Information Retrieval - Assignment 2

# Main program

assignment2.Main 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

# General Classification Information

All 3 classification are using one-vs-all approach.

All 3 classification are using StopWords (assignment2.StopWords.scala) and Stemming (com.github.aztek.porterstemmer.PortStemmer.scala)

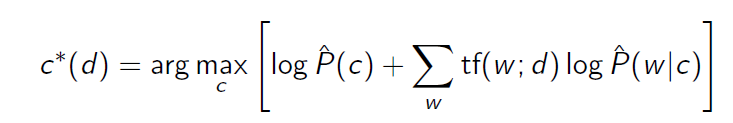
For all 3 classification top 3 topics are returned. Why 3. After an evaluation over ….

# Naive Bayse

Class: assignment2.naivebayse.NaiveBayseClassification.scala

In a first pass a 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 a second pass NaiveBayseClassification goes over test data and for each document it computes for all topics the probability.



|  |  |
| --- | --- |
|  | incl. la place smoothing |

Best result using Naive Bayse:

Precision= 0.7194131709337228 , R= 0.7333289634183215 , F1= 0.7020213093418058

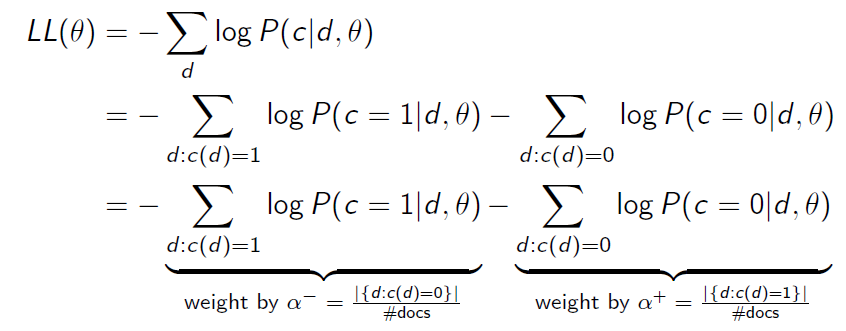
# Logistic Regression

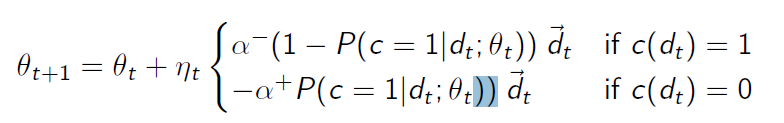
Class: assignment2.regression.LogisticRegressionClassification.scala

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) of randomly picked train features and updates vector theta.





In prediction step:

For each test document LogisticRegression goes over all topic thetas and computes logistic. Top 3 scores are returned.

TODO TODO Logistic Loss Function!!!!!

Best result using Logistic Regression:

Precision= 0.22266917745103393 , Recall= 0.3768042188256521 , F1= 0.2683379337145363

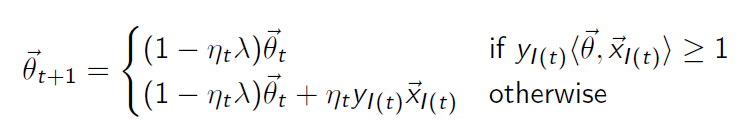
# SVM - Support Vector Machines

Class: assignment2.svm.SvmClassification

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

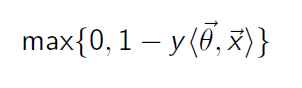
In training step:

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



In prediction step:

For each test document SVM goes over all topic thetas and computes hingeLoss. Top 3 scores are returned.



Best result using SVM:

Precision= 0.5904149471800447 , Recall= 0.601059109400312 , F1= 0.5744665782352627