MACHINE LEARNING FOR DATA STREAMS

BAYESIAN NETWORKS

NAÏVE BAYES

- Although Naive Bayes classifier could be applied on evolving data stream, it cannot adapt to concept drift directly. (Paper "An Effective pattern-based Bayesian classifier for evolving data stream").
- It is common practice, for Naïve Bayes classification, to discretise all numeric features so that all features for NB classification are categorical. This leads to a straight forward implementation of (1) (Paper "Fast Feature Selection for Naive Bayes Classification in Data Stream Mining (2013)").
- 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**
    - o 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. **Fast Feature Selection for Naive Bayes Classification in Data Stream Mining (2013):**
    - o This paper reports **studies that were conducted to identify efficient computational methods for selecting relevant features for NB classification based on the sliding window method of stream mining**. The paper also provides experimental results which demonstrate that **continuous feature selection for NB stream mining provides high levels of predictive performance compared to once-off feature selection**.
    - o In order to identify irrelevant features, methods for measuring correlations between qualitative features need to be employed. One such

- method is the use of the **symmetrical uncertainty (SU) coefficient** which is defined in terms of the entropy function (same interpretation as **Pearson's product moment correlation coefficient** for quantitative variables).
- **Contingency tables** can be used to **store data (frequencies) for the computation of the SU coefficients for feature selection** as well as the **computation of the probability terms for Naïve Bayes classification**. Various statistical measures can be derived from a contingency table.
- in order to characterise the association (correlation) between X and C.
- A common approach to the implementation of the Naïve Bayes classifier is to use two tables for the model. One table stores the class prior probability estimates Pr(c j) while the second table stores the likelihood estimates Pr(xi | c j) for each feature value. The use of the same computational data structures for the feature selection and Naïve Bayes computations results in fast and efficient implementation of feature selection for Naïve Bayes classification. This approach is especially desirable for stream mining, and it is the approach that was used for the
- studies reported in this paper.
- The objectives of the studies reported in this paper were **to establish whether continuous feature selection for stream mining using the Naïve Bayes classifier and the sliding window technique leads to improved predictive performance compared to a once-off feature selection approach**.
- Three alternative approaches to incremental Naïve Bayes classification were used for the feature selection studies:
  - To add newly arriving instances to the training dataset for the model without removing old instances.
  - To use a sliding window where a small number of old instances are removed whenever new instances are added to the training dataset for the model.
  - Use a sliding window where a large number of old instances are removed whenever new instances are added to the training dataset.
- Two alternatives for feature selection were studied:
  - Predictive features were selected at the start of the mining process, using the initial batch of training data. These features were used for NB classification for all subsequent time windows.
  - To conduct feature selection at the beginning of each time window.
- Two main data structures were implemented for stream mining. The first data structure is the list of features where each entry in the list stores a description of a feature as (name, type, category count, categories, SU coefficient, relevant). The second data structure is a list of contingency tables.
- Numeric features were each discretised into 10 intervals **using equal-width binning**.
- Cohen [18] has recommended that correlations with a magnitude less than 0.1 have no practical significance. For this reason, **features with an SU**

coefficient less than 0.1 were considered to be irrelevant and **were excluded from the classification process**.

- m estimate of probability solves the problem of having cells with zero
- counts or vey small counts in a contingency table.
- A fast method of feature selection for Naïve Bayes stream mining has been presented in this paper. This method uses the same up-to-date data, stored in contingency tables, for both feature selection and Naïve Bayes classification.
- One weakness of the Naïve Bayes algorithm is to due to the inclusion of irrelevant features. Irrelevant features have a very small or no correlation with the class variable, and so, have very little or no predictive power. The second weakness for Naïve Bayes classification is that for some xi values that appear in the training data, the frequency counts for these values may be too small to produce a reliable estimate of Pr(xi | c j).
- Comparison between the alternatives previously mentioned:
    - The experimental results reported in Section IV have indicated that for the dataset used in the experiments, continuous feature selection leads to improved predictive performance.
    - Table IV.

2. **Incremental Weighted Naive Bayes Classifiers (2014):**
    - 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".
    - 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).
    - Its memory consumption is also low since it requires only one conditional probability density estimation per variable.
    - 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.
    - 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.
    - 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.

- 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.

3. **RGNBC: Rough Gaussian Naïve Bayes Classifier for Data Stream Classification with Recurring Concept Drift (2016):**
    - 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.
    - The problem considered here is to perform the data stream classification by considering the recurring concept drift.
    - The important problem considered here is the recurring concept drift where the dimension of the features is varied for every time.
    - **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.

4. 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.

TAN

1. **Incremental Learning of Tree Augmented Naive Bayes Classifiers (2002):** **No aparece en los surveys (1), (2), (3) y (4)**
    - They use an **incremental algorithm** for **learning tree-shaped Bayesian Networks** to obtain an **incremental Tree Augmented Naive Bayes classifier**.
    - an incremental Tree Augmented Naive Bayes classifier
    - The algorithm rebuilds the network structure from the Branch which is found to be invalidated, in some sense, by data.
    - They show,that the incremental version obtains most of times **the same accuracy than the batch classifier while saving computational time**.
    - Their incremental approach revises an already learnt tree-shaped Bayesian Network without processing the old data instances. Roughly speaking, they state that **new data invalidates the old tree-shaped structure when the branches are not anymore in decreasing cost order**. Then, **the tree is rebuilt from the first branch found in a bad position into the order**. In this way, our algorithm, can both detect the need of updating and update the current network.We will call our proposal **ACO heuristic** (Arches in Correct Order).
    - In this paper, they give a direction to all the branches of the tree.We take as the root of the tree one of the nodes of the first branch and the direction of the branches introduced afterwards goes from the node already into the structure to the one recently introduced.
    - They divide their algorithm into two steps. In the first step, the algorithm **calculates the sufficient statistics for both old D and new D' data instances**, and in the second, it **revises the tree structure according to the new sufficient statistics**.
    - In the first step of the algorithm, we assume that sufficient statistics of the old data set D are stored. Thus, in order to recover the sufficient statistics, of the whole set of data instances (including the new ones) the algorithm does not need to go through the old ones.

- The second step uses a heuristic which decides to update the structure only when the arches are not in correct order.
- They use Chow and Liu's algorithm to build the tree for the very first time and to rebuild it from a certain arch that has not the highest cost among all candidates arches present in the former structure.
- Another cause which may influence the time spent is the order in which the instances are presented. Usually, when similar instances are presented consecutively, the network structures learned from data are **not good models of the probability distribution of the entire database**. Thereof, the incremental algorithm spends more time as it must update the network structure.
- Their algorithm is very reactive, that is, it is able to quickly detect changes in new data and to correctly update the structure.
- The major benefit of our incremental proposition is that it saves computing time.
- Compared to batch CL algorithm:
  - ACO approximates the best solution very well.
  - Their incremental classifier behaves as well as the batch one in accuracy.
  - If the accuracy curves of the three different orders are compared, the accuracy is best when data is randomly presented while it is worse when similar instances are presented consecutively.

SEVERAL TYPES OF BAYESIAN CLASSIFIERS

2. **Mining Complex Models from Arbitrarily Large Databases in Constant Time (2002): No aparece en los surveys (1), (2), (3) y (4)**
   - Their method falls in the general category of **sequential analysis**.
   - This paper proposes that the time used by a learning algorithm should **depend only on the complexity of the model being learned, not on the size of the available training data**.
   - Our framework further allows transforming the algorithm to work incrementally, to give results anytime, to fit within memory constraints, to support interleaved search, to adjust to time changing data, and to support active learning.
   - They present a framework for semi-automatically scaling any learning algorithm that performs a discrete search over model space to be able to learn from an arbitrarily large database in constant time.
   - They propose a scaling-up method that is applicable to essentially any induction algorithm based on **discrete search** (at each search step a number of candidate models or model components are considered, and the

best one or ones are selected based on their performance on an iid sample from the domain of interest).

- The result of applying the method to an algorithm is that its **running time becomes independent of the size of the database**, while the decisions made are essentially **identical to those that would be made given infinite data**.
- The method works within **pre-specified memory limits** and, as long as the data is iid, **only requires accessing it sequentially**. It gives **anytime results**, and can be used to produce batch, **stream**, time-changing and active-learning versions of an algorithm.
- We apply the method to learning Bayesian networks, developing an algorithm that is faster than previous ones by orders of magnitude, while achieving essentially the same predictive performance.
- By applying their method, HGC's running time can be made independent of the size of the training set.
- 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.
- HGC and VFBN1 share a major limitation: they are unable to learn Bayesian networks in domains with more than 100 variables or so. The reason is that the space and time complexity of their search increases quadratically with the number of variables (since, at each search step, each of the d variables considers on the order of one change with respect to each of the other d variables).
- This complexity (previous bullet) can be greatly reduced by noticing that, because of the decomposability of the BD score, many of the alternatives considered in a search step are independent of each other. That is, except for avoiding cycles, the best arc change for a particular variable will still be best after changes are made to other variables. The VFBN2 algorithm exploits this by carrying out a separate search for each variable in the domain, interleaving all the searches. VFBN2 takes advantage of our method's ability to reduce I/O time reading a data block just once for all of its searches.
- Future work.
- Compared to VFBN1, VFBN2 and true networks:
  - VFBN2 is thus able to learn on domains with many more variables than HGC or VFBN1. Further, HGC and VFBN1.
  - Further, HGC and VFBN1perform redundant work by repeatedly evaluating O(d2) alternatives, selecting the best one, and discarding the rest. This is wasteful because some of the discarded alternatives are independent of the one chosen, and will be selected as winners at a later step. VFBN2 avoids much of this redundant work and learns structure up to a factor of d faster than VFBN1 and HGC.

        - The systems achieve approximately the same likelihoods for the small networks. Further, their likelihoods were very close to those of the true networks, indicating that our scaling method can achieve high quality models. VFBN1 and VFBN2 both completed all four runs within the allotted 5 days, while HGC did so only twice. VFBN2 was an order of magnitude faster than VFBN1, which was an order of magnitude faster than HGC.
        - For the large networks, we found VFBN2 to improve significantly on the initial networks, and to learn networks with
        - likelihoods similar to those of the true networks.

3. **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
    - 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.

4. **An effective pattern-based Bayesian classifier for evolving data stream (2018)_PBDS:**
    - An effective Pattern-based Bayesian classifier for Data Stream (PBDS) is proposed. Due to the space limitation, a **window-based model for frequent pattern mining should be selected**. To their knowledge, there are three basic stream processing models: landmark window model, sliding window model and damped window model. The sliding window considers a fixed number of stream records, compared with landmark window model, it detects changes in the properties of stream records faster (e.g. concept drift), and does not have to assign different weights to stream records (as damped window model), so the **sliding window model is employed for efficient pattern mining in this paper**.
        - First, they propose a **summary data structure for compact representation of data**, and to find patterns more efficiently for each class (simple but effective summary data structure for each class based on the sliding window model is proposed, which means the probability approximation for PBDS is separately tailored to each class). An effective summary data structure for processing records over sliding windows.
        - A data-driven lazy learning strategy is employed to discover local frequent patterns for each test record (since the computing of lazy classifier is performed on a demand driven basis, only the "useful"portion of the training data (in a sliding window) is mined for generating patterns applicable to the test instance, which increases the chance of achieving the most significant patterns that are useful for classifying the test case). It can provide a more complete description for each record and avoid the extra computation adopted by eager algorithms
        - Greedy search and minimum description length combined with Bayesian network are applied to evaluating extracted patterns (for a pattern-based Bayesian classifier, a set of long and not overlapped patterns that fully covers the given test case should be found, so it could be considered as a set covering problem essentially. As set covering problems are NP-hard in general, a heuristic pattern extraction mechanism is adopted, which is based on greedy search and the minimum description length (MDL) for the Bayesian classifier to reduce the generation of candidate itemsets, and to ensure the fitness between extracted patterns and original data records).

    SEE CONTRIBUTIONS IN PAGE 2.

- A promising approach to Bayesian classification is based on **exploiting frequent patterns**, i.e., patterns that frequently occur in the training data set, to estimate the Bayesian probability. Pattern-based Bayesian classification focuses on building and evaluating reliable probability approximations by exploiting a subset of frequent patterns tailored to a given test case. (Definition from the paper “Enbay: a novel pattern-based Bayesian classifier”).
- Most of the exiting classifiers for non-stationary data stream pay more attention to adaptive tree models, decision rules, ensemble algorithms and kNN. However, the above algorithms ignore the potential of incorporating pattern in graphical models. (Not an inconvinient for this proposal, but for others).
- Sometimes the discovered patterns of CFI i are not enough to cover test record since some items are infrequent. Hence it arise the zero frequency count or non-frequency problem. For Bayesian classifier, multiplied by zero probability will ignore the contributions of other patterns, so Laplacian smoothing is used to address this issue.
- Although PBDS is effective on concept drift data stream, outperforms most state-of-the-art classifiers on accuracy, there are some limitations. First, how to explore numerical- attributes-based patterns for classification directly is a big challenge for pattern Discovery. Second, PBDS is still time-consuming though a compressed tree structure is proposed. More advanced and efficient data storage system should be considered in the future. Third, the parameter setting (e.g. the size of sliding window N) affects the final result notably, another future direction of our work could involve in- vestigating how to create faster and more accurate pattern-based Bayesian classifier using adaptive sliding windows.

- Comparison with MOA default settings of Nave Bayes (NB), Naive Bayes Multinomial (NBM) [30] , k NN, k NNwithPAW (PAW) [22] , RuleClassifierNBayes (RCNB) [17] , HoeffdingTree (HT) [15] , and Accuracy Weighted Ensemble classifier [21] (AWE).:
  - Naive Bayes is a special case of PBDS.
  - Previous works on streaming data mining either focus on finding interesting patterns efficiently [10–13,25] or creating classifiers effectively [15–17,19,20,22] . However, none of them combines pattern with classifier practically. Unlike these listed methods, our objec-tive is to learn an efficient and effective Pattern-based Bayesian classifier for Data Stream (PBDS) that adapts to concept drift.
  - PBDS outperforms other classical methods.
  - In addition, our lazy PBDS outperforms other lazy classifiers (like k NN and PAW).
  - PBDS is more robust to concept drifts compared with other classifiers.