

Classificação de SPAM usando Naive Bayes e SVM

Marcos Treviso

07 de novembro de 2015

Aprendizado de Máquina - Universidade Federal do Pampa

- Multinomial Naive Bayes
- Ambiente de Teste
- Resultados
- Conclusão

- Multinomial Naive Bayes
- Ambiente de Teste
- Resultados
- Conclusão

Multinomial Naive Bayes

- Indicado para classificação de documentos
- Captura a frequência das palavras
- Assume independência entre palavras

Multinomial Naive Bayes

$$P(x|y) = \prod_{j=1}^V P(w_j|y)^{n_j(x)}$$

$$\hat{P}(y_k) = \frac{|J_k|}{M}, \quad \hat{P}(w_j|y_k) = \frac{\sum_{m \in J_k} n_j(x^m)}{\sum_{i=1}^V \sum_{m \in J_k} n_i(x^m)}$$

- Aplicando *log*: soma de probabilidades
- M = número de documentos
- V = tamanho do vocabulário
- x^i = i -ésimo documento
- y^i = classe de x^i
- J_k = índices das instâncias pertencendo a k -ésima classe
- $n_j(x)$ = número de ocorrências da palavra w_j no documento x

- Multinomial Naive Bayes
- Ambiente de Teste
- Resultados
- Conclusão

- Dell Inspiron 7348

Processador	Intel i7-5500U 2.40GHz - 4 cores
Memória Principal	8 GB RAM DDR3
Cache	L3 Cache 4 MB
Processador	Intel Xeon X5690 3.40GHz - 24 cores
Memória Principal	64 GB RAM DDR3
Cache	L3 Cache 12 MB

- Python 3.4.3
- Numpy 1.9.2

Ambiente de Teste

- Multinomial Naive Bayes:
 - *smooth* de 10^{-9}
 - Palavras em *lowercase*
 - Números para *NUMBER*
 - Endereços web para *URL*
 - Emails web para *EMAIL*
 - \$ web para *DOLLAR*
 - Remoção de HTML e pontuação
- Redes Neurais
 - Regularização de 0.1
 - Palavras em *lowercase*
 - Números para *NUMBER*
 - Endereços web para *URL*
 - Emails web para *EMAIL*
 - \$ web para *DOLLAR*
 - Remoção de HTML e pontuação

- Multinomial Naive Bayes
- Ambiente de Teste
- Resultados
- Conclusão

Resultados - Acurácia

- Média de 10 execuções

Alg.	Enron1		Enron2		Enron3	
	Treino	Teste	Treino	Teste	Treino	Teste
MNV	99.25%	98.56%	99.04%	98.64%	98.88%	98.53%
SVM	98.68%	98.12%	98.28%	97.87%	98.27%	97.94%

Alg.	Enron4		Enron5	
	Treino	Teste	Treino	Teste
MNV	99.12%	98.83%	99.33%	99.18%
SVM	98.26%	96.10%	97.44%	96.17%

Resultados - Pontuação F1

- Média de 10 execuções

Alg.	Enron1		Enron2		Enron3	
	Treino	Teste	Treino	Teste	Treino	Teste
MNV	0.99%	0.99%	0.98%	0.98%	0.98%	0.99 %
SVM	0.96%	0.95%	0.94%	0.91%	0.94%	0.89%

Alg.	Enron4		Enron5	
	Treino	Teste	Treino	Teste
MNV	0.99%	0.97%	0.98%	0.97%
SVM	0.97%	0.95%	0.97%	0.96%

- Multinomial Naive Bayes
- Ambiente de Teste
- Resultados
- Conclusão

Conclusão

- MNB parece ser muito bom para a tarefa
- MNB é bem mais rápido para treinar que o SVM
- A acurácia de ambos ficou boa
- Os filtros de pré-processamento ajudam a deixar o problema mais fácil
- O código do trabalho está em:
<<https://github.com/meitcher/spam-classification>>

Referências

- LxMLS - Lab Guide, Jul, 19, 2015. Lisboa, Portugal.
- Machine Learning, Andrew Ng. Coursera.

Classificação de SPAM usando Naive Bayes e SVM

Marcos Treviso

Aprendizado de Máquina

7 de novembro de 2015

Universidade Federal do Pampa