Naive Bayes Spam Filtering Using Word-Position-Based Attributes

Johan Hovold

Dejun Qian

Outline

Introduction

- The problem of spam gets worse every year
  - Waste resources on the Internet
  - Waste time for user
  - May expose children to unsuitable contents (e.g. pronography)

Introduction

- The problem of spam gets worse every year
  - Waste resources on the Internet
  - Waste time for user
  - May expose children to unsuitable contents (e.g. pronography)

- The solution to the spam problem
  - Automatic spam filter

Techniques for Spam Filters

- Instances of knowledge engineering
  - Hand-crafted rules (e.g. presence of string "buy now" indicates spam)

Techniques for Spam Filters

- Instances of knowledge engineering
  - Hand-crafted rules (e.g. presence of string "buy now" indicates spam)
- Machine learning
  - Supervised learning algorithm is presented with a mailbox and outputs a filter

Techniques for Spam Filters

- Instances of knowledge engineering
  - Hand-crafted rules (e.g. presence of string "buy now" indicates spam)
- Machine learning
  - Supervised learning algorithm is presented with a mailbox and outputs a filter
- Naive Bayes classifier in this paper

Naive Bayes Using Word-Position-Based Attributes

- NB: $C_{NB} = \arg\max_{c \in C} P(c) \prod_i P(a_i|c)$

Naive Bayes Using Word-Position-Based Attributes

- NB: $C_{NB} = \arg\max_{c \in C} P(c) \prod_i P(a_i|c)$
- One attribute for each word position in a document

Naive Bayes Using Word-Position-Based Attributes

- NB: $C_{NB} = \arg\max_{c \in C} P(c) \prod_i P(a_i|c)$
- One attribute for each word position in a document
- Probability of certain word w_k at position i, given target c_j: $P(a_i = w_k|c_j)$

Naive Bayes Using Word-Position-Based Attributes

- NB: $C_{NB} = \arg\max_{c \in C} P(c) \prod_i P(a_i|c)$
- One attribute for each word position in a document
- Probability of certain word w_k at position i, given target c_j: $P(a_i = w_k|c_j)$
- Sparseness: $P(a_i = w_k|c_j) = P(a_m = w_k|c_j)$

Naive Bayes Using Word-Position-Based Attributes

- NB: $C_{NB} = \arg\max_{c \in C} P(c) \prod_i P(a_i|c)$
- One attribute for each word position in a document
- Probability of certain word w_k at position i, given target c_j: $P(a_i = w_k|c_j)$
- Sparseness: $P(a_i = w_k|c_j) = P(a_m = w_k|c_j)$
- Estimate $P(a_i = w_k|c_j)$ with $P(w_k|c_j)$

Naive Bayes Using Word-Position-Based Attributes

- NB: $C_{NB} = \arg\max_{c \in C} P(c) \prod_i P(a_i|c)$
- One attribute for each word position in a document
- Probability of certain word w_k at position i, given target c_j: $P(a_i = w_k|c_j)$
- Sparseness: $P(a_i = w_k|c_j) = P(a_m = w_k|c_j)$
- Estimate $P(a_i = w_k|c_j)$ with $P(w_k|c_j)$
- $P(w_k|c_j) = \frac{C_j(w_k)+1}{n_j+|Vocabulary|}$

Benchmark Corpora

corpus	messages	spam ratio
PU1	1099	44%
PU2	721	20%
PU3	4139	44%
PUA	1142	50%
SA	6047	31%

- PU corpora[1]
- SpamAssassin corpus[2]

[1] `http://www.iit.demokritos.gr/skel/i-config/`
[2] `http://spamassassin.org/publiccorpus/`

Attribute Selection - Infrequent

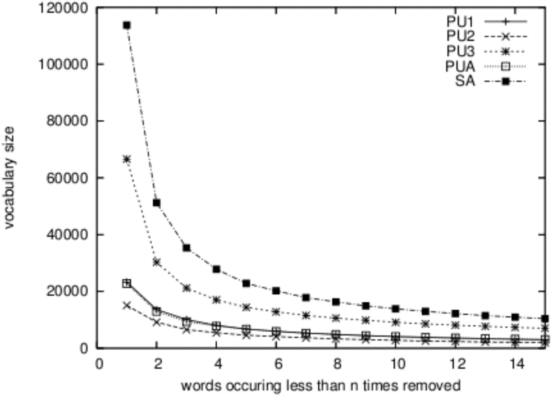

Figure 1: Impact on vocabulary size when removing infrequent words (from nine tenths of each corpora).

- Slightly increased precision at the expense of slightly reduced recall as n grew

Attribute Selection - Frequent

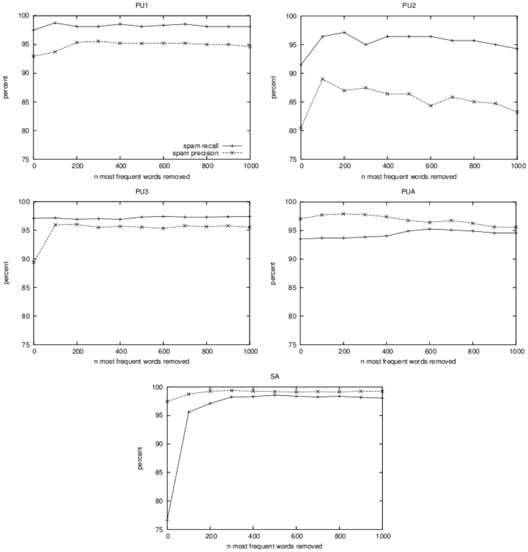

Figure 2: Impact on spam precision and recall when removing the most frequent words.

n-grams

Table 2: Comparison of classification results when using only unigram attributes and uni-, bi- and trigram attributes, respectively. In the experiment n-grams occurring less than three times and the 200 most frequent n-grams were removed. The second n-gram row for the SA corpus shows the result when the 5000 most frequent n-grams were removed.

n-grams	R	P	Acc
PU1			
$n = 1$	98.12	95.35	97.06
$n = 1, 2, 3$	99.17	96.19	97.89
PU2			
$n = 1$	97.14	87.00	96.20
$n = 1, 2, 3$	95.00	93.12	96.90
PU3			
$n = 1$	96.92	96.02	96.83
$n = 1, 2, 3$	96.59	97.83	97.53
PUA			
$n = 1$	93.68	97.91	95.79
$n = 1, 2, 3$	94.74	97.75	96.23
SA			
$n = 1$	97.12	99.25	98.95
$n = 1, 2, 3$	92.26	98.70	97.42
$n = 1, 2, 3$	98.46	99.66	99.46

Cost-Sensitive Classification

$$\frac{P(spam|d)}{P(legit|d)} > \lambda,$$

$$\frac{P(spam|d)}{P(legit|d)} > w^{|d|},$$

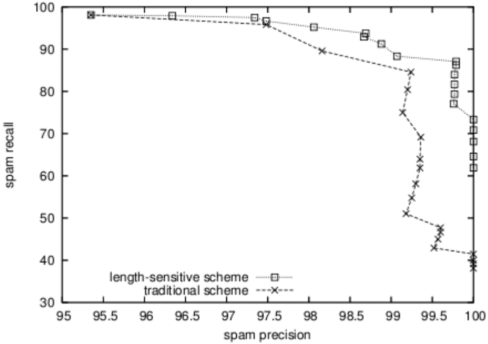

Figure 4: Example recall/precision curves of the two weighting schemes from cost-sensitive classification on the PU1 corpus.

Conclusion

- Possible to achieve very good classification performance using a word-position-based variant of naive Bayes
- Attribute selection has been stressed: memory requirements may be lowered and classification performance increased
- By extending the attribute set with bi- and trigrams, better classification performance may be achieved
- Simple weighting scheme boost precision further

Question?