicação rápida e bastante popular, desempenhando um papel muito importante no mundo empresarial e na vida social individual. Por ter um papel crucial nos negócios, frequentemente esse tipo de comunicação é alvo de agentes maliciosos, que enviam *emails* não solicitados, ou *spam*.

Esse tipo de *email* apresenta frequentemente como conteúdo ataques do tipo *malware* e tentativas de *phishing*. Recentemente, foi constatado que mais de 86% das mensagens recebidas por usuários são consideradas *spam* e cada vez mais novos ataques com táticas mais sofisticadas estão sendo criados, como *whaling* e falsificação de identidade [Bhardwaj e Sapra, 2020].

A proteção contra o *spam* é uma preocupação contínua para organizações de qualquer tipo, e com métodos cada vez mais sofisticados e complexos, as técnicas de identificação desse tipo de *email*, como aquelas baseadas em técnicas de aprendizagem de máquina e processamento de linguagem natural (PLN) necessitam ser cada vez mais desenvolvidas para mitigar esse problema eficientemente.

Nesse sentido, Yitagesu e Tijare [2016] apontam que existem dois tipos de classificação de *spam*: a realizada em relação ao remetente e em relação ao conteúdo do *email*. A primeira deve considerar informações como o nome de usuário e o endereço de *email* do remetente. Por sua vez, uma classificação baseada em conteúdo busca analisar as palavras que compõem o *email*.

No presente trabalho, consideramos o conteúdo do *email* como dados de entrada para sua classificação. O algoritmo Naive Bayes, que utiliza o Teorema de Bayes com uma premissa que os eventos individuais são independentes, teve grande sucesso nas primeiras soluções da literatura. Partimos da hipótese que a premissa de independência dos eventos é falha e propomos um modelo, intitulado Intersect Bayes, que utiliza algoritmos de agrupamento das palavras de acordo com sua ocorrência concomitante na base a priori de *emails*, sendo este agrupamento o guia para a identificação das dependências entre as palavras e uma melhor classificação das mensagens analisadas.

O restante do trabalho está organizado da seguinte forma: a Seção 2 apresenta trabalhos relacionados à classificação de *emails*. A Seção 3 descreve o Problema de Agrupamento por Interseção de Conjuntos (PAIC) e propõe soluções exata e heurística. Adiante, a Seção 4 descreve a metodologia empregada no trabalho para classificação de *emails* usando soluções do PAIC. A Seção 5 apresenta os experimentos computacionais realizados e seus resultados. E, por fim, a Seção 6 traz as considerações finais e trabalhos futuros.

2. Classificação de *emails*

Muitos métodos da literatura para classificação de *emails* empregam técnicas de aprendizagem de máquina supervisionada. [Ma et al., 2020] propõe a classificação dos *emails* da base de dados Enron, utilizando os métodos Multinomial Naive Bayes e Máquina de Vetores de Suporte (SVM). Os autores também descrevem as etapas de pré-processamento dos dados, nas quais as palavras foram reduzidas ao seu radical, *tags* de HTML, *stopwords* e palavras não significativas foram removidas. Esse pré-processamento é necessário para preparar os dados e deixar apenas as palavras necessárias. Os autores utilizaram o método *bag-of-words* para extrair características de uma lista de palavras dos dados textuais, tornando-os utilizáveis em algoritmos de aprendizagem de máquina.

[Occhipinti et al., 2022] abordam a classificação de *emails* com 12 modelos de aprendizagem de máquina. Eles também utilizam a base de dados Enron. Para o pré-processamento desta, foram aplicadas técnicas de Processamento Natural de Linguagem (PLN), incluindo técnicas de stemização (ou *stemming*, em inglês) e lematização. Além disso, foram removidas *tags* em HTML e *stopwords*. O objetivo do pré-processamento é reduzir o ruído dos dados e sua dimensionalidade.

No trabalho [Occhipinti et al., 2022] os modelos utilizados foram: KNN, MPNN, Regressão Logística, Random Forest, XGBoost, Naive Bayes (Multinomial, Gaussiano e Bernoulli), SVM (Linear, Polinomial, RBF e Sigmoid). Destaca-se também o tempo necessário para que cada classificador pudesse chegar a um resultado satisfatório. Dentre eles, os classificadores que se destacaram foram o XGBoost, o MPNN e o SVM RBF.

A classificação de texto não se restringe apenas a *emails*, ela pode se estender a outras formas de texto, como notícias digitais. O trabalho de [Lima, 2023] utiliza técnicas de PLN e de *Deep Learning*, que recebem os *tokens* dos textos representados matematicamente em vetores (*Word2Vec*). A hipótese deste trabalho é encontrar similaridades entre palavras, pois as palavras que compartilham contextos similares tendem a ter significados semelhantes.

3. Soluções para o Problema de Agrupamento por Interseção de Conjuntos (PAIC)

O PAIC foi proposto por [Galvão Jr. et al., 2023] e procura encontrar grupos de objetos cuja interseção de recursos compartilhados seja maximizada. Nesta seção, apresentamos formalmente o PAIC, propomos uma nova formulação matemática e uma heurística para sua resolução.

3.1. Definição do PAIC

Sejam $X = \{x_1, x_2, \dots, x_n\}$ um conjunto de n objetos, L o conjunto de recursos compartilhados entre os objetos, $L_i \subseteq L$ o conjunto de recursos utilizados pelo objeto $x_i \in X$, $P = \{C_1, C_2, \dots, C_m\}$ um particionamento de X ,

$$L_{C_k} = \begin{cases} \bigcap_{x_i \in C_k} L_i & , \text{ se } |C_k| > 1 \\ \emptyset & , \text{ caso contrário,} \end{cases}$$

o conjunto interseção dos recursos utilizados pelos objetos contidos em uma parte C_k . No PAIC deseja-se encontrar um particionamento P que

$$\begin{aligned} & \text{Maximize} \quad \sum_{k=1}^m |L_{C_k}| \\ & \text{s.a.} \quad \bigcup_{k=1}^m C_k = X, \end{aligned} \tag{1}$$

$$C_k \neq \emptyset, \quad \forall k \in \{1, 2, \dots, m\}, \tag{2}$$

$$C_k \cap C_l = \emptyset, \quad \forall k, l \in \{1, 2, \dots, m\}, k \neq l. \tag{3}$$

Neste modelo o agrupamento é um particionamento de X obtida pelas restrições (1), (2) e (3), ou seja, tem-se como saída do problema um agrupamento P onde todo objeto $x_i \in X$ está em uma e somente uma parte de P , e a soma do tamanho das interseções de cada parte é máxima.

Por fim, a seguinte proposição foi provada em [Galvão Jr. et al., 2023]:

Proposição 1 *Seja (X, L) uma instância para o PAIC, existe pelo menos um particionamento ótimo P^* de X com no máximo dois objetos em cada parte.*

3.2. Reformulação matemática

Definição 1 (*Grafo interseção*) É um grafo orientado $G = (V, A)$, onde cada vértice $i \in V$ está associado ao objeto $x_i \in X$, o conjunto de arcos $A = \{(i, j) \mid i, j \in V, i \leq j\}$ representa a relação de interseção de recursos entre x_i e x_j . Os arcos são ponderados pelo peso

$$w_{ij} = \begin{cases} |L_i \cap L_j| & , \text{ se } i < j \\ 0 & , \text{ se } i = j. \end{cases}$$

Sejam $G = (V, A)$ um grafo de incidência, $A_v = \{(i, j) \in A \mid v = i \text{ ou } v = j\}$ o conjunto de arcos de A que possuem v como vértice extremo, a variável binária de decisão $y_{ij} = 1$ indicando que os vértices i, j formam uma parte $C = \{x_i, x_j\}$ contida na solução P e considerando a Proposição 1, temos que no PAIC deseja-se encontrar um particionamento P que

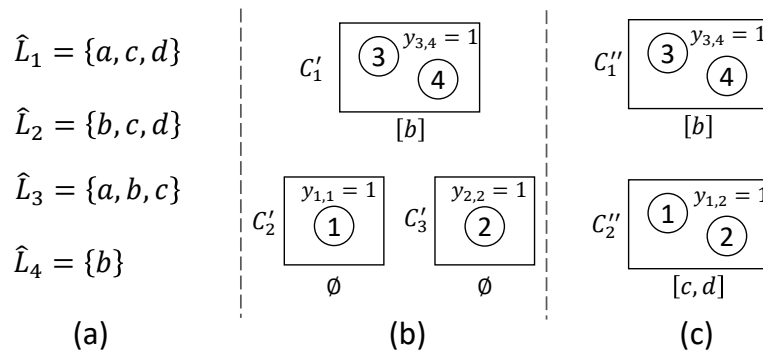
$$\text{Maximize} \quad \sum_{(i,j) \in A} w_{ij} y_{ij} \quad (4)$$

$$\begin{aligned} \text{s.a.} \quad & \sum_{(i,j) \in A_v} y_{ij} \leq 1, & \forall v \in V, \\ & y_{ij} \in \{0, 1\}, & \forall (i, j) \in A. \end{aligned} \quad (5)$$

A partição P formada por este modelo é composta por grupos que contém no máximo 2 objetos, sendo a solução ótima garantida pela Proposição 1. A função objetivo (4) procura maximizar a interseção dos recursos compartilhados entre pares de objetos que compõem P . Já as restrições (5) garantem que cada vértice esteja no máximo em um grupo da partição P . Todo vértice v cujo $\sum_{(i,j) \in A_v} y_{ij} = 0$ ou $y_{vv} = 1$ será adicionado em um grupo unitário $C = \{x_v\}$ em P sem perda de otimalidade.

A Figura 1(a) ilustra uma instância exemplo $\hat{X} = \{x_1, x_2, x_3, x_4\}$ para o PAIC através dos conjuntos de recursos $\hat{L}_i$. A Figura 1(b) ilustra um particionamento $P' = \{C'_1, C'_2, C'_3\}$ de $\hat{X}$, onde $C'_1 = \{x_3, x_4\}$, definido pela variável $y_{3,4} = 1$, e possui apenas o recurso b em sua interseção, logo $w_{3,4} = 1$. As partes $C'_2 = \{x_1\}$ e $C'_3 = \{x_2\}$, definidas por $y_{1,1} = y_{2,2} = 1$, formam grupos unitários e não possuem recursos em sua interseção, logo, P' contém apenas um recurso. Já a Figura 1(c) apresenta o particionamento ótimo de $\hat{X}$, pois $C''_2 = \{x_1, x_2\}$, formado por $y_{1,2} = 1$, possui $L_{C''_2} = \{c, d\}$, ou seja, $w_{1,2} = 2$, aumentando para três os recursos nas interseções de P'' .

Figura 1: Soluções para o PAIC. (a) Recursos dos objetos da instância $\hat{X} = \{x_1, x_2, x_3, x_4\}$, (b) solução P' de $\hat{X}$ e (c) solução ótima P'' de $\hat{X}$.



3.3. Heurística para o PAIC

Considerando a Proposição 1 definimos uma heurística gulosa para resolução do PAIC chamada *ParticaoGulosa*. O algoritmo consiste em, a cada iteração, adicionar à partição solução P os dois objetos ainda não pertencentes a P que contém a maior interseção. O Algoritmo 1 descreve a heurística *ParticaoGulosa*, recebendo como entrada o conjunto de objetos X e o grafo de interseção $G = (V, A)$. Na linha 1 inicializamos a solução P vazia, já na linha 2 construímos o conjunto C com todos os arcos $(i, j) \in A$ cujos objetos associados possuem interseção não vazia ($w_{ij} > 0$). No pior caso, todo objeto possui interseção de recursos com outro objeto, logo,

$|C| \approx O(|X|^2)$. Em seguida ordenamos C de modo não crescente pelo tamanho desta interseção. No laço das linhas 4-8 temos que enquanto $C \neq \emptyset$ pegamos o arco (i, j) com maior peso de C , adicionamos o par de objetos $\{x_i, x_j\}$ em P , e atualizamos C removendo os arcos $A_i \cup A_j$ para garantir que os objetos x_i e x_j não sejam colocados em outro grupo de P . Por fim, na linha 9, todos objetos $x \in X$ que não estiverem em P serão adicionados em um grupo com interseção vazia.

Algoritmo 1: ParticaoGulosa($X, G = (V, A)$)

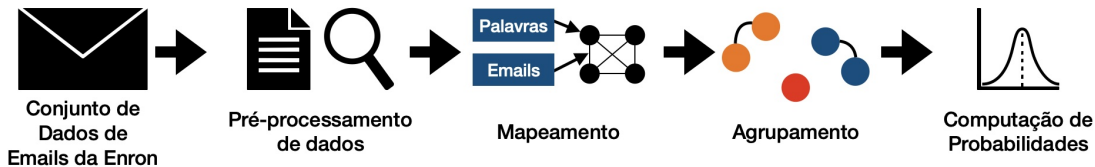
```

1  $P \leftarrow \emptyset$ 
2  $C \leftarrow \{(i, j) \in A \mid w_{ij} > 0\}$  // pares de objetos de  $X$  com interseção não vazia
3 Ordene  $C$  não crescente pelo peso  $w_{ij}$ 
4 enquanto  $C \neq \emptyset$  faça
5    $(i, j) \leftarrow$  primeiro arco de  $C$ 
6    $P \leftarrow P \cup \{x_i, x_j\}$ 
7    $C = C \setminus (A_i \cup A_j)$ 
8 fim
9  $P \leftarrow P \cup \{x \in X \mid x \text{ não está em } P\}$ 
10 retorna  $P$ 
  
```

4. Metodologia para classificação de *emails*

Nesta seção, apresentamos a metodologia adotada no trabalho para a classificação de *emails* como *spam*. A Figura 2 ilustra as etapas envolvidas no processo de classificação proposto, sendo cada etapa melhor detalhada nas subseções seguintes.

Figura 2: Metodologia do modelo Intersect Bayes para classificação de *emails*



Neste estudo utilizamos o conjunto de dados Enron *spam corpus* [Metsis et al., 2006]. Para evitar viés no conjunto de dados, definimos usar no processo de aprendizado a mesma proporção de *emails spam* e não-*spam*. A base original possui 17.171 *emails spam* e 16.545 *emails não-spam*, logo realizamos uma subamostragem da classe majoritária, sendo selecionados aleatoriamente 16.545 *emails spam*, permitindo uma proporção igual das duas classes.

A primeira etapa do método proposto (Seção 4.1) consiste em uma limpeza do conjunto de dados. A segunda etapa (Seção 4.2) realiza um mapeamento do conjunto de *emails* e palavras em estruturas para utilização pelo PAIC. Com essas estruturas, a terceira etapa consiste em criar particionamento das palavras pelo algoritmo *ParticaoGulosa*. Por fim, na quarta etapa (Seção 4.3) propomos um modelo para computar a probabilidade dos *emails* da base de teste serem *spam* guiado pelos particionamentos das palavras na base de treinamento.

4.1. Pré-processamento

Efetuamos uma etapa de pré-processamento dos dados para reduzir os ruídos e melhorá-los para a etapa de classificação. Para isso, a coluna relacionada à data foi removida e as colunas do assunto e conteúdo do texto foram concatenadas para a aplicação de técnicas de Processamento Natural de Linguagens (PLN). Com fins de reduzir a quantidade de palavras presentes na base de dados (vocabulário) e remover palavras desnecessárias foram empregadas as seguintes técnicas:

- Tokenização: divide o texto em unidades menores, geralmente palavras;
- Conversão de todas as letras para minúsculas;
- Lematização [Cambria e White, 2014]: simplifica palavras flexionadas para suas formas básicas, como transformar 'seguir' em 'seguir', mantendo apenas palavras que são adjetivos, advérbios, verbos auxiliares, substantivos, pronomes e verbos;
- Remoção de ruídos: exclui elementos textuais sem dígitos ou compostos por uma única letra, bem como remove pontuação, caracteres não alfanuméricos e *stop words* em português usando a lista do modelo spaCy [Honnibal e Montani, 2017];
- Filtragem contextual: elimina palavras específicas ao contexto dos *emails*, como endereço do *email*, *tags* em HTML e URLs;

Por fim, realizamos uma codificação do rótulo para valores numéricos, mapeando valores *spam* para "1" e não-*spam* para "0".

Ao final das etapas de pré-processamento, tem-se que o conjunto de dados é composto por um vocabulário de 125.365 palavras, o que significa uma redução de 98,8% em relação ao vocabulário original, com uma média de palavras por *email* $\mu = 131$ e desvio padrão $\sigma = 379$. Nesse caso, a distribuição do número de palavras por *email* apresenta uma forma de cauda longa à esquerda, com uma concentração de *emails* com poucas palavras, sendo os valores dos percentis 50 e 90 respectivamente 61 e 270, o que explica a discrepância entre o desvio padrão e a média.

4.2. Mapeamento

Para que possamos gerar um particionamento de palavras para cada *email* do conjunto de dados, realizamos um mapeamento desta base para as estruturas de entrada do PAIC. Cada palavra do vocabulário é mapeada para o conjunto de objetos X , enquanto que cada *email* é mapeado para o conjunto de recursos L , sendo assim, para toda palavra $w \in X$ temos que L_w representa o conjunto de *emails* que possuem w . No entanto, considerando que existem duas categorias de *emails*, definimos L^c como a parte do conjunto de *emails* rotulados na classe c e L_w^c a parte do conjunto de *emails* rotuladas em c que w aparece. Ou seja, L^{spam} é o conjunto de *emails* rotulados como *spam* e $L^{não-spam}$ é o conjunto de *emails* não-*spam*. Observe que como nos dados deste trabalho $c \in \{spam, não-spam\}$, tem-se que $L^{spam} \cup L^{não-spam} = L$ e $L_w^{spam} \cup L_w^{não-spam} = L_w$.

Como iremos trabalhar com uma técnica de aprendizado supervisionado, a base de *emails* rotulados (a priori) será utilizada para construir o grafo de interseção $G = (V, A)$ como definido na Seção 3.2. Servindo assim na classificação de uma base de *emails* que não possuem rotulação prévia (posteriori). Para isto, seja e um *email* não rotulado qualquer, e M o conjunto de vértices associados às palavras de e , definimos o subgrafo de interseção induzido $H = G[M]$ formado pelas arcos (i, j) de G cujos vértices $i, j \in M$. Logo, com o auxílio da heurística proposta na Seção 3.3, fazemos $P = ParticaoGulosa(M, H)$, sendo P a partição das palavras de M que será utilizada para a classificação do *email* e como descrito na próxima seção.

4.3. Teorema de Bayes para classificação

Em trabalhos anteriores, diversos autores demonstraram que o uso do Teorema de Bayes é eficaz para problemas de classificação [Domingos e Pazzani, 1996; Langley et al., 1992; Sakkis et al., 2004]. Neste trabalho, propomos utilizá-lo para computar a probabilidade condicionada de um *email* e ser *spam* caso apresente como conteúdo deste o conjunto de palavras M .

Do Teorema de Bayes e da Lei da Probabilidade Total, a probabilidade de um *email* e , cujas palavras $M = \{w_1, \dots, w_n\}$ fazem parte de seu conteúdo, pertencer a categoria c é:

$$Pr(e = c | M) = \frac{Pr(M | e = c) \cdot Pr(e = c)}{\sum_{k \in \{spam, não-spam\}} Pr(M | e = k) \cdot Pr(e = k)}. \quad (6)$$

Computar $Pr(M | e = c)$ não é uma tarefa trivial, e por isto, o classificador Naive Bayes assume de forma simplista que o evento de participação individual de cada palavra $M = w_i$ é independente das outras, logo, tem-se que

$$Pr(M | e = c) = \prod_{i=1}^n Pr(M = w_i | e = c).$$

As probabilidades a priori básicas contidas nos dados são dadas por:

$$Pr(e = c) = \frac{|L^c|}{|L|} \quad (7)$$

$$Pr(M = w_i | e = c) = \frac{k + |L_{w_i}^c|}{2k + |L^c|}, \quad (8)$$

onde $k = 0,01$ é uma constante aplicada para mitigar erros de classificação quando a ocorrência do evento $M = m_i$ for rara no conjunto de dados.

Modelo Intersect Bayes

Conforme mencionado anteriormente, a suposição de independência dos eventos $M = m_i$ é simplista, e para o modelo Intersect Bayes proposto supomos que existe um relacionamento entre palavras que caracterizam *emails* do tipo *spam*. Para quantificar estas relações contidas nos dados, propomos a construção do particionamento P das palavras $M = \{m_1, \dots, m_n\}$ através do algoritmo *ParticaoGulosa*, sendo cada grupo de palavras $S \in P$ uma relação de dependência medida pela suas ocorrências concomitantes em outros *emails*. Logo, reescrevemos a probabilidade condicionada do evento de presença das palavras M no conteúdo de um *email* de rótulo c como

$$Pr(M | e = c) = \prod_{S \in P} Pr(M = S | e = c).$$

Por sua vez, a probabilidade a priori do evento $M = S$, quando $L_S \neq \emptyset$, pode ser computada por:

$$Pr(M = S | e = c) = \frac{k + |L_S^c|}{2k + |L_S|}, \quad (9)$$

com a aplicação da mesma constante k com fins de tratamento de eventos raros. Observe que a probabilidade a priori do evento $M = S$ utiliza no denominador L_S , que é o conjunto dos *emails* dos quais cada grupo de palavras da partição aparece concomitantemente. Com tal mudança, a probabilidade a priori é calculada com base em um universo reduzido que é o conjunto de *emails* na interseção que necessariamente está contido no conjunto de *emails* original.

Por fim, para casos onde $L_S = \emptyset$, a Equação (9) é substituída por:

$$Pr(M = S | e = c) = \prod_{w \in S} Pr(M = w | e = c)$$

que é justamente a computação da probabilidade a priori definida pela Equação (8) aplicada para palavras cujos eventos são independentes.

Para o entendimento do efeito da identificação de dependências entre eventos pelo PAIC para a classificação de *emails*, propomos computar a probabilidade condicionada $Pr(\hat{e} = spam | \hat{M})$ do *email* exemplo $\hat{e}$ cuja lista de palavras $\hat{M} = \{"grátis", "jogo", "bet", "frete"\}$ tem seu histórico em *emails* rotulados ilustrado na Figura 3(a). Neste exemplo ilustrativo, consideramos a constante de eventos raros $k = \frac{1}{2}$ para facilitar o entendimento.

Figura 3: Dados a priori para classificação do *email* exemplo $\hat{e}$. (a) *Emails* da base rotulada que contém as palavras de $\hat{M}$ e (b) probabilidades a priori contidas no histórico de $\hat{M}$.

<i>emails</i>					$\hat{M}$	$Pr(\hat{M} spam)$	$\hat{M}$	$Pr(\hat{M} \overline{spam})$
"grátis"	e_1	e_2	e_3	e_5	"grátis"	$\frac{1/2 + 2}{1 + 4} = \frac{1}{2}$	"grátis"	$\frac{1/2 + 2}{1 + 4} = \frac{1}{2}$
"jogo"	e_1	e_3	e_4	e_6	"jogo"	$\frac{1/2 + 2}{1 + 4} = \frac{1}{2}$	"jogo"	$\frac{1/2 + 2}{1 + 4} = \frac{1}{2}$
"bet"	e_2	e_3	e_4	e_5	"bet"	$\frac{1/2 + 2}{1 + 4} = \frac{1}{2}$	"bet"	$\frac{1/2 + 2}{1 + 4} = \frac{1}{2}$
"frete"	e_1	e_2	e_4	e_6	"frete"	$\frac{1/2 + 3}{1 + 4} = \frac{7}{10}$	"frete"	$\frac{1/2 + 1}{1 + 4} = \frac{3}{10}$
$\hat{L}^{\overline{spam}} = \{e_3, e_5, e_6\}$					$Pr(\hat{e} = spam) = \frac{1}{2}$		$Pr(\hat{e} = \overline{spam}) = \frac{1}{2}$	
$\hat{L}^{spam} = \{e_1, e_2, e_4\}$								

(a)
(b)

A Figura 3(b) apresenta as probabilidades a priori extraídas do histórico de *emails* que contém as palavras de $\hat{M}$. Por exemplo, considerando a Equação (8) e $k = \frac{1}{2}$, a probabilidade da palavra "grátis" está contida em um *spam* é $Pr(\hat{M} = "grátis" | \hat{e} = spam) = \frac{1/2+2}{1+4} = \frac{1}{2}$, pois dos 4 *emails* que "grátis" aparece (e_1, e_2, e_3, e_5), 2 deles são rotulados como *spam* (e_1, e_2). A chance de um *email* $\hat{e}$ qualquer ser *spam* é $Pr(\hat{e} = spam) = \frac{1}{2}$. A rotulação destes *emails* está ilustrada nas listas $\hat{L}^{spam}$ e $\hat{L}^{\overline{spam}}$ da Figura 3(a).

Inicialmente, para efeito de comparação, assumiremos que os eventos são independentes, logo,

$$Pr(\hat{M} | \hat{e} = spam) = \frac{1}{2} \cdot \frac{1}{2} \cdot \frac{1}{2} \cdot \frac{7}{10} = \frac{7}{80},$$

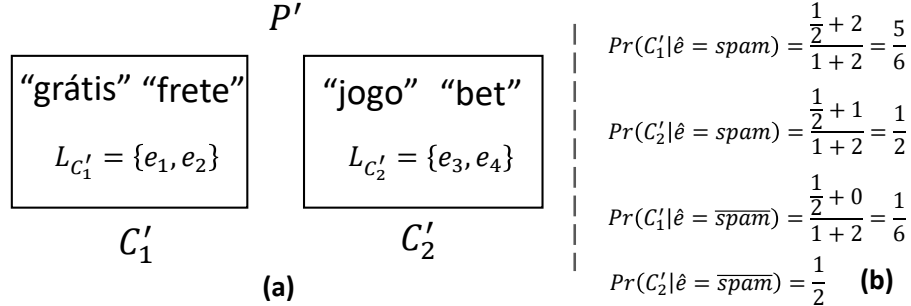
e de forma análoga computamos $Pr(\hat{E} | \hat{e} = \overline{spam}) = \frac{3}{80}$, e aplicando a Equação (6) temos que $Pr(\hat{e} = spam | \hat{M}) = \frac{7}{10} = 70\%$.

Agora considerando a existência de dependência dos eventos entre as palavras de $\hat{M}$ e utilizando a heurística *ParticaoGulosa* do PAIC construímos o particionamento $P' = \{C'_1, C'_2\}$ ilustrado na Figura 4. Temos também as probabilidades a priori dos grupos C'_1 e C'_2 , como por exemplo, considerando a Equação (9) e $k = \frac{1}{2}$, podemos observar que a probabilidade $Pr(\hat{M} = C'_1 | \hat{e} = spam) = \frac{1/2+2}{1+2} = \frac{5}{6}$, pois $L_{C'_1} = L_{C'_1}^{spam} = \{e_1, e_2\}$.

De posse das probabilidades a priori das partições, podemos recalculer

$$Pr(\hat{M} | \hat{e} = spam) = \frac{5}{6} \cdot \frac{1}{2} = \frac{5}{12}$$

Figura 4: Particionamento P' de $\hat{M}$ gerado pela heurística *ParticaoGulosa*. (a) Particionamento $P' = \{C'_1, C'_2\}$ e (b) as probabilidades a priori dos grupos C'_1 e C'_2 .



e $Pr(\hat{M} | \hat{e} = \overline{spam}) = \frac{1}{12}$. Por fim, aplicando a Equação (6) temos que $Pr(\hat{e} = spam | \hat{M}) = \frac{5}{6} = 83,3\%$, ou seja, com o modelo Intersect Bayes houve um aumento da probabilidade do *email* $\hat{e}$ ser *spam* que estava em 70% com o modelo Naive Bayes.

5. Experimentos computacionais

O algoritmo proposto na Seção 3.3 e os algoritmos de aprendizagem de máquina foram implementados na linguagem de programação Python versão 3.9.6 com auxílio da biblioteca *scikit-learn* [du Boisberranger et al., 2022]. Todos os experimentos computacionais deste trabalho foram executados em um computador com processador Intel Core i5, de 4 núcleos de 2,3 GHz, a máquina dispõe de 8 GB de memória RAM, e possui como sistema operacional o MacOS Sonoma.

5.1. Resultados do PAIC

Para a avaliação das técnicas de agrupamento propostas, realizamos testes em instâncias artificiais propostas por [Galvão Jr. et al., 2023]. Selecionamos 31 instâncias com $|X|$ entre 40 e 50 objetos, que fica próximo da mediana, em palavras, para *emails* do conjunto de dados utilizado.

Dentro do limite de execução de 60 s, em nenhuma instância o modelo exato, proposto na Seção 3.2, foi capaz de encontrar a solução ótima. Ao mesmo tempo, em todas as instâncias a heurística *ParticaoGulosa* foi capaz de encontrar uma solução em até 10 ms. Em relação a qualidade da solução, em nenhuma instância a heurística obteve um particionamento com soma das interseções superior à solução primal do modelo exato. Considerando $GAP = (1 - \frac{\text{solução heurística}}{\text{solução primal}}) \times 100\%$, a heurística obteve um GAP médio de 1,63 % com desvio padrão $\sigma = 1,25\%$.

Tais valores mostram que é possível sacrificar um pouco a qualidade da solução em troca de obter soluções válidas com relativo baixo esforço computacional, uma vez que a heurística *ParticaoGulosa* possui complexidade $O(|X|^2)$. Sendo assim, considerando o tamanho do conjunto de dados de *emails* da literatura, optamos por utilizar a heurística *ParticaoGulosa* para guiar o classificador proposto.

5.2. Resultados da classificação de *emails*

Comparamos o método proposto para classificação de *emails*, intitulado Intersect Bayes, com a versão original do Naive Bayes e o Multinomial Naive Bayes [Murphy, 2012] sob a base de dados de *emails* da *Enron*. Para a escolha do melhor modelo aplicamos a técnica *k-fold cross validation* ($k = 5$). Nela o conjunto de dados são divididos em k pedaços (*fold*), e para cada modelo são executadas k iterações, ao final é computada a média das métricas de classificação das iterações, sendo considerado o melhor modelo aquele que obtiver a melhor média. Em cada iteração um pedaço dos dados é chamado de base de validação e o restante base de treinamento, na iteração seguinte troca-se o pedaço da validação com um dos pedaços do treinamento. A base de treinamento

foi utilizada pelos algoritmos, conforme descrito na Seção 4, enquanto a base de validação foi usada para medir a qualidade dos classificadores treinados, definindo como métricas básicas dos classificadores: vp indicando o número de classificações corretas das amostras de teste como *spam*, vn indicando o número de classificações incorretas das amostras de teste como *spam*, fp indicando o número de classificações corretas das amostras de teste como não-*spam*, e fn indicando o número de classificações incorretas das amostras de teste como não-*spam*.

A Tabela 1 apresenta, para cada método, os valores médios para as seguintes métricas de classificação: precisão = $\frac{vp}{vp+vn}$, $recall = \frac{vp}{vp+fn}$, $f1 = 2 \times \frac{\text{precisão} \times \text{recall}}{\text{precisão} + \text{recall}}$, acurácia = $\frac{vp+fp}{vp+vn+fp+fn}$ e o *ROC-AUC* que representa a área abaixo da curva ROC que é um gráfico que mostra a relação entre a taxa de verdadeiros positivos e a taxa de falsos positivos para diferentes limiares de decisão do classificador, o melhor valor para esta métrica é 1 [Fawcett, 2006]. Para cada métrica adotada, a melhor média entre os métodos avaliados está marcado em negrito.

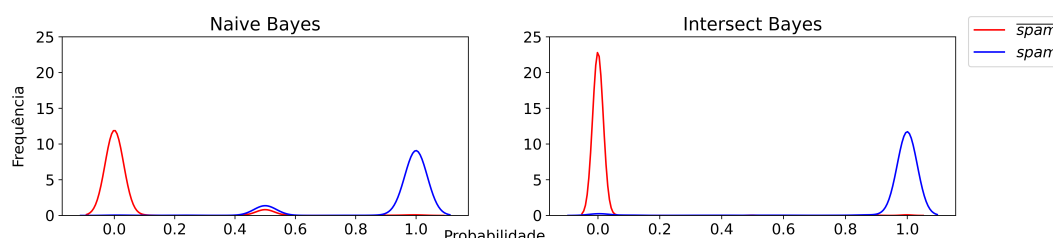
Tabela 1: Métricas dos modelos de classificação de *e-mails* sob a base de dados Enron.

Modelo	Métrica				
	precisão	<i>recall</i>	<i>f1</i>	acurácia	ROC-AUC
Intersect Bayes	0,992	0,975	0,984	0,984	0,998
Naive Bayes	0,987	0,860	0,919	0,924	0,992
Multinomial Naive Bayes	0,980	0,985	0,982	0,982	0,996

Da Tabela 1 é possível observar que, o modelo Intersect Bayes apresenta melhores médias em 4 das 5 métricas avaliadas, sendo a exceção o *recall*. Entretanto, observe que o modelo proposto produz um ganho médio de aproximadamente 6,5% na métrica *f1* em relação ao Naive Bayes e de 0,2% em relação ao Multinomial Naive Bayes, sendo o *f1* uma média harmônica entre a precisão e o *recall*, o que indica um melhor equilíbrio do modelo. Ao mesmo tempo, observe que o modelo Intersect Bayes produz um ganho médio de aproximadamente 6% na acurácia em relação ao Naive Bayes e de 0,2% em relação ao Multinomial Naive Bayes. Em geral, as métricas mostram que o modelo proposto é superior na distinção entre os diferentes tipos de *emails*.

Para tentar entender melhor o ganho das métricas de classificação do Intersect Bayes sobre o Naive Bayes, selecionamos um conjunto validação V contendo 6.618 amostras, separamos seus *emails* pelo rótulo formando os conjuntos V^{spam} e $V^{non-spam}$ e criamos, para cada modelo, uma estimativa de densidade por *kernel* (EDK) [Silverman, 1986] contendo os valores das probabilidades, computadas pelo classificador de cada modelo, de um *email* ser considerado *spam*. A Figura 5 ilustra os EDKs de cada modelo. A linha azul representa a distribuição de valores da probabilidade fornecida pelos classificadores sobre os *emails* de V^{spam} , já a linha vermelha apresenta a distribuição dos valores obtidos com a classificação dos *emails* de $V^{non-spam}$.

Figura 5: Estimativa de densidade por *kernel* para os modelos Naive Bayes e Intersect Bayes.



Os dois modelos, como esperado, apresentam maior concentração de valores de suas probabilidades posteriores em 0 ou 1, o que indica uma confiança do modelo na classificação proposta. Entretanto, nota-se um comportamento especial no Naive Bayes tradicional que é uma concentração de instâncias com probabilidades próximas de 0,5. Isto não ocorre no Intersect Bayes, levando seu classificador a errar menos como demonstrado nas métricas anteriormente. A seguir apresentamos duas possíveis explicações para este fenômeno:

1. O modelo Intersect Bayes possui uma redução implícita no ruído de sua classificação devido à diminuição da perda de precisão numérica, obtida com o agrupamento de palavras. Com este agrupamento, temos menos elementos no produto das probabilidades dos eventos independentes, o que reduz a série de multiplicações entre números no intervalo (0,1). Para exemplificar, em média, uma instância do conjunto de dados Enron contém 131 palavras, enquanto que as soluções do PAIC, utilizado pelo Intersect Bayes, contém 40 grupos de palavras.
2. Quando, em um *email* específico e , as palavras que o compõem possuem probabilidades a priori individuais próximas de 0,5, isto leva o Naive Bayes a produzir uma probabilidade posteriori próxima de 0,5 para e . No Intersect Bayes ocorre a identificação da relação, a priori, de um grupo de palavras com uma classe de *emails* c , mesmo que esta relação com a classe c não ocorra com as palavras separadamente. Logo, se este grupo ocorrer em e aumentará muito a probabilidade de e ser classificada como c . Este tipo de melhoria da classificação de *emails* ao agrupar palavras pode ser observada no exemplo da Seção 4.3.

Por fim, é importante salientar que o Intersect Bayes possui um custo computacional de classificação maior em relação aos outros modelos. Em nossos testes, em média, o tempo gasto em treinamento e predição foi 62 vezes superior no Intersect Bayes em comparação ao Naive Bayes.

6. Considerações finais e trabalhos futuros

O *spam* é um *email* indesejado e de origem duvidosa que pode conter ataques como *malware* ou tentativas de *phishing*. Trabalhos anteriores sobre classificação de *emails* fazem uso tanto de técnicas de PLN como modelos de classificação baseadas no Teorema de Bayes. Neste trabalho foi proposto um modelo para classificação de *emails*, denominado Intersect Bayes, que parte do algoritmo Naive Bayes, não aceitando a premissa de seu modelo em que os eventos de ocorrência das palavras nos *emails* são independentes. Para identificar estas dependências entre as palavras, mapeamos a base de *emails* de entrada à uma heurística proposta para o Problema de Agrupamento por Interseção de Conjuntos (PAIC).

Os experimentos conduzidos revelaram que, em um conjunto de dados de *emails* corporativos, o classificador proposto foi capaz de obter métricas de classificação melhores em comparação as técnicas clássicas baseadas no Teorema de Bayes (Naive Bayes e Multinomial Naive Bayes). No entanto, o custo computacional do Intersect Bayes é significativamente maior comparado aos outros algoritmos.

Sendo assim, como trabalhos futuros propomos o estudo de técnicas de agrupamentos das palavras que melhorem a qualidade das soluções do PAIC e diminuam seu custo computacional, podendo assim, melhorar a classificação e diminuir o tempo computacional do classificador Intersect Bayes. Em outro ponto, pretendemos estudar a aplicação do modelo Intersect Bayes em outros conjuntos de dados envolvendo, por exemplo, a identificação de notícias falsas e análise de sentimento.

Referências

- Bhardwaj, A. e Sapra, V. (2020). Why is phishing still successful? *Computer Fraud & Security*, 2020:15–19.
- Cambria, E. e White, B. (2014). Jumping nlp curves: A review of natural language processing research [review article]. *IEEE Computational Intelligence Magazine*, 9(2):48–57.
- Domingos, P. M. e Pazzani, M. J. (1996). Beyond independence: Conditions for the optimality of the simple bayesian classifier. In *International Conference on Machine Learning*.
- du Boisberranger, J., Van den Bossche, J., Estève, L., e J. Fan, T. (2022). scikit-learn: machine learning in python. URL <https://scikit-learn.org/stable/>.
- Fawcett, T. (2006). An introduction to roc analysis. *Pattern Recognition Letters*, 27(8):861–874. ISSN 0167-8655. ROC Analysis in Pattern Recognition.
- Galvão Jr., D. P., Sousa Filho, G. F., e Cabral, L. A. F. (2023). Classificação de fraudes em licitações públicas através do agrupamento de empresas em conluíus. In *Anais do XLIII Congresso da Sociedade Brasileira de Computação*, p. pp. 13–24, João Pessoa, PB. Sociedade Brasileira de Computação.
- Honnibal, M. e Montani, I. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear, 2017.
- Langley, P., Iba, W., and, e Thompson, K. (1992). An analysis of bayesian classifiers. In *Proceedings of the Tenth National Conference on Artificial Intelligence*, p. 223–228. AAAI Press. ISBN 0262510634.
- Lima, G. d. S. (2023). Classificação de notícias digitais utilizando processamento de linguagem natural.
- Ma, T. M., YAMAMORI, K., e Thida, A. (2020). A comparative approach to naïve bayes classifier and support vector machine for email spam classification. p. 324–326.
- Metsis, V., Androutsopoulos, I., e Paliouras, G. (2006). Enron-spam datasets. Online. Available from: http://nlp.cs.aueb.gr/software_and_datasets/Enron-Spam/index.html.
- Murphy, K. P. (2012). *Machine Learning: A Probabilistic Perspective*. MIT Press.
- Occhipinti, A., Rogers, L., e Angione, C. (2022). A pipeline and comparative study of 12 machine learning models for text classification. *Expert Systems with Applications*, 201:117193. ISSN 0957-4174.
- Sakkis, G., Androutsopoulos, I., Paliouras, G., Karkaletsis, V., Spyropoulos, C. D., e Stamatopoulos, P. (2004). A memory-based approach to anti-spam filtering for mailing lists. *Information Retrieval*, 6:49–73.
- Silverman, B. W. (1986). *Density Estimation for Statistics and Data Analysis*, volume 26 of *Mono-graphs on Statistics and Applied Probability*. Chapman and Hall, London.
- Yitagesu, M. e Tijare, M. (2016). Email classification using classification method. *International Journal of Engineering Trends and Technology*, 32:142–145. ISSN 2231-5381.