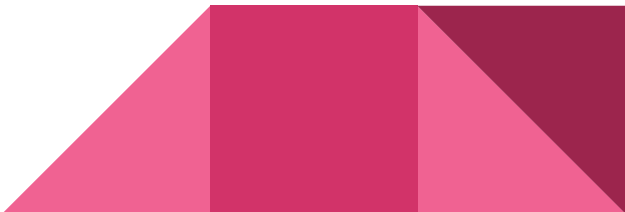


Implementação de Kernel Customizado Aplicado à Análise de Sentimentos em Resenhas de Filmes

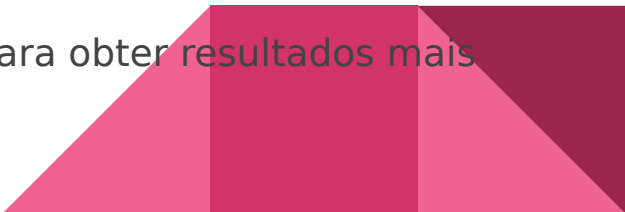
Luciana Kayo e Paulo Mei

Prof. Dr. Marco Dimas Gubitoso

Introdução

- ❑ Inspiração na competição “*When Bag of Words Meets Bags of Popcorn*” do Kaggle
 - ❑ Competição iniciada em outubro de 2014
 - ❑ Análise de sentimentos em resenhas de filmes
 - ❑ Uso da ferramenta Word2Vec do Google, que busca significado semântico em textos
 - ❑ Base de dados do IMDb
- 

Proposta

- ☐ Fazer análise de sentimentos nas resenhas dos filmes utilizando SVM (*Support Vector Machine*)
 - ☐ Conjunto de dados é o mesmo disponibilizado para a competição do Kaggle
 - ☐ Objetivo é comparar kernels diferentes quanto a acurácia em determinar uma resenha em positiva ou negativa
 - ☐ Kernel linear vs. Kernel customizado
 - ☐ Estudo de formas melhores de tratamento de dados para obter resultados mais significativos
- 

Conceitos

- ☐ Mineração de dados (*Data mining*)

 - ☐ Análise de sentimentos

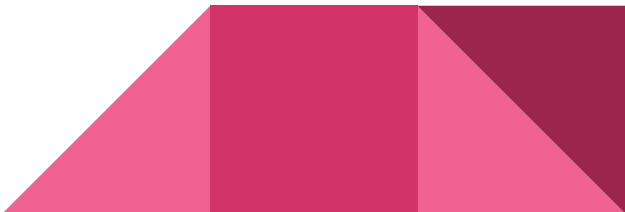
- ☐ Aprendizado supervisionado

 - ☐ SVM

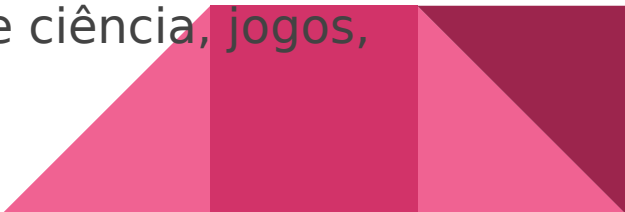
 - ☐ Função de *kernel*

- ☐ *Bag of Words*

- ☐ Tf-idf



Mineração de Dados

- ☐ Análise de um grande conjunto de dados em busca de padrões e conhecimento
 - ☐ Foco em extrair informações do conjunto e transformá-las em estruturas que podem ser melhor entendidas e usadas posteriormente
 - ☐ Mescla aprendizagem de máquina (*Machine Learning*), inteligência artificial e estatística
 - ☐ Aplicação em diversas áreas, como medicina e ciência, jogos, vigilância, finanças e negócios etc
- 

Análise de Sentimentos

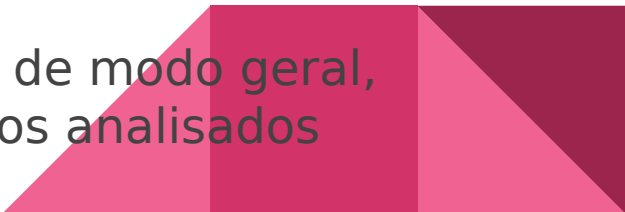
- ☐ Campo da mineração de dados
- ☐ Sinônimo de mineração de opinião
 - ☐ Uso de processamento de linguagem natural e análise de textos
 - ☐ Objetivo é identificar e extrair informações subjetivas em textos
 - ☐ Tentativa de entender o sentimento do autor no momento da escrita, determinando a polaridade do texto: positiva ou negativa

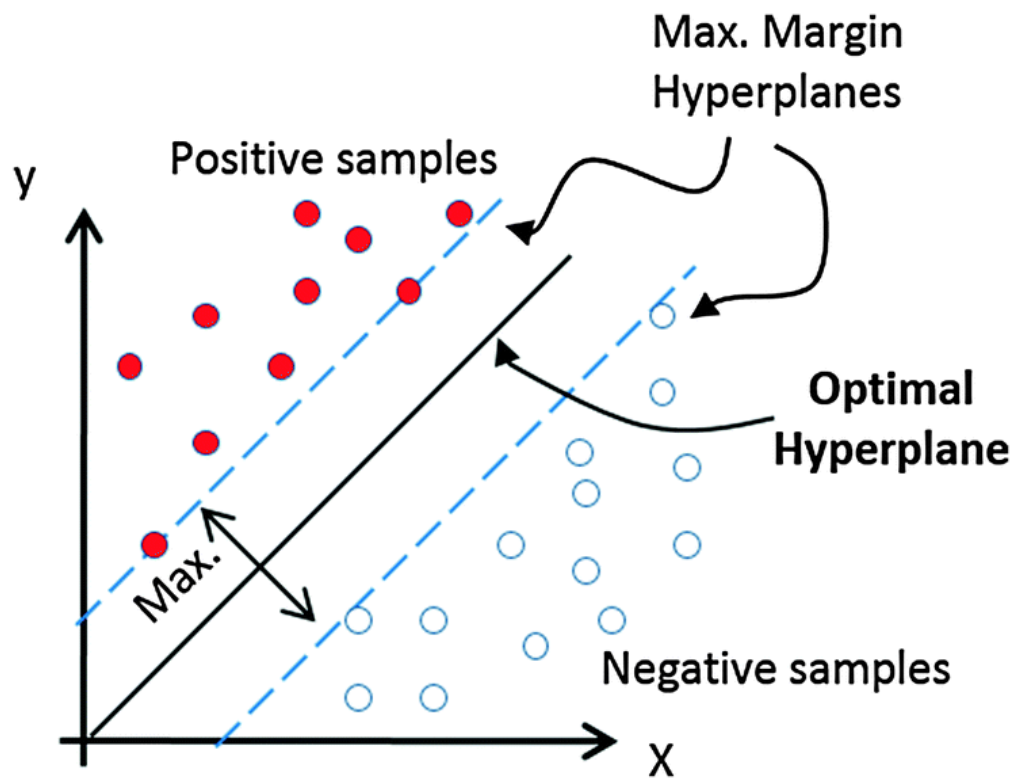


Aprendizado Supervisionado

- ☐ Subcampo de aprendizagem de máquina
 - ☐ Busca analisar um conjunto de dados de treino rotulados para encontrar uma função
 - ☐ Dados consistem de uma informação de entrada e um resultado esperado (rótulo)
 - ☐ Função encontrada deve ser capaz de mapear novos dados não rotulados
- 

SVM

- ❑ Modelo de aprendizado supervisionado
 - ❑ Algoritmo que busca padrões em dados para classificação e análise de regressão
 - ❑ Analisa um conjunto de dados de treino rotulados e cria um modelo para definir se um novo dado pertence a uma categoria ou a outra
 - ❑ Basicamente cria um hiperplano que separa o conjunto em duas classificações
 - ❑ Baseia-se em uma função de *kernel*, que tenta, de modo geral, computar um produto interno a partir dos dados analisados
- 



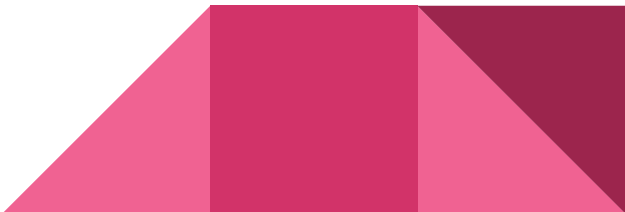
Exemplo de definição de hiperplano pelo SVM

Bag of Words

- ☐ Modelo utilizado em processamento de linguagem natural
- ☐ Texto é representado em um saco de palavras
- ☐ Considera a multiplicidade das palavras, desconsiderando gramática ou a ordem em que ocorrem
- ☐ Técnica usada na classificação de textos baseadas na frequência de palavras como vetor de características

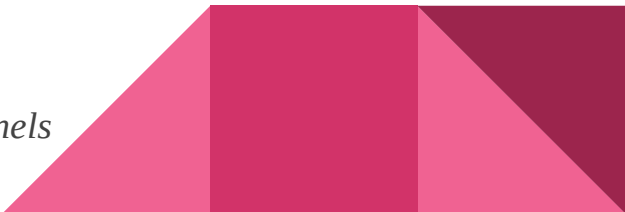


Tf-idf

- ☐ Do inglês, *Term frequency-inverse document frequency*
 - ☐ Cálculo da frequência de um termo (palavra) levando em conta tanto a frequência num único texto quanto no conjunto de textos
 - ☐ Termo que aparece muitas vezes num texto ganha um peso alto, pois é importante
 - ☐ Termo que aparece em muitos textos no conjunto ganha um peso baixo, pois não é bom para ajudar a separar as amostras
 - ☐ Peso final do termo depende dos dois fatores
- 

Implementação

☐ Fases:

- ☐ Definição do conjunto de dados a serem estudados
 - ☐ Definição da linguagem e ferramentas a serem usadas
 - ☐ Pré-processamento (limpeza) dos dados
 - ☐ Uso do SVM com dois *kernels* diferentes
 - ☐ Análise dos resultados obtidos
 - ☐ Conclusão com base no desempenho de ambos os *kernels*
- 

Conjunto de Dados

- ☐ Dados são resenhas do IMDb, previamente organizados para a necessidade da competição do Kaggle
- ☐ Cada dado é uma linha de um arquivo contendo:
 - ☐ Identificador
 - ☐ Rótulo binário: 1 para positivo (nota ≥ 7) e 0 para negativo (nota < 5)
 - ☐ Texto da resenha, em inglês
- ☐ Conjunto inicial de 25 mil dados rotulados



Linguagem e Ferramentas

☐ Linguagem Python:

- ☐ Linguagem de alto nível
 - ☐ Geralmente utilizada para mineração de dados e SVM
 - ☐ Possui várias ferramentas que podem ser acopladas aos programas e diversas bibliotecas com funções pré-definidas para uso de SVM
 - ☐ Pandas para manipulação de dados, análise e uso de estruturas como *data frame*
 - ☐ NLTK para processamento de linguagem natural, classificação, *tagging* e tokenização de palavras
 - ☐ Scikit-Learn, baseado em NumPy, SciPy e matplotlib, para processamento dos dados e classificação com algoritmos como SVM
- 

Pré-processamento dos dados

☐ Passos principais:

- ☐ Remover *tags* HTML

- ☐ Remover *stop-words*: termos como “*the*”, “*is*” etc

☐ Alternativas para melhorar o resultado dos algoritmos:

- ☐ *POS-Tagging (Part-of-Speech Tagging)*: classificar termos de uma frase segundo o tipo de palavra: adjetivo, substantivo, advérbio, verbo etc

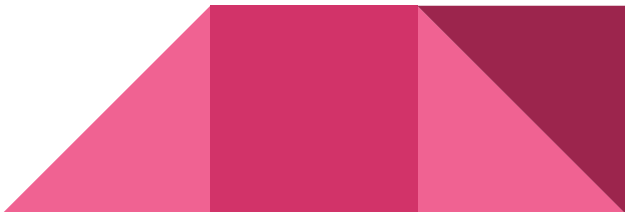
- ☐ Separar apenas adjetivos, uma vez que tem peso em definir algo como positivo e negativo

- ☐ Separar apenas adjetivos e advérbios, dada a intensificação que os advérbios trazem aos adjetivos

SVM

- ☐ Dados limpos foram submetidos ao aprendizado pelo SVM
- ☐ 60% da amostra rotulada separada para treino
- ☐ 40% restante da amostra para testes
- ☐ Aplicação dos dois *kernels* com o SVM
- ☐ Cálculo do *score-in* e *score-out*, porcentagem de acerto dentro da amostra de treino e fora dela, respectivamente, para comparação dos *kernels*

SVM com *Kernel Linear*

- ☐ *Kernel* linear basicamente define uma função linear para a classificação
 - ☐ Uso do SVM do Scikit-Learn
 - ☐ Dados foram submetidos ao *Bag of Words* e depois ao *Tf-idf*
 - ☐ Foco na frequência de adjetivos e advérbios no texto, independente da ordem
 - ☐ Permitiu trabalhar com amostra maior
 - ☐ Bom resultado de acertos
- 

Idéia do *Kernel* Customizado

- ❑ Surgiu em uma conversa com o professor Gubi
- ❑ Ele sugeriu um esboço de *kernel* que levasse em conta a ordenação de palavras e as sequências delas que poderiam se repetir em mais de uma resenha
- ❑ A idéia era dar peso maior às palavras que aparecessem na mesma ordem em diferentes resenhas, fazendo algo próximo de um *cluster* em torno das mais significativas



	Film e	muit o	engr açad o	diver tido
Foi	0	0	0	0
muit o	0	8	8	0
engr açad o	0	8	10	8
diver tido	0	0	8	8
surpr eend ente	0	0	0	0

Esboço do que seria o *kernel* com um *cluster*

Gap-weighted subsequence kernel

- ❑ Durante as pesquisas sobre *string kernels*, foi encontrado um algoritmo parecido com o sugerido pelo professor Gubi
- ❑ *String kernel* apresentado no livro *Kernel Methods for Pattern Analysis* e no paper *Text Classification Using String Kernels*
- ❑ Idéia é procurar por substrings comuns nos textos, porém dando margem a um espaçamento entre as letras, ou seja, a nem todas as letras da sequência precisam estar uma imediatamente após a outra
- ❑ Porém se as letras da substring estiverem na sequência buscada, a substring recebe um peso maior

	c-a	c-t	a-t	b-a	b-t	c-r	a-r	b-r
f(c at)	lambda b d a^2	lambda b d a^3	lambda b d a^2	0	0	0	0	0
f(c ar)	lambda b d a^2	0	0	0	0	lambda b d a^3	lambda b d a^2	0
f(b at)	0	0	lambda b d a^2	lambda b d a^2	lambda b d a^3	0	0	0
f(b ar)	0	0	0	lambda b	0	0	lambda b	lambda b

Exemplo
do par

ence kernel, retirado

Solução

- ☐ Adaptar o algoritmo já existente
- ☐ Substituir as comparações com substrings com comparações por palavras seguidas no texto
- ☐ Levar em conta a ordem das palavras
- ☐ Levar em conta a distância entre palavras



Resultados

SVM Count Bag of Words			
	Sem stop- words	Só adjetiv os	Adjetiv os + Advérb ios
Score- in * Amostra de 25 mil	0.9963	0.8403	0.8996
Score- out	0.8284	0.7955	0.8205

Resultados

SVM <i>Tf-idf</i> Bag of Words			
	Sem <i>stop-words</i>	Só adjetivos	Adjetivos + Advérbios
<i>Score-in</i> * Amostra de 25 mil	0.9394	0.8359	0.8815
<i>Score-out</i>	0.8732	0.7981	0.8377

Resultados

SVM com <i>kernel</i> customizado	
	Adjetivos + Advérbios
<i>Score-in</i> * Amostra de 1 mil	0.9967
<i>Score-out</i>	0.5275

Resultados

Tempo de execução			
	Sem <i>stop-words</i>	Só adjetivos	Adjetivos + Advérbios
Limpeza	17m 20.79s	10h 00m 35.22s	10h 01m 03.41s
SVM <i>Count</i>	22m 26.75s	07m 49.56s	14m 33.00s
SVM <i>Tf-idf</i>	23m 06.03s	05m 35.15s	09m 38.79s

Conclusões

- ☐ Filtragem por adjetivos melhora o desempenho, em relação ao uso de todo tipo de palavra, reduzindo o vetor de características, porém a um custo alto de pré-processamento
- ☐ Filtrar por adjetivo e advérbio melhora um pouco mais os resultados, porém demora ainda mais tempo para o pré-processamento
- ☐ Kernel customizado tem execução muito custosa em termos de processamento de tempo de execução
- ☐ Hipótese de que o tempo e o desempenho melhorariam com o uso apens de adjetivos e advérbios
- ☐ Mesmo com a redução, o custo computacional é muito alto para um

Fim!

Obrigado!

