Feature Selection and Text Clustering

Ayoub Bagheri

Before we start

k-fold cross validation in R

Method 1:

```
library(e1071)

#specify the cross-validation method
tune.control <- tune.control(random             = F,
                             nrepeat            = 1,
                             sampling           = c("cross'
                             sampling.aggregate = mean,
                             cross              = 5,
                             best.model         = T,
                             performances       = T)

# fit a model and use k-fold CV to evaluate performance
model <- naiveBayes(outcome ~ ., data, tune.control)
```

Before we start

k-fold cross validation in R

Method 2:

```
library(caret)

# specify the cross-validation method
ctrl <- trainControl(method = "cv",
                     number = 5)

# fit a model and use k-fold CV to evaluate performance
model <- train(y ~ x1 + x2, data = df, method = "lm", trCo
```

Lecture Plan

1. Features in text? And how to do text feature selection?
2. What is text clustering?
3. What are the applications?
4. How to cluster text data?

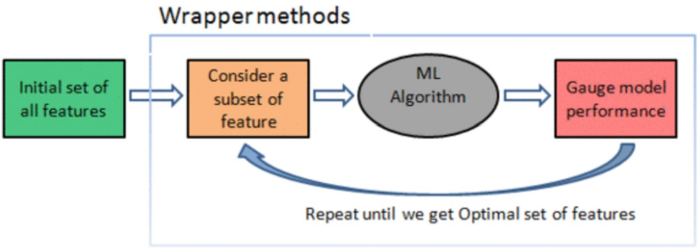
Feature Selection

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



Feature selection methods

- 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

Filter Methods

Filter merhods

- Document frequency
- Information gain
- Chi-squared
- F-score
- Relief
- Rough Sets consistency
- Binary consistency
- Inconsistent Examples consistency
- Inconsistent Examples Pairs consistency
- Determination Coefficient
- Mutual information
- Gain ratio
- Symmetrical uncertain
- Gini index

Document frequency

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

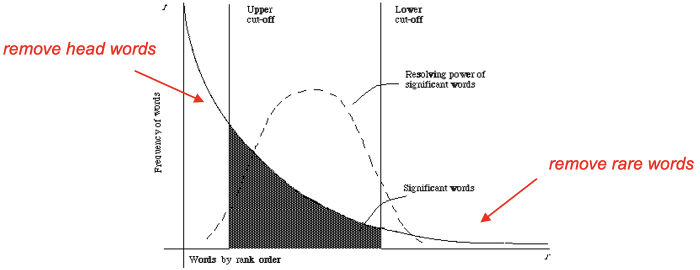


Figure 2.1. A plot of the hyperbolic curve relating f, the frequency of occurrence and r, the rank order (Adaped from Schultz[44] page 120)

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.

Information gain

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

$$
\begin{aligned}
IG(t) = &-\sum_{c} p(c)\log p(c) \\
&+p(t)\sum_{c} p(c|t)\log p(c|t) \\
&+p(\bar{t})\sum_{c} p(c|\bar{t})\log p(c|\bar{t})
\end{aligned}
$$

Entropy of class label along

Entropy of class label if t is present

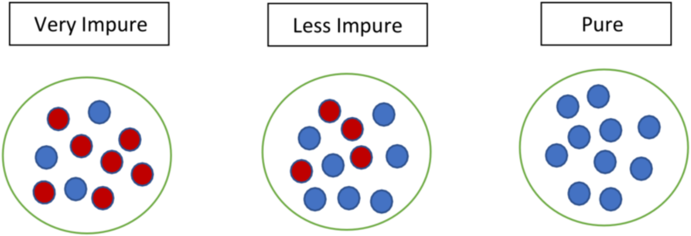
Entropy of class label if t is absent

probability of seeing class label c in documents where t occurs

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

Information gain

- The higher the information gain the greater discriminative power of the term t



In R

```
library(caret)
library(tm)
library(FSinR)

data <- c('Cats like to chase mice.',
          'Dogs like to eat big bones.')

# convert data to vector space model
corpus <- VCorpus(VectorSource(data))
# create a dtm object
dtm <- DocumentTermMatrix(corpus,
                          list(removePunctuation = TRUE,
                               stopwords = TRUE,
                               stemming = TRUE,
                               removeNumbers = TRUE))

# add the dependent variable
train_data <- as.matrix(dtm)
```

In R

```
# Feature Selection
evaluator       <- filterEvaluator('giniIndex')
directSearcher <- directSearchAlgorithm('selectKBest', list
 
# results
results <- directFeatureSelection(train_data, 'y', directSe
results$bestFeatures
```

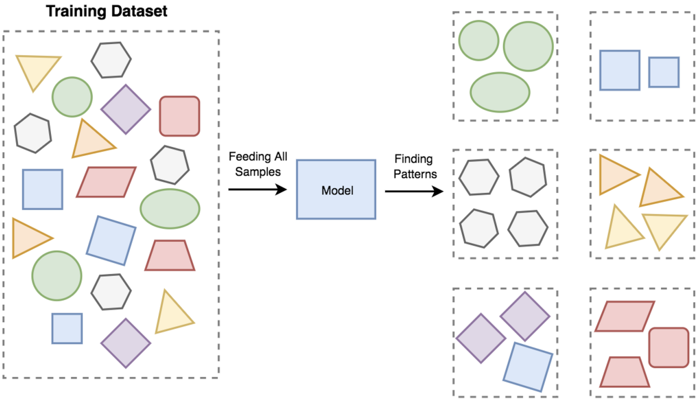
```
##      big bone cat chase dog eat like mice
## [1,]   1    1   1     0   0   0    0    0
```

```
results$featuresSelected
```

```
## [1] "big"  "bone" "cat"
```

Text Clustering

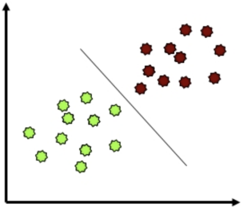
Unsupervised learning



Clustering versus classification

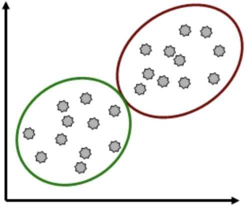
CLASSIFICATION

- Labeled data points
- Want a "rule" that assigns labels to new points
- Supervised learning



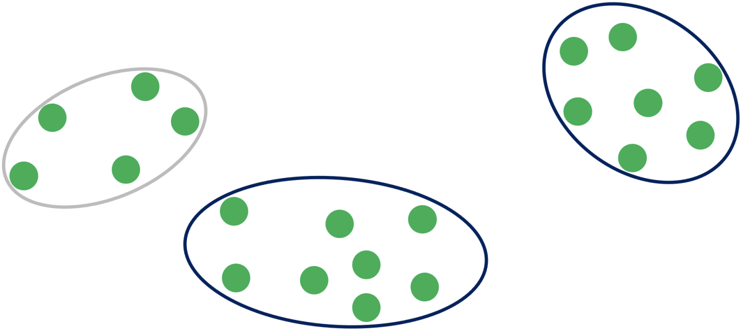
CLUSTERING

- Data is not labeled
- Group points that are "close" to each other
- Identify structure or patterns in data
- Unsupervised learning



Clustering

- Clustering: the process of grouping a set of objects into clusters of similar objects
- Discover "natural structure" of data
  - What is the criterion?
  - How to identify them?
  - How to evaluate the results?



Question

Which one is not a text clustering task?

- Finding similar patterns in customer reviews
- Grouping political tweets and finding their hidden topics
- Detection of heart failure (yes or no) using discharge letters
- Grouping scientific articles

Clustering

- Basic criteria
  - high intra-cluster similarity
  - low inter-cluster similarity
- No (little) supervision signal about the underlying clustering structure
- Need similarity/distance as guidance to form clusters

Clustering algorithms

Categories

- Hard versus soft clustering
- Partitional clustering
- Hierarchical clustering
- Topic modeling

Hard versus soft clustering

- Hard clustering: Each document belongs to exactly one cluster
  - More common and easier to do
- Soft clustering: A document can belong to more than one cluster.

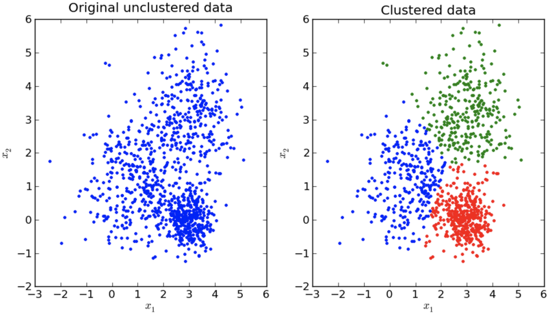
Partitional clustering

Partitional clustering algorithms

- Partitional clustering method: Construct a partition of n documents into a set of K clusters
- Given: a set of documents and the number K
- Find: a partition of K clusters that optimizes the chosen partitioning criterion
  - Globally optimal
    - Intractable for many objective functions
    - Ergo, exhaustively enumerate all partitions
  - Effective heuristic methods: K-means and K-medoids algorithms

Partitional clustering algorithms

- Typical partitional clustering algorithms
  - k-means clustering
    - Partition data by its closest mean



K-Means algorithm

- Assumes documents are real-valued vectors.
- Clusters based on centroids of points in a cluster, c:

$$\vec{\mu}(c) = \frac{1}{|c|} \sum_{\vec{a} \in c} \vec{x}$$

- Reassignment of instances to clusters is based on distance to the current cluster centroids.

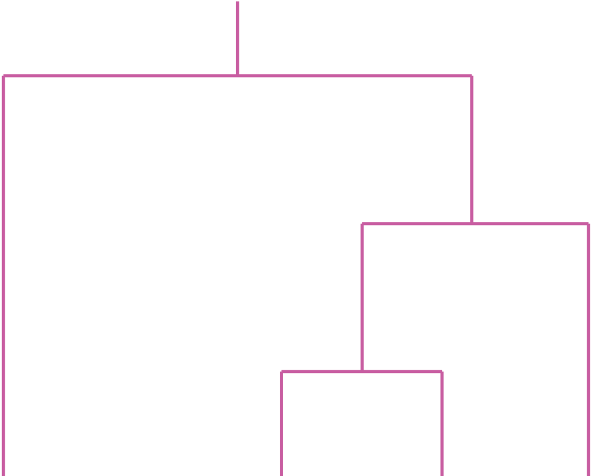
K-Means algorithm

- Select K random docs $\{s_1, s_2, \ldots s_K\}$ as seeds.
- Until clustering converges (or other stopping criterion):
  - For each document d_i:
    - Assign d_i to the cluster c_j such that $dist(x_i, s_j)$ is minimal.
  - (Next, update the seeds to the centroid of each cluster)
  - For each cluster cj
    - $s_j = \mu(c_j)$

Hierarchical Clustering

Dendrogram: Hierarchical clustering

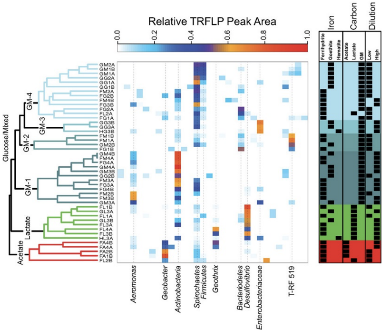
- Build a tree-based hierarchical taxonomy (dendrogram) from a set of documents.
- Clustering obtained by cutting the dendrogram at a desired level: each connected component forms a cluster.



Clustering algorithms

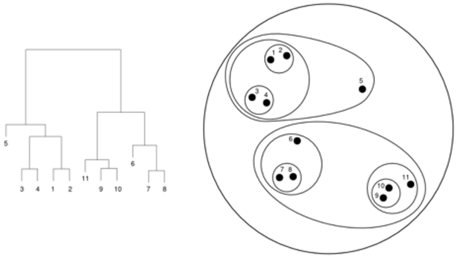
- Typical hierarchical clustering algorithms
  - Bottom-up agglomerative clustering
    - Start with individual objects as separated clusters
    - Repeatedly merge closest pair of clusters

Most typical usage: gene sequence analysis



Clustering algorithms

- Typical hierarchical clustering algorithms
  - Top-down divisive clustering
    - Start with all data as one cluster
    - Repeatedly splitting the remaining clusters into two



Hierarchical Agglomerative Clustering (HAC)

- Starts with each document in a separate cluster
  - then repeatedly joins the closest pair of clusters, until there is only one cluster.
- The history of merging forms a binary tree or hierarchy.

Closest pair of clusters

- Many variants to defining closest pair of clusters (linkage methods):
  - Single-link
    - Similarity of the most cosine-similar
  - Complete-link
    - Similarity of the "furthest" points, the least cosine-similar
  - Centroid
    - Clusters whose centroids (centers of gravity) are the most cosine-similar
  - Average-link
    - Average cosine between pairs of elements
  - Ward's linkage
    - Ward's minimum variance method, much in common with analysis of variance (ANOVA)
    - The distance between two clusters is computed as the increase in the "error sum of squares" (ESS) after fusing two clusters into a single cluster.

Summary

Summary

- Feature Selection
- Text Clustering
- Evaluation

Practical 5