Technische Universität Berlin

Report

Advanced Information Management III: Scalable Data Analytics and Data Mining

A Comparison of Online learning Naïve Bayes Classifier on RSS Feeds using SPARK

Authors:

Ahmet Anil Pala

Franziska Adler

February 4, 2015

Table of Contents

1 Theory

1.1 Motivation

In Text Classification Naïve Bayes Classifiers are popular due to their simplicity and find application in the field of spam email detection and descovery of specified web content (Ertel, 2008, p. 225). Classifier aim the prediction of a categorie like spam or ham on the basis of previous examples. This can be easily realized with a probalistic classifier like Naïve Bayes assuming that the data distribution is static over time, all data is always accessible and query time does not play a major role. For adapting real world problems and their dynamics this might not be sufficient. The developement of the internet and the need of processing huge amounts of available data and answering queries in time-critical situations needs to be taken in account for handeling model construction and model updating. In Online Machine Learning different methods for dealing with sequentially arriving data and updating the model with new incoming observations exist. This is called "Concept Drift" and referes to the problem of the variation of statistical properties (Astudillo and Gonzalez, 2013). Also aspects of scalability might be considered since Concept Drift occurs mainly in situations of large data qantities arriving via stream (Tsymbal, 2004, p. 4) and the underlying data processing system is distibuted.
In this report we are using RSS feeds from BCC for classification on a distributed system. Focus is the evaluation of several Naïve Bayes classifiers which implement the online learning paradigm by using streaming techniques. The evalutation concentrates on the handling of Concept Drift. In a first theoretically part of this report challanges, concepts and methods as well as evaluation measures are described. The second part consits of the explanation of our implementation using SPARK to show how the different classifier approaches work. Furthermore the evaluation of their performances are analysed and presented.

1.2 Naïve Bayes classifier

As mentioned Naïve Bayes classifiers are probalistic classifiers which are simple but effictive in text classification. They operate on the basis of the Bayes Theorem and assume independance of features. Here the feature values are normalized words frequencies occuring in a document. The probability of a word w belonging to class $y \in Y$ is given by $P(y|w_i)$ and can be reformulated with Bayes Theorem to its conditional probality and a priori probability $P(y)$ of class y. Both can be derived from frequencies of documents and words.

$$P(y|w_i) \propto P(y)P(w_i|y)$$

The computation of the joint probability over a documents features will give the probability for a document d belonging to a certain class y. Instead of multiplication the logarithmen is used to avoid underflow by multiplying with zero:

$$P(y|d) \propto P(y) \prod_i P(w_i|y), \text{ bzw.: } P(y|d) \propto logP(y)log \sum_i P(w_i|y),$$

Applying for all classes will give the decision rule which categorize a new text document according to its most likely class which is the one with the highest joint probability value.

$$\arg\max_y \{logP(y)log \sum_i P(w_i|y)\}$$

1.3 Challanges/ Concepts

texttexttext

1.3.1 Concept Drift

texttexttext

1.3.2 Streaming

texttexttext

1.3.3 Distributed Systems

With the Internet data availability seems not to constitude a problem anymore. Aspects of storing, processing and mining those data amounts were reconsidered in the past years and led to the raise of new technologies and the mostly unloved buzzword Big Data. A single CPU can not accomblish the processing of those data quantities, especially if the algorithms are computationally intensive like most of the prediction algorithms from the field of Machine Learning which are widely used in Datamining. To handle that the parallel execution of calculations is spread over a cluster of machines with a underlying system responsible for scheduling, load balance and fault tolerance (**?**, p. 10) . The forerunner of this cluster model was Hadoop, now several frameworks which extend this work are developed and allow wider functionality and significant performance improvements. Even though those frameworks are centered around data processing and ease of use software developers need to be aware of the distributed nature of such systems and parallel computation execution. This might be clear to those coming from a Distributed Systems background but challanging for data engineers or developers related to Datamining who are more used to sequencial arrangement of data processing.

1.4 Methods

1.4.1 Sliding Window

texttexttext

1.5 Evaluation Measurse

texttexttext

2 Empirics

2.1 Data

texttexttext

2.2 Software

For realising the stated approaches we are using Apache Spark and MLlib. Apache Spark is a open source processing engine for parallel processing of large scale data. Spark works on top of a distributed storage and uses a cluster manager cluster manager like Yarn or Mesos. For the purpose of this project the locale storage was used as simulated distributed storage which is integrated in spark for developing and testing reasons.
While the Spark core handels scheduling and load balancing the on top working modules provide additional functionality for streaming, Machine Learning algorithims and graph computation. The main programming abstraction in Spark is called RDD (resilient distributed dataset), a collection of objects partitioned across different machines for parallel programming. Beside map and reduce parallel operations on RDDs like filter, collect and foreeach etc. are provided. For shared variables broadcast variables and accumulators can be used.
TODO: Something about Streaming
TODO: MLlib if used

2.3 Data

texttexttext

2.4 Implementation

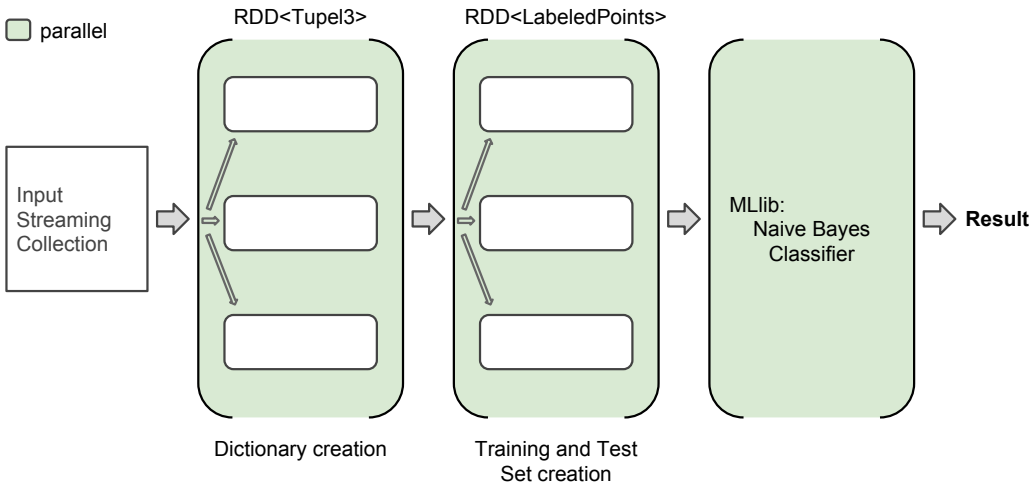

Figure 1: Offline Workflow

2.5 Evaluation

texttexttext

2.6 Results

texttexttext

2.7 Summary

texttexttext

References

Wolgang Ertel. *Grundkurs Künstliche Intelligenz: Eine praxisorientierte Einführung*. Vieweg + Teubner, Wiesbaden, 2008.

César A. Astudillo and Javier I. Gonzalez. Concept drift detection using online bayesian classifier. In *Proceedings of the XXXII International Conference of The Chilean Computer Science Society*, 2013. URL: http://jcc2013.inf.uct.cl/wp-content/proceedings/SCCC/Concept Drift Detection Using Online Bayesian Classifier.pdf,last accessed on 22.01.2015.

Alexey Tsymbal. The problem of concept drift: Definitions and related work. Technical report, Department of Computer Science, Trinity College Dublin, Ireland, 2004. URL: https://www.cs.tcd.ie/publications/tech-reports/reports.04/TCD-CS-2004-15.pdf, last accessed on 21.01.2015.