Análise de Sentimento Usando uma Máquina de Vetor de Suporte com Função de Núcleo Linear

Lucas Coelho de Almeida - 19/0136871, João Tribouillet Marcial de Menezes - 19/0136839

***Resumo*— Trabalho que procura utilizar Aprendizado de máquina utilizando *Support Vector Machines* junto com processamento de linguagem natural para classificar como positivos ou negativos comentários de filmes.**

***Palavras-Chave*— Análise de Sentimento; Aprendizado de Máquina; Máquinas de Vetor de Suporte; Processamento de Linguagem Natural.**

I. Introdução

A linguagem é um campo de estudo de interesse de linguistas e, principalmente hoje, de vários outros cientistas e profissionais, para as mais diversas finalidades comerciais e acadêmicas. A concepção de linguagens como sistemas permitiu a separação para análise de palavras em frases a partir de seus relacionamentos, repetições, estrutura gramatical e contrastes para verificação de aspectos sobre a própria frase. Assim, com a evolução da sociedade, ocorreram avanços nesses estudos, permitindo aplicações reais com técnicas mais avançadas e elegantes, como *Machine Learning*.

Natural Language Processing (NLP) é um campo da linguística que, junto com a Computação, objetiva a interpretação e processamento de textos escritos (ou falados). Trabalhos na área são verificados desde Alan Turing em [1], tendo evoluído ao ponto de permitir a criação de algoritmos complexos com o suporte de plataformas computadorizadas que interpretam e auxiliam usuários de forma automática.

A era da informação ampliou a expressividade de opiniões e comentários em diversos meios de comunicação em vários (se não todos) os meios sociais. Diante disso, várias entidades empresariais, governamentais e até indivíduos privados tiveram a necessidade de obter *feedbacks* sobre os comentários feitos pelas pessoas sobre algo de interesse, mesmo que esse problema potencialmente envolvesse milhares de usuários. Nesse sentido, NLP se mostrou uma ótima abordagem que permite a análise de sentimentos de forma sistemática. Com isso, é possível, de forma rápida e acurada, determinar quais comentários foram positivos e quais foram negativos.

”Aprendizado supervisionado” é uma área de aprendizado de máquina que permite ao programa melhorar uma medida de desempenho previamente escolhida. Isso é feito a partir da análise das relações entre parâmetros existentes no contexto do problema e o *output* referente a cada conjunto de parâmetros constituintes das entradas providas ao sistema/algoritmo. Ela pode ser dividida em duas áreas:

1) *Classificação*
   Permite a identificação de categorias previamente separadas a partir de um conjunto de dados que já contenha tanto os valores dos parâmetros quanto a categoria resultante.
2) *Regressão*
   Permite estimar o valor de algum parâmetro mediante a análise da relação entre os parâmetros de um conjunto de dados coletados sobre um problema tendo, nesse conjunto, os valores ocorridos do dado que se deseja estimar.

Departamento de Engenharia Elétrica (ENE), Universidade de Brasília (UnB), E-mails: luccoelhoo@gmail.com, jotatri@gmail.com

Dentro dos algoritmos de aprendizado supervisionado existe o *Support Vector Machine* (SVM), que pode ser utilizado tanto para problemas de classificação quanto regressão. SVMs são máquinas que separam os dados em diferentes rótulos em um plano (denomidado *hyperplan*), o qual possui uma margem que é maximizada para a separação dos dados. A Figura 1 mostra um exemplo de classificação com a utilização de um *hyperplane*.

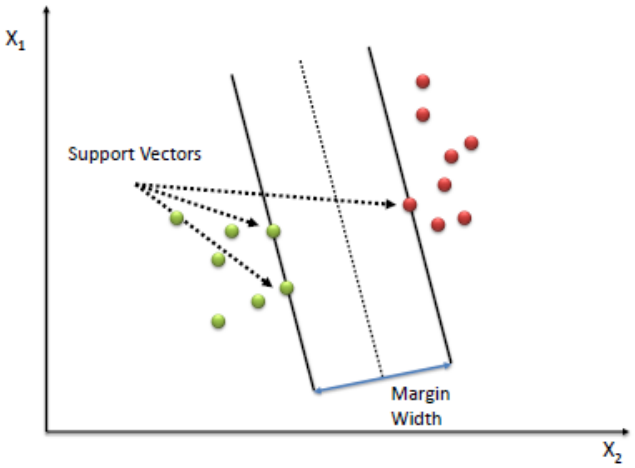

Fig. 1
SVM *Hyperplane*

Tal método para classificação utiliza, portanto, planos N-dimensionais lineares para a separação em diferentes classes. Entretanto, também é possível separar grupos por funções não lineares. Isso é feito variando as funções de *kernel*, ou núcleo, que modificam a forma de processamento dos dados para permitir tal separação. Uma delas é a *kernel trick*, que permite a transformação de dados para uma dimensão de dados maior do que a apresentada no problema a partir de uma função que calcula os produtos internos entre pares de dados, o que permite a separação em um plano de dimensão maior porém sem a necessidade de calcular ou levar em consideração as novas coordenadas dos parâmetros. Nesse novo espaço, é

possível a separação com uma função linear. Um exemplo dessa transformação está apresentado na Figura 2.

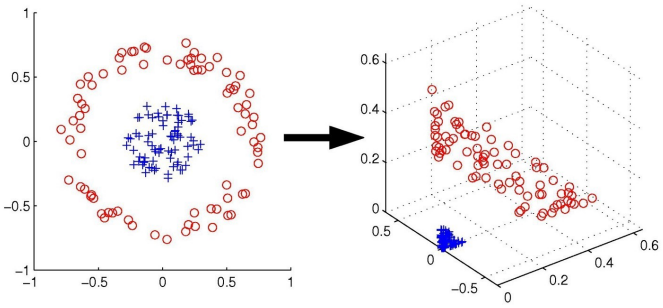

Fig. 2
KERNEL TRICK

Para a utilização de SVM para análise de sentimentos em frases, é necessário a transformação das informações contidas nas orações em dados numéricos. Para isso, pode ser utilizado a vetorização de palavras. Existem vários modos de realizar essa transformação como:

1) *Bag of Words*
   Consiste na identificação das palavras existentes no conjunto de frases em um vetor e na utilização dos índices das palavras neste vetor para indicar a existência da palavra em uma frase (com o número 1) ou não (com o número 0), gerando vetores para cada sentença.
2) *Word Counts*
   Consiste na identificação de palavras de interesse em um vetor e utilização de seus índices para determinar a quantidade de vezes que tal palavra foi falada em uma frase.
3) *Term Frequency and Inverse Document Frequency*
   Consiste na determinação de uma pontuação dada às palavras que permite inferir a sua importância em um escopo a partir da avaliação da frequência que a palavra é mencionada em um documento e a frequência com que palavras aparecem em diferentes documentos, o que sinalizaria que são menos importantes.

A seguir, a implementação e detalhes de projeto da SVM será abordada, bem como detalhes da base de dados usada e o tipo de vetorização escolhido para o problema.

II. IMPLEMENTAÇÃO DE UMA MÁQUINA DE VETOR DE SUPORTE COM FUNÇÃO DE NÚCLEO LINEAR PARA ANÁLISE DE SENTIMENTO

Em primeiro lugar, é importante entender qual o objetivo da máquina e os dados que serão usados para treinamento e teste. A análise de sentimento pode se tornar um problema extremamente complexo ao envolver questões subjetivas como nível de satisfação de um consumidor, entretanto, para a aplicação deste artigo, o objetivo será a avaliação de um comentário em linguagem natural como positivo ou negativo. Ou seja, o contexto de classificação da máquina será binário, em que ”positivo” significa a classe positiva (classe ”1”), e o que não for classificado como positivo será ”negativo” (classe ”0”).

A base de dados usada será a ”Polarity dataset v2.0” (disponível em [2]), a qual contém uma lista de avaliações de filmes previamente classificadas em relação ao sentimento em ”positivas” e ”negativas” (comentários escritos na língua inglesa). Um exemplo de como os dados são apresentados segue na imagem 3.

	Content	Label
0	every once in a while you see a film that is s...	pos
1	the love for family is one of the strongest dr...	pos
2	after the terminally bleak reservoir dogs and ...	pos

Fig. 3
EXEMPLO DE APRESENTAÇÃO DAS TRÊS PRIMEIRAS LINHAS DOS DADOS DE TREINO DA BASE ”POLARITY DATASET V2.0”.

A implementação de uma SVM (Support Vector Machine - Máquina de Vetor de Suporte) pode ser bastante facilitada dependendo da escolha da linguagem de programação e de suas bibliotecas, ou pacotes de funções. Para este projeto, a linguagem Python versão ”3” foi escolhida, bem como os pacotes SKLearn e Pandas. Uma breve descrição destes segue:

1) *Pandas [3]:* Pacote em Python criado para facilitar o uso e manejo de dados estruturados e séries temporais.
2) *SKLearn [4]:* Pacote em Python criado durante uma competição promovida pela empresa Google e que mais tarde seria liberado de forma oficial para a comunidade de software aberto. Seu objetivo é fornecer ferramentas para o desenvolvimento de programas baseados em aprendizado de máquina. Contém diversas funções para classificação, regressão, máquinas de vetor de suporte, aumento de gradiente, entre outros, além de trabalhar de forma conjunta com os pacotes ”Numpy” e ”SciPy”, padrões da linha científica de pesquisa usando a linguagem escolhida.

O pacote SKLearn contém uma função denominada ”svm”, a qual é usada na criação de máquinas de vetor de suporte que utilizam treinamento baseado na mínima distância quadrática. Por padrão, as máquinas são criadas usando função de núcleo do tipo ”*Radial Basis Function*” (RBF), ou função de base radial, contudo, para a aplicação de análise de sentimento, dado que as classificações são binárias e não contêm sobreposição (não existem, a priori, comentários neutros na base de dados), a função de núcleo do tipo ”linear” foi a escolhida. Na discussão dos resultados, as performances das duas funções, e adicionalmente da função de núcleo ”sigmoide” serão apresentadas.

Por fim, mas não menos importante, o método de vetorização dos dados. Como descrito na seção I, o processamento de linguagem natural (aquela usada para comunicação entre humanos, tanto escrita como falada) não se dá diretamente usando os caracteres e palavras, mas sim, usando algum método matemático que gera artefatos numéricos que representam a informação contida nas palavras e/ou sentenças. O mais comumente usado é o uso de algoritmos que transformem as sentenças em vetores, e o método escolhido para o problema foi o TF-IDF - ”Term Frequency and Inverse Document

Frequency”, ou Frequência de Termos e Frequência Inversa de Documento, disponível através da função ”TfidfVectorizer” (mais informações a respeito dos parâmetros da função podem ser encontrados em [5]). Nesse método, as palavras não são simplesmente contadas, é dado maior esforço para identificar aquelas mais importantes em cada sentença, ou seja, palavras comuns entre os comentários, mesmo que muito repetidas, são desinteressantes justamente por não acrescentarem novidade à análise. Assim, o algoritmo procura pela frequência de palavras dentro das sentenças e que não se repetem com facilidade entre os diferentes comentários, e monta vetores reais (sequência de números reais) representando a aparição dessas palavras. Cada termo é calculado segundo a equação na Figura 4. Na Figura 5, exemplo da nova representação dos dados de treino após vetorização.

$$\text{idf}(t) = log\frac{1+n_d}{1+\text{df}(d,t)} + 1,$$

Fig. 4

EQUAÇÃO PARA O CÁLCULO DA FREQUÊNCIA DE CADA TERMO NO ALGORITMO TF-IDF. O ”LOG” TEM COMO BASE ”E”, ”ND” É O NÚMERO DE COMENTÁRIOS/SENTENÇAS E ”DF(D,T)” É A CONTAGEM RELATIVA AO TERMO OBJETO DA DADA ANÁLISE [6].

```
(0, 12442)    0.027020558383921 24
(0, 1128)     0.025715898904247 15
(0, 4118)     0.050991425626187 31
(0, 5847)     0.050375367812363 24
(0, 4138)     0.067155710365791 93
(0, 8958)     0.049366264453454 58
(0, 5246)     0.047329706296509 98
(0, 9680)     0.042692033850973 9
(0, 7331)     0.049788657454804 42
(0, 3603)     0.067155710365791 93
(0, 2758)     0.053819458804157 67
(0, 9350)     0.027248265217541 45
(0, 3136)     0.037197627451227 91
(0, 11835)    0.058044054222750 94
(0, 2106)     0.044698774746501 58
(0, 2540)     0.062761978345868 64
(0, 3429)     0.082587982415150 57
(0, 3094)     0.042218183549368 52
(0, 9583)     0.073063374008690 95
(0, 11883)    0.048179601120214 625
(0, 4989)     0.070505550026024 47
(0, 4800)     0.026533944026732 918
(0, 11024)    0.042452830433279 88
(0, 12214)    0.028324563115548 446
(0, 9619)     0.027550719799348 96
:     :
```

Fig. 5

EXEMPLO DA NOVA REPRESENTAÇÃO DOS DADOS DE TREINO APÓS VETORIZAÇÃO

Portanto, uma vez que tem-se os detalhes de projeto, o fluxo de trabalho sugerido para o programa segue (bem como sua devida implementação em código, baseada em [7]) e apresentados nas Figuras 6 até 11]:

1) Incluir os pacotes Python necessários para o projeto. Presente na imagem 6.
2) Fazer o ”download” da base de dados (dados de treinamento disponíveis em [8] e de teste em [9]) e armazená-la em estruturas úteis para o programa usando a função ”read_csv” do pacote Pandas. Presente na imagem 7.
3) Criar os vetores reais usando a função ”TfidfVectorizer” do pacote ”sklearn” de cada sentença/comentário da base de dados de treinamento e de teste. Presente na imagem 8.
4) Criar a máquina de vetor de suporte com função de núcleo do tipo ”linear” usando a função ”svm” e a subfunção ”SVC” do pacote ”sklearn”. Presente na imagem 9.
5) Treinar a máquina criada com os vetores gerados através dos dados de treinamento. Presente na imagem 10.
6) Predizer a classificação de cada vetor gerado através dos dados de teste e comparar com a classificação previamente dada na base original, chegando, por fim, à assertividade do modelo. Presente na imagem 11.

```
from sklearn.feature_extraction.text import TfidfVectorizer
import time
from sklearn import svm
from sklearn.metrics import classification_report
import pandas as pd
```

Fig. 6

PASSO ”1”, INCLUSÃO DOS PACOTES.

```
trainData = pd.read_csv("https://raw.githubusercontent.com/Vasistareddy/sentiment_analysis/master/data/train.csv")
testData = pd.read_csv("https://raw.githubusercontent.com/Vasistareddy/sentiment_analysis/master/data/test.csv")
```

Fig. 7

PASSO ”2”, AQUISIÇÃO E ARMAZENAMENTO DOS DADOS DE TREINO E TESTE.

```
vectorizer = TfidfVectorizer(min_df = 5,
                             max_df = 0.8,
                             sublinear_tf = True,
                             use_idf = True)

train_vectors = vectorizer.fit_transform(trainData['Content'])
test_vectors = vectorizer.transform(testData['Content'])
```

Fig. 8

PASSO ”3”, VETORIZAÇÃO DOS DADOS USANDO ALGORITMO ”TF-IDF”.

```
classifier_linear = svm.SVC(kernel='linear')
```

Fig. 9

PASSO ”4”, CRIAÇÃO DA MÁQUINA DE VETOR DE SUPORTE COM FUNÇÃO DE NÚCLEO LINEAR.

```
classifier_linear.fit(train_vectors, trainData['Label'])
```

Fig. 10

PASSO ”5”, TREINAMENTO DA MÁQUINA SVM COM OS VETORES DOS DADOS DE TREINAMENTO.

```
prediction_linear = classifier_linear.predict(test_vectors)
report = classification_report(testData['Label'], prediction_linear, output_dict=True)
```

Fig. 11

PASSO ”6”, PREDIÇÃO DO SENTIMENTO DOS VETORES DOS COMENTÁRIOS DE TESTE.

Na próxima seção, serão brevemente discutidos e apresentados os resultados do treinamento e validação do projeto descrito após implementado.

III. DISCUSSÃO DOS RESULTADOS

Na Figura 12, é possível ver os resultados da aplicação da SVM com função de núcleo linear. Nas Figuras 13 e 14, é possível ver o resultado com o uso das funções ”RBF” e ”sigmoide”, respectivamente. É interessante notar que o o uso da função linear, a mais simples, foi também a que obteve a melhor assertividade em ambas as classes. Isso se dá pela natureza do problema e dos dados: provavelmente, existe pouca superposição entre os sentimentos positivos e negativos (para a base de dados em questão) e o espaço que os separa não é ”sinuoso”, ou seja, não é uma função de ordem muito grande, mesmo que possivelmente tenha dimensão grande (muitas variáveis).

```
Results for SVC(kernel=linear)
Training time: 8.548708s; Prediction time: 0.937417s
positive:  {'precision': 0.9191919191919192, 'recall': 0.91, 'f1-score': 0.91457
28643216081, 'support': 100}
negative:  {'precision': 0.9108910891089109, 'recall': 0.92, 'f1-score': 0.91542
28855721394, 'support': 100}
>>>
```

Fig. 12

RESULTADOS DA IMPLEMENTAÇÃO DA MÁQUINA DE CLASSIFICAÇÃO DE SENTIMENTO APLICADA AOS COMENTÁRIOS SOBRE FILMES DA BASE DE DADOS ”POLARITY DATASET V2.0”. PRECISÃO DE CERCA DE 91% PARA AMBAS AS CLASSES.

```
Results for SVC(kernel=linear)
Training time: 10.637846s; Prediction time: 0.999929s
positive:  {'precision': 0.7787610619469026, 'recall': 0.88, 'f1-score': 0.82629
10798122066, 'support': 100}
negative:  {'precision': 0.8620689655172413, 'recall': 0.75, 'f1-score': 0.80213
90374331552, 'support': 100}
>>>
```

Fig. 13

USO DE UMA SVM COM FUNÇÃO DE NÚCLEO DO TIPO ”RBF” (RADIAL BASIS FUNCTION - FUNÇÃO DE BASE RADIAL). PRECISÃO DE CERCA DE 77% PARA A CLASSE POSITIVA E 86% PARA A CLASSE NEGATIVA.

IV. CONCLUSÕES

A interpretação de frases, escritas ou faladas, de forma automática, bem como suas construções, são de extrema importância hoje em dia.Essa análise de frases é muito facilitada com a vetorização das palavras, pois permite o uso de funções matemáticas para espacialmente separar diferentes rótulos. A partir da vetorização com análise TF-IDF do ”dataset” usado, foi possível a criação de uma SVM com várias funções de separação lineares, visto que existiam duas classes de interesse: Positivos e Negativos. As precisões obtidas para cada função utilizada foram:

1) *Linear Function*
   Comentários positivos: 91.91%
   Comentários negativos: 91.08%
2) *Radial Basis Function*
   Comentários positivos: 77.87%
   Comentários negativos: 86.20%
3) *Sigmoid*
   Comentários positivos: 77.87%
   Comentários negativos: 86.20%

Obteve-se um melhor resultado com a utilização da função linear, porém os outros resultados de acurácia não foram baixos. Isso pode ser ao fato da existência de uma diferença significante entre as duas classes, tornando este um problema binário sem muita superposição e permitindo sua correta classificação.

```
Results for SVC(kernel=linear)
Training time: 9.166630s; Prediction time: 1.015576s
positive:  {'precision': 0.7787610619469026, 'recall': 0.88, 'f1-score': 0.82629
10798122066, 'support': 100}
negative:  {'precision': 0.8620689655172413, 'recall': 0.75, 'f1-score': 0.80213
90374331552, 'support': 100}
>>>
```

Fig. 14

USO DE UMA SVM COM FUNÇÃO DE NÚCLEO DO TIPO ”SIGMOIDE”. PRECISÃO DE CERCA DE 77% PARA A CLASSE POSITIVA E 86% PARA A CLASSE NEGATIVA, ASSIM COMO DO TIPO ”RBF”.

REFERÊNCIAS

[1] “Computing machinery and intelligence.” [Online]. Available: https://www.csee.umbc.edu/courses/471/papers/turing.pdf

[2] “Movie review data,” Cornell University. [Online]. Available: http://www.cs.cornell.edu/people/pabo/movie-review-data/

[3] “Pandas project,” NumFOCUS. [Online]. Available: https://pandas.pydata.org/index.html

[4] “Scikit-learn - machine learning in python,” Scikit-learn’s development and maintenance team. [Online]. Available: https://scikit-learn.org/stable/#

[5] “Refeerência - sklearn.svm.svc,” Scikit-learn’s development and maintenance team. [Online]. Available: https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html

[6] “Preparing the text data with scikit-learn,” Medium - Vasista Reddy. [Online]. Available: https://medium.com/@vasista/preparing-the-text-data-with-scikit-learn-b31a3df567e

[7] “Sentiment analysis using svm,” Medium - Vasista Reddy. [Online]. Available: https://medium.com/@vasista/sentiment-analysis-using-svm-338d418e3ff1

[8] “Dados treinamento - olarity dataset v2.0”,” Medium - Vasista Reddy. [Online]. Available: https://raw.githubusercontent.com/Vasistareddy/sentiment_analysis/master/data/train.csv

[9] “Dados teste - olarity dataset v2.0”,” Medium - Vasista Reddy. [Online]. Available: https://raw.githubusercontent.com/Vasistareddy/sentiment_analysis/master/data/test.csv