UNIVERSIDADE FEDERAL DA FRONTEIRA SUL
CAMPUS CHAPECÓ
CURSO DE CIÊNCIA DA COMPUTAÇÃO

MARCELO ACORDI

SELEÇÃO DE AMOSTRAS DE DADOS MENOS REPRESENTATIVAS USANDO APRENDIZADO ATIVO

CHAPECÓ
2021

MARCELO ACORDI

SELEÇÃO DE AMOSTRAS DE DADOS MENOS REPRESENTATIVAS USANDO APRENDIZADO ATIVO

Trabalho de conclusão de curso de graduação apresentado como requisito para obtenção do grau de Bacharel em Ciência da Computação da Universidade Federal da Fronteira Sul.

Orientador: Prof. Dr. Guilherme Dal Bianco

CHAPECÓ
2021

Acordi, Marcelo

Seleção de Amostras de Dados Menos Representativas Usando Aprendizado Ativo / por Marcelo Acordi. – 2021.
44 f.: il.; 30 cm.

Orientador: Guilherme Dal Bianco
Monografia (Graduação) - Universidade Federal da Fronteira Sul, Ciência da Computação, Curso de Ciência da Computação, SC, 2021.

1. Aprendizado ativo. 2. Aprendizado de máquina. 3. Aprendizado semi-supervisionado. 4. Aprendizado ativo baseado em regras. I. Bianco, Guilherme Dal. II. Título.

E-mail: pawnsimple@gmail.com

MARCELO ACORDI

SELEÇÃO DE AMOSTRAS DE DADOS MENOS REPRESENTATIVAS USANDO APRENDIZADO ATIVO

Trabalho de conclusão de curso de graduação apresentado como requisito para obtenção do grau de Bacharel em Ciência da Computação da Universidade Federal da Fronteira Sul.

Orientador: Prof. Dr. Guilherme Dal Bianco

Este trabalho de conclusão de curso foi defendido e aprovado pela banca em: 10 / 05 / 21

BANCA EXAMINADORA:

Dr. Guilherme Dal Bianco - UFFS

Dr. Denio Duarte - UFFS

Me. Adriano Sanick Padilha - UFFS

RESUMO

Dados estão cada vez mais disponíveis de serem coletados e armazenados, consequência da grande quantidade produzida por diversos dispositivos interconectados. Tais dados podem, por exemplo, serem usados em tarefas de aprendizado supervisionado, cujo objetivo é prever um comportamento com base nos dados rotulados, previamente fornecidos. Métodos supervisionados são utilizados no contexto de classificação de informações, como por exemplo classificar se um e-mail é SPAM ou não, ou categorizar documentos de texto em categorias predefinidas - esporte, política, etc. Porém caso os dados não apresentarem rótulos, dentro do contexto de aprendizado supervisionado, pode ser difícil sua possibilidade de uso. A rotulagem é um processo que pode ser custoso em questão de tempo ou em recursos financeiros. Dessa forma, encontrar exemplos informativos e representativos pode representar uma redução de custos. Neste contexto, a aprendizagem ativa consiste no estudo de técnicas para redução no número de instâncias presentes no treinamento, selecionando somente as mais informativas para rotulagem. Este trabalho buscou explorar configurações de um algoritmo de aprendizado ativo ambicionando selecionar mais instâncias positivas e a redução da quantidade de instâncias selecionadas. Com os experimentos verificou-se a possibilidade de incremento de instâncias positivas e redução de negativas.

Palavras-chave: Aprendizado ativo. Aprendizado de máquina. Aprendizado semi-supervisionado. Aprendizado ativo baseado em regras.

ABSTRACT

Data is increasingly available to be collected and stored, a consequence of the large amount produced by several interconnected devices. This data, for example, be used in supervised learning tasks, the purpose of which is to predict behavior based on previously labeled data. Supervised methods are used in the context of classification, such as classifying whether an email is SPAM or not, or categorizing text documents into predefined categories - sports, politics, etc. However, if the data does not have labels, on supervised learning, the possibility of use may be difficult. Labeling is a process that can be costly in a matter of time or financial resources. However, finding informative and representative examples represents a cost reduction. In this context, active learning consists of studying techniques to reduce the number of instances present in the training, selecting only the most informative ones for labeling. This work sought to explore configurations of an active learning algorithm in order to select more instances of the non-dominant class and reduce the total of selected instances. With the experiments, it was possible to increase the non-dominant class instances and reduce the quantity of dominant class instances.

Keywords: Active learning algorithm,Active learning, Machine learning, Semi-supervised learning, Active rules-based learning.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

AAA	Algoritmos de Aprendizagem Ativa
SSAR	Amostragem Ativa Usando Regras de Associação
QBC	Consulta por Comitê
SVM	Máquina de Vetores de Suporte

SUMÁRIO

1 INTRODUÇÃO

Aprendizado ativo consiste na seleção de instâncias que melhor representem um corpo de dados para constituir um conjunto de treinamento, para um algoritmo de aprendizado supervisionado. Durante o processo o algoritmo solicitará a um professor (oráculo) o rótulo de cada instância selecionada. Este tipo de aprendizado é usado quando há escassez de instâncias com rótulos, visando assim, atingir predição desejável com menos instâncias no conjunto de treinamento.

Um algoritmo de aprendizado supervisionado busca a predição de uma variável dependente de acordo com um conjunto de variáveis independentes, por exemplo: a predição do preço de um imóvel de uma região de acordo com os preços dos imóveis vendidos na mesma região.

Em tarefas de aprendizado supervisionado são necessários rótulos, respostas desejadas, para as instâncias presentes no conjunto de treinamento, a fim de que seja possível o treinamento do método de aprendizado. Porém, em diversas tarefas existem dados abundantes mas sem rótulos [Settles, 2010].

Partindo das questões de pesquisa definidas, tanto como o método de aprendizado ativo escolhido (SSAR), este trabalho propõe explorar configurações e ajustes para selecionar mais as instâncias menos representativas ou reduzir a seleção ativa. Em que as instâncias menos representativas são consideradas as instâncias da classe não dominante do conjunto de entrada, já a seleção ativa se trata da coleta das instâncias mais informativas do conjunto de entrada.

1.1 Questões de Pesquisa

- QP1- A alteração da discretização das características altera o resultado do método de seleção ativa?
- QP2- A aplicação de pesos como penalização, no total de regras da projeção, tem efeito sobre o resultado da seleção ativa?
- QP3- A aplicação de pesos variáveis de acordo com a razão de instâncias de cada classe sobre o treinamento, tem efeito na seleção ativa?

1.2 Contextualização

Cada vez mais informações estão sendo produzidas por meio de diversos dispositivos conectados devido à disponibilidade de conexão, capacidade de armazenamento e processamento que está aumentando ao longo do tempo. Dessa forma, foi impulsionado o uso e estudo de algoritmos de aprendizado de máquina para análise e predição de informações, para uso na academia e indústria.

Um caso de aprendizado de máquina se trata do aprendizado supervisionado, onde é fornecido a um algoritmo de aprendizagem as entradas, que neste trabalho é referido como instâncias, e o suas saídas respectivas. Com o objetivo de criar uma generalização das entradas em relação as saídas desejadas, assim desta forma predizer qual seria a saída de acordo com uma nova entrada.

Tarefas de aprendizado supervisionado geralmente necessitam de grande quantidade de dados rotulados para o treinamento do algoritmo, para alcançar um grau de predição aceitável. No entanto, diversos dados disponíveis para a maioria das tarefas de aprendizado não possuem um rótulo. Como resultado motivou o estudo e desenvolvimento do aprendizado ativo. Partindo da observação de que se um algoritmo de aprendizado puder consultar um professor para rotular instâncias que o mesmo considere informativas e a partir das mesmas criar o conjunto de treinamento, é possível atingir generalização igual o melhor com menos instâncias no conjunto de treinamento.

Existem vários cenários de aplicações de aprendizado ativo, tanto quanto para algoritmos para seleção das instâncias mais informativas, como pode ser visto em [Settles, 2010]. Entretanto podem existir classe(s) que estejam menos representadas em um conjunto de treinamento. Como o exemplo apresentado na Tabela 1.1.

Tabela 1.1: Base de dados de Notícias

Quantidade de Instâncias	Classe
1500	Futebol
20	Negócios
1200	Religião
1000	Moda
950	Ciência

A classificação de documentos é considerada uma tarefa de aprendizado supervisionado. Na Tabela 1.1, tem-se a classe de *Negócios* com uma quantidade de instâncias menor em rela-

ção as demais classes. A falta de representatividade da classe citada na base, torna difícil a seleção de uma amostragem representativa do mesmo, para a composição de um conjunto de treinamento, deste modo influenciando no modelo gerado pelo algoritmo de aprendizado.

Outra questão importante é que a classe de *Negócios* pode ter relevância maior que as outras classes, como por exemplo um feed de notícias em que o modelo de aprendizado é responsável por recomendar as notícias preferidas de um usuário que podem pertencer a classe menos representada na base de dados.

Desta forma, este trabalho mostra a possibilidade de alteração de um método do aprendizado ativo para seleção de mais instâncias menos representadas.

1.3 Objetivos

1.3.1 Objetivo Geral

Desenvolver ajustes em método de aprendizado ativo baseado em regras para encontrar mais instâncias de subamostras.

1.3.2 Objetivos Específicos

- Avaliar mudanças que podem ser feitas para o método escolhido;
- Desenvolver experimentos com diferentes configurações do método selecionado, aplicando as mudanças propostas;
- Avaliar com métricas de desempenho o método, com as mudanças propostas dentro do objetivo desejado;
- Comparar resultados das diferentes mudanças propostas para o método.

1.4 Justificativa

As tarefas de aprendizado supervisionado geralmente demandam de grande de instâncias rotuladas para o seu treinamento, porém algumas dessas tarefas não existe a quantidade necessária de instâncias rotuladas, e obter tais rótulos é custoso financeiramente ou pelo tempo consumido pelo processo de rotulagem. Segundo [Settles, 2010], os sistemas de aprendizado ativo visam minimizar esse problema, perguntando a um professor o rótulo de instâncias con-

sideradas informativas, desse jeito é reduzido o custo de obter instâncias rotuladas, visando atingir alta performance no treinamento com um conjunto menor de instâncias.

Em alguns métodos de aprendizado ativo, são usadas abordagens de seleção por incerteza do rótulo da instância, de modo a completar a informatividade do conjunto de treinamento [Silva, 2012] e [Cohn and Schohn, 2000]. A importância do trabalho está na seleção de instâncias que são consideradas informativas, mas são raras no conjunto de dados, pois estas podem ter maior importância em uma tarefa de aprendizado.

1.5 Estrutura do Trabalho

O trabalho está disposto da forma seguinte. O Capítulo 2 apresenta uma introdução ao conceito de aprendizado de máquina e seus principais cenários. Também conceitua o aprendizado ativo e configurações em que é possível seu uso. Adicionalmente é descrito o conceito de seleção por regras. O Capítulo 3 apresenta os trabalhos relacionados, descrevendo os métodos e a intuição por trás dos mesmos. O Capítulo 4 detalha a proposta desenvolvida. O Capítulo 5 apresenta os experimentos realizados e resultados alcançados. Finalizando, o Capítulo que traz a conclusão.

2 FUNDAMENTAÇÃO TEÓRICA

Neste capítulo é apresentado a fundamentação para o entendimento do que se propõe neste trabalho. Uma introdução ao aprendizado de máquina, seus tipos de aprendizado e diferenças entres os mesmos, assim como o conceito de regras de associação. Destacando o aprendizado ativo, os cenários presentes no mesmo e sua diferença dos demais.

2.1 Aprendizagem de Máquina

De acordo com [Mohri et al., 2018], aprendizado de máquina é definido por métodos computacionais utilizando experiência para melhorar sua performance ou para fazer predições mais exatas. A experiência se trata da informação disponível para o algoritmo de aprendizado, em que a quantidade, também como a qualidade são fundamentais para o grau de precisão da predição do algoritmo em relação a uma tarefa.

Uma tarefa de aprendizado de máquina pode ser a classificação de um e-mail como spam, ou a classificação de objetos em uma imagem. Entretanto, em uma tarefa não é possível ter total certeza da predição. Mas podem existir padrões nos dados, assim sendo possível chegar em uma aproximação, entendendo os padrões e regularidades. Dessa forma é possível fazer previsões, assumindo que o futuro próximo não terá muita diferença das informações passadas usadas para o aprendizado [Alpaydin, 2010].

Alguns tipos de aprendizado de máquina, de acordo com [Mohri et al., 2018]:

- **Supervisionado**: quando no conjunto de treinamento tem-se todas as instâncias rotuladas, ou seja, com uma resposta desejada referente a cada instância. Este tipo de aprendizado tem como objetivo gerar um modelo para fazer predições sobre instâncias ainda não vistas.

- **Não-Supervisionado**: neste caso tem-se um conjunto de instâncias de treinamento não rotulado, em que o algoritmo de aprendizagem busca descobrir padrões e similaridades entre as instâncias. Desta forma produzindo um modelo para predição de instâncias não vistas, porém, neste tipo de aprendizado pode ser difícil avaliar a performance do modelo pelo fato de não ter dados rotulados. Os exemplos de tarefas são os agrupamentos (agrupamento de instâncias de acordo com similaridade) e detecção de anomalias (quando algumas instâncias são incomuns no conjunto de dados).

- **Aprendizado por reforço**: nesse tipo de aprendizagem, o algoritmo interage com o ambiente, recebendo recompensas ou punições de acordo com cada ação feita ao longo do tempo. O principal objetivo é maximizar a recompensa obtida ao longo de todas as ações com o ambiente. Em algum ponto será necessário decidir se continua a obter mais possíveis desconhecidas recompensas com novas ações, ou se será explorado ações conhecidas que levam a recompensas esperadas.

2.2 Aprendizagem Ativa

Segundo [Settles, 2010], o aprendizado ativo é um subcampo de aprendizagem de máquina. Tipo de aprendizagem utilizada quando há poucas instâncias com rótulos e dificuldade de se obter rótulos para as que não possuem, pois para determinadas tarefas pode ser muito custoso rotular todas as instâncias. Deste modo o algoritmo de aprendizagem ativa seleciona as instâncias que julga informativa para serem rotuladas por um professor e inseridas no conjunto de treinamento. Assim sendo possível que o aprendizado ocorra com um número menor de instâncias, buscando atingir acurácia melhor ou semelhante.

Este tipo de aprendizado também pode ser considerado um tipo de aprendizado semi-supervisionado, por utilizar dados rotulados e não rotulados. A ideia central da área é que se forem usadas somente instâncias consideradas informativas, escolhidas pelo algoritmo de aprendizado para rotulagem por um professor, pode-se atingir acurácia melhor ou semelhante perante o uso de todo o conjunto de dados para treinamento.

Os principais cenários de aprendizado ativo, de acordo com [Settles, 2010]:

- **Stream-based Selective Sampling**: há um fluxo contínuo de instâncias (sem rótulo) de entrada. Conforme o algoritmo de aprendizado decide através de um critério se vai consultar o rótulo da instância, para adicionar no conjunto de treinamento, ou se vai descartá-la.

- **Pool-based Sampling**: há um conjunto de instâncias não rotuladas em que o algoritmo de aprendizado decide através de um critério de informatividade aplicado sobre o conjunto inteiro, qual instância deve ser selecionada para rotulagem e adição no conjunto de treinamento.

O estudo do aprendizado ativo surge para lidar com o problema da escassez de rótulos para determinadas tarefas de aprendizado complexas, criando e otimizando métodos para

melhor selecionar instâncias para compor um conjunto de treinamento. Enquanto, o algoritmo de aprendizagem envia a solicitação de rótulo a um professor, rótulo das instâncias ainda não rotuladas, consideradas mais informativas de acordo com algum critério, para assim, atingir a melhor generalização possível com o mínimo de instâncias rotuladas.

A principal diferença entre o aprendizado ativo e o aprendizado passivo é a seleção ativa das instâncias para o conjunto de treinamento. Já que no modo tradicional, passivo, o objetivo é apenas reunir a maior quantidade possível de instâncias rotuladas, não considerando nenhuma medida de informatividade destas.

2.3 Regras de Associação

De acordo com [Alpaydin, 2010], regras de associação é método para descobrir associações entre itens de uma base de dados. Possibilita também mostrar o quão frequente o item ocorre. Uma regra de associação é uma implicação da forma $X \rightarrow Y$ onde X é o antecedente, e Y é o consequente. Tanto antecedente quanto consequente são um conjunto de itens.

O método é bastante usado para análise de compras. Dados t_i como identificador de uma transação, e o valor 1 corresponde a compra do item, 0 a não compra do item. A Tabela 2.1 apresenta algumas transações com itens: Leite, Cerveja, Massa e Pão, como exemplo.

Existem três principais medidas usadas para medir a qualidade das regras de associação: **Suporte**, **Confiança** e o **Lift**.

Tabela 2.1: Transações - Compras de Mercadorias

ID	Leite	Cerveja	Massa	Pão
t_1	1	1	1	1
t_2	1	1	0	1
t_3	0	1	1	0
t_4	1	0	1	1
t_5	0	1	1	0
t_6	1	0	0	1

- **Suporte de** $X \rightarrow Y$: fração de transações que contém os itens de X e Y. Pode ser descrito da seguinte forma: Suporte $(X, Y) = P(X, Y) = (t_i \; com \; X \; e \; Y \;)/ \; (Total \; de \; t_i)$. **Exemplo**: com base na Tabela 2.1, cálculo do suporte do suporte de t_3 = {Cerveja, Massa}, é $3/6 = 0.5$, possuí um suporte de 50% pois está contido em metade das transações (t_1, t_3, t_5).

- **Confiança de** $X \rightarrow Y$: Suporte de X e Y, dividido pelo suporte de X. Pode ser descrito da seguinte forma: Confiança $(X \rightarrow Y) = P(Y|X) = (t_i \; com \; X \; e \; Y \;)/ \; (t_i \; com \; X)$. **Exemplo**: com base na Tabela 2.1, a confiança de {Massa $\rightarrow$ Pão} é $((2/6)/(4/6)) = 0,5$, ou seja, a confiabilidade da regra é de 50%, onde 50% das transações que contém Massa tem Pão.

- **Lift de** $X \rightarrow Y$: Suporte de X e Y, dividido pela multiplicação do suporte de X e do suporte de Y. Pode ser descrito da seguinte forma: Lift $(X \rightarrow Y) = P(X,Y) \; / \; P(X)P(Y)$. **Exemplo**: com base na Tabela 2.1, o lift de {Massa $\rightarrow$ Pão } é $(2/6)/((4/6) * (4/6)) = 0,75$. Em que valores menores que 1 do lift, indicam menor correlação entre os itens da regra, já maiores que 1 indicam maior correlação e lift igual a 1 indica nenhuma associação entre os itens da regra.

As regras de associação podem ser usadas para quantificar a informatividade de uma instância, em um método de aprendizado ativo. Um exemplo de uso é no método SSAR explicado na seção seguinte.

3 TRABALHOS RELACIONADOS

Nesta seção, são descritos brevemente alguns Algoritmos de Aprendizagem Ativa (AAA), que tem a finalidade de selecionar instâncias consideradas informativas. A aprendizagem ativa é um subcampo da aprendizagem de máquina, em que estuda-se a possibilidade de um método de aprendizado performar satisfatoriamente com uma quantidade menor de instâncias em seu treinamento. Desde que, o mesmo possa escolher quais instâncias formaram o conjunto de treinamento, sendo selecionadas de acordo com uma estratégia e rotuladas por um professor. Essa abordagem é útil para tarefas de aprendizado com escassez de instâncias rotuladas, em que é custoso em tempo ou caro obter rótulos [Settles, 2010].

3.1 Amostragem Ativa Baseada em Regras - SSAR

De acordo com [Silva et al., 2011], o método SSAR (Amostragem Seletiva Usando Regras de Associação) é uma estratégia para enfrentar o alto custo de rotulagem em grandes quantidades de dados. A característica principal do método é a seleção ativa de um número reduzido de instâncias que são informativas e não redundantes, capazes de representar a coleção sem (ou com reduzida) perda de informação. De modo simplificado, o SSAR, através da projeção, com base no conjunto rotulado, seleciona novas instâncias, que não estão rotuladas, para complementar as que estão presentes no conjunto de treinamento.

Inicialmente, o conjunto de treinamento está vazio. O método inicia a partir da seleção de uma instância não rotulada, que tenha mais atributos em comum com todas as instâncias não rotuladas, para a rotulagem e inserção no conjunto de treinamento. Na sequência é construído a projeção de cada instância do conjunto não rotulado sobre o atual treinamento, ou seja, na projeção permanecem somente os atributos que estão presentes no conjunto de treinamento.

As regras extraídas são quantizadas para representar a informatividade da instância não rotulada. Caso sejam geradas muitas regras, o documento tem caráter menos informativo pois os atributos estão redundantes em relação ao atual conjunto de treinamento. Após ter-se as regras quantizadas de cada instância, escolhe-se a instância que gerou menos regras, rotula-se a mesma, e insere-se no conjunto de treinamento. O objetivo é selecionar o menor conjunto de instâncias capaz de representar a coleção. Explorando a característica de que algumas instâncias podem compartilhar mesma informação, logo não seria necessário rotular todas. Consequentemente se consegue retratar com mais semelhança todas as instâncias presentes na coleção.

O método converge automaticamente, não necessitando de interferência, devido a informatividade medida pela quantização das regras. A finalização do método ocorre quando for selecionado uma instância que já está presente no conjunto de treinamento, não acrescentando mais informatividade ao conjunto, pois já está rotulada, [Silva, 2012].

3.1.1 Método e Exemplo

Algoritmo 1: AMOSTRAGEM SELETIVA USANDO REGRAS DE ASSOCIAÇÃO [SILVA ET AL., 2011]

Require: Conjunto de documentos não rotulados D e $\sigma_{min}(\approx 0)$
Ensure : Conjunto de treinamento T

```
1  while true do
2      forall documento d_i ϵ D do
3          T_{d_i} ⇐ T projetado de acordo com d_i
4          R_{d_i} ⇐ Regras geradas de acordo com T_{d_i} | σ ≥ σ_min
5      end
6      if T = ∅ then
7          Γ_i ⇐ d_i de tal modo que ∀ d_j : |D_di| ≥ |D_dj|
8      else
9          Γ_i ⇐ d_i de tal modo que ∀ d_j : |R_di| ≤ |R_dj|
10     end
11     if Γ_i ϵ D then
12         Break;
13     else
14         ColocarRotulo(Γ_i)
15         T ⇐ T ⋃ {Γ_i}
16     end
17 end
```

No Algoritmo 1 pode-se visualizar dois conjuntos. Um conjunto de documentos não rotulados que será representando por D, e o conjunto de treinamento, representado por T. No começo da iteração do método mostra-se a projeção de um documento d_i pertencente a D, de acordo com o conjunto de treinamento atual T (Linhas 3 e 4). Em seguida as regras de associação da projeção de d_i são geradas. Caso o conjunto de treinamento esteja vazio, na primeira execução do método, é selecionado o documento d_i mais representativo de D (Linha 7) para ser rotulado e inserido no conjunto de treinamento (Linhas 14 e 15). Se conjunto de treinamento não estiver vazio, deve ser escolhido o d_i que gerou menos regras, considerado o mais informativo (Linha 9). Deve ser rotulado e inserido no conjunto de treinamento (Linhas 14 e 15). As iterações continuam até que seja selecionado um documento que já esteja no conjunto

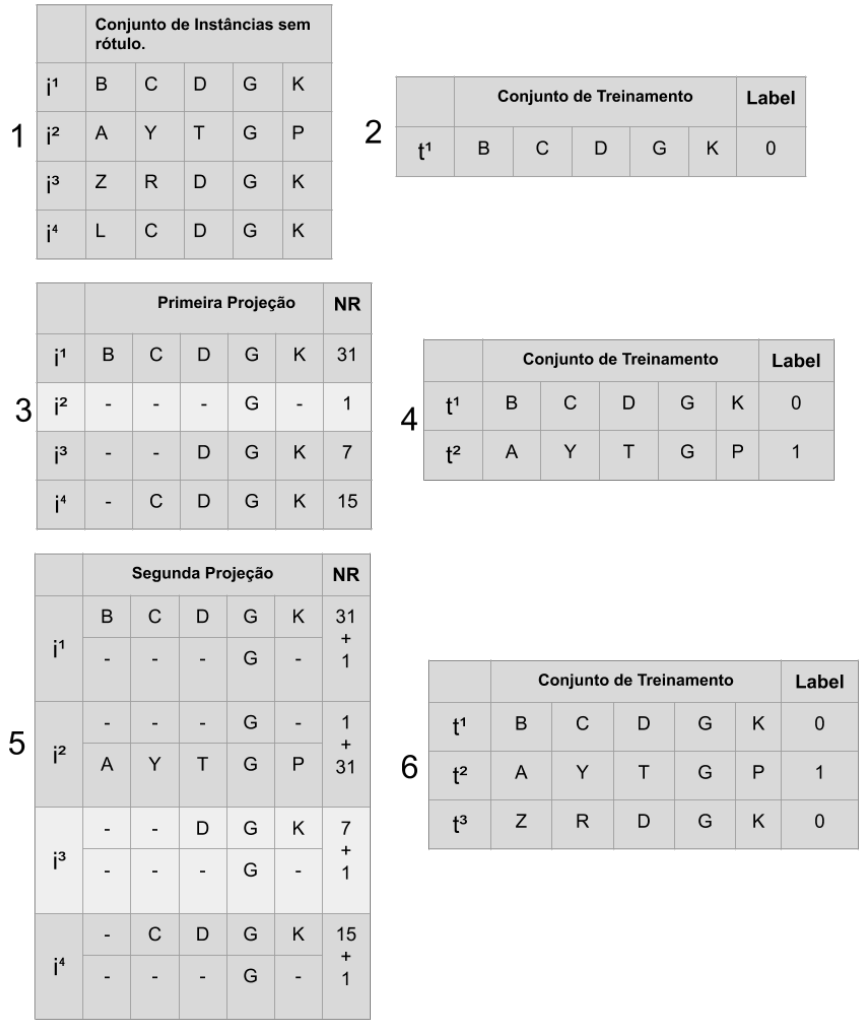

Figura 3.1: Exemplo do Método SSAR. Projeções e rotulagem feitas na sequência definida pelas etapas.

de treinamento, consequentemente o método finaliza (Linha 11).

Na Figura 3.1 é apresentado um exemplo de execução do método SSAR. Na primeira execução do método, é inserido a instância mais representativa pois o conjunto de treinamento está inicialmente vazio (Etapa 2). No exemplo, foi selecionado a instância i_1 por ter mais atributos em comum com as demais não rotuladas. Em seguida, é realizado a projeção de cada instância não rotulada sobre o conjunto de treinamento (que atualmente contém somente uma instância), mantendo os valores de atributos que estão no atual treinamento (Etapa 3). Dessa forma, é possível gerar as regras de associação de cada projeção (ilustradas na coluna *NR* - Número de Regras). Na etapa 3, a projeção da instância i_1 gerou 31 regras, já a projeção da instância i_2 gerou uma regra. Assim, a instância i_2 é considerada mais informativa e será rotulada pelo professor pois gerou menos regras. Então na etapa 4 a instância i_2 é inserida no

conjunto de treinamento com seu rótulo, *label* (rótulo da instância). Bem como nas etapas 5 e 6 é repetido o processo de projeção, onde o número de regras é computado novamente com base no conjunto de treinamento atualizado, da etapa 4. Logo é escolhido a instância que gerou menos regras para ser rotulada e inserida no conjunto de treinamento. Essa iteração ocorre até que seja escolhida uma instância para rotulagem que já esteja no conjunto de treinamento, assim o método finaliza.

É importante notar que conforme o conjunto de treinamento for recebendo mais instâncias rotuladas o número de regras geradas pelas projeções vai aumentar, resultando no acréscimo de custo computacional para o cálculo das regras. Consequentemente em grandes bases de dados se torna inviável o uso do método.

3.2 Consulta por Comitê- QBC

Segundo [Settles, 2010], o método ativo de Consulta por Comitê (*Query by Committe*) consiste na geração de um grupo de hipóteses (modelos) para selecionar as instâncias mais informativas. A seleção de instâncias é baseada no desacordo entre um conjunto de classificadores que pode ser formado por Árvore de Decisão, Naïve Bayes, Artificial Neural Networks, entre outros.

Uma deficiência relevante do método é que o mesmo pode considerar um ruído como uma instância informativa. Ruído é uma instância pertencente a uma classe que tem diferença significativa das outras instâncias da mesma classe.

O QBC necessita de um conjunto de treinamento inicial para começar sua execução. Os classificadores são treinados com este conjunto inicial. Uma nova instância é escolhida para ser rotulada baseada no desacordo em relação ao rótulo dado a mesma pelos classificadores. Esta instância, após rotulada, é adicionada no conjunto de treinamento e os classificadores são novamente treinados sobre o conjunto de treinamento para a escolha de novas instâncias para a rotulagem. O processo se repetirá até que não se tenha discordância entre os classificadores em relação ao rótulo de uma instância.

A Figura 3.2 apresenta um exemplo de seleção no método QBC, no qual se tem um conjunto de instâncias sem rótulo e um conjunto de treinamento inicial. Em uma primeira rodada, os membros do comitê são treinados no conjunto de treinamento inicial, que apresenta somente uma instância (Etapa 1). É possível observar que a maioria dos classificadores concordam com o rótulo da instância i_1 e i_2 (Etapa 2). Mas quanto a instância i_3, existe uma discordância em

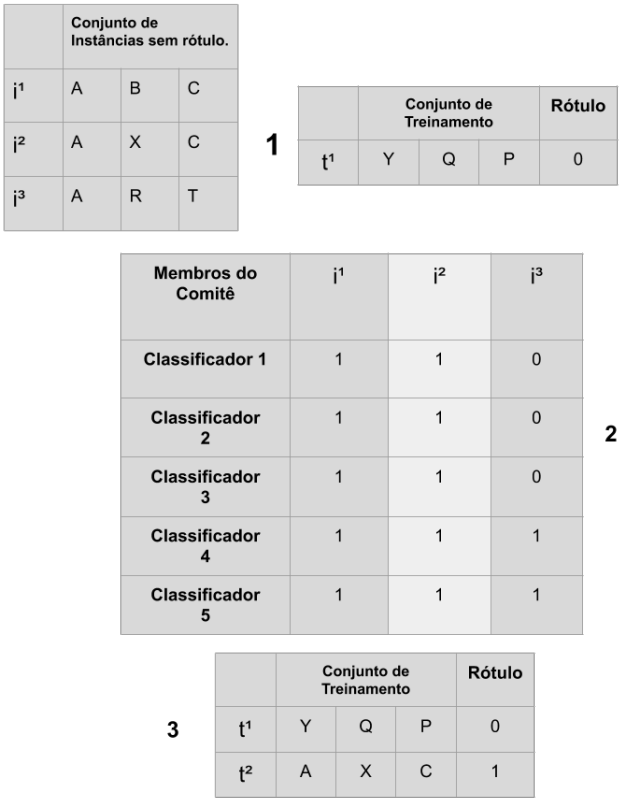

Figura 3.2: Exemplo do QBC dividido em três etapas. Na etapa 1 ocorre o treinamento, na 2 acontece a votação, e na etapa 3 se tem a rotulagem.

relação ao rótulo. Logo, a mesma é escolhida para ser rotulada e inserida no conjunto de treinamento (Etapa 3). O processo se repete até que seja atingido o número desejado de instâncias rotuladas no treinamento. Em suma, para melhores resultados é necessário levar em conta a quantidade inicial de instâncias para o treinamento, tanto como a possibilidade de seleção de instâncias consideradas ruídos que não são realmente informativas.

3.3 Amostragem por Incerteza

Abordagem inicialmente proposta por [Lewis and Gale, 1994], cujo o objetivo é selecionar a instância do qual se têm mais incerteza em relação ao seu rótulo. No trabalho foi testado um modelo probabilístico de classificação binário, que seleciona como instância mais informativa aquela que tem probabilidade posterior positiva, mais próxima de 0.5. O método pode ser usado em qualquer tipo de classificador que prediz um rótulo e uma medida de quão certa é a predição, como por exemplo os classificadores probabilísticos.

Para problemas onde existem três ou mais classes foram propostos critérios para seleção de instâncias informativas, usando a abordagem de seleção por incerteza. Conforme a

distribuição de probabilidades dos rótulos em relação as instâncias, dados por um modelo θ.

Least Confidence [Culotta and McCallum, 2005]: sendo $\hat{y}$ o rótulo mais provável para x de acordo com o modelo θ. Nessa estratégia será selecionado a instância em que se tiver menos confiança em relação ao seu rótulo mais provável.

$$x^*_{LC} = argmax_x 1 - P_\theta(\hat{y}|x)$$

Smallest Margin [Scheffer et al., 2001]: sendo $y1$ e $y2$ os rótulos mais prováveis para x de acordo com modelo θ. Nessa estratégia é selecionado a instância em que tiver a menor diferença entre o primeiro e segundo rótulo mais prováveis.

$$x^*_{SM} = argmin_x P_\theta(y1|x) - P_\theta(y2|x)$$

Entropy [Dagan and Engelson, 1995]: é uma medida de incerteza de uma variável aleatória. Com objetivo de utilizar as probabilidades de todos os rótulos. Então é calculado sobre cada instância, e a que tiver maior valor é selecionada.

$$x^*_{LE} = argmax_x - \sum_i P_\theta(yi|x) log P_\theta(yi|x)$$

A instância y_i alcança todos os rótulos possíveis. Selecionar a instância em que a entropia do rótulo é a maior.

Tabela 3.1: Distribuição de Probabilidades de Classificador para as instâncias i_1 e i_2, com rótulos A, B, C

Instância	A	B	C
i1	0,356	0,554	0,49
i2	0,9	0,213	0,135

Considerando o critério de seleção por **Least Confidence**, a instância selecionada seria a instância $i1$ pois apresenta menor confiança em relação ao seu rótulo mais provável, B com probabilidade de 0.554, com base na Tabela 3.1.

Agora, considerando **Smallest Margin**. Em $i1$ visualiza-se uma diferença de $0.554 - 0.49 = 0,064$, já em $i2$ visualiza-se uma diferença de $0.9 - 0.213 = 0,687$, logo será selecionado a instância $i1$. De acordo com a Tabela 3.1.

Para considerar a probabilidade de todos os rótulos na seleção, com base na Tabela 3.1. É necessário usar **Entropy**, que aplicando sobre $i1$, produz $-1 * (0.356 * log(0.356) + 0.554 * log(0.554) + 0.49 * log(0.49)) = 0,453583312$. Já $i2$, gera $-1 * (0.9 * log(0.9) + 0.213 *$

$log(0.213) + 0.135 * log(0.135)) = 0,301641827$. Como $i1$ teve um valor maior no cálculo da entropia, essa instância é considerada para seleção.

3.4 Máquinas de Vetores de Suporte - SVM

SVM em geral são considerados classificadores de bom desempenho com o objetivo de encontrar a melhor separação entre classes. Porém, mesmo que não seja possível separar as mesmas, usando nesse caso ajustes. Dependendo do conjunto de dados, parâmetros escolhidos, o treinamento pode levar um tempo considerável. O classificador tem como objetivo obter um hiperplano (entre os vários que podem ser construídos) capaz de separar as classes do conjunto, maximizando as margens entre as instâncias que são extremos de cada classe e o hiperplano [Mohri et al., 2018].

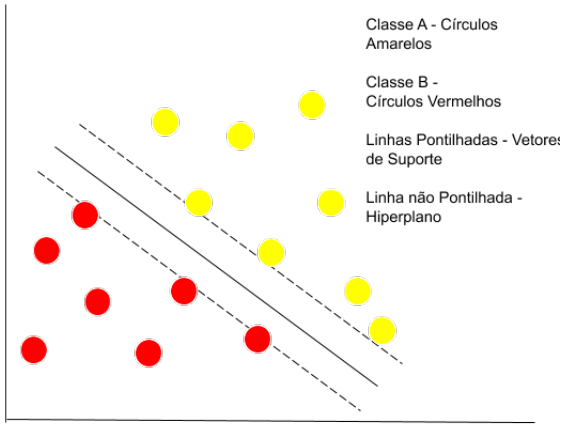

Figura 3.3: Exemplo - Hiperplano de Separação Ótima - SVM. Duas linhas pontilhadas representando os vetores de suporte e uma linha não pontilhada representando o hiperplano.

A Figura 3.3 apresenta um exemplo de hiperplano ótimo, que separa as classes A e B. Vetores de suporte são os vetores que delimitam as instâncias no extremo de cada classe. Um grande número de hiperplanos pode ser obtido em diferentes ângulos e posições. Enquanto mantenha a margem em relação aos vetores de suporte. Caso não seja possível obter um hiperplano de separação ótima, é utilizado um hiperplano que separe o melhor possível as instâncias das classes, penalizando as instâncias que violarem as margens e o hiperplano. Através de um limite de erro em relação a soma de todas as penalizações.

Em [Cohn and Schohn, 2000], é proposto uma heurística, como critério de seleção de instâncias no aprendizado ativo com SVM. Está baseada no impacto esperado da seleção da instância na mudança do hiperplano (linha divisória) entre as classes. Assim considerando a ins-

tância mais informativa, aquela que gerar uma mudança maior no hiperplano caso seja rotulada. Uma vantagem dessa abordagem é a menor propensão da seleção de instâncias consideradas ruídos, devido ao método considerar o corpo inteiro de dados.

No trabalho de [Cohn and Schohn, 2000], os resultados empíricos mostraram desempenho superior em vários domínios em comparação com o uso da seleção randômica de instâncias. Observando que além do ganho referente a melhor generalização das instâncias, há também o ganho em relação ao tempo de treinamento das SVM, já que o mesmo está relacionado ao tamanho do conjunto de treinamento.

Ainda em [Cohn and Schohn, 2000] é usado a forma de aprendizado ativo, chamada amostragem seletiva (selective sampling). Onde são selecionadas instâncias não rotuladas consideradas mais informativas de um conjunto para compor um subconjunto, que almeja representar a informação do conjunto em sua totalidade.

A heurística pode ser calculada de forma simples, e não faz uso de informação do conjunto de dados. Para escolha da instância mais informativa não rotulada, é utilizado o critério da distância da instância em relação ao hiperplano. Pode ser obtido com o produto escalar, que se caso for 0 resulta em um ângulo de $90°$. A determinação da escolha da instância mais informativa é definida como aquela que esteja mais próxima do hiperplano. Pois resulta do fato de que as instâncias mais próximas do hiperplano determinam o ângulo e posição do mesmo.

Como critério de parada, [Cohn and Schohn, 2000] partem da ideia de que se o conjunto de instâncias for linearmente separável, somente as instâncias dentro da margem terão efeito sobre o aprendizado do classificador. Após ser definido uma classe a uma instância dentro da margem, é possível que ocorra uma mudança no hiperplano e vetores de suporte, fazendo assim que instâncias que estavam fora da margem, estejam dentro da mesma.

Através dessa observação pode-se concluir que após rotular todas as instâncias que estiverem dentro da margem, nenhuma rotulagem de instância irá afetar o SVM, no mínimo. Logo, é razoável assumir que quando não se tiver mais instâncias não rotuladas dentro da margem, é possível parar de selecionar novas instâncias para rotulagem. Pode-se computar o critério de parada da seguinte forma, de todas as instâncias, se a que estiver mais próxima do hiperplano não estar a uma distância menor do que qualquer vetor de suporte em relação ao hiperplano, não existe instância sem rótulo dentro da margem. Então finalizando o processo de rotulagem.

4 PROPOSTA

Conforme apresentado, este trabalho tem como objetivo selecionar instâncias em cenários de desbalanceamento, através de um método de seleção ativa baseado em regras. Para isso, é proposto uma abordagem que explora ajustes na quantização das regras de associação, afim de avaliar o comportamento do método de Amostragem Ativa Baseada em Regras (SSAR). O SSAR foi escolhido pois se trata de um trabalho recente, que está entre os métodos no estado-da-arte de aprendizagem ativa. As seções a seguir descrevem os detalhes da proposta.

4.1 Método Proposto

Está seção tem como objetivo descrever as mudanças propostas para o método de seleção ativa baseado em Regras, SSAR (explicado na Seção 3.1). As alterações aqui desenvolvidas, são focadas na parte da quantização das regras, que será explicado a seguir em um exemplo de execução.

Conjunto de instâncias sem rótulo					**Conjunto de Treinamento**				**Label**
i^1	A	B	C		**t^1**	A	B	C	0
i^2	A	B	R						
i^3	A	X	Z						

Figura 4.1: Exemplo Penalização - Conjunto Inicial

Na Figura 4.1, é visto os dois conjuntos necessários para ser possível iniciar o método. O conjunto de instâncias não rotuladas (esquerda) e o conjunto de treinamento (direita). No conjunto de treinamento, inicialmente está armazenado uma instância considerada a mais representativa do conjunto não rotulado, pois a mesma tem mais atributos em comum com as demais instâncias, estando a mesma rotulada.

Já na Figura 4.2, ocorre a primeira projeção do conjunto de instâncias sem rótulo sobre o conjunto de treinamento. Na projeção de i^1 são geradas 7 regras, em i^2 são geradas 3 regras e em i^3 é gerado 1 regra. Para calcular a quantidade de regras deve-se fazer a seguinte operação, exemplo considerando i^1, $2^3 - 1 = 7$, onde 3 é o total de atributos da instância.

Estas são associadas ao rótulo da instância do conjunto de treinamento sobre o qual foram projetadas. A projeção Pi^3 gerou menos regras, então a instância i^3 será rotulada de

acordo com um professor e inserida no conjunto de treinamento como t^2.

Primeira Projeção														
Pi^1	A	B	C	0		**Pi^2**	A	B	-	0		**Pi^3**	A	-	-	0
7 Regras						**3 Regras**						**1 Regra**				

Figura 4.2: Exemplo Penalização - Primeira Projeção do Conjunto não rotulado.

Na segunda projeção é feito novamente a projeção do conjunto de instâncias não rotuladas sobre o conjunto de treinamento, conforme descrito na Figura 4.3.

Segunda Projeção															
Pi^1	A	B	C	0		**Pi^2**	A	B	-	0		**Pi^3**	A	-	-	0
	A	-	-	1			A	-	-	1			A	X	Z	1
(7+1) Regras						**(3+1) Regras**						**(1+7) Regras**				

Figura 4.3: Exemplo Penalização - Segunda Projeção do Conjunto não rotulado.

Na segunda projeção, agora com duas instâncias no conjunto de treinamento, a projeção Pi^2 gera um número menor de regras. Então i^2 é rotulada de acordo com um professor e inserida no conjunto de treinamento. Em uma terceira projeção, será selecionada uma instância do conjunto não rotulado que já está presente no conjunto de treinamento, fazendo assim o método parar sua execução. O conjunto final de treinamento pode ser visualizado na Figura 4.4.

Conjunto de Treinamento				Label
t^1	A	B	C	0
t^2	A	X	Z	1
t^3	A	B	R	0

Figura 4.4: Exemplo Penalização - Conjunto Final de Treino

É observável no exemplo, que no método as regras geradas da projeção de uma instância i^n (Conjunto não rotulado) sobre uma instância t^n(Conjunto de treinamento) podem ser associadas com o rótulo da instância t^n. Logo, é possível perceber que na quantização das regras caso verificar-se instâncias de diferentes classes no conjunto de treinamento, é possível penalizar as instâncias que possuírem uma maior quantidade de regras com a classe negativa na quantização. São propostas duas estratégias: Penalização Fixa e Penalização Variável.

4.1.1 Penalização Fixa

No Algoritmo 2 é demonstrado a função desenvolvida para contabilizar a quantidade de regras para cada rótulo na projeção de uma instância. Levando em consideração que o método seja usado com duas classes (positiva e negativa). Como foi visto, a seleção da instância que será rotulada, é sempre a que gerar menos regras na projeção.

Algoritmo 2: Quantificar número de labels das regras geradas da projeção de *i*

```
   Input: regras[]
 1 qdtPositivo = 0;
 2 qtdNegativo = 0;
 3 u = 0;
 4 while regras[u] do
 5 |  if regras[u][label] == 1 then
 6 |  |  qdtPositivo += 1;
 7 |  else
 8 |  |  qtdNegativo += 1;
 9 |  end
10 |  u++;
11 end
```

A função desenvolvida no Algoritmo 2 tem como entrada uma variável nomeada como $regras$, que armazena todas as regras geradas da projeção de uma instância com seus detalhes como suporte mínimo, rótulo, etc. No início do algoritmo (Linhas 1, 2 e 3) são atributos valores de inicialização as variáveis necessárias, para contar as regras positivas e negativas.

No Laço (Linha 4 - 11) ocorre a iteração sobre as regras, já a condição (Linha 5) verifica se a regra tem rótulo positivo, caso sim, incrementa o contador de regras positivas (Linha 6). Caso não, incrementa o contador de regras negativas (Linha 8). A contabilização ocorre quando acabar a quantidade de regras da variável de entrada, iteradas por u.

$$qtdRegras = qtdRNegativas * PESO + qtdRPositivas \tag{4.1}$$

A estratégia de penalização fixa, se baseia em aumentar o número total de regras da projeção de uma instância. Sendo implementada a partir da multiplicação do número de regras do rótulo negativo de uma projeção por um *PESO*, demonstrado na Equação 4.1. Dessa forma, as instâncias que tiverem mais regras de rótulo negativo tendem a ter um total de regras maior por causa da penalização, forçando assim que assim não sejam selecionadas.

4.1.2 Penalização Variável

No trabalho de [Yu et al., 2019] é proposto uma forma de penalização, que usa a proporção da classe com mais instâncias ou da classe com menos instâncias, sobre o atual conjunto de instâncias rotuladas. Assim, a variação dos pesos da penalização contempla o desequilíbrio entre a proporção das classes, mesmo enquanto forem adicionadas novas instâncias rotuladas.

$$wi = \begin{cases} \frac{|N^+|}{|N^+|+|N^-|}, & \text{Se } xi \text{ pertence a classe com mais instâncias} \\ \frac{|N^-|}{|N^+|+|N^-|}, & \text{Se } xi \text{ pertence a classe com menos instâncias} \end{cases} \tag{4.2}$$

Na Equação 4.2, $|N^+|$ e $|N^-|$ representam o número de instâncias positivas (classe minoritária) e negativas(classe majoritária) com rótulo presentes no conjunto de treinamento. Já xi é uma instância que ainda não foi rotulada.

Neste trabalho, será testado a penalização da Equação 4.2, como também uma alteração proposta para a penalização sobre a quantização das regras geradas na projeção de uma instância. A alteração proposta consiste em inverter as condições de penalização, dessa forma com maior possibilidade de penalizar a classe dominante no SSAR, resultando na Equação 4.3.

$$wi = \begin{cases} \frac{|N^+|}{|N^+|+|N^-|}, & \text{Se } xi \text{ pertence a classe com menos instâncias} \\ \frac{|N^-|}{|N^+|+|N^-|}, & \text{Se } xi \text{ pertence a classe com mais instâncias} \end{cases} \tag{4.3}$$

Assim, na quantização das regras da projeção de uma instância, a quantidade de regras geradas com rótulo positivo será penalizada pela proporção de instâncias com rótulo positivo do conjunto rotulado. Já a quantidade de regras geradas com rótulos negativos será penalizada com a proporção de instâncias de rótulo negativo do conjunto rotulado. Consequentemente é esperado que com essa penalização dinâmica, se mantenha certa proporção de instâncias selecionadas dentre positivas e negativas.

5 EXPERIMENTOS E RESULTADOS

Este capítulo define as condições, especificações dos experimentos realizados e resultados. Na seção 5.1 são detalhadas informações sobre as bases de dados utilizadas para realizar os experimentos. E nas seções 5.2 e 5.4 são apresentados os métodos de avaliação usados e respectivamente os resultados da execução dos experimentos, direcionados pelas questões de pesquisa definidas na introdução.

5.1 Base de Dados

Neste trabalho, foram utilizadas duas bases de dados: IMDBxNetflix e DBLPxCiteseer. A base IMDBxNetflix foi criada consultando o serviço de interface público de aplicação (API) do Netflix e IMDB, que representam acervos sobre filmes [Dal Bianco et al., 2013]. Foram integrados os pares através dos atributos em comum como título, diretor e ano de lançamento.

Já a base DBLPxCiteseer foi criada através da junção dos conjuntos de dados do Citeseer e DBLP, que são ambos repositórios com foco em ciência da computação. DBLPxCiteseer foi produzida usando os atributos título, autor, e ano de publicação. As bases IMDBxNetflix e DBLPxCiteseer, são compostas por 3.009 e 3.037 pares correspondentes respectivamente, a primeira com 2.011 rótulos negativos e 998 positivos, a segunda com 1.803 rótulos negativos e 1.234 rótulos positivos. Foram criadas a partir da rotulagem dos pares feitas por cinco estudantes de ciência da computação [Dal Bianco et al., 2013].

```
1  @relation whatever
2  @attribute  0 numeric
3  @attribute  1 numeric
4  @attribute  2 numeric
5  @attribute classe {0,1}
6  @data
7  0.22222222, 0.4117647, 0.0, 0
8  0.022727273, 1.0, 0, 0
9  0.26923078, 0.2857143, 0.33333334, 0
```

Figura 5.1: Exemplo de formato do Arff.

A base de dados foi pré-processada utilizando o formato de dados *arff*, esse formato é necessário para ser processado pela implementação do método do SSAR. Na Figura 5.1 é visto um exemplo de como um arquivo *arff* é composto. Bases de dados no formato *arff* são compostas por duas seções: uma seção de cabeçalho; e uma seção de dados. No cabeçalho é definido o nome da base, especificado por @*relation*, e também os atributos, especificados por

@attribute. Já na seção de dados, definido com *@data*, marca o início das linhas com os dados separados por vírgula.

Para que as regras de associação sejam aplicadas, é necessário que os dados sejam discretizados em categorias. O método SSAR utiliza o algoritmo TUBE [Schimidberger and Frank, 2009] para realizar a discretização dos dados. O TUBE usa uma árvore de decisão (binária) para separar valores de um intervalo em sub-intervalos de tamanho variado. Dessa forma, a base de dados deve ser discretizada para intervalos, por exemplo, uma base de dados composta de registros com os valores entre 1 a 10, como na Figura 5.2.

$$\begin{bmatrix} 1 & 1 & 2 & 7 & 8 & 9 & 9 \\ 3 & 4 & 5 & 7 & 8 & 8 & 10 \\ & & & \cdots & & & \end{bmatrix}$$

Figura 5.2: Base não discretizada.

Os valores foram discretizados na seguinte configuração com 5 intervalos, valores entre 1 e 2 pertencem ao primeiro intervalo, valores entre 3 e 4 pertencem ao segundo intervalo, valores entre 5 e 6 pertencem ao terceiro intervalo, valores entre 7 e 8 pertencem ao quarto intervalo e valores entre 9 e 10 pertencem ao quinto intervalo. O resultado pode ser visto na Figura 5.3.

$$[1,2] = 1, [3,4] = 2, [5,6] = 3, [7,8] = 4, [9,10] = 5$$

$$\begin{bmatrix} 1 & 1 & 1 & 4 & 4 & 5 & 5 \\ 2 & 2 & 3 & 4 & 4 & 4 & 5 \\ & & & \cdots & & & \end{bmatrix}$$

Figura 5.3: Base discretizada.

5.2 Métricas de Avaliação

No trabalho é determinado *Precisão*, *Revocação* e *F1*, como medidas de avaliação [Manning et al., 2008], conforme definido a seguir:

- **Precisão(Prec)**: é a porcentagem de instâncias classificadas como positivas que realmente são positivas. Formalizada na Equação 5.1, em que VP são as instâncias classifi-

cadas como Positivas que realmente são Positivas e FP as instâncias classificadas como Positivas que são Negativas:

$$Prec = \frac{VP}{VP + FP} \tag{5.1}$$

- **Revocação(Rev)**: é a porcentagem de instâncias positivas que foram classificadas como positivas. Formalizada na Equação 5.2, em que FN representa as instâncias classificadas como Negativas que são Positivas.

$$Rev = \frac{VP}{VP + FN} \tag{5.2}$$

- **F1**: medida de avaliação que equilibra precisão e revocação. Formalizada na Equação 5.3.

$$F1 = \frac{2 \times (Prec \times Rev)}{Prec \times Rev} \tag{5.3}$$

5.3 Configurações dos Experimentos

As linguagens usadas para o desenvolvimento do trabalho foram as linguagens de programação C e Shell Script, no que se refere a seleção das instâncias. Já para a avaliação dos conjuntos selecionados de instâncias foi utilizada a linguagem de programação Python.

A avaliação foi feita com algoritmos de classificação presentes na biblioteca **scikit-learn**[1], que é amplamente utilizada para o aprendizado de máquina. Adicionalmente, a leitura dos dados foi feita com pacote específico para o formato *arff* presente na biblioteca **SciPy**[2] e para o processamento dos dados foi utilizada a biblioteca **Pandas**[3].

Em todos os experimentos a execução do método de seleção ativa foi repetido 10 vezes, para se obter um valor médio de instâncias positivas e negativas selecionadas. Também foi calculado o desvio padrão da seleção de instâncias positivas e negativas, que foi nulo em todas as execuções.

Os experimentos de avaliação com os classificadores foram executados 10 vezes, para se obter um valor médio da Precisão (*Prec*), da Revocação (*Rev*) e *F1*. Além disso para o F1 foi

[1] www.scikit-learn.org/stable/
[2] www.scipy.org/
[3] www.pandas.pydata.org/

calculado o desvio padrão dentro das 10 execuções, que foi nulo em todas as execuções.

Nos experimentos de avaliação foram utilizados dois algoritmos de classificação. O *SVM* com os hiper parâmetros definidos da seguinte forma: $kernel = rbf$, porém o C (Custo) e g (gamma) foram definidos de acordo com o GridSearchCV(função do scikit-learn) para cada conjunto de instâncias positivas e negativas avaliada. Os hiper parâmetros de entrada do GridSearchCV foram $kernel = rbf$, $cv = 5$, intervalo de procura do custo $[0.001, 0.01, 0.1, 1, 10]$, intervalo de procura do gamma $[0.001, 0.01, 0.1, 1]$. Já para o algortimo de classificação Random Forest, foram utilizados os parâmetros $n\_estimators = 100$ e $random\_state = 0$.

5.4 Execução dos Experimentos

A seguir são apresentados os experimentos realizados, em todos existem duas classes, positiva e negativa, sendo sempre a classe negativa dominante. Em geral, os experimentos são compostos por duas etapas: na primeira é visto a seleção de instâncias positivas e negativas de acordo com a abordagem adotada; já na segunda etapa, é avaliado o desempenho usando os algoritmos de classificação *SVM* e *RF* usando como treino o conjunto de instâncias selecionadas e como teste a base completa. Os experimentos foram conduzidos para avaliar o efeito das abordagens propostas sobre a seleção ativa do método SSAR. Buscando maior seleção de instâncias positivas (classe com menor proporção no treino) ou a redução do treino.

5.4.1 Avaliação dos Intervalos

Neste experimento, o objetivo é avaliar como a discretização impacta na seleção das instâncias positivas e negativas. As Tabelas 5.1 e 5.2 mostram a quantidade de instâncias positivas e negativas selecionadas de acordo com cada intervalo. Por exemplo com 10,20,30 ,40 ou 100 intervalos para discretização dos dados. As colunas *Instâncias Positivas* e *Instâncias Negativas* reportam o número de selecionadas de cada classe. Já os intervalos testados estão dispostos na coluna *Intervalos*.

Nessas Tabelas, 5.1 e 5.2 , é visto que a seleção de instâncias em ambas as bases *IMDBxNetflix* e *DBLPxCiteseer*, dentro do conjunto de intervalos definidos, mostra um padrão de aumento na quantidade total de instâncias selecionadas. Por exemplo, considerando a Tabela 5.1 com a quantidade de intervalos definida como 10 resulta em um total de 58 (52+6) instâncias, agora se considerarmos o intervalo definido em 40, o total de instâncias sobe para 168 (161+7).

O mesmo é visto na Tabela 5.2, o intervalo definido como 10 resulta no total de 43 instâncias, e quando o intervalo é determinado como 40, o resultado sobe para 153 instâncias. Logo, este acréscimo no total selecionado de instâncias pode ser explicado pelo maior número de regras geradas nas projeções, devido ao maior número de intervalos.

Intervalos	**Instâncias Negativas**	**Instâncias Positivas**
10	52	6
20	109	8
30	159	9
40	161	7
50	189	11
60	230	12
70	253	13
80	220	16
90	242	21
100	278	24

Tabela 5.1: Seleção Ativa por Intervalos de IMDBxNetflix

Intervalos	**Instâncias Negativas**	**Instâncias Positivas**
10	13	30
20	37	31
30	61	54
40	81	72
50	88	89
60	97	129
70	74	102
80	87	107
90	99	126
100	120	123

Tabela 5.2: Seleção Ativa por Intervalos de DBLPxCiteseer

Os resultados dos conjuntos de instâncias selecionadas de cada intervalo, como treino perante a base inteira correspondente, são apresentados nas Tabelas 5.3 e 5.4. Denotados pelas colunas *Intervalos*, *Classificador*, *Prec*,*Rev* e *F1*.

Considerando a maioria dos intervalos testados nas Tabelas 5.3 e 5.4, em ambos os algoritmos, *SVM* e *RF*, a métrica de avaliação principal *F1* teve acréscimo ou se manteve acima do *F1* do intervalo padrão 10, conforme a quantidade de intervalos aumentava.

A quantidade de 50 intervalos foi escolhida como intervalo padrão para os próximos experimentos em decorrência de eliminações, buscando selecionar o intervalo mais promissor para ambas as bases. A eliminação começou pelos intervalos menores, pois todos selecionaram

uma quantidade menor de instâncias positivas e a média de *F1* foi igual ou abaixo. Já para os intervalos maiores que 50, a média mais alta de *F1* dos classificadores apresentou variação de menos de $\%1$, então foram eliminados os intervalos com a quantidade total de instâncias maior e também com maior diferença entre o total de instâncias de cada classe.

Intervalos	**SVM**			**RF**			**Treino**
	Prec	**Rev**	**F1**	**Prec**	**Rev**	**F1**	**Total**
10	0,83	0,69	0,76	0,95	0,69	0,80	58
20	0,87	0,77	0,81	0,95	0,85	0,90	117
30	0,96	0,86	0,91	0,96	0,86	0,91	168
40	0,98	0,69	0,81	0,97	0,84	0,90	168
50	0,93	0,92	0,92	0,97	0,87	0,92	200
60	0,97	0,87	0,91	0,99	0,83	0,90	242
70	0,98	0,67	0,79	0,98	0,86	0,92	266
80	0,95	0,90	0,92	0,97	0,89	0,93	236
90	0,95	0,90	0,93	0,97	0,87	0,92	263
100	0,94	0,91	0,92	0,98	0,87	0,92	302

Tabela 5.3: Avaliação Intervalos de IMDBxNetflix

Intervalos	**SVM**			**RF**			**Treino**
	Prec	**Rev**	**F1**	**Prec**	**Rev**	**F1**	**Total**
10	0,87	0,97	0,92	0,83	0,98	0,90	43
20	0,90	0,95	0,93	0,93	0,92	0,93	68
30	0,92	0,94	0,93	0,94	0,92	0,93	115
40	0,89	0,95	0,92	0,93	0,93	0,93	153
50	0,93	0,94	0,94	0,94	0,90	0,92	177
60	0,93	0,94	0,93	0,95	0,92	0,93	226
70	0,92	0,94	0,93	0,96	0,92	0,94	176
80	0,95	0,92	0,93	0,96	0,88	0,92	194
90	0,93	0,94	0,93	0,96	0,91	0,93	225
100	0,94	0,92	0,93	0,96	0,91	0,94	243

Tabela 5.4: Avaliação Intervalos de DBLPxCiteseer

5.4.2 Avaliação da Penalização Fixa com Intervalo

Neste experimento o objetivo é analisar se a penalização fixa sobre a quantização das regras, tem efeito sobre a seleção das instâncias.

Peso	**Instâncias Negativas**	**Instâncias Positivas**
1	189	11
2	200	22
3	129	14
4	97	9
5	44	7
6	39	7
7	36	6
8	36	6
9	34	6
10	27	4

Tabela 5.5: Seleção Ativa por Penalização Fixa de IMDBxNetflix.

Peso	**Instâncias Negativas**	**Instâncias Positivas**
1	88	89
2	32	70
3	23	67
4	18	63
5	15	61
6	15	61
7	15	61
8	15	61
9	15	61
10	15	61

Tabela 5.6: Seleção Ativa por Penalização Fixa de DBLPxCiteseer.

Nas Tabelas 5.5 e 5.6, está disposto a seleção por cada *Peso* aplicado como penalização. Desconsiderando a penalização neutra, ou seja, a penalização por 1, que não altera o total de regras geradas. Com efeito da penalização, o total de instâncias selecionadas reduz pois a penalização eleva o número total de regras das instâncias em geral na quantização. Isso faz com que existam menos chances de selecionar uma instância que ainda não foi escolhida e que tenha escapado da penalização gerando menos regras que as demais instâncias.

O resultado das seleções de instâncias de cada penalização usados como treino podem ser vistos nas Tabelas 5.7 e 5.8. É visto que mesmo com a redução no total de instâncias selecionadas, a métrica $F1$ continua acima de 90% em pelo menos um dos classificadores,

Pesos	SVM			RF			Treino
	Prec	Rev	F1	Prec	Rev	F1	Total
1	0,93	0,92	0,92	0,97	0,87	0,92	200
2	0,93	0,74	0,82	0,96	0,88	0,92	222
3	0,95	0,87	0,91	0,96	0,88	0,92	143
4	0,97	0,85	0,91	0,97	0,85	0,91	106
5	0,93	0,90	0,91	0,96	0,88	0,92	51
6	0,93	0,90	0,91	0,96	0,89	0,92	46
7	0,93	0,88	0,91	0,96	0,86	0,91	42
8	0,93	0,88	0,91	0,97	0,85	0,91	42
9	0,93	0,88	0,91	0,97	0,86	0,91	40
10	0,95	0,72	0,82	0,99	0,79	0,88	31

Tabela 5.7: Avaliação da Penalização Fixa de IMDBxNetflix

Pesos	SVM			RF			Treino
	Prec	Rev	F1	Prec	Rev	F1	Total
1	0,93	0,94	0,94	0,94	0,90	0,92	177
2	0,91	0,95	0,93	0,96	0,90	0,93	102
3	0,90	0,96	0,93	0,95	0,91	0,93	90
4	0,89	0,96	0,93	0,95	0,89	0,92	81
5	0,84	0,98	0,91	0,93	0,89	0,91	76
6	0,84	0,98	0,91	0,93	0,89	0,91	76
7	0,84	0,98	0,91	0,93	0,89	0,91	76
8	0,84	0,98	0,91	0,93	0,89	0,91	76
9	0,84	0,98	0,91	0,93	0,89	0,91	76
10	0,84	0,98	0,91	0,93	0,89	0,91	76

Tabela 5.8: Avaliação da Penalização Fixa de DBLPxCiteseer

considerando até a penalização por 9 das avaliações em ambas as bases.

Avaliando a redução do treino e *F1*, o resultado mais promissor para *IMDBxNetflix* pode ser considerado a penalização por 6, já para *DBLPxCiteseer* pode ser considerado a penalização por 3.

5.4.3 Avaliação da Penalização Variável com Intervalo

Neste experimento é testado a penalização proposta, vide seção 4.1.2, por [Yu et al., 2019], assim como uma proposta de alteração sobre a mesma. A quantidade de intervalos escolhida é de 50 conforme definido na Seção 5.4.1. O objetivo é verificar se a proporção das instâncias rotuladas, do conjunto de instâncias selecionadas, positivas e negativas, pode ser um fator de penalização promissor.

- Penalização proposta sem alteração:

Seleção Ativa	Instâncias Negativas	Instâncias Positivas
IMDBxNetflix	25	2
DBLPxCiteseer	42	52

Tabela 5.9: Seleção Penalização Variável

Base	SVM			RF			Treino
	Prec	Rev	F1	Prec	Rev	F1	Total
IMDBxNetflix	0,90	0,93	0,91	0,95	0,82	0,88	27
DBLPxCiteseer	0,92	0,94	0,93	0,94	0,91	0,93	94

Tabela 5.10: Avaliação Penalização Variável

- Penalização proposta com alteração:

Seleção Ativa	Instâncias Negativas	Instâncias Positivas
IMDBxNetflix	57	10
DBLPxCiteseer	49	63

Tabela 5.11: Seleção Penalização Variável com Alteração

Base	SVM			RF			Treino
	Prec	Rev	F1	Prec	Rev	F1	Total
IMDBxNetflix	0,92	0,92	0,92	0,96	0,88	0,92	67
DBLPxCiteseer	0,91	0,95	0,93	0,90	0,93	0,91	112

Tabela 5.12: Avaliação Penalização Variável com Alteração

Como pode ser visto nas Tabelas 5.11 e 5.12, a penalização proposta conseguiu selecionar mais instâncias positivas em ambas as bases e atingir valor de $F1$ igual ou maior em pelo menos um dos classificadores, pois as instâncias que tivessem em sua projeção uma quantização de regras maior de rótulo negativo eram mais penalizadas, postergando deste modo, sua seleção pelo SSAR.

Considerando o experimento de penalização fixa por intervalo, este experimento apresentou melhor resultado para *IMDBxNetflix* mas não para *DBLPxCiteseer*, porém a vantagem dessa abordagem está contida no detalhe de não ser necessário definir um intervalo e seus valores de penalização.

5.5 Discussão sobre os resultados

O objetivo desta seção é apresentar um resumo sobre os resultados atingidos nos experimentos. Como foi apresentado nas questões de pesquisa, três pontos foram investigados na aplicação do método SSAR. Sendo que foram realizados 44 experimentos, levando em conta as duas bases.

Em *QP1* é visualizável uma tendência de aumento do número de instâncias selecionadas conforme a quantidade de intervalos incrementava. Desta maneira mais instâncias positivas foram selecionadas e o *F1* resultante do conjunto de instâncias escolhidas na maioria dos intervalos teve resultado melhor que a escolha do intervalo padrão 10.

Já em *QP2*, com intervalo definido de acordo com *QP1* foi possível reduzir a quantidade de instâncias selecionadas no total, principalmente negativas. Também foi mantido *F1* acima de 90% em pelo menos um dos classificadores utilizados dentro do intervalo [2-9]. Por final, na *QP3* a penalização de [Yu et al., 2019] modificada, definida na Equação 4.3, teve desempenho melhor do que a penalização sem modificações, a grande vantagem dessa abordagem residiu em não precisar especificar um intervalo de penalização, pois os pesos são a proporção das classes no conjunto de instâncias rotuladas.

Nas Tabelas 5.13 e 5.14 estão dispostos os melhores resultados em cada experimento.

Experimento		**Instâncias Negativas**	**Instâncias Positivas**	**SVM F1**	**RF F1**
Intervalos	50	189	11	0,92	0,92
Penalização Fixa	6	39	7	0,91	0,92
Penalização Variável	Com Alteração	57	10	0,92	0,92

Tabela 5.13: Comparativo de Resultados IMDBxNetflix

Experimento		**Instâncias Negativas**	**Instâncias Positivas**	**SVM F1**	**RF F1**
Intervalos	50	88	89	0,94	0,92
Penalização Fixa	3	23	67	0,93	0,93
Penalização Variável	Com Alteração	49	63	0,93	0,91

Tabela 5.14: Comparativo de Resultados DBLPxCiteseer

É visto que com o uso da penalização foi possível reduzir o total do conjunto de instâncias selecionadas, diminuindo mais a quantidade de instâncias negativas. A penalização variável foi mais efetiva em *IMDBxNetflix* se considerarmos o *F1* dos dois classificadores utilizados, já em *DBLPxCiteseer* teve melhor resultado a penalização fixa. Concluindo, o desempenho da

penalização pode ser explicado, pelo fato de que as instâncias com um maior número de regras de rótulo negativo na quantização, foram penalizadas com mais frequência, removendo ou postergando sua seleção para o conjunto final.

6 CONCLUSÃO

Este trabalho teve como objetivo aprimorar um método de aprendizagem ativa para promover a seleção de instâncias menos representadas. Foi utilizado o *SSAR*, método ativo proposto por [Silva, 2012], para o desenvolvimento dos experimentos. Este seleciona apenas instâncias que considera informativa para montar um conjunto de treinamento, a seleção é pelo critério de quantização das regras geradas usando regras de associação. A contribuição do trabalho se trata da forma de penalização que reduziu de forma significativa a proporção de instâncias negativas selecionadas.

A partir do *SSAR*, nos experimentos, verificou-se que a alteração da discretização interfere na quantidade de instâncias menos representadas selecionadas. Além disso, um resultado não desejável do aumento no intervalo de discretização é a seleção de um número maior de instâncias negativas, o que foi tratado no experimento de penalização fixa e variável, reduzindo a quantidade de instâncias negativas de forma promissora.

Como trabalho futuro, pode-se verificar mais casos com diferentes bases de dados, com diferentes proporções entre classes. Adicionalmente, é necessário realizar testes estatísticos para validar a vantagem real da abordagem proposta.

REFERÊNCIAS

[1] E. Alpaydin. *Introduction to Machine Learning*. The MIT Press, 2nd edition, 2010. ISBN 026201243X, 9780262012430.

[2] D. Cohn and G. Schohn. Less is more: Active learning with support vector machines. *ICML Proceedings of the Seventeenth International Conference on Machine Learning*, 2000.

[3] A. Culotta and A. McCallum. Reducing labeling effort for structured prediction tasks. *In Proceedings of the National Conference on Artificial Intelligence(AAAI)*, 2005.

[4] I. Dagan and S. P. Engelson. Committee-based sampling for training probabilistic classifiers. *In Proceedings of the Twelfth International Conference on Machine Learning*, 1995.

[5] G. Dal Bianco, R. Galante, C. A. Heuser, and M. A. Gonçalves. Tuning large scale deduplication with reduced effort. pages 1–12, 2013.

[6] D. D. Lewis and W. A. Gale. A sequential algorithm for training text classifiers. *In Proceedings of the 17th annual international ACM SIGIR conference on Research and development in information retrieval.*, 1994.

[7] C. D. Manning, P. Raghavan, and H. Schütze. Introduction to information retrieval. *Cambridge University Press*, 2008.

[8] M. Mohri, A. Rostamizadeh, and A. Talwalkar. *Foundations of Machine Learning*. Adaptive Computation and Machine Learning. MIT Press, 2 edition, 2018.

[9] T. Scheffer, C. Decomain, , and S. Wrobel. Active hidden markov models for information extraction. *In Proceedings of the International Conference on Advances in Intelligent Data Analysis (CAIDA)*, 2001.

[10] G. Schimidberger and E. Frank. Unsupervised discretization using tree-based density estimation. *In Proceedings of the 9th European Conference on Principles and Practice of Knowledge Discovery in Databases, Berlin*, 2009.

[11] B. Settles. Active learning literature survey. *Computer Sciences Technical Report*, 2010. URL http://burrsettles.com/pub/settles.activelearning.pdf.

[12] R. Silva, M. A. Gonçalves, and A. Veloso. Rule-based active sampling for learning to rank. *ECML PKDD*, 2011.

[13] R. D. M. Silva. Aprendizado ativo para ordenação de resultados. *Instituto de Ciências Exatas*, 2012.

[14] H. Yu, X. Yang, S. Zheng, and C. Sun. Active learning from imbalanced data: A solution of online weighted extreme learning machine. *IEEE Transactions on Neural Networks and Learning Systems*, 30:1088–1103, 2019.