MACHINE LEARNING FOR DATA STREAMS

BAYESIAN NETWORKS

NAÏVE BAYES

- See "A Survey on Supervised Classification on Data Streams" for information about naïve Bayes for data streams.
    - ➢ The naïve Bayes approach is intrinsically incremental and can be easily updated on-line. To do so, **it is sufficient to update some counts in order to estimate the univariate conditional probabilities**. These probabilities are based on an estimate of the densities; this problem can be viewed as **the incremental estimation of the conditional densities**
    - ➢ The naive Bayes classifier uses conditional probabilities estimates. **These estimates are usually done after a discretization of the explicative variables**. In the stream mining context this first step needs dedicated methods as most off-line methods usually need to load all the data in memory. In the literature, two kinds of publication related to incremental discretization can be found. **The articles related to data-mining are not numerous** but the literature related to database provides much more articles because the database systems (DBMS) need to have good estimates of the data distribution.

1. **Incremental Weighted Naive Bayes Classifiers (2014):**
    - o Presents a new method based on a graphical model close to a neural network which computes the weights on the input variables using a stochastic estimation. The method is incremental and produces an Weighted Naive Bayes Classifier for data stream. This weighting produces a single model close to an \averaged naive Bayes classifier" a "weighted naive Bayes classifier".
    - o Its complexity to predict is very low which makes it suitable and widely used for stream mining prediction (only depends on the number of explanatory variables).
    - o Its memory consumption is also low since it requires only one conditional probability density estimation per variable.
    - o The method used to update the weights is the one used usually for a standard back-propagation and three main parameters have to be considered (Lecun et al. (1998)) in the case of stochastic gradient descent: (i) the cost function; (ii) the number of iterations; (iii) the learning rate.
    - o The conditional density probabilites used as inputs on their graphical model are estimated using three methods: their two layers incremental discretization method based on order statistics as described in (Salperwyck

and Lemaire (2013)) (ii) a two layer discretization method "cPiD" which is a modified version of the PiD method of Gama (Gama and Pinto (2006)) (iii) and a Gaussian approximation.

- o Their WNB approach uses a low amount of memory thanks to the two level approach to estimate the conditional densities, and is purely incremental thanks to the graphical model and the stochastic gradient descent to estimate the weights.

- o Comparison (Weighted Naive Bayes trained online with the two level discretization method which uses GKClass (level 1) and the MODL discretization (level 2) and our method based on the graphical method to estimate the weights.)with naive Bayes (NB) trained offline with the MODL discretization and all the data in memory; (2) an Averaged Naive Bayes (ANB) trained offline with the MODL discretization and a Naive Bayes trained online with the two level discretization method which uses GKClass (level 1) and the MODL discretization (level 2):
    - It improves the performance compared to the non-weighted version and is close to the off-line averaged version of the naive Bayes classifier.
    - They think they can improve the results in future work:
        - Use the GK Class summaries as "mini-batch" (Cotter et al. (2011)) and do several iterations to speed up the gradient descent.
        - Use an adaptive learning rate: high at the beginning and low after, or to take into account the error rate as in (Kuncheva and Plumpton (2008))
    - See results for more specific information.

2. **RGNBC: Rough Gaussian Naïve Bayes Classifier for Data Stream Classification with Recurring Concept Drift (2016):**
    - o Two new contributions are made to handle the challenges of recurring concept drift. The first contribution is to **utilize the rough set theory for detecting the concept drift**. Then, gaussian naïve classifier is modified mathematically to **handle the dynamic data without using the historic data**. Also, the classification is performed using the posterior probability and the objective function which considers the multiple criteria.
    - o The problem considered here is to perform the data stream classification by considering the recurring concept drift.
    - o The important problem considered here is the recurring concept drift where the dimension of the features is varied for every time.
    - o **Methodology**: At first, input data stream is directly read out by the proposed method at every time and the GNBC is built up initially by constructing the information table. Then, for new incoming data stream, concept of change (COD) is detected using rough set theory which have the accuracy of approximation. Once the COD is detected, the GNBC

model is updated based on the new mathematical model developed in this work. This model updates the existing model based on the new data stream without storing the data tuples. Also, the proposed method utilizes the feature evaluation function and information table to handle the recurring concept drift.

- The Gaussian naïve bayes classifier performs the classification using two important steps such as, construction of model and classification. At the model construction, the information table is constructed by including mean and variance of the every attributes. In the classification stage, the posterior probability is computed to find the class label of the input data. The classification is performed using the updated GNBC model by considering recurring concept drift.
- The updating of model happen only if the newly arrived data have the **concept drift** which means that the boundary of the data is changed heavily either inside or outside. The dynamic updating of GNBC model has three important steps, such as, detecting COD by rough set theory, updating of GNBC model and updating of important features.
- The detecting concept change is performed using the lower approximation and upper approximation which is taken from the rough set theory.
- Conclusions section.
- Comparison with MReC-DFS (Ensemble):
  - MReC-DFS utilized the Naive Bayes (NB) algorithm with ensemble weighting mechanism to handle the recurring concept drift for data stream classification. In their method, the ensemble weight mechanism considered the accuracy and error values. But, due to the dynamic nature of data, classes and data samples are not constant over the period of time. So, considering accuracy and error may affect the performance of the classification if one class attribute has bigger data samples. So, the multiple objective criteria like, sensitivity, specificity should be included to ensemble weighting.
  - This graph ensured that the proposed RGNBC model outperformed the existing algorithm in two datasets even for the various sizes of chunks.

3. Classification Of Massive Data Streams Using Naïve Bayes (2018)

- This paper implements highly efficient and popular algorithm "Naïve Bayes Algorithm" on huge complex data to acquire knowledge. An addition reduction technique is used to eliminate similar data which in turn reduces the computation time, demands less memory space comparatively and enhances the performance of Naïve Bayes Algorithm. HDFS provides a platform with clusters of distributed systems which provides the functionality of storage and the processing. The unstructured or semi-structured data are reconstructed to required file format by converting into CSV, followed by converting into ARFF. Weka tool is a machine learning tool which is used to apply the proposed algorithm on the massive data streams.

- Initially, all the data streams obtained from the client are stored in HDFS (Hadoop Distributive File System). Here the data streams are bisected equally with respect to the cluster of distributed systems available in the HDFS platform. Once the data streams are stored, they are converted into Comma-Separated Systems format followed by converting them into ARFF file format. A machine learning tool is used to classify each attribute and place it in the respective attribute type. Then by using a reduction technique, the duplicate data are eliminated which intern provide only unique data stream. Now by using Naïve Bayes Algorithm, the probability of each attribute in the data stream is estimated with respect to its outcome. With the acquired probabilities of the individual attribute, the system is able to predict the non-existent attribute set. The result is provided in the form of a ratio that provides the outcome of the non-existent stream that was given by the client.
- The accuracy of the Naïve Bayes algorithm is much more than the existing system. It acquires knowledge from data in the form of probabilities which intern provides the output with high accuracy.
- Large data streams are high acceptable by the Naïve Bayes algorithm because it does not sort any data. Each and every attribute and its outcome is determined individually which in turn does not produce any noisy data.
- In the proposed system before applying Naïve algorithm on the large data stream, a reduction technique is used which eliminates the duplicate data and provides data which unique for further process.

SEVERAL TYPES OF BAYESIAN CLASSIFIERS

1. **Mining Complex Models from Arbitrarily Large Databases in Constant Time (2002):**
   - Mentioned by the proposal "Voting Massive Collections of Bayesian Network Classifiers for Data Streams" (Bayesian networks): The Hoeding criterion can be applied to Bayesian network learning (this proposal) thereby incrementally learning the network structure. Still, many statistics need to be kept in memory thereby slowing down the number of instances that can be processed per second.

2. **Voting Massive Collections of Bayesian Network Classifiers for Data Streams (2006)**
   - In this article, they analyze how to vote a collection of forest augmented naive Bayes (FAN) under some mild conditions. The number of FANs is exponential in the number of attributes making it seemingly impossible to use in a practical way. However, they show how to calculate the vote in quadratic time instead of exponential.
   - Single Bayesian networks tend to be less accurate classifiers than ensembles like AODE. This last general observation leads us to try to use a collection of a **large number of Bayesian networks**.
   - Conclusions sectos

- o Comparison between Bayesian classifiers:
  - Batch mode experiments (small datasets) were performed with naive Bayes, TAN, AODE, FANC, SPC, SPCr and TC. Classification performance of TAN and AODE is better than naive Bayes and that AODE tends to perform similar to TAN but with lower train time. Further, of the collections (FANC, SPC, SPCr and TC) classified all consistently better than naive Bayes.
  - Better performance comes from being able to capture more dependencies, while diminished performance for very small datasets is caused by the increased variance in parameter estimation since for the collections a much larger number of parameters need to be established. Training and testing times as well as memory requirements are consistently larger than naive Bayes. Compared with TAN andAODE, FANC classification performs is about the same, SPC and TC slightly better and SPCr considerably better overall.
  - In short, though SPCr classification performance is the best overall, this comes at a cost of training time. If memory is not a restriction, TC performs best; but if it is, SPC tends to perform quite well.
  - Large datasets: For the naive Bayes data source, all algorithms perform very similar. For the TAN data source, naive Bayes is consistently outperformed by the other algorithms and SPC and TC outperforms the other algorithms. For the BAN data source results are similar to the TAN data source, except that TC outperforms SPC. This leads to two conclusions; the collections perform well on large data sets and with increasing complexity of the concept to be learned increasingly complex collections help in performance but cost in memory and classification time. This allows for balancing between collection performance and computational complexity.

3. **An effective pattern-based Bayesian classifier for evolving data stream (2018) NOT READ YET**
   - o