Universidade de São Paulo
Instituto de Matemática e Estatística
MAC 5739 - Inteligência Artificial

Exercício Programa 3: Aprendizagem Supervisionada

Autor:

Walter Perez Urcia

São Paulo

Novembro 2015

1 Dados de entrada

Para este exercício foi considerado um conjunto de dados com informação sobre sms spam[1]. O conjunto de dados tem 5559 instâncias e 2 características (ou atributos): Type (o tipo de sms: ham ou spam) e Text (o texto original do sms).

Da forma que o conjunto de dados se encontra, não é fácil usá-lo com um classificador porque o atributo Text é uma cadeia de texto muito diferente em cada instância do conjunto, mas poderia ser pré-processada de forma que seja um vetor de características como é definido a continuação:

- Term Frequency (TF): Número de vezes que cada palavra aparece no texto do sms
- Tf_Idf: Relevância da palavra na instância comparada com o conjunto total

$$Tf\_Idf(word) = TF(word) * log(N/DF(word))$$

onde N é o número total de instâncias. Então o tamanho do vetor de características vai depender do número total de palavras no conjunto, mas isso gera os seguintes problemas para fazer o pré-processamento:

- Palavras muito parecidas ou que são conjugações de outras (e.g. rise-rises, eat-ate)
- Muitas abreviações de palavras (e.g. u-you, vry-very)
- Palavras que aparecem en quase todos os textos ou em quase nenhum (também chamadas stop-words)

Para solucionar cada um dos problemas foram desenvolvidas as seguintes soluções:

- Foi usada a livraria NLTK (Python)[2] para fazer lemmatization (mudar a palavra a uma conjugação padrão) e stemming (encontrar a raiz da palavra)
- Foi usado um arquivo com sentenças equivalentes onde cada abreviação tinha um conjunto de palavras assignado[3]
- Foi usado um arquivo com as palavras que não dam muita informação sobre o texto, como pronomes ou interjeições[4]

Por último, os conjuntos de dados novos (com representação de vetor de características) foram divididos em duas partes: treinamento (70%) e validação (30%) para os experimentos das seguintes seções.

[1] Extraido de http://www.dt.fee.unicamp.br/~tiago/smsspamcollection/
[2] Official site: http://www.nltk.org
[3] O arquivo está em: https://github.com/NonWhite/IA_EP3/blob/master/data/equivalents.txt
[4] O arquivo está em: https://github.com/NonWhite/IA_EP3/blob/master/data/stop_words.txt

2 Configuração de experimentos

Para todos os experimentos foi usada a livraria scikit-learn (Python)[5]. Além disso, para cada classificador foram usadas as seguinte configurações:

- K-Nearest Neighbors
  - metric: euclidean, manhattan e hamming. Função da distancia entre vizinhos
  - k: 1, 5 , 10 e 50. Número de vizinhos a considerar por ponto
- Decision Tree
  - criterion: gini e entropy. Critério para seleção de atributos
  - depth: 10, 50 e 100. Máxima profundidade da árvore
- Naive Bayes
  - alpha: 1.0 , 1.0 , 100.0. Constante para Laplace smoothing

Os resultados para cada uma das configurações serão mostrados nas seções 3, 4 e 5 usando só o conjunto de treinamento com validação cruzada. Na seção 6, as melhores configurações serão usadas com ambos conjuntos (treinamento e validação) para encontrar o melhor classificador para o domínio.

3 Nearest Neighbors

Para comparar as diferentes configurações do experimento usamos a métrica F-Measure que é uma média armónica das métricas precisão (Precision) e cobertura (Recall). A tabela 1 mostra que a melhor configuração é aquela que usa distancia Euclidean e número de vizinhos igual a 1. Além disso, aquela configuração é a que tem o menor erro de todas. Em geral, pode-se ver que enquanto o valor K aumenta, a cobertura cae e portanto também F-Measure.

Por outro lado, usando a representação Tf-Idf temos que a melhor configuração também usa $K = 1$, mas a função de distancia é Hamming. O comportamento do algoritmo continua sendo o mesmo enquanto o valor de K aumenta.

[5]Official Site: http://scikit-learn.org/stable/

Metric	K	Precision	Recall	F-Measure	Mn. Abs. Error	Mn. Sqr. Error
Euclidean	1	0.99	0.70	0.82	0.041 ± 0.009	0.041 ± 0.009
	5	0.99	0.44	0.60	0.075 ± 0.013	0.075 ± 0.013
	10	1.00	0.26	0.41	0.0987 ± 0.0104	0.0987 ± 0.0104
	50	0.70	0.03	0.06	0.130 ± 0.004	0.130 ± 0.004
Manhattan	1	0.99	0.68	0.81	0.044 ± 0.007	0.044 ± 0.007
	5	1.00	0.41	0.58	0.079 ± 0.009	0.079 ± 0.009
	10	1.00	0.22	0.36	0.105 ± 0.009	0.105 ± 0.009
	50	0.50	0.02	0.04	0.131 ± 0.003	0.131 ± 0.003
Hamming	1	0.99	0.66	0.79	0.047 ± 0.007	0.047 ± 0.007
	5	1.00	0.39	0.56	0.082 ± 0.009	0.082 ± 0.009
	10	1.00	0.21	0.34	0.107 ± 0.006	0.107 ± 0.006
	50	0.50	0.02	0.04	0.131 ± 0.003	0.131 ± 0.003

Tabela 1: Comparação de K-Nearest Neighbor com Term Frequency

Metric	K	Precision	Recall	F-Measure	Mn. Abs. Error	Mn. Sqr. Error
Euclidean	1	0.99	0.64	0.77	0.049 ± 0.009	0.049 ± 0.009
	5	0.99	0.38	0.54	0.084 ± 0.008	0.084 ± 0.008
	10	1.00	0.17	0.29	0.111 ± 0.008	0.111 ± 0.008
Manhattan	1	0.99	0.62	0.76	0.051 ± 0.007	0.051 ± 0.007
	5	1.00	0.36	0.53	0.086 ± 0.007	0.086 ± 0.007
	10	1.00	0.16	0.27	0.113 ± 0.007	0.113 ± 0.007
Hamming	1	0.99	0.66	0.79	0.047 ± 0.007	0.047 ± 0.007
	5	1.00	0.39	0.56	0.082 ± 0.008	0.082 ± 0.008
	10	1.00	0.20	0.34	0.107 ± 0.006	0.107 ± 0.006

Tabela 2: Comparação de K-Nearest Neighbor com Tf-Idf

4 Árvores de decisão

Da mesma forma que fizemos a comparação das configurações na seção anterior, a melhor configuração será aquela que tenha o maior valor de F-measure. Por exemplo, na Tabela 3 mostra que usando como criterio de escolha de atributos Entropy e máxima profundidade 100 obtemos um melhor modelo de classificação.

Por outro lado, se usamos a representação Tf-idf, existem tres configurações que tem o maior valor F-measure, mas escolhemos aquela que tem o menor erro médio. Então a melhor configuração para árvore de decisão é criterio Entropy e máxima profundidade 50.

Criterion	Depth	Precision	Recall	F-Measure	Mn. Abs. Error	Mn. Sqr. Error
Gini	10	0.92	0.74	0.82	0.044 ± 0.009	0.044 ± 0.008
	50	0.89	0.82	0.84	0.039 ± 0.008	0.039 ± 0.008
	100	0.89	0.82	0.85	0.039 ± 0.009	0.039 ± 0.009
Entropy	10	0.94	0.70	0.81	0.044 ± 0.008	0.044 ± 0.007
	50	0.91	0.79	0.84	0.039 ± 0.008	0.038 ± 0.007
	100	0.91	0.80	0.86	0.037 ± 0.007	0.038 ± 0.008

Tabela 3: Comparação de Decision Tree com Term Frequency

Criterion	Depth	Precision	Recall	F-Measure	Mn. Abs. Error	Mn. Sqr. Error
Gini	10	0.92	0.73	0.81	0.043 ± 0.008	0.043 ± 0.008
	50	0.90	0.81	0.85	0.038 ± 0.007	0.037 ± 0.008
	100	0.88	0.81	0.84	0.038 ± 0.008	0.038 ± 0.010
Entropy	10	0.94	0.70	0.80	0.046 ± 0.006	0.046 ± 0.008
	50	0.91	0.79	0.85	0.037 ± 0.008	0.038 ± 0.005
	100	0.91	0.80	0.85	0.039 ± 0.007	0.038 ± 0.008

Tabela 4: Comparação de Decision Tree com Tf-Idf

5 Naive Bayes

Os resultados na Tabela 5 mostram que aumentar o valor alpha acrescenta a precisão mas diminui a cobertura e portanto também F-measure. Então a melhor configuração usando a representação de frequência de palavras usa alpha igual a 1.

Alpha	Precision	Recall	F-Measure	Mn. Abs. Error	Mn. Sqr. Error
1	0.87	0.91	0.89	0.031 ± 0.009	0.031 ± 0.009
10	0.99	0.72	0.83	0.038 ± 0.009	0.038 ± 0.009
100	1.00	0.14	0.24	0.116 ± 0.005	0.116 ± 0.005

Tabela 5: Comparação de Naive Bayes com Term Frequency

Na outra representação, o comportamento do classificador e similar para precisão e cobertura, mas em geral F-measure aumenta porque as mudanças de valores entre precisão e cobertura não são tão significativas como no caso anterior. Portanto a melhor configuração para Tf-Idf usa alpha igual a 100.

6 Validação

Para validar que tão bem classificam novas instâncias, os modelos serão testados usando o conjunto de treinamento usando as melhores configurações encontradas nas seções anteriores. Os resultados mostram que para a representação com frequência de palavras usar

Alpha	Precision	Recall	F-Measure	Mn. Abs. Error	Mn. Sqr. Error
1	0.61	0.96	0.75	0.089 ± 0.021	0.089 ± 0.021
10	0.66	0.94	0.77	0.075 ± 0.018	0.075 ± 0.018
100	0.88	0.81	0.85	0.040 ± 0.007	0.040 ± 0.007

Tabela 6: Comparação de Naive Bayes com Tf-Idf

o classificador Naive Bayes ou Decision Tree é quase igual, mas o primeiro tem melhor precisão para o conjunto de dados. Por outro lado, se tf-idf é usado, a tabela 8 mostra que usar Naive Bayes ou Decision Tree é indistinto.

Classifier	Precision	Recall	F-Measure	Mn. Abs. Error	Mn. Sqr. Error
Nearest Neighbor	0.96	0.96	0.96	0.040	0.040
Decision Tree	0.96	0.97	0.97	0.028	0.028
Naive Bayes	0.97	0.97	0.97	0.028	0.028

Tabela 7: Comparação de classificadores com Term Frequency

Classifier	Precision	Recall	F-Measure	Mn. Abs. Error	Mn. Sqr. Error
Nearest Neighbor	0.96	0.96	0.95	0.043	0.043
Decision Tree	0.96	0.96	0.96	0.040	0.040
Naive Bayes	0.96	0.96	0.96	0.040	0.040

Tabela 8: Comparação de classificadores com Tf-Idf

Por último, todos os classificadores obtiveram melhores valores para as métricas comparado com as retornadas usando validação cruzada só no conjunto de treinamento. Isto pode ser explicado porque fazendo validação cruzada a proporção das classes não sempre ficou igual entre as instâncias usadas para treinar e validar.

7 Conclusões

Com os resultados dos experimentos pode-se concluir que o melhor classificador para o domínio de sms spam é Naive Bayes para ambas representações usadas. A possível razão que aconteça isso é que as características representem bastante bem cada classe usando probabilidades, mas poderia ser comprovado usando alguma outra representação de vetor de características. Além disso, a comparação feita é confiável porque apesar que os classificadores usados fazem coisas diferentes, as métricas usadas são padrão na área de aprendizagem de máquina e portanto podem ser comparadas.