CS317
Information Retrieval
Week 11

Muhammad Rafi
April 08, 2020

Text Classification

Chapter No. 13

Agenda

- Ad hoc vs. Standing Query
- Text/Document Classification
- Naïve Bayes Classification
- Variations of Naïve Bayes Classification
- Evaluation of Classification
- Classification model & Accuracy
- Feature Selection
  - Mutual Information
  - Chi Square Method
  - Frequency based feature
- Conclusion

Agenda

- Vector Space Text Classification
- Rocchio’s Algorithm
- KNN algorithm
- Conclusion

Ad hoc vs. Standing Query

- Ad hoc Query
  - Ad hoc retrieval, where users have transient information needs that they try to address by posing one or more queries to a search engine.
- Standing Query
  - A standing query is like any other query except that it is periodically executed on a collection to which new documents are incrementally added over time.

Text/Document Classification

- Text/Document classification is the process of assigning a predefine set of classes to the new instance of text/document.
- *Input*:
  - a text piece/ a document d
  - a fixed set of classes $C = \{c_1, c_2, \ldots, c_J\}$
- *Output*: a predicted class $c_i \in C$

Text/Document Classification

- Manual
- Supervised
- Semi-supervised
- Unsupervised

Text/Document Classification

- Manual Hand coded rules
  - Rules based on combinations of words or other features
    - spam: black-list-address OR (“dollars” AND“have been selected”)
  - Accuracy can be high
    - If rules carefully refined by expert
  - But building and maintaining these rules is expensive

Text/Document Classification

- Supervised
  - *Input:*
    - a document d
    - a fixed set of classes $C = \{c_1, c_2, \ldots, c_J\}$
    - A training set of m hand-labeled documents $(d_1,c_1),\ldots,(d_m,c_m)$
  - *Output:*
    - a learned classifier $\gamma: d \rightarrow c$

Text/Document Classification

- Semi-supervised
  - Started as supervised
  - Tune in such a way that it will continue to learn from the functionality of its works <classification>

Text/Document Classification

- Unsupervised
  - It will learn of it own how to classify the text/doc by implicitly learned the features.

Naïve Bayes Classification

- The first supervised learning method we introduce is the multinomial Naive Bayes or multinomial NB model, a probabilistic learning method.
- The probability of a document d being in class c is computed as

$$P(c|d) \propto P(c) \prod_{1 \le k \le n_d} P(t_k|c)$$

Naïve Bayes Classifier

$$c_{MAP} = \underset{c \in C}{\operatorname{argmax}}\, P(c \mid d)$$

MAP is "maximum a posteriori" = most likely class

$$= \underset{c \in C}{\operatorname{argmax}} \frac{P(d \mid c)P(c)}{P(d)}$$

Bayes Rule

$$= \underset{c \in C}{\operatorname{argmax}}\, P(d \mid c)P(c)$$

Dropping the denominator

Naïve Bayes Classification

- First attempt: maximum likelihood estimates
  - simply use the frequencies in the data

$$\hat{P}(c_j) = \frac{doccount(C = c_j)}{N_{doc}}$$

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

Naïve Bayes Classification

$$\hat{P}(w_i \mid c) = \frac{count(w_i, c) + 1}{\sum_{w \in V} \left(count(w, c) + 1 \right)}$$

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

Multinomial NB

- Training

```
TrainMultinomialNB(ℂ, 𝔻)
V ← ExtractVocabulary(𝔻)
N ← CountDocs(𝔻)
for each c ∈ ℂ
do N_c ← CountDocsInClass(𝔻, c)
    prior[c] ← N_c / N
    text_c ← ConcatenateTextOfAllDocsInClass(𝔻, c)
    for each t ∈ V
    do T_ct ← CountTokensOfTerm(text_c, t)
    for each t ∈ V
    do condprob[t][c] ← (T_ct + 1) / Σ_t' (T_ct' + 1)
return V, prior, condprob
```

Multinomial NB

- Testing

```
APPLYMULTINOMIALNB(ℂ, V, prior, condprob, d)
W ← EXTRACTTOKENSFROMDOC(V, d)
for each c ∈ ℂ
do score[c] ← log prior[c]
    for each t ∈ W
    do score[c] += log condprob[t][c]
return arg max_{c∈ℂ} score[c]
```

Example

$$\hat{P}(c) = \frac{N_c}{N}$$

$$\hat{P}(w \mid c) = \frac{count(w,c)+1}{count(c)+|V|}$$

	Doc	Words	Class
Training	1	Chinese Beijing Chinese	c
	2	Chinese Chinese Shanghai	c
	3	Chinese Macao	c
	4	Tokyo Japan Chinese	j
Test	5	Chinese Chinese Chinese Tokyo Japan	?

Priors:
P(*c*)= 3/4
P(*j*)= 1/4

Conditional Probabilities:
P(Chinese|*c*) = (5+1) / (8+6) = 6/14 = 3/7
P(Tokyo|*c*) = (0+1) / (8+6) = 1/14
P(Japan|*c*) = (0+1) / (8+6) = 1/14
P(Chinese|*j*) = (1+1) / (3+6) = 2/9
P(Tokyo|*j*) = (1+1) / (3+6) = 2/9
P(Japan|*j*) = (1+1) / (3+6) = 2/9

Choosing a class:
P(c|d5) $= 3/4 * (3/7)^3 * 1/14 * 1/14 \approx 0.0003$

P(j|d5) $= 1/4 * (2/9)^3 * 2/9 * 2/9 \approx 0.0001$

Variations of Naïve Bayesian

- There are two ways in which we can setup NB Classifiers
  - Multinomial NB, that we discussed along with an example
  - Multivariate Bernoulli model

Bernoulli model

- Training of Model

```
TrainBernoulliNB(ℂ, 𝔻)
V ← ExtractVocabulary(𝔻)
N ← CountDocs(𝔻)
for each c ∈ ℂ
do N_c ← CountDocsInClass(𝔻, c)
    prior[c] ← N_c/N
    for each t ∈ V
    do N_ct ← CountDocsInClassContainingTerm(𝔻, c, t)
        condprob[t][c] ← (N_ct + 1)/(N_c + 2)
return V, prior, condprob
```

Bernoulli model

- Testing

```
ApplyBernoulliNB(ℂ, V, prior, condprob, d)
V_d ← ExtractTermsFromDoc(V, d)
for each c ∈ ℂ
do score[c] ← log prior[c]
    for each t ∈ V
    do if t ∈ V_d
          then score[c] += log condprob[t][c]
          else score[c] += log(1 − condprob[t][c])
return arg max_{c∈ℂ} score[c]
```

Example

	Doc	Words	Class
Training	1	Chinese Beijing Chinese	c
	2	Chinese Chinese Shanghai	c
	3	Chinese Macao	c
	4	Tokyo Japan Chinese	j
Test	5	Chinese Chinese Chinese Tokyo Japan	?

Priors:

$P(c)$= 3/4

$P(j)$=1/4

Conditional Probabilities:

The conditional probabilities are:

$$\hat{P}(\text{Chinese}|c) = (3+1)/(3+2) = 4/5$$
$$\hat{P}(\text{Japan}|c) = \hat{P}(\text{Tokyo}|c) = (0+1)/(3+2) = 1/5$$
$$\hat{P}(\text{Beijing}|c) = \hat{P}(\text{Macao}|c) = \hat{P}(\text{Shanghai}|c) = (1+1)/(3+2) = 2/5$$
$$\hat{P}(\text{Chinese}|\overline{c}) = (1+1)/(1+2) = 2/3$$
$$\hat{P}(\text{Japan}|\overline{c}) = \hat{P}(\text{Tokyo}|\overline{c}) = (1+1)/(1+2) = 2/3$$
$$\hat{P}(\text{Beijing}|\overline{c}) = \hat{P}(\text{Macao}|\overline{c}) = \hat{P}(\text{Shanghai}|\overline{c}) = (0+1)/(1+2) = 1/3$$

Choosing a class:

$$\begin{aligned}\hat{P}(c|d_5) &\propto \hat{P}(c) \cdot \hat{P}(\text{Chinese}|c) \cdot \hat{P}(\text{Japan}|c) \cdot \hat{P}(\text{Tokyo}|c) \\ &\quad \cdot (1 - \hat{P}(\text{Beijing}|c)) \cdot (1 - \hat{P}(\text{Shanghai}|c)) \cdot (1 - \hat{P}(\text{Macao}|c)) \\ &= 3/4 \cdot 4/5 \cdot 1/5 \cdot 1/5 \cdot (1-2/5) \cdot (1-2/5) \cdot (1-2/5) \\ &\approx 0.005\end{aligned}$$

$$\begin{aligned}\hat{P}(\overline{c}|d_5) &\propto 1/4 \cdot 2/3 \cdot 2/3 \cdot 2/3 \cdot (1-1/3) \cdot (1-1/3) \cdot (1-1/3) \\ &\approx 0.022\end{aligned}$$

Multinomial Vs. Bernoulli

Table 13.3 Multinomial versus Bernoulli model.

	multinomial model	Bernoulli model
event model	generation of token	generation of document
random variable(s)	$X = t$ iff t occurs at given pos	$U_t = 1$ iff t occurs in doc
document representation	$d = \langle t_1, \ldots, t_k, \ldots, t_{n_d} \rangle, t_k \in V$	$d = \langle e_1, \ldots, e_i, \ldots, e_M \rangle,$ $e_i \in \{0, 1\}$
parameter estimation	$\hat{P}(X = t \mid c)$	$\hat{P}(U_i = e \mid c)$
decision rule: maximize	$\hat{P}(c) \prod_{1 \le k \le n_d} \hat{P}(X = t_k \mid c)$	$\hat{P}(c) \prod_{t_i \in V} \hat{P}(U_i = e_i \mid c)$
multiple occurrences	taken into account	ignored
length of docs	can handle longer docs	works best for short docs
# features	can handle more	works best with fewer
estimate for term the	$\hat{P}(X = \text{the} \mid c) \approx 0.05$	$\hat{P}(U_{\text{the}} = 1 \mid c) \approx 1.0$

Evaluation of Classification Task

- Lets start our discussion on this with e-mail classification-simple binary classification.
- There are two classes Spam/Ham
- The task of classification produced a matrix of its work-called contingency matrix or confusion matrix
- Lets our e-mail training dataset contains 120 e-mails and 70 Hams and 50 Spams

Evaluation of Classification Task

	Spam	Ham
Predicted Spam	35	16
Predicted Ham	15	54

- Accuracy
  - Fraction of correctly classified items
    - (35+54)/120
- Error
  - Fraction of error in classification
    - (15+16)/120

Evaluation of Classification Task

- Multiclass Classification

	C1	C2	C3	C4	C5
Pre C1	10	1	0	1	0
Pre C2	2	22	1	1	1
Pre C3	0	2	20	2	2
Pre C4	2	0	12	21	0
Pre C5	0	0	0	0	25

Evaluation of Classification Task

Recall:

Fraction of docs in class *i* classified correctly: $\dfrac{c_{ii}}{\sum_j c_{ij}}$

Precision:

Fraction of docs assigned class *i* that are actually about class *i*: $\dfrac{c_{ii}}{\sum_j c_{ji}}$

Accuracy: (1 - error rate)

Fraction of docs classified correctly: $\dfrac{\sum_i c_{ii}}{\sum_j \sum_i c_{ij}}$

Evaluation of Classification Task

- If we have more than one class, how do we combine multiple performance measures into one quantity?
- Macroaveraging: Compute performance for each class, then average.
- Microaveraging: Collect decisions for all classes, compute contingency table, evaluate.

Evaluation of Classification Task

Class 1

	Truth: yes	Truth: no
Classifier: yes	10	10
Classifier: no	10	970

Class 2

	Truth: yes	Truth: no
Classifier: yes	90	10
Classifier: no	10	890

Micro Ave. Table

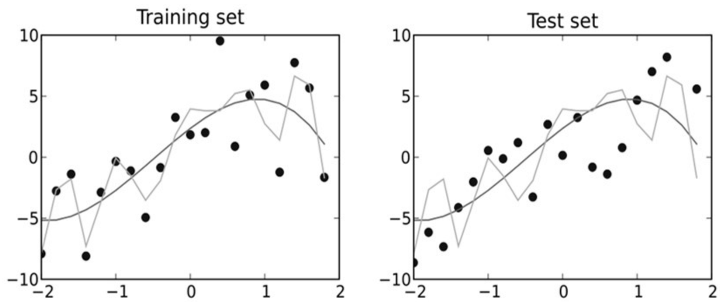
	Truth: yes	Truth: no
Classifier: yes	100	20
Classifier: no	20	1860

- Macroaveraged precision: (0.5 + 0.9)/2 = 0.7
- Microaveraged precision: 100/120 = .83
- Microaveraged score is dominated by score on common classes

Supervised Learning -Datasets

- Training / Validation / Testing sets
  - Model fitting – Feature selection/Parameter estimates
  - Validation dataset
    - Holdout (fixed splits 70/30)
    - Cross validation
      - 10-fold (k-fold)
  - Variance Vs. Bias
    - Error due to Bias: difference between expected and actual
    - Error due to variance: variability of the model
    - Bulls –eye example

Example



A training set (left) and a test set (right) from the same statistical population are shown as blue points. Two predictive models are fit to the training data. Both fitted models are plotted with both the training and test sets. In the training set, the MSE of the fit shown in orange is 4 whereas the MSE for the fit shown in green is 9. In the test set, the MSE for the fit shown in orange is 15 and the MSE for the fit shown in green is 13. The orange curve severely overfits the training data, since its MSE increases by almost a factor of four when comparing the test set to the training set. The green curve overfits the training data much less, as its MSE increases by less than a factor of 2.

Variance vs. Bias

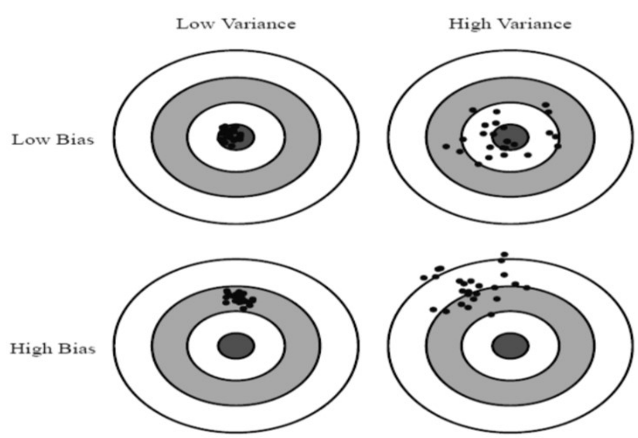


Fig. 1 Graphical illustration of bias and variance.

Classification Model & Accuracy

- Classification Model learning is a challenging task.
- Textual documents are rich in features.
- Feature Selection is helpful in simplifying the model and improving accuracy of the model.
- Textual Features
  - Words/ Lexemes/ Tokens
  - Phrases/ Bi-grams/ sequences
  - Graphs
  - NLP based Features

Features

- A feature is an individual measurable property of a phenomenon being observed.
- Choosing informative, discriminating and independent features is a crucial step for effective algorithms in pattern recognition, classification and regression.
- Features are usually numeric, but structural features such as strings and graphs are used in syntactic pattern recognition.

Feature Selection

- Feature selection is the process of selection of subset of features for the training set and using only this subset as features in text classification.
- Feature selection serves two main purposes:
  - It makes training and applying a classifier more efficient by decreasing the size of the effective vocabulary
  - Feature selection often increases classification accuracy as noise features are eliminated through it.

Feature Selection

- There are mainly two types of feature selection methods in machine learning
- Wrappers
  - Wrappers use the classification accuracy of some learning algorithm as their evaluation function. Since wrappers have to train a classifier for each feature subset to be evaluated, they are usually much more time consuming especially when the number of features is high.
  - So wrappers are generally not suitable for text classification.

Feature Selection

- Filters
  - filters perform feature selection independently of the learning algorithm that will use the selected features.
  - In order to evaluate a feature, filters use an evaluation metric that measures the ability of the feature to differentiate each class.
  - In general filters are much less time consuming than wrappers and have been widely used in text classification
- Feature selection metric should consider problem domain and algorithm characteristics

Feature Selection

- How it Works
  - We can view feature selection as a method for replacing a complex classifier (using all features) with a simpler one (using a subset of the features).
  - The purpose of a feature selection algorithm is to select only those features that For a given class c, we compute a utility measure A(t, c) for each term of the vocabulary and select the k terms that have the highest values of A(t, c).

Features Selection

```
SELECTFEATURES(D, c, k)
V ← EXTRACTVOCABULARY(D)
L ← []
for each t ∈ V
do A(t,c) ← COMPUTEFEATUREUTILITY(D, t, c)
    APPEND(L, ⟨A(t,c), t⟩)
return FEATURESWITHLARGESTVALUES(L, k)
```

▶ **Figure 13.6** Basic feature selection algorithm for selecting the k best features.

Feature Selection

- We will discuss three approaches to feature selection
  - mutual information, A(t, c) = I (U_t ;C_c);
  - X^2, A(t, c) = X^2 (t,c);
  - Frequency based features, A(t, c) = N(t,c);

Mutual Information

- A common feature selection method is to compute A(t, c) as the expected mutual information (MI) of term t and class c.
- MI measures how much information the presence/absence of a term contributes to making the correct classification decision on c.

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

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

Mutual Information

Example 13.3: Consider the class *poultry* and the term export in Reuters-RCV1. The counts of the number of documents with the four possible combinations of indicator values are as follows:

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

After plugging these values into Equation (13.17) we get:

$$\begin{aligned} I(U;C) &= \frac{49}{801{,}948}\log_2\frac{801{,}948 \cdot 49}{(49+27{,}652)(49+141)} \\ &+ \frac{141}{801{,}948}\log_2\frac{801{,}948 \cdot 141}{(141+774{,}106)(49+141)} \\ &+ \frac{27{,}652}{801{,}948}\log_2\frac{801{,}948 \cdot 27{,}652}{(49+27{,}652)(27{,}652+774{,}106)} \\ &+ \frac{774{,}106}{801{,}948}\log_2\frac{801{,}948 \cdot 774{,}106}{(141+774{,}106)(27{,}652+774{,}106)} \\ &\approx 0.0001105 \end{aligned}$$

Mutual Information – Example

UK		*China*		*poultry*	
london	0.1925	china	0.0997	poultry	0.0013
uk	0.0755	chinese	0.0523	meat	0.0008
british	0.0596	beijing	0.0444	chicken	0.0006
stg	0.0555	yuan	0.0344	agriculture	0.0005
britain	0.0469	shanghai	0.0292	avian	0.0004
plc	0.0357	hong	0.0198	broiler	0.0003
england	0.0238	kong	0.0195	veterinary	0.0003
pence	0.0212	xinhua	0.0155	birds	0.0003
pounds	0.0149	province	0.0117	inspection	0.0003
english	0.0126	taiwan	0.0108	pathogenic	0.0003

coffee		*elections*		*sports*	
coffee	0.0111	election	0.0519	soccer	0.0681
bags	0.0042	elections	0.0342	cup	0.0515
growers	0.0025	polls	0.0339	match	0.0441
kg	0.0019	voters	0.0315	matches	0.0408
colombia	0.0018	party	0.0303	played	0.0388
brazil	0.0016	vote	0.0299	league	0.0386
export	0.0014	poll	0.0225	beat	0.0301
exporters	0.0013	candidate	0.0202	game	0.0299
exports	0.0013	campaign	0.0202	games	0.0284
crop	0.0012	democratic	0.0198	team	0.0264

► **Figure 13.7** Features with high mutual information scores for six Reuters-RCV1 classes.

Chi Square Method

- In feature selection, the two events are occurrence of the term and occurrence of the class. We then rank terms with respect to the following quantity:

$$X^2(\mathbb{D}, t, c) = \sum_{e_t \in \{0,1\}} \sum_{e_c \in \{0,1\}} \frac{(N_{e_t e_c} - E_{e_t e_c})^2}{E_{e_t e_c}}$$

- X^2 is a measure of how much expected counts E and observed counts N deviate from each other. A high value of X^2 indicates that the hypothesis of independence, which implies that expected and observed counts are similar, is incorrect.

Chi Square Method

$$X^2(\mathbb{D},t,c) = \frac{(N_{11}+N_{10}+N_{01}+N_{00}) \times (N_{11}N_{00} - N_{10}N_{01})^2}{(N_{11}+N_{01}) \times (N_{11}+N_{10}) \times (N_{10}+N_{00}) \times (N_{01}+N_{00})}$$

X^2 Example

Example 13.4: We first compute E_{11} for the data in Example 13.3:

$$\begin{aligned} E_{11} &= N \times P(t) \times P(c) = N \times \frac{N_{11}+N_{10}}{N} \times \frac{N_{11}+N_{01}}{N} \\ &= N \times \frac{49+141}{N} \times \frac{49+27652}{N} \approx 6.6 \end{aligned}$$

where N is the total number of documents as before.
We compute the other $E_{e_t e_c}$ in the same way:

	$e_{poultry} = 1$		$e_{poultry} = 0$	
$e_{\text{export}} = 1$	$N_{11} = 49$	$E_{11} \approx 6.6$	$N_{10} = 27{,}652$	$E_{10} \approx 27{,}694.4$
$e_{\text{export}} = 0$	$N_{01} = 141$	$E_{01} \approx 183.4$	$N_{00} = 774{,}106$	$E_{00} \approx 774{,}063.6$

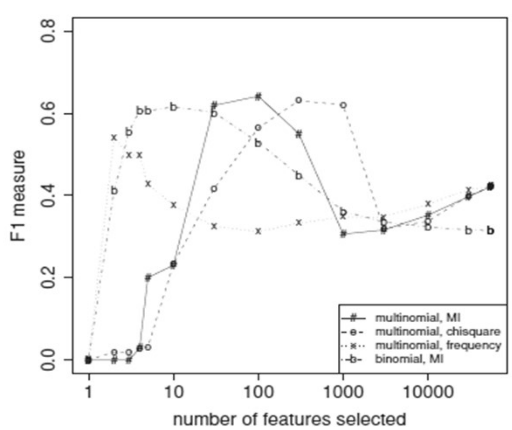
Plugging these values into Equation (13.18), we get a X^2 value of 284:

$$X^2(\mathbb{D},t,c) = \sum_{e_t \in \{0,1\}} \sum_{e_c \in \{0,1\}} \frac{(N_{e_t e_c} - E_{e_t e_c})^2}{E_{e_t e_c}} \approx 284$$

Frequency based features

- A third feature selection method is frequency-based feature selection, that is, selecting the terms that are most common in the class.
- Frequency-based feature selection selects some frequent terms that have no specific information about the class.
- Frequency can be either defined as document frequency (the number of documents in the class c that contain the term t) or as collection frequency (the number of tokens of t that occur in documents in c). Document frequency is more appropriate for the Bernoulli model, collection frequency for the multinomial model.

Features Selection – Experiment



► **Figure 13.8** Effect of feature set size on accuracy for multinomial and Bernoulli models.

Features Generation

- The idea is to generate high-level features (more abstract) from the low-level features.
- Example: sentence dependency graph from sentence.

Naïve Bayes

- Very Fast, low storage requirements
- Robust to Irrelevant Features

  Irrelevant Features cancel each other without affecting results
- Very good in domains with many equally important features

  Decision Trees suffer from *fragmentation* in such cases – especially if little data
- Optimal if the independence assumptions hold: If assumed independence is correct, then it is the Bayes Optimal Classifier for problem
- A good dependable baseline for text classification

Vector Space Model

- A document is represented as a feature vector in feature-dimensional space.
- Distance is generally the notions of similarity for this model.
- General assumptions for classification:
  - Contiguity hypothesis: Documents in the same class form a contiguous region and regions of different classes do not overlap.

Classification –Vector Space Model

- Corpus-Classification Dataset
  - Pre-processing
- Feature Selection & weighting
- Similarity function
- Classification algorithm
- Performance evaluation

Rocchio's Algorithm

- Rocchio's algorithm can be used for text classification as well.
- Rocchio classification divides the vector space into regions centered on centroids or prototypes, one for each class, computed as the center of mass of all documents in the class.
- Rocchio classification is simple and efficient, but inaccurate if classes are not approximately spheres with similar radii.

Rocchio's Algorithm

TRAINROCCHIO($\mathbb{C}, \mathbb{D}$)
1 **for each** $c_j \in \mathbb{C}$
2 **do** $D_j \leftarrow \{d : \langle d, c_j \rangle \in \mathbb{D}\}$
3 $\quad \vec{\mu}_j \leftarrow \frac{1}{|D_j|} \sum_{d \in D_j} \vec{v}(d)$
4 **return** $\{\vec{\mu}_1, \ldots, \vec{\mu}_J\}$

APPLYROCCHIO($\{\vec{\mu}_1, \ldots, \vec{\mu}_J\}, d$)
1 **return** $\arg\min_j |\vec{\mu}_j - \vec{v}(d)|$

K-Nearest Neighbor Learning

- k-NN learning is an example based learning, it is also a memory based method in which learning is just storing the representations of the training examples in D.
- Testing instance x: Compute similarity between x and all examples in D. Assign x the category of the most similar example in D
- It does not explicitly calculate a class/category prototype descriptor.

k-NN Classification Algorithm

- Training
  - For each training example <d_i, C_j> ε D_{train}
  - Compute the corresponding Feature vector -> d_i, for document d_i
- Testing
  - Computer vector for d_j using the same feature vector
  - For each <d_i, C_j> ε D_{train} calculate x[i]=Cosine(d_j,d_i), sort x[] by decreasing value.
  - Let N be the closest (i.e. first) k examples in D. (get k most similar neighbors) Return the majority class of examples in N

k-NN Classification

TRAIN-KNN($\mathbb{C}$, $\mathbb{D}$)
$\mathbb{D}' \leftarrow$ PREPROCESS($\mathbb{D}$)
$k \leftarrow$ SELECT-K($\mathbb{C}$, $\mathbb{D}'$)
return $\mathbb{D}', k$

APPLY-KNN($\mathbb{C}$, $\mathbb{D}'$, k, d)
$S_k \leftarrow$ COMPUTENEARESTNEIGHBORS($\mathbb{D}'$, k, d)
for each $c_j \in \mathbb{C}$
do $p_j \leftarrow |S_k \cap c_j|/k$
return $\arg\max_j p_j$

Example

	Doc	Words	Class
Training	1	Chinese Beijing Chinese	c
	2	Chinese Chinese Shanghai	c
	3	Chinese Macao	c
	4	Tokyo Japan Chinese	j
Test	5	Chinese Chinese Chinese Tokyo Japan	?

Dictionary = <bejing, chinese, japan, macao,shanghai,tokyo>
Doc#1 < 1,2,0,0,0,0>
Doc#2 <0,2,0,0,1,0>
Doc#3 <0,1,0,1,0,0>
Doc#4 <0,1,1,0,0,1>
Doc#5 <0,3,0,0,0,1>

Cos(d1,d5) = dot-product (d1,d5) ÷ |d1| |d5|
Cos(d1,d5) = 0.848
Cos(d2,d5) =0.848
Cos(d3,d5) =0.424
Cos(d4,d5) =0.953

For 1-NN d5 will belong to j
For 3-NN d5 will belong to c

K-Nearest Neighbor Learning

- Advantages
  - Simple, intuitive, easy to implement.
  - Only one hyper-parameter
- Disadvantages
  - Sensitive to value of k, distance function, and noisy data
  - Lazy learner – no precompute model
  - No training time but large computation time for large dataset.