

# IR – Document Classification

*Cyrill Zadra*

## Main program

assignment2.Main.scala is main program for document classification and requires following arguments:

-trainData [directory]	(directory for trainData)
-testData [directory]	(directory for testData)
-labeled [true false]	(does testData contains labels/topics, if yes then true otherwise false)
-type [NB LR SVM]	(NB for NaiveBayse, LR for Logistic Regression, SVM for Support Vector Machines)

For instance:

-trainData C:/IR/trainData/ -testData C:/IR/test-with-labels/ -labeled true -type SVM

It's important to set following VM Arguments: -Xss400m -Xms2g -Xmx4g -XX:-UseGCOverheadLimit

This project uses one data structure class from Breeze: breeze.linalg.SparseVector.

## General Classification Information

All 3 classification models are one-vs-all approach. All 3 classification are using StopWords (assignment2.StopWords.scala) and Stemming (com.github.aztek.porterstemmer.PortStemmer.scala)

## Naive Bayse

Class: assignment2.naivebayse.NaiveBayseClassification.scala

In a first pass the assignment2.index.IndexBuilder collects all relevant information from train data, such as nr of documents, topic counts, topic length (total number of tokens for each topic) and topicTfIndex ( collection frequency for each topic ), and puts it in Memory.

In classification step:

NaiveBayseClassification goes over test data and for each document it computes for all topics the probability by using the information prepared in first step from memory.

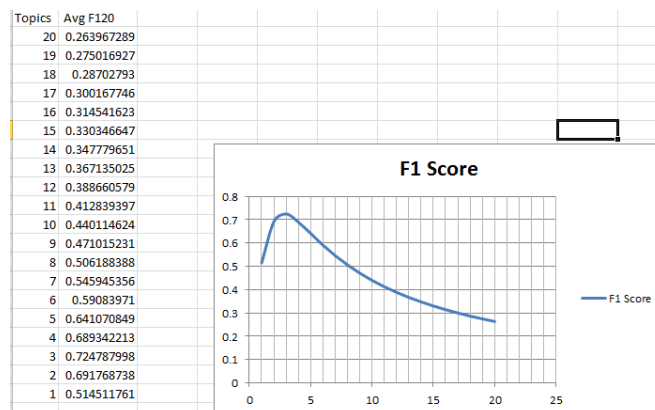
$$c^*(d) = \arg \max_c \left[ \log \hat{P}(c) + \sum_w \text{tf}(w; d) \log \hat{P}(w|c) \right]$$

$$\hat{P}(c) = \frac{\#\{d : d \in c\}}{\#\{d\}}$$

$$\hat{P}(w|c) = \frac{\sum_{d \in c} \text{tf}(w; d) + \alpha}{\sum_{d \in c} \text{len}(d) + \alpha \#\{w\}}$$

inclusive la place smoothing

NaiveBayseClassification selects always top 3 (highest probability that topic occurs in document). In screen below it is shown why I have chosen 3 topics. It results in the best F1 score.



Best avg result using Naive Bayse:

Precision= 0.7194131709337228 , R= 0.7333289634183215 , F1= 0.7020213093418058

## Logistic Regression

Class: assignment2.regression.LogisticRegressionClassification.scala

In a first pass LogisticRegressionClassification uses assignment2.index.FeatureBuilder to collect separately all features (term frequencies) from train and test data.

In training step:

For each topic (theta) in train data SVM goes over a number (NUMBER\_OF\_ITERATIONS=10'000) of randomly picked train features and updates vector theta. There is also a weight factor added for imbalanced classes.

$$\text{weight by } \alpha^- = \frac{|\{d:c(d)=0\}|}{\#docs} \quad \text{weight by } \alpha^+ = \frac{|\{d:c(d)=1\}|}{\#docs}$$

$$\theta_{t+1} = \theta_t + \eta_t \begin{cases} \alpha^-(1 - P(c = 1|d_t; \theta_t)) \vec{d}_t & \text{if } c(d_t) = 1 \\ -\alpha^+ P(c = 1|d_t; \theta_t) \vec{d}_t & \text{if } c(d_t) = 0 \end{cases}$$

In classification step:

For each test document LogisticRegression goes over all topic thetas and applies classification function below to see if topic occurs in document or not. Threshold is set to 0.6, which seemed to be best value after playing around with different values.

$$\frac{1.0}{1.0 + \exp(-\langle \vec{x}, \vec{\theta} \rangle)} \geq threshold$$

Best avg result using Logistic Regression:

Precision= 0.11602640607536213, Recall=0.7146578813051772, F1= 0.18199537485321735

## SVM - Support Vector Machines

Class: assignment2.svm.SvmClassification.scala

In a first pass SvmClassification uses assignment2.index.FeatureBuilder to collect separately all features (term frequencies) from train and test data.

In training step:

For each topic (theta) in train data SVM goes over a number (NUMBER\_OF\_ITERATIONS=10'000) of randomly picked train features and updates vector theta.

$$\vec{\theta}_{t+1} = \begin{cases} (1 - \eta_t \lambda) \vec{\theta}_t & \text{if } y_{I(t)} \langle \vec{\theta}, \vec{x}_{I(t)} \rangle \geq 1 \\ (1 - \eta_t \lambda) \vec{\theta}_t + \eta_t y_{I(t)} \vec{x}_{I(t)} & \text{otherwise} \end{cases}$$

In classification step:

For each test document SVM goes over all topic thetas and applies classification function below to see if topic occurs in document or not. Only those topics are selected which are greater equal then 0.

$$\langle \vec{\theta}, \vec{x} \rangle \geq 0$$

Best avg result using SVM:

Precision= 0.776048635620022, Recall= 0.35593345731025067, F1= 0.46301152014155816