Feature Selection

September 8, 2024

Why feature selection

In text classification, feature selection is typically used to achieve two objectives:

- Reduce the size of the feature set
    - in order to optimize the use of computing resources and to
- Remove noise from the data
    - in order to optimize the classification performance.

Common feature selection methods for both supervised and unsupervised applications

- Stop-word removal
  - we determine the common words in the documents which are not specific or discriminatory to the different classes.
- Stemming, different forms of the same word are consolidated into a single word.
  - singular, plural and different tenses are consolidated into a single word.
- Features are often scored and ranked using some feature weighting scheme that reflects the importance of the feature for a given task
- These methods are not specific to the case of the classification problem,
- Often used in a variety of unsupervised applications such as clustering and indexing.

Feature selection

- How to represent documents for text classification?
- Option 1: represent documents with all the terms (recall the term-document matrix)
    - Very high-dimensional space, with each dimension corresponding to a term.
    - Many dimensions correspond to rare words.
    - Rare words can mislead the classifier.
    - Rare misleading features are called noise features.
    - Very common words may not be good as well.
- Eliminating noise features from the representation increases efficiency and effectiveness of text classification.
- Eliminating features is called feature selection.

Example for a noise feature

- Let's say we're doing text classification for the class *China*.
- Suppose a rare term, say ARACHNOCENTRIC, has no information about *China* . . .
- . . . but all instances of ARACHNOCENTRIC happen to occur in *China* documents in our training set.
- Then we may learn a classifier that incorrectly interprets ARACHNOCENTRIC as evidence for the class *China*.
- Such an incorrect generalization from an accidental property of the training set is called overfitting.
- Feature selection reduces overfitting and improves the accuracy of the classifier.

Basic feature selection algorithm

```
SelectFeatures($\mathbb{D}$, $c$, $k$)
  $V \leftarrow$ ExtractVocabulary($\mathbb{D}$)
  $L \leftarrow []$
  for each $t \in V$
  do $A(t,c) \leftarrow$ ComputeFeatureUtility($\mathbb{D}$, $t$, $c$)
      Append($L$, $\langle A(t,c), t \rangle$)
  return FeaturesWithLargestValues($L$, $k$)
```

Basic feature selection algorithm

```
SelectFeatures($\mathbb{D}, c, k$)
    $V \leftarrow$ ExtractVocabulary($\mathbb{D}$)
    $L \leftarrow []$
    for each $t \in V$
    do $A(t, c) \leftarrow$ ComputeFeatureUtility($\mathbb{D}, t, c$)
        Append($L, \langle A(t, c), t \rangle$)
    return FeaturesWithLargestValues($L, k$)
```

How do we compute A, the feature utility?

Different feature selection methods

- A feature selection method is mainly defined by the feature utility measure it employs
- Feature utility measures:
  - Frequency – select the most frequent terms
  - Mutual information – select the terms with the highest mutual information
  - Mutual information is also called information gain in this context.
  - Chi-square (see book)
-
-
-

Feature functions

- These functions capture the intuition that the best terms for c_i are the ones distributed most differently in the sets of positive and negative examples of c_i.
- interpretations of this principle vary across different functions.
- χ^2 and MI: measure how the results of an observation differ (i.e. are independent) from the results expected according to an initial hypothesis

Mutual information

- Compute the feature utility $A(t, c)$ as the mutual information (MI) of term t and class c.
- MI tells us “how much information” the term contains about the class and vice versa.
- For example, if a term’s occurrence is independent of the class (same proportion of docs within/without class contain the term), then MI is 0.
- Starting point: PMI (point-wise mutual information)

PMI

- Definition of PMI

$$PMI(t, c) = \log \frac{N_{tc}}{\hat{N}_{tc}} \tag{1}$$

- N_{tc}: observed count of term t in class c.
- $\hat{N}_{tc}$: expected count if t is random.
- When $\hat{N}_{tc} = N_{tc}$, t is independent of c, hence MI=0.
- How to estimate $\hat{N}_{tc}$?
- By the MLE estimator,

$$\hat{N}_{tc} = \frac{N_t N_c}{N} \tag{2}$$

  - N_t: total count of term t (document frequency of t)
  - N_c: documents in class c.
  - N: total number of documents.

MI

- Definition:

$$I(U;C)=\sum_{e_t\in\{1,0\}}\sum_{e_c\in\{1,0\}}P(U=e_t,C=e_c)\log_2\frac{P(U=e_t,C=e_c)}{P(U=e_t)P(C=e_c)}$$

- Based on maximum likelihood estimates, the formula we actually use is:

$$\begin{aligned} I(U;C) &= \frac{N_{11}}{N}\log_2\frac{NN_{11}}{N_{1.}N_{.1}}+\frac{N_{01}}{N}\log_2\frac{NN_{01}}{N_{0.}N_{.1}} \\ &\quad +\frac{N_{10}}{N}\log_2\frac{NN_{10}}{N_{1.}N_{.0}}+\frac{N_{00}}{N}\log_2\frac{NN_{00}}{N_{0.}N_{.0}} \end{aligned}$$

- N_{xy} denote the number of docs that
  - N_{10}: contain t ($e_t = 1$) and are not in c ($e_c = 0$);
  - N_{11}: contain t ($e_t = 1$) and are in c ($e_c = 1$);
  - N_{01}: do not contain t ($e_t = 0$) and are in c ($e_c = 1$);
  - N_{00}: do not contain t ($e_t = 0$) and are not in c ($e_c = 0$);
- $N = N_{00} + N_{01} + N_{10} + N_{11}$.

	Observed			Expected	
	poultry	not poultry	SUM	poultry	no poultry
export	49	27652	27701	6.56	27694.43
no export	141	774106	774247	183.43	774063.56
sum	190	801758	801948		

For 'poultry' class,

$$\hat{export} = \frac{190 * 27701}{801948} \approx 6.56 \tag{3}$$

mutual information intermediate data:

	P(tc)	Obs/Expected	
11	6.11012E-05	7.466090337 2.900352965	0.000177215
10	0.000175822	0.768656296 -0.379589451	-6.67401E-05
01	0.034481039	0.998467671 -0.002212379	-7.62851E-05
00	0.965282038	1.000054824 7.90916E-05	7.63457E-05
sum			0.000110536

How to compute MI values (2)

- Alternative way of understanding MI:

$$I(U;C) = \sum_{e_t \in \{1,0\}} \sum_{e_c \in \{1,0\}} P(U = e_t, C = e_c) \log_2 \frac{N(U = e_t, C = e_c)}{E(U = e_t, C = e_c)}$$

- $N(U = e_t, C = e_c)$ is the count of documents with values e_t and e_c .
- $E(U = e_t, C = e_c)$ is the expected count of documents with values e_t and e_c if we assume that the two random variables are independent.

MI example for *poultry*/EXPORT in Reuters

	$e_c = e_{poultry} = 1$	$e_c = e_{poultry} = 0$
$e_t = e_{\text{EXPORT}} = 1$	$N_{11} = 49$	$N_{10} = 27{,}652$
$e_t = e_{\text{EXPORT}} = 0$	$N_{01} = 141$	$N_{00} = 774{,}106$

Plug these values into formula:

$$
\begin{aligned}
I(U;C) &= \frac{49}{801{,}948}\log_2\frac{801{,}948\cdot 49}{(49+27{,}652)(49+141)}\\
&\quad+\frac{141}{801{,}948}\log_2\frac{801{,}948\cdot 141}{(141+774{,}106)(49+141)}\\
&\quad+\frac{27{,}652}{801{,}948}\log_2\frac{801{,}948\cdot 27{,}652}{(49+27{,}652)(27{,}652+774{,}106)}\\
&\quad+\frac{774{,}106}{801{,}948}\log_2\frac{801{,}948\cdot 774{,}106}{(141+774{,}106)(27{,}652+774{,}106)}\\
&\approx 0.000105
\end{aligned}
$$

MI feature selection on Reuters

Class: *coffee*

term	MI
COFFEE	0.0111
BAGS	0.0042
GROWERS	0.0025
KG	0.0019
COLOMBIA	0.0018
BRAZIL	0.0016
EXPORT	0.0014
EXPORTERS	0.0013
EXPORTS	0.0013
CROP	0.0012

Class: *sports*

term	MI
SOCCER	0.0681
CUP	0.0515
MATCH	0.0441
MATCHES	0.0408
PLAYED	0.0388
LEAGUE	0.0386
BEAT	0.0301
GAME	0.0299
GAMES	0.0284
TEAM	0.0264

χ^2

$$\chi^2 = \sum_{e_t \in 0,1} \sum_{e_c \in 0,1} \frac{(N_{e_t e_c} - E_{e_t e_c})^2}{E_{e_t e_c}} \tag{4}$$

	Observed			Expected	
	poultry	not poultry	SUM	poultry	no poultry
export	49	27652	27701	6.56	27694.43
no export	141	774106	774247	183.43	774063.56
sum	190	801758	801948		

chi^2 for term *export* and class *poultry*:

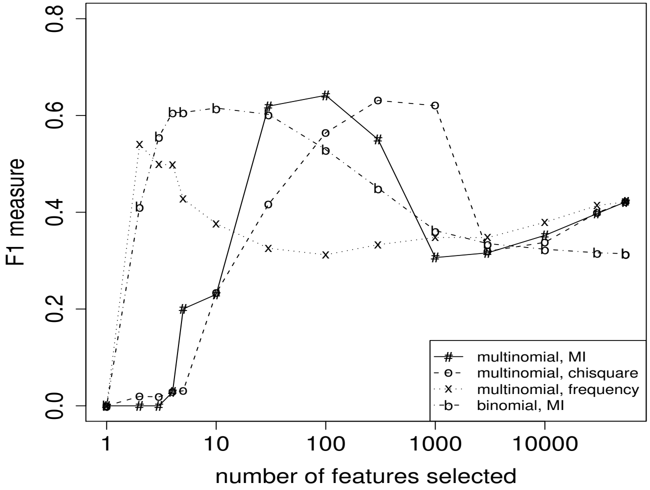
$$\chi^2 = \frac{(49 - 6.56)^2}{6.56} + \frac{(141 - 183)^2}{183} + \dots \tag{5}$$

$$\approx 274.4 + 9.8 + \dots \tag{6}$$

$$= 284.2 \tag{7}$$

observed - expected		
	poultry	no poultry
export	42.43699342	(42.44)
not export	(42.43699342)	42.44
square/expected		
	poultry	no poultry
export	274.40143308	0.06502744
not export	9.81753122	0.00232655
sum		284.28631830

Naive Bayes: Effect of feature selection



multinomial = multinomial Naive Bayes
binomial = Bernoulli Naive Bayes

Feature selection for Naive Bayes

- In general, feature selection is necessary for Naive Bayes to get decent performance.
- Also true for many other learning methods in text classification: you need feature selection for optimal performance.

Exercise

- Compute the "export"/POULTRY contingency table for the "Kyoto"/JAPAN in the collection given below.
- Make up a contingency table for which MI is 0 – that is, term and class are independent of each other.

"export"/POULTRY table:

	$e_c = e_{poultry} = 1$	$e_c = e_{poultry} = 0$
$e_t = e_{\text{EXPORT}} = 1$	$N_{11} = 49$	$N_{10} = 27{,}652$
$e_t = e_{\text{EXPORT}} = 0$	$N_{01} = 141$	$N_{00} = 774{,}106$

Collection:

	docID	words in document	in c = Japan?
training set	1	Kyoto Osaka Taiwan	yes
	2	Japan Kyoto	yes
	3	Taipei Taiwan	no
	4	Macao Taiwan Shanghai	no
	5	London	no

Feature Transformation Methods: Supervised LSI

- Feature selection: reduce the dimensionality of the data by picking from the original set of attributes,
- Feature transformation: create a new (and smaller) set of features as a function of the original set of features.
- Typical examples of feature transformation methods
    - Latent Semantic Indexing (LSI), and its probabilistic variant PLSA .
- LSI method transforms the text space of a few hundred thousand word features to a new axis system
- Principal Component Analysis techniques are used to determine the axis-system which retains the greatest level of information about the variations in the underlying attribute values.
- Disadvantage: unsupervised, blind to the underlying class distribution.

- Reading: P251-P265. IIR
- References: