IR – Document Classification

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 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 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 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.

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

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

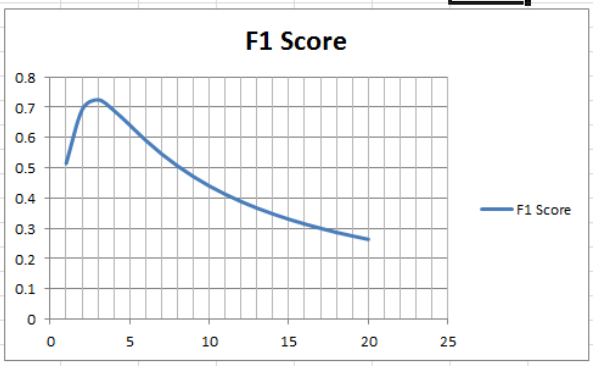
$$\hat{P}(w|c) = \frac{\sum_{d \in c} \mathrm{tf}(w;d) + \alpha}{\sum_{d \in c} \mathrm{len}(d) + \alpha \#\{w\}}$$

incl. la place smoothing

Cyrill Zadra

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 is producing the best F1 score.

Topics	Avg F120
20	0.263967289
19	0.275016927
18	0.28702793
17	0.300167746
16	0.314541623
15	0.330346647
14	0.347779651
13	0.367135025
12	0.388660579
11	0.412839397
10	0.440114624
9	0.471015231
8	0.506188388
7	0.545945356
6	0.59083971
5	0.641070849
4	0.689342213
3	0.724787998
2	0.691768738
1	0.514511761



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. It is also a weighting factor implemented for imbalanced classes.

$$
\begin{aligned}
LL(\theta) &= -\sum_{d} \log P(c|d,\theta) \\
&= -\sum_{d:c(d)=1} \log P(c=1|d,\theta) - \sum_{d:c(d)=0} \log P(c=0|d,\theta) \\
&= \underbrace{-\sum_{d:c(d)=1} \log P(c=1|d,\theta)}_{\text{weight by } \alpha^- = \frac{|\{d:c(d)=0\}|}{\#\text{docs}}} \underbrace{-\sum_{d:c(d)=0} \log P(c=0|d,\theta)}_{\text{weight by } \alpha^+ = \frac{|\{d:c(d)=1\}|}{\#\text{docs}}}
\end{aligned}
$$

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

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.

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

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.

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

In classification step:

For each test document SVM goes over all topic thetas and 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.

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

Best avg result using SVM:

Precision= 0.776048635620022, Recall= 0.35593345731025067, F1= 0.46301152014155816