Feature Engineering and Text Classification

Ayoub Bagheri

Introduction to Text Mining with R

Table of Contents

Lecture’s Plan

1. How to represent a document?
2. What are vector space and bag-of-words models?
3. Features in text? And how to do text feature selection?
4. How to classify text data?
5. How to evaluate a classifier?

Text Classification

Text classification

- Supervised learning: Learning a function that maps an input to an output based on example input-output pairs.
  - infer a function from labeled training data
  - use the inferred function to label new instances
- Human experts annotate a set of text data
  - Training set

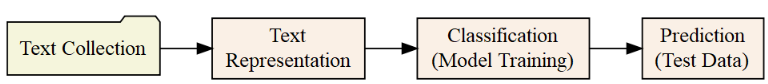
Document	Class
Email1	Not spam
Email2	Not spam
Email3	Spam
…	…

Text classification?

- Which problem is not a text classification task? (less likely to be)
  - Author’s gender detection from text
  - Finding about the smoking conditions (yes/no) of patients from clinical letters
  - Grouping similar news articles
  - Classifying reviews into positive and negative sentiment

Go to www.menti.com and use the code 9594 3321

Pipeline



Text Representation

How to represent a document

- Represent by a string?
  - No semantic meaning
- Represent by a list of sentences?
  - Sentence is just like a short document (recursive definition)
- Represent by a vector?
  - A vector is an ordered finite list of numbers.

Vector space model

- A vector space is a collection of vectors
- Represent documents by concept vectors
  - Each concept defines one dimension
  - k concepts define a high-dimensional space
  - Element of vector corresponds to concept weight

Vector space model

- Distance between the vectors in this concept space
  - Relationship among documents
- The process of converting text into numbers is called Vectorization

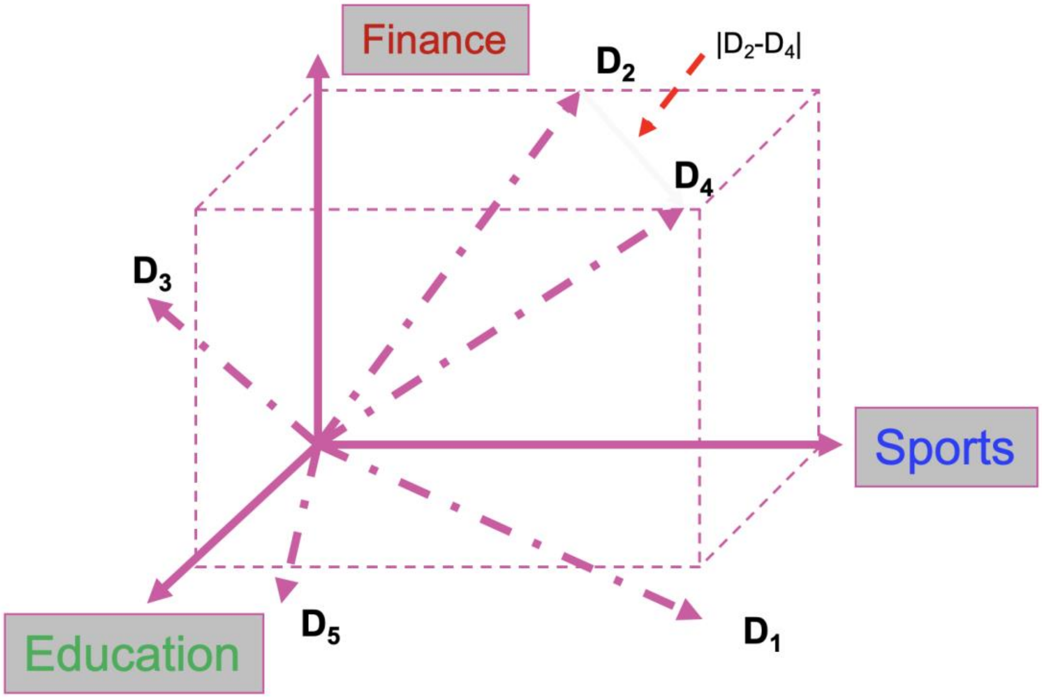
Vector space model

- Terms are generic features that can be extracted from text
- Typically, terms are single words, keywords, n-grams, or phrases
- Documents are represented as vectors of terms
- Each dimension (concept) corresponds to a separate term

$$d = (w_1, \ldots, w_n)$$

An illustration of VS model

- All documents are projected into this concept space



Vector space model

- Bag of Words
- Topics
- Word Embeddings

Bag of Words (BOW)

- With Bag of Words (BOW), we refer to a Vector Space Model where:
  - Terms: words (more generally we may use n-grams, etc.)
  - Weights: number of occurrences of the terms in the document

BOW representation

- Term as the basis for vector space
  - Doc1: Text mining is to identify useful information.
  - Doc2: Useful information is mined from text.
  - Doc3: Apple is delicious.

	text	information	identify	mining	mined	is	useful	to	from	apple	delicious
Doc1	1	1	1	1	0	1	1	1	0	0	0
Doc2	1	1	0	0	1	1	1	0	1	0	0
Doc3	0	0	0	0	0	1	0	0	0	1	1

–>

BOW weights: Binary

- Binary
  - with 1 indicating that a term occurred in the document, and 0 indicating that it did not

BOW weights: Term frequency

- Idea: a term is more important if it occurs more frequently in a document
- TF Formulas
  - Let $t(c, d)$ be the frequency count of term t in doc d
  - Raw TF: $tf(t, d) = c(t, d)$

BOW weights: TFiDF

- Idea: a term is more discriminative if it occurs a lot but only in fewer documents

Let $n_{d,t}$ denote the number of times the t-th term appears in the d-th document.

$$TF_{d,t} = \frac{n_{d,t}}{\sum_i n_{d,i}}$$

Let N denote the number of documents annd N_t denote the number of documents containing the t-th term.

$$IDF_t = log(\frac{N}{N_t})$$

TF-IDF weight:

$$w_{d,t} = TF_{d,t} \cdot IDF_t$$

In R

```
library(tm)

## Loading required package: NLP

data <- c('Text mining is one of the Utrecht summer school courses.',
          'There are other data science courses in Utrecht summer school')

# convert data to vector space model
```

```r
corpus <- VCorpus(VectorSource(data))

# create a dtm object
dtm <- DocumentTermMatrix(corpus,
                          list(removePunctuation = TRUE,
                               stopwords = TRUE,
                               stemming = TRUE,
                               removeNumbers = TRUE))
```

In R

```r
inspect(dtm)

## <<DocumentTermMatrix (documents: 2, terms: 9)>>
## Non-/sparse entries: 13/5
## Sparsity           : 28%
## Maximal term length: 7
## Weighting          : term frequency (tf)
## Sample             :
##     Terms
## Docs cours data mine one school scienc summer text utrecht
##    1     1    0    1   1      1      0      1    1       1
##    2     1    1    0   0      1      1      1    0       1
```

Feature Selection

Feature selection for text classification

- Feature selection is the process of selecting a specific subset of the terms of the training set and using only them in the classification algorithm.
- high dimensionality of text features
- Select the most informative features for model training
  - Reduce noise in feature representation
  - Improve final classification performance
  - Improve training/testing efficiency
    - Less time complexity
    - Fewer training data

Feature selection methods

- Wrapper methods
  - Find the best subset of features for a particular classification method
  - Sequential forward selection or genetic search to speed up the search

- Filter methods
  - Evaluate the features independently from the classifier and other features
  - Feasible for very large feature se
  - Usually used as a preprocessing step
- Embedded methods
- e.g. Regularized regression, Regularized SVM

Document frequency

- Rare words: non-influential for global prediction, reduce vocabulary size

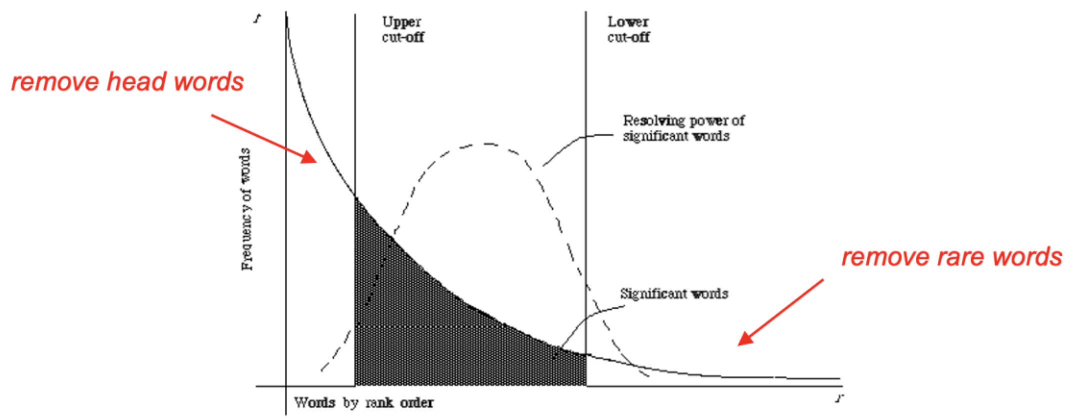


Figure 2.1. A plot of the hyperbolic curve relating f, the frequency of occurrence and r, the rank order (Adaped from Schultz[44] page 120)

Information gain

- Decrease in entropy of categorical prediction when the feature is present or absent

$$IG(t) = -\sum_{c} p(c) \log p(c)$$ Entropy of class label along

$$+p(t) \sum_{c} p(c|t) \log p(c|t)$$ Entropy of class label if t is present

$$+p(\bar{t}) \sum_{c} p(c|\bar{t}) \log p(c|\bar{t})$$ Entropy of class label if t is absent

probability of seeing class label c in documents where t occurs

probability of seeing class label c in documents where t does not occur

Gini Index

Let $p(c|t)$ be the conditional probability that a document belongs to class c, given the fact that it contains the term t. Therefore, we have:

$$\sum_{c=1}^{k} p(c|t) = 1$$

Then, the gini-index for the term t, denoted by $G(t)$ is defined as:

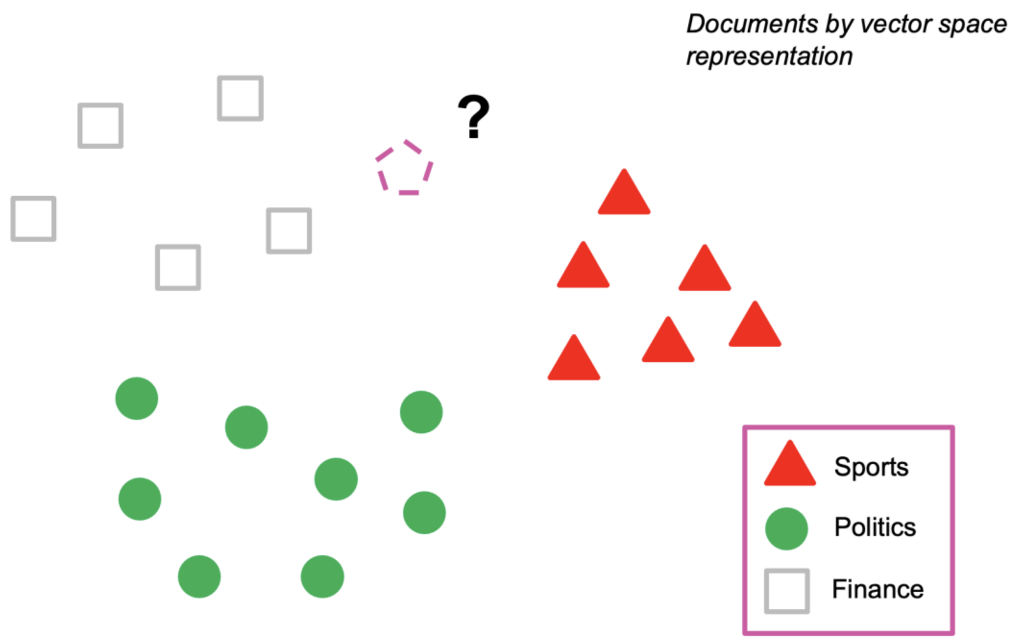
$$G(t) = \sum_{c=1}^{k} p(c|t)^2$$

Gini Index

- The value of the gini-index lies in the range $(1/k, 1)$.
- Higher values of the gini-index indicate a greater discriminative power of the term t.
- If the global class distribution is skewed, the gini-index may not accurately reflect the discriminative power of the underlying attributes.
- Other methods
  - Normalized gini-index
  - Mutual Information
  - χ^2-Statistic

Classification Algorithms

How to classify this document?



Text Classification: definition

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

Hand-coded rules

- Rules based on combinations of words or other features
- Rules carefully refined by expert
- But building and maintaining these rules is expensive
- Data/Domain specifics
- Not recommended!

Supervised Machine Learning

- Logistic regression

- K-nearest neighbors
- Naïve Bayes
- Support vector machines
- Neural networks

Rocchio Classifier (Nearest Centroid)

Each class is represented by its centroid, with test samples classified to the class with the nearest centroid. Using a training set of documents, the Rocchio algorithm builds a prototype vector, centroid, for each class. This prototype is an average vector over the training documents' vectors that belong to a certain class.

$$\boldsymbol{\mu}_{\mathbf{c}} = \frac{1}{|D_c|} \sum_{\mathbf{d} \in D_c} \mathbf{d}$$

Where D_c is the set of documents in the corpus that belongs to class c and d is the vector representation of document d.

Rocchio Classifier (Nearest Centroid)

The predicted label of document d is the one with the smallest (Euclidean) distance between the document and the centroid.

$$\hat{c} = \underset{c}{\operatorname{argmin}} ||\boldsymbol{\mu}_{\mathbf{c}} - \mathbf{d}||$$

K-Nearest Neighbor

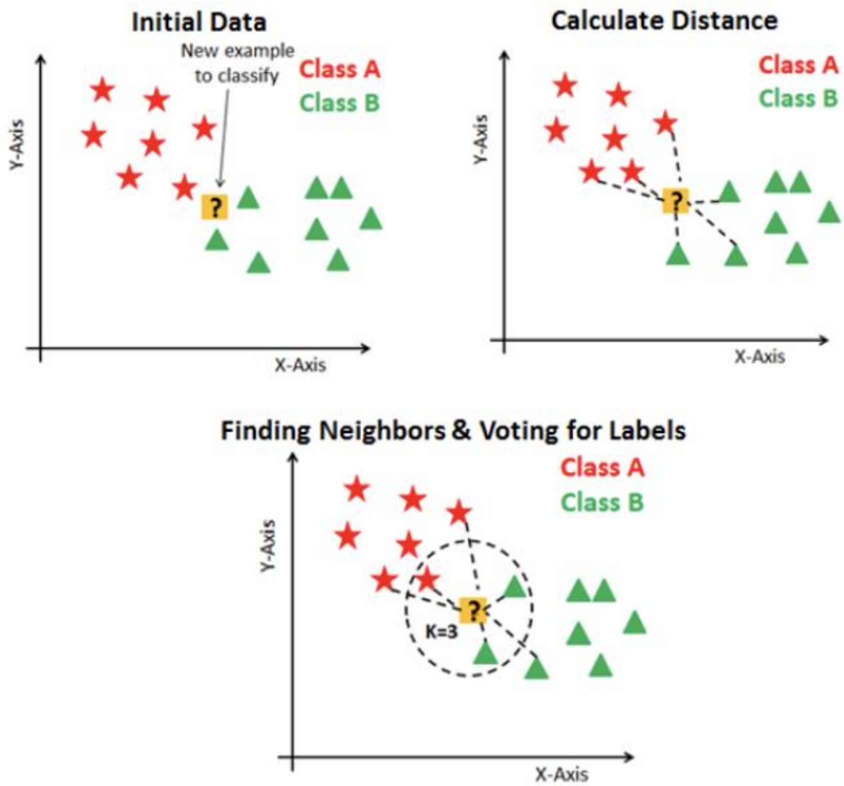
- Given a test document d,. the KNN algorithm finds the k nearest neighbors of d among all the documents in the training set, and scores the category candidates based on the class of the k neighbors.

- After sorting the score values, the algorithm assigns the candidate to the class with the highest score.

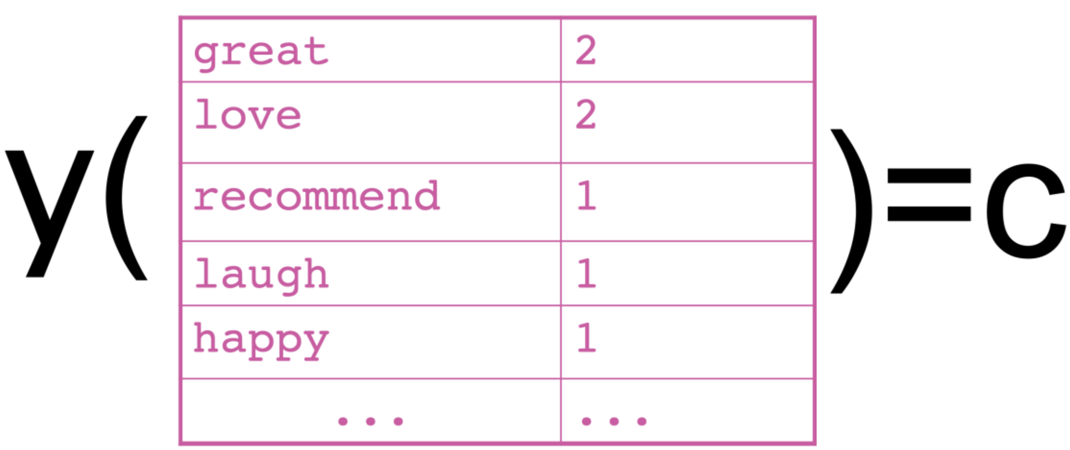
- The basic nearest neighbors classification uses uniform weights: that is, the value assigned to a query point is computed from a simple majority vote of the nearest neighbors. C

- Can weight the neighbors such that nearer neighbors contribute more to the fit.

K-Nearest Neighbor



Naïve Bayes



Bayes' Rule

- Applied to documents and classes
- For a document *d* and a class *c*

$$P(c|d) = \frac{P(d|c)P(c)}{P(d)}$$

Multinomial Naïve Bayes Independence Assumptions

- Bag of Words assumption: Assume position doesn't matter
- Conditional Independence: Assume the feature probabilities $P(w_i|c_j)$ are independent given the class c.

$$P(w_1, \dots, w_n|c) = P(w_1|c) \cdot P(w_2|c) \cdot P(w_3|c) \cdot \dots \cdot P(w_n|c)$$

Multinomial Naïve Bayes Classifier

$$C_{MAP} = \underset{c \in C}{\operatorname{argmax}} P(w_1, w_2, \dots, w_n|c)P(c)$$

$$C_{NB} = \underset{c \in C}{\operatorname{argmax}} P(c_j) \prod_{w \in V} P(w|c)$$

$$C_{NB} = \underset{c \in C}{\operatorname{argmax}} P(c_j) \prod_{i \in positions} P(w_i|c_i)$$

Parameter estimation

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

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

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

- fraction of times word w_i appears among all words in documents of topic c_j

Problem with Maximum Likelihood

What if we have seen no training documents with the word coffee and classified in the topic positive (thumbs-up)?

$$\hat{P}(\text{"coffee"}|\text{positive}) = \frac{count(\text{"coffee"}, \text{positive})}{\sum_{w \in V} count(\text{w}, \text{positive})}$$

Zero probabilities cannot be conditioned away, no matter the other evidence!

$$C_{MAP} = \underset{c}{\text{argmax}} \hat{P}(c) \prod_{i} \hat{P}(w_i|c)$$

Laplace (add-1) smoothing for Naïve Bayes

$$\begin{aligned} \hat{P}(w_i|c) &= \frac{count(w_i, c) + 1}{\sum_{w \in V} (count(w, c) + 1)} \\ &= \frac{count(w_i, c) + 1}{(\sum_{w \in V} count(w, c) + |V|)} \end{aligned}$$

Multinomial Naïve Bayes: Learning

- From training corpus, extract Vocabulary
- Calculate $P(c_j)$ terms
  - For each c_j in C do

  $docs_j \leftarrow$ all docs with class = c_j

$$P(c_j) \leftarrow \frac{|docs_j|}{|total\ \#\ documents|}$$

- Calculate $P(w_k|c_j)$ terms
  - $Text_j \leftarrow$ single doc containing all $docs_j$
  - For each word w_k in Vocabulary

    $n_k \leftarrow$ # of occurrences of w_k in $Text_j$

$$P(w_k|c_j) \leftarrow \frac{n_k + \alpha}{n + \alpha|Vocabulary|}$$

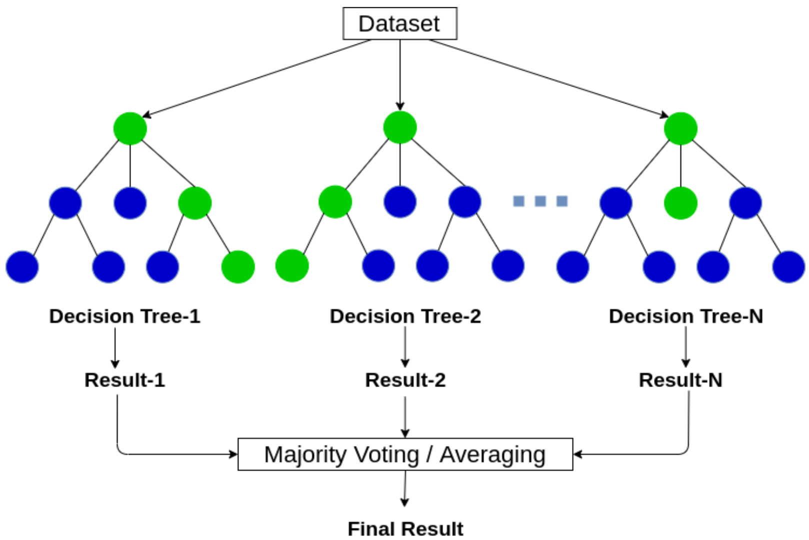
Decision Tree

- A decision tree is a hierarchical decomposition of the (training) data space, in which a condition on the feature value is used in order to divide the data space hierarchically.

- Top-down, by choosing a variable at each step that best splits the set of items.
- Different algorithms to measure the homogeneity of the target variable within the subsets.
  - Gini impurity
  - Information gain

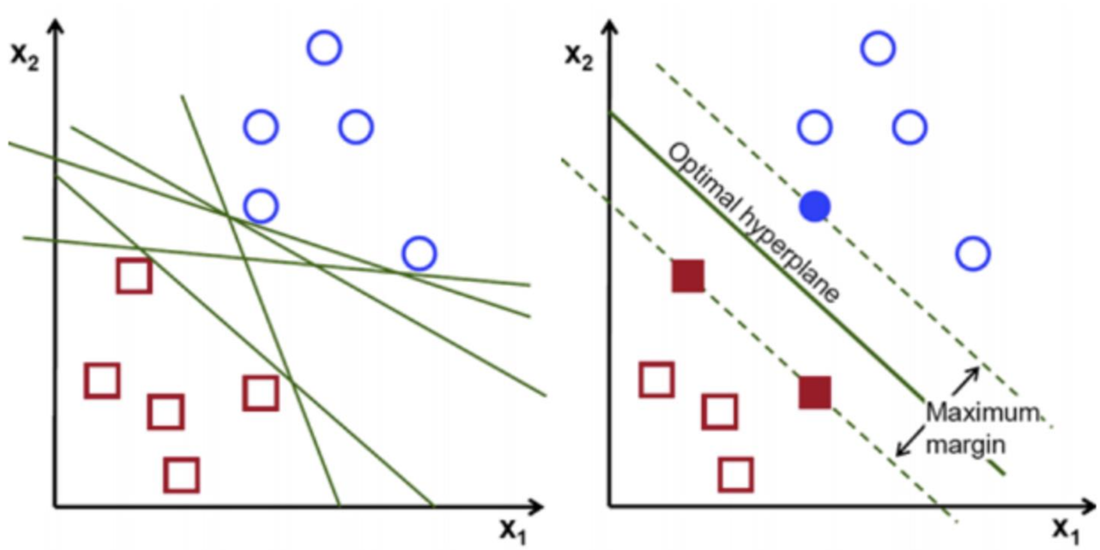
Random Forest

- Random forests are an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time.
- Fit multiple trees to bootstrapped samples of the data AND at each node select best predictor from only a random subset of predictors. Combine all trees to yield a consensus prediction



Support Vector Machine

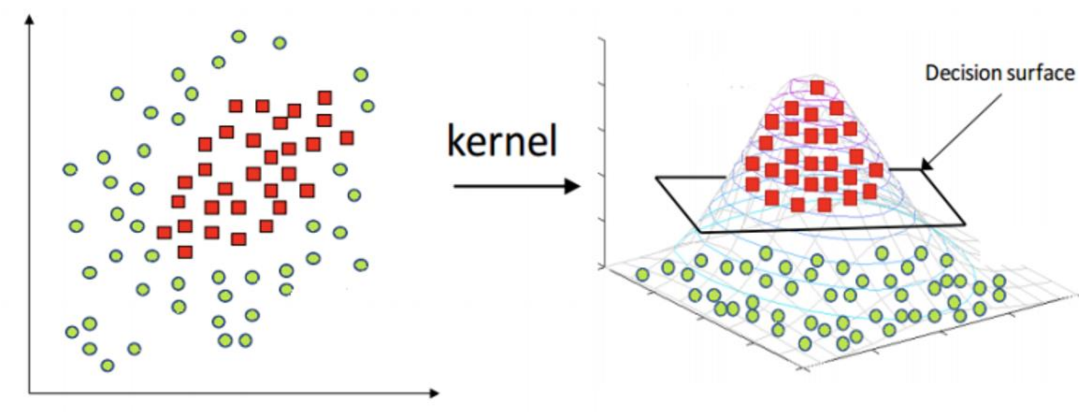
- The main principle of SVM is to determine separators in the search space which can best separate the different classes.
- SVM tries to make a decision boundary in such a way that the separation between the two classes is as wide as possible.



Support Vector Machine

- It is not necessary to use a linear function for the SVM classifier.
- With the kernel trick, SVM can construct a nonlinear decision surface in the original feature space by mapping the data instances non-linearly to a new space where the classes can be separated linearly with a hyperplane.

- In practice, linear SVM is used most often because of their simplicity and ease of interpretability.
- SVM is quite robust to high dimensionality.

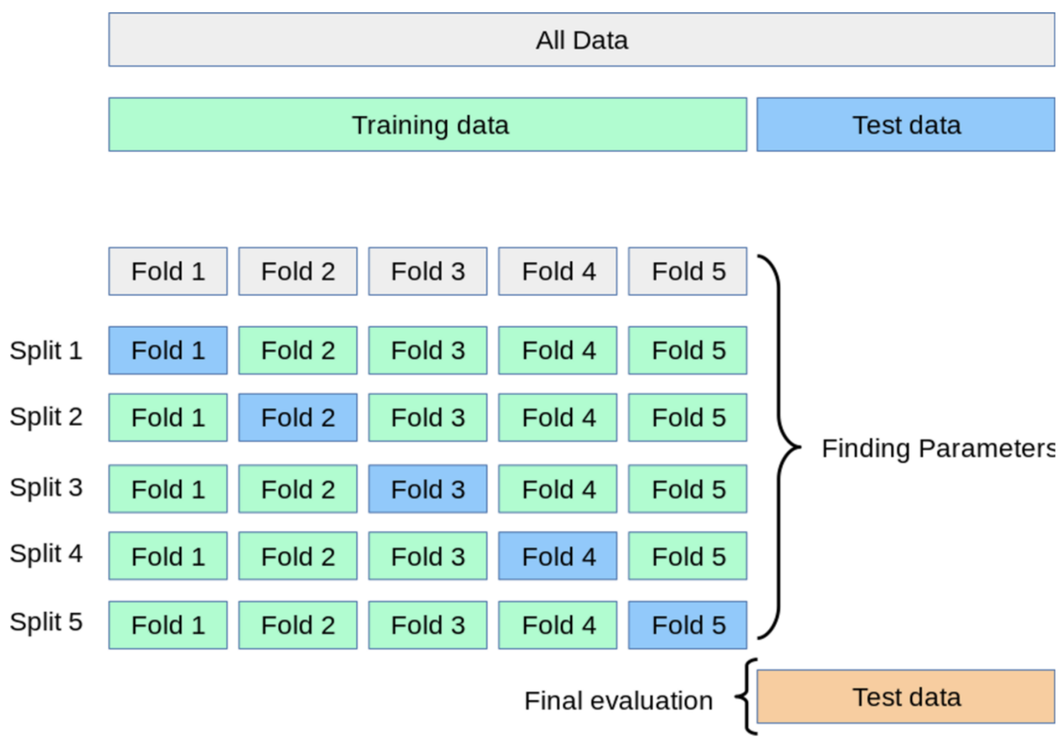


Evaluation

Data Splitting

- Training set
  - Validation set (dev set)
    - A validation dataset is a dataset of examples used to tune the hyperparameters (i.e. the architecture) of a classifier. It is sometimes also called the development set or the “dev set”.
- Test set

Cross Validation



https://scikit-learn.org/stable/modules/cross_validation.html

Confusion matrix

		Predicted Class: Positive	Predicted Class: Negative	
Actual Class	**Positive**	True Positive (TP)	False Negative (FN) **Type II Error**	**Sensitivity** $\frac{TP}{(TP+FN)}$
	Negative	False Positive (FP) **Type I Error**	True Negative (TN)	**Specificity** $\frac{TN}{(TN+FP)}$
		Precision $\frac{TP}{(TP+FP)}$	**Negative Predictive Value** $\frac{TN}{(TN+FN)}$	**Accuracy** $\frac{TP+TN}{(TP+TN+FP+FN)}$

Accuracy

- What proportion of instances is correctly classified?
  TP + TN / TP + FP + FN + TN
- Accuracy is a valid choice of evaluation for classification problems which are well balanced and not skewed.

- Let us say that our target class is very sparse. Do we want accuracy as a metric of our model performance? What if we are predicting if an asteroid will hit the earth? Just say "No" all the time. And you will be 99% accurate. The model can be reasonably accurate, but not at all valuable.

Precision and recall

- Precision: % of selected items that are correct Recall: % of correct items that are selected
- Precision is a valid choice of evaluation metric when we want to be very sure of our prediction.
- Recall is a valid choice of evaluation metric when we want to capture as many positives as possible.

A combined measure: F

A combined measure that assesses the P/R tradeoff is F measure (weighted harmonic mean):

$$F = \frac{1}{\alpha\frac{1}{P} + (1-\alpha)\frac{1}{R}} = \frac{(\beta^2+1)PR}{\beta^2 P + R}$$

The harmonic mean is a very conservative average;

Balanced F1 measure - i.e., with $\beta = 1$ (that is, $\alpha = 1/2$): $F = 2PR/(P+R)$

The Real World

No training data?

- Manually written rules

  If (x or y) and not (w or z) then categorize as class1

  - Need careful crafting
  - Low accuracy
  - Domain-specific
  - Time-consuming
- Active learning
- Unsupervised methods

Very little data?

- Use Naïve Bayes, KNN, Rocchio
  - Naïve Bayes is a “high-bias” algorithm (Ng and Jordan 2002 NIPS)
- Get more labeled data
- Find ways to label data for you
- Try semi-supervised methods:
  - e.g. active learning, bootstrapping

A reasonable amount of data?

- Perfect for all the complex classifiers
  - SVM
  - Regularized Logistic Regression
  - Random forest

A huge amount of data?

- Can achieve high accuracy!
- At a cost:
  - SVMs (train time) or KNN (test time) can be too slow
  - Regularized logistic regression
  - Naïve Bayes again!
  - Deep learning

Accuracy as a function of data size

- With enough data
  - Classifier may not matter

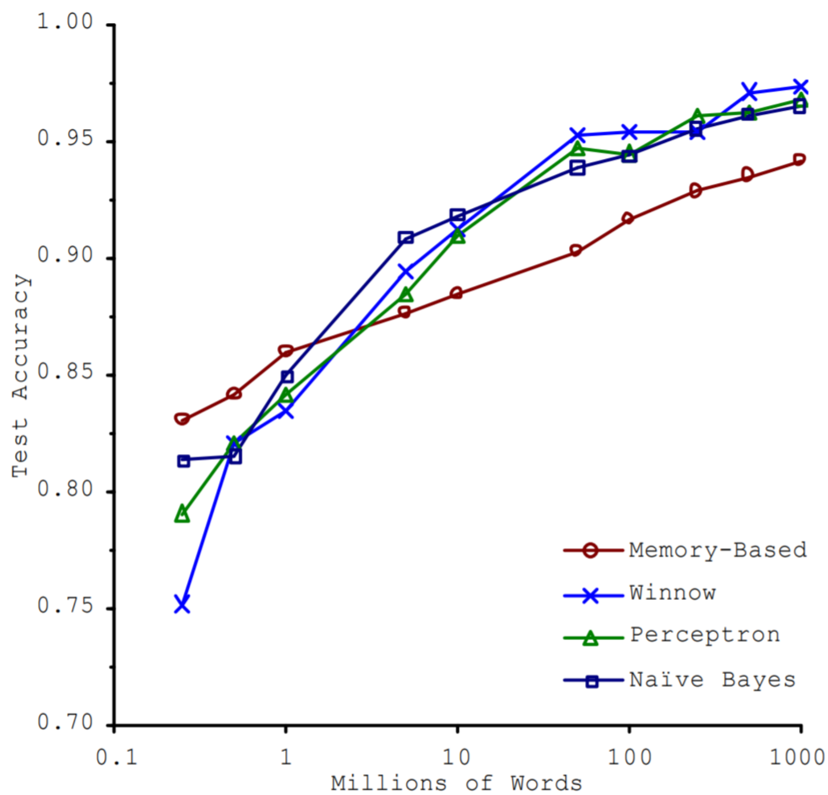


Figure 1. Learning Curves for Confusion Set Disambiguation

https://aclanthology.org/P01-1005.pdf

How to tweak performance

- Domain-specific features and weights: very important in real performance
- Sometimes need to collapse terms:
  - Part numbers, chemical formulas, …
  - But stemming generally doesn't help
- Upweighting: Counting a word as if it occurred twice:
  - Title words
  - First sentence of each paragraph (Murata, 1999)
  - In sentences that contain title words
- Hyperparameter optimization

Terminology

Some terminology

Corpus: is a large and structured set of texts

Stop words: words which are filtered out before or after processing of natural language data (text)

Unstructured text: information that either does not have a pre-defined data model or is not organized in a pre-defined manner.

Tokenizing: process of breaking a stream of text up into words, phrases, symbols, or other meaningful elements called tokens (see also lexical analysis)

Natural language processing: field of computer science, artificial intelligence, and linguistics concerned with the interactions between computers and human (natural) languages

Term document (or document term) matrix: is a mathematical matrix that describes the frequency of terms that occur in a collection of documents

Supervised learning: is the machine learning task of inferring a function from labeled training data

Unsupervised learning: find hidden structure in unlabeled data

Summary

Summary

- Vector space model & BOW
- Feature Selection
- Text Classification
- Evaluation

Practical 3