

Classificação

Prof. Dr. Thiago de Paulo Faleiros

Roteiro da apresentação

- 1 Classificação de texto
- 2 Avaliação

Classificação de texto

Atribua categorias, tópicos ou gênero

Detecção de spam

Identificação de autoria

Identificação de linguagem

Análise de sentimento

Classificação de texto: definição

Entrada:

- um documento d
- um conjunto fixo de classes $C = \{c_1, c_2, \dots, c_j\}$

Saída: uma classe predita $c \in C$

Método básico de Classificação

Regras baseadas na combinação de palavras ou outras características

- spam: black-list-address ou (“dollar” e “have been selected”)

A acurácia pode ser alta

- em domínios específicos
- Se as regras são claramente definidas por experts

mas:

- construir e manter regras é caro
- são muito literais e específico: muita precisão e pouca revocação

Método de Classificação: Aprendizagem de Máquina supervisionada

Entrada:

- um documento d
- um conjunto fixo de classes $C = \{c_1, c_2, \dots, c_j\}$
- Um conjunto de treinamento m manualmente anotado $(d_1, c_1), \dots, (d_m, c_m)$

Saída:

- um classificador $\lambda : d \rightarrow c$

Método de Classificação: Aprendizagem de Máquina supervisionada

Vários tipos de classificadores!

- Naive Bayes (essa aula)
- Regressão Logística
- Redes Neurais
- K-vizinhos mais próximos
- ...

Podemos também usar modelos de linguagem pré-treinados!

- Fine-tuned um modelo para classificação
- Usar prompt para classificação (Zero or Few shot learning)

Vamos ser ingênuos

Naive Bayes é um classificador simples baseado em regras
Baseia-se em uma representação simples do documento

- **Bag-of-words**

A representação em Bag-of-Words

I love this movie! It's sweet, but with satirical humor. The dialogue is great and the adventure scenes are fun... It manages to be whimsical and romantic while laughing at the conventions of the fairy tale genre. I would recommend it to just about anyone. I've seen it several times, and I'm always happy to see it again whenever I have a friend who hasn't seen it yet!





it	6
I	5
the	4
to	3
and	3
seen	2
yet	1
would	1
whimsical	1
times	1
sweet	1
satirical	1
adventure	1
genre	1
fairy	1
humor	1
have	1
great	1
...	...

A representação em Bag-of-Words

$Y($

seen	2
sweet	1
whimsical	1
recommend	1
happy	1
...	...

$) = C$

Lei de Bayes aplicada na classificação de documentos

- Para um documento d e uma classe c

$$P(c|d) = \frac{P(d|c)P(c)}{P(d)}$$

Classificador Naive Bayes

$$\begin{aligned}c_{MAP} &= \operatorname{argmax}_{c \in C} P(c|d) \\&= \operatorname{argmax}_{c \in C} \frac{P(d|c)P(c)}{P(d)} \\&= \operatorname{argmax}_{c \in C} P(d|c)P(c)\end{aligned}$$

* MAP é “maximum at posteriori” = a classe mais provável

Classificador Naive Bayes

$$\begin{aligned}c_{MAP} &= \operatorname{argmax}_{c \in C} P(d|c)P(c) \\ &= \operatorname{argmax}_{c \in C} P(x_1, x_2, \dots, x_n|c)P(c)\end{aligned}$$

$P(d|c)$ é a probabilidade do documento dada a classe

$P(c)$ é o conhecimento a priori da classe

O documento é representado pelas n características

$$d = \{x_1, x_2, \dots, x_n\}$$

Classificador Naive Bayes Multinomial

$$P(x_1, x_2, \dots, x_n | c)$$

Pressupõe representação Bag-of-Words: A posição das palavras não importa

Independência condicional: Assume que as probabilidades das características $P(x_i | c_j)$ são independentes dada a classe c .

$$P(x_1, x_2, \dots, x_n | c) = P(x_1 | c) \times P(x_2 | c) \times P(x_3 | c) \times \dots \times P(x_n | c)$$

Naive Bayes Multinomial

$$c_{MAP} = \arg \max_{c \in C} P(x_1, x_2, \dots, x_n | c) P(c)$$

$$c_{NB} = \arg \max_{c \in C} P(c_j) \prod_{x \in X} P(x | c)$$

Aplicando o MNB

posição \leftarrow todas as posições de palavras no documento de teste

$$c_{NB} = \arg \max_{c_j \in C} P(c_j) \prod_{i \in \text{posição}} P(x_i | c_j)$$

Problemas por multiplicar várias probabilidades

Existe um problema com essa equação:

$$c_{NB} = \arg \max_{c_j \in C} P(c_j) \prod_{i \in \text{posição}} P(x_i | c_j)$$

Multiplicar várias probabilidades pode resultar em *float-point underflow*!

$$0.0006 \times 0.0007 \times 0.009 \times 0.1 \times 0.5 \times 0.000008$$

Ideia: Usar *logs*, notem que $\log(ab) = \log a + \log b$ Iremos somar várias probabilidades em vez de multiplicar

Na verdade tudo é feito em espaço log

Em vez de:

$$c_{NB} = \arg \max_{c_j \in C} P(c_j) \prod_{i \in \text{posição}} P(x_i | c_j)$$

Faça:

$$c_{NB} = \arg \max_{c_j \in C} \left[\log P(c_j) + \sum_{i \in \text{posição}} \log P(x_i | c_j) \right]$$

Notas:

- 1 Calcular o log não muda o ranking das classes!
A classe com maior probabilidade tem a maior probabilidade no espaço logarítimo
- 2 É um modelo linear: Apenas uma soma máxima de pesos:
uma função linear nas entradas
Então Naive Bayes é um classificador linear!

Aprendizagem com o Naive Bayes Multinomial

Primeira tentativa: estimar a probabilidade máxima

- simplesmente utilize a frequência

$$P(c_j) = \frac{N_{c_j}}{N_{total}}$$

$$P(w_i|c_j) = \frac{count(w_i, c_j)}{\sum_{w \in V} count(w, c_j)}$$

Estimando Parâmetros

$$P(w_i|c_j) = \frac{\text{count}(w_i, c_j)}{\sum_{w \in V} \text{count}(w, c_j)}$$

Fração de vezes que uma palavra w_i aparece entre todas as palavras de um documento da classe c_j .

Crie uma mega documento para a classe j concatenando todos os documentos neste tópico

- Use a frequência de w no mega documento

Problemas com a probabilidade máxima

E se não pudermos ver nenhum documento no treinamento com a palavra “fantástico” e classificá-la com uma classe positiva?

$$P(\text{"fantástico"}|\text{positivo}) = \frac{\text{count}(\text{"fantástico"}, \text{positivo})}{\sum_{w \in V} \text{count}(w, \text{positivo})} = 0$$

Probabilidades com valor zero não pode ser condicionadas, não importa a evidência!

$$c_{MAP} = \arg \max_c \frac{P(c)}{\prod_i P(x_i|c)}$$

Suavização de Laplace

$$\begin{aligned} P(w_i|c) &= \frac{\text{count}(w_i, c) + 1}{\sum_{w \in V} (\text{count}(w, c) + 1)} \\ &= \frac{\text{count}(w_i, c) + 1}{(\sum_{w \in V} \text{count}(w, c)) + |V|} \end{aligned}$$

Naive Bayes Multinomial

- Do corpus de treinamento extraia o Vocabulário

Calculando o termo $P(c_j)$

- Para cada classe c_j em C
 - $docs_j \leftarrow$ todos os documentos com classe = c_j

$$P(c_j) \leftarrow \frac{|docs_j|}{|total\ n\ docs|}$$

Calculando o termo $P(w_k|c_j)$

- $text_j \leftarrow$
documento simples contendo todos os documentos $docs_j$
- Para cada palavra w_k no vocabulário
 - $n_k \leftarrow n.$ ocorrências de w_k em $text_j$
 - $P(w_k|c_j) \leftarrow \frac{n_k + \alpha}{n + \alpha|V|}$

Palavras desconhecidas

E palavras desconhecidas?

- que aparecem no conjunto de teste
- mas não no conjunto de treinamento?

Ignore!

- Remova do documento de teste
- Finja que elas não estão lá!
- Não inclua nenhuma probabilidade para elas

E se construirmos um modelo com palavras desconhecidas?

- Não resolve! Sabendo qual classe tem mais palavras desconhecidas não é geralmente útil.

Stop words

Alguns sistemas ignoram stop words

- **Stop words:** palavras muito frequente como “uma” e “The” (ingles).
- Chame as top 10 ou 50 palavras de **stopword list**.
- Remove todas as stop words do treinamento e do conjunto de teste
 - Como se elas não existissem!

Mas remover as stop words não ajuda

- Na prática, a maioria das implementações do NB usa todas as palavras e não remove a lista de stopwords.

Vamos fazer um classificador de sentimentos

	class	documentos
Treino	–	simplesmente chato
	–	totalmente previsível e sem energia
	–	não surpreendente e poucas risadas
	+	muito poderoso
	+	o filme mais divertido do verão
Teste	?	previsível e sem graça

Vamos fazer um classificador de sentimentos

	class	documentos
Treino	-	simplesmente chato
	-	totalmente previsível e sem energia
	-	não surpreendente e poucas risadas
	+	muito poderoso
	+	o filme mais divertido do verão
Teste	?	previsível e não é divertido

3. Probabilidades do treinamento

$$p(w_i|c) = \frac{\text{count}(w_i, c) + 1}{(\sum_{w \in V} \text{count}(w, c) + |V|)}$$

$$p(\text{previsível}|-) = \frac{1+1}{14+20} \quad p(\text{previsível}|+) = \frac{0+1}{9+20}$$

$$p(\text{não})|-) = \frac{1+1}{14+20} \quad p(\text{não}|+) = \frac{0+1}{9+20}$$

$$p(\text{divertido})|-) = \frac{0+1}{14+20} \quad p(\text{divertido}|+) = \frac{0+1}{9+20}$$

1. A priori do treinamento:

$$P(c_j) = \frac{n_{c_j}}{n_{total}}$$

$$P(-) = 3/5 \quad P(+) = 2/5$$

2. Remove “e”, “é”

4. Score do conjunto de teste:

$$p(-)p(S|-) = \frac{3}{5} \times \frac{2 \times 2 \times 1}{34^3} = 6.1 \times 10^{-5}$$

$$p(+)p(S|+) = \frac{2}{5} \times \frac{1 \times 1 \times 2}{29^3} = 3.2 \times 10^{-5}$$

Otimizando para análise de sentimentos

Para tarefas como análise de sentimentos, a ocorrência das palavras parece mais importante que a frequência.

- A ocorrência da palavra “fantástico” nos diz muito
- O fato da palavra ocorrer 5 vezes não diz muito

Naive Bayes Multinomial Binário

- contagem de palavras até 1

Naive Bayes Multinomial

- Do corpus de treinamento extraia o Vocabulário
- Calculando o termo $P(c_j)$

- Para cada classe c_j em C
 - $docs_j \leftarrow$ todos os documentos com classe = c_j

$$P(c_j) \leftarrow \frac{|docs_j|}{|total\ n\ docs|}$$

Remove palavras duplicadas em cada documento

Calculando o termo $P(w_k|c_j)$

- $text_j \leftarrow$ documento simples contendo todos os documentos $docs_j$
- Para cada palavra w_k no vocabulário
 - $n_k \leftarrow n.$ ocorrências de w_k em $text_j$
 - $P(w_k|c_j) \leftarrow \frac{n_k + \alpha}{n + \alpha|V|}$

Binary Multinomial Naive Bayes no conjunto de teste

Primeiro remove todas as palavras duplicadas
Então compute NB usando a mesma equação:

$$c_{NB} = \arg \max_{c_j \in C} P(c_j) \prod_{i \in \text{posição}} P(w_i | c_j)$$

Binary Multinomial Naive Bayes

Four original documents:

- it was pathetic the worst part was the boxing scenes
- no plot twists or great scenes
- + and satire and great plot twists
- + great scenes great film

After per-document binarization:

- it was pathetic the worst part boxing scenes
- no plot twists or great scenes
- + and satire great plot twists
- + great scenes film

	NB Counts		Binary Counts	
	+	-	+	-
and	2	0	1	0
boxing	0	1	0	1
film	1	0	1	0
great	3	1	2	1
it	0	1	0	1
no	0	1	0	1
or	0	1	0	1
part	0	1	0	1
pathetic	0	1	0	1
plot	1	1	1	1
satire	1	0	1	0
scenes	1	2	1	2
the	0	2	0	1
twists	1	1	1	1
was	0	2	0	1
worst	0	1	0	1

Figure 4.3 An example of binarization for the binary naive Bayes algorithm.

Classificação de Sentimentos: Tratando Negação

Eu gosto desse filme

Eu não gosto desse filme

Negação muda o significado do “gosto” para a negação

Negação pode também mudar negativo para positivo

- Não descarte esse filme
- Não te deixa entendiado

Classificação de Sentimentos: Tratando Negação

Um método simples

Adicione NAO_ para toda palavra entre as negações e seguindo a pontuação:

Não gostei desse filme, mas eu

↓↓↓↓

NAO_gostei NAO_desse NAO_filme mas eu

- Das, Sanjiv and Mike Chen. 2001. Yahoo! for Amazon: Extracting market sentiment from stock message boards. In Proceedings of the Asia Pacific Finance Association Annual Conference (APFA)
- Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up? Sentiment Classification using Machine Learning Techniques. EMNLP-2002, 79—86.

Classificação de Sentimentos: Lexicons

Algumas vezes não temos dados de treinamento suficientes
Nesses casos, podemos usar listas pré-definidas chamadas **lexicons**
Existem várias listas de lexicons para análise de sentimentos disponíveis

MPQA Subjectivity Cues Lexicon

Theresa Wilson, Janyce Wiebe, and Paul Hoffmann (2005). Recognizing Contextual Polarity in Phrase-Level Sentiment Analysis. Proc. of HLT-EMNLP-2005.

Riloff and Wiebe (2003). Learning extraction patterns for subjective expressions. EMNLP-2003.

Home page:

https://mpqa.cs.pitt.edu/lexicons/subj_lexicon/
6885 palavras de 8221 lemas, anotadas por intensidade
(forte/fraca)

- 2718 positivas
- 4912 negativas

+ : admirable, beautiful, confident, dazzling, ecstatic, favor, glee, great

- : awful, bad, bias, catastrophe, cheat, deny, envious, foul, harsh, hate

The General Inquirer

Philip J. Stone, Dexter C Dunphy, Marshall S. Smith, Daniel M. Ogilvie. 1966. The General Inquirer: A Computer Approach to Content Analysis. MIT Press

- Home page: <http://www.wjh.harvard.edu/~inquirer>
- Lista de Categorias:
<http://www.wjh.harvard.edu/~inquirer/homecat.htm>
- Planilha:
<http://www.wjh.harvard.edu/~inquirer/inquirerbasic.xls>

Categorias:

- Positivo (1915 palavras) e Negativo (2291 palavras)
- Forte vs Fraca, Ativo vs Passivo, Exagerado vs discreto
- Prazer, Dor, Virtude, Vício, Motivação, Orientação Cognitiva, etc.

Livre para uso em pesquisa!

Usando Lexicons na Classificação de Sentimentos

Adicione características que contem as ocorrências dos léxicons

- Essa *feature* X conta a ocorrência das palavras que ocorrem nos lexicons positivos, Y conta a ocorrência das palavras que ocorrem nos lexicons negativos

Agora todas as palavras positivas ou negativas contam na determinada característica

Usar apenas 1-2 *feature* não é bom quanto usar todas as palavras

- Mas quando o conjunto de treinamento é esparso ou não representativo no conjunto de teste, lexicons densos pode ajudar

Naive Bayes em outras tarefas: Detectando Spam

Features da base *SpamAssassin*:

- Menções a dolar
- From: começa com muitos números
- Assunto está todo e maiúsculo
- HTML tem pouco texto e muita imagem
- “One hundred percent guaranteed”
- afirma que você pode ser removido da lista

Naive Bayes na Identificação de Linguagem

Determinar qual linguagem um trecho do texto é escrito
Características baseadas em n-gramas trazem bons resultados

Sumário: Naive Bayes is Not So Naive

Muito rápido, requer pouca memória

Trabalha bem com pouca quantidade de dados de treinamento

Robusto para características irrelevantes

- Características irrelevantes podem se cancelar sem afetar o resultado

Ótimo se a suposição de independência se mantém

Um bom baseline para classificador de texto

- Mas existem outros que dão melhores valores de acurácia

Naive Bayes como um modelo de linguagem

- O classificador Naive Bayes pode usar várias características do texto
 - URL, e-mail, dicionário
- Mas se
 - Usar todas as palavras como características
- Então
 - Naive Bayes tem uma similaridade importante com modelo de linguagem

Cada classe = Mode de Linguagem em Unigramas

- Atribua para cada palavra: $P(\text{palavra}|c)$
- Atribua cada sentença: $P(s|c) = \prod P(\text{palavra}, c)$

Class *pos*

0.1	I	<u>I</u>	<u>love</u>	<u>this</u>	<u>fun</u>	<u>film</u>
0.1	love	0.1	0.1	.05	0.01	0.1
0.01	this					
0.05	fun					
0.1	film					

$$P(s \mid \text{pos}) = 0.0000005$$

Naive Bayes como Modelo de Linguagem

- Qual classe atribui a maior probabilidade para s?

Model pos

0.1	I
0.1	love
0.01	this
0.05	fun
0.1	film

Model neg

0.2	I
0.001	love
0.01	this
0.005	fun
0.1	film

I	love	this	fun	film
0.1	0.1	0.01	0.05	0.1
0.2	0.001	0.01	0.005	0.1

$$P(s|\text{pos}) > P(s|\text{neg})$$

Avaliando o Classificador: Como dizer que o classificador está bom?

- Vamos primeiro considerar um classificador binário
 - Esse e-mail é spam?
spam (+) not spam (-)
 - Esse poste é sobre a Empresa X?
- Precisamos saber
 - 1 O que o nosso classificador diz sobre cada e-mail ou o poste?
 - 2 O que o classificador respondeu? A resposta deve ser igual ao rotulado pelo humano

Primeiro passo para avaliação: Matriz de Confusão?

		<i>gold standard labels</i>	
		gold positive	gold negative
<i>system output labels</i>	system positive	true positive	false positive
	system negative	false negative	true negative

$$acurácia = \frac{tp + tn}{tp + fp + tn + fn}$$

Por que não usar acurácia?

Acurácia não funciona bem quando estamos tratando com dados desbalanceados

Suponha que os 1000000 postes sobre a empresa X

- 100 falam bem
- 999900 falam mal

Imagine um classificador simples que diz: “fala mal”

Acurácia = $999900/1000000 = 99.99 \%$

Em vez de acurácia use precisão e recall

		<i>gold standard labels</i>		
		gold positive	gold negative	
<i>system output labels</i>	system positive	true positive	false positive	$\text{precision} = \frac{tp}{tp+fp}$
	system negative	false negative	true negative	
		$\text{recall} = \frac{tp}{tp+fn}$		$\text{accuracy} = \frac{tp+tn}{tp+fp+tn+fn}$

Precisão: % dos itens selecionados que estão corretos

Recall: % dos itens corretos que são selecionados

Precisão/Recall não te engana

Classificador estúpido: Apenas diz não: todo poste é “não sobre a empresa X”

- 100 postes falam sobre X, 999900 não falam
- Acurácia = $999900/1000000 = 99.99\%$

Mas o Recall e Precisão para esse classificador é terrível:

$$\text{Recall} = \frac{\text{true positive}}{\text{true positive} + \text{false positive}}$$

$$\text{Precisão} = \frac{\text{true positive}}{\text{true positive} + \text{false positives}}$$

Uma métrica combinando Precisão e Recall

Métrica F1-score

$$F1 = \frac{2 \times Precision}{Precision + Recall}$$

Casos especiais do F1

F-measure weighted

Média Harmônica (MH)

$$MH(a_1, a_2, a_3, \dots, a_n) = \frac{n}{\frac{1}{a_1} + \frac{1}{a_2} + \frac{1}{a_3} + \dots + \frac{1}{a_n}}$$

$$F = \frac{1}{\alpha \frac{1}{Precision} + (1 - \alpha) \frac{1}{Recall}}$$

Para $\beta^2 = \frac{1-\alpha}{\alpha}$

$$F = \frac{(\beta^2 + 1) Precision \times Recall}{\beta^2 \times Precision + Recall}$$

F1 é um caso especial da medida F com $\beta = 1, \alpha = \frac{1}{2}$

Suponha que tenhamos mais de duas classes

Vários problemas de classificação têm mais de duas classes
Podemos definir precisão e recall para múltiplas classes

		<i>gold labels</i>			
		urgent	normal	spam	
<i>system output</i>	urgent	8	10	1	$\text{precision}_u = \frac{8}{8+10+1}$
	normal	5	60	50	$\text{precision}_n = \frac{60}{5+60+50}$
	spam	3	30	200	$\text{precision}_s = \frac{200}{3+30+200}$
		$\text{recall}_u = \frac{8}{8+5+3}$	$\text{recall}_n = \frac{60}{10+60+30}$	$\text{recall}_s = \frac{200}{1+50+200}$	

Como combinar Precisão e Revocação

Class 1: Urgent

	true urgent	true not
system urgent	8	11
system not	8	340

$$\text{precision} = \frac{8}{8+11} = .42$$

Class 2: Normal

	true normal	true not
system normal	60	55
system not	40	212

$$\text{precision} = \frac{60}{60+55} = .52$$

Class 3: Spam

	true spam	true not
system spam	200	33
system not	51	83

$$\text{precision} = \frac{200}{200+33} = .86$$

Pooled

	true yes	true no
system yes	268	99
system no	99	635

$$\text{microaverage precision} = \frac{268}{268+99} = .73$$

$$\text{macroaverage precision} = \frac{.42+.52+.86}{3} = .60$$

Danos da classificação

- Classificadores, como qualquer algoritmo de NLP, pode ter viés
- Isso é verdade para qualquer classificador, se NB ou qualquer outro

Danos Representacional

- Viés causado por sistemas que degradam um grupo social
 - Perpetuação de estereótipo negativo
- Estudo de Kiritchenko e Mohammad 2018
 - Examinaram 200 sistemas de análise de sentimentos em pares de sentenças
 - Nomes Afro-Americano ou Americo-Europeu deram resultados diferentes
 - Like “I talked to Shaniqua yesterday” vs “I talked to Stephanie yesterday”
- Resultado: Sistema atribui sentimento mais baixo e emoções negativas para sentenças com nomes Afro-Americano
- Consequências:
 - Perpetuar estereótipos sobre Afro-Americanos
 - Afro-Americanos tratados diferentes pelas ferramentas de NLP como analisador de sentimentos (amplamente usados em marketing, estudos de saúde mental, etc.)

Ameaça de Censura

- Detecção de Toxidades é uma tarefa de classificação de texto de identificar discurso de ódio, abuso, ameaça, ou outra forma de linguagem tóxica.
 - Amplamente usada em moderação de conteúdo online
- Classificador de toxidades incorretamente identificam sentenças com menções simples a identidades minoritárias (como as palavras “cedo” ou “gay”)
 - Mulher (Park et al., 2018),
 - pessoas com deficiência (Hutchinson et al., 2020)
 - população LGBTQIAP+ (Dixon et al., 2018; Oliva et al., 2021)
- Consequências
 - Censurar discursos de grupos de pessoas
 - Discurso desses grupos se torna menos visíveis online
 - Escritos podem ser estimulados por esses algoritmos a evitar certas palavras, fazendo as pessoas escreverem menos sobre elas ou outros grupos

Disparidade de desempenho

- 1 O desempenho do classificador de texto é pior em várias linguagens devido a falta de dados rotulados
- 2 Classificadores são piores em português do que inglês

Viés em Classificação de texto

- Causa:
 - Problemas nos dados; Sistemas de NLP amplificam o viés de treinamento
 - Problemas nos rótulos
 - Problemas nos algoritmos (o que o modelo é treinado para otimizar)
- Prevalência: O mesmo problema ocorre em outras tarefas de NLP (incluindo os LLM's)
- Solução: Não existe uma forma solução geral de mitigação desses problemas
 - Mitigação desses riscos é uma área ativa e precisa de mais pesquisas
 - Existem conjuntos de dados e ferramentas que podem ser usadas para medir esses problemas