Concept Drift Handling of Imbalanced Data Streams: a Literature Review

R. Anderson, K. Chen and S. Jin

Abstract

We review the literature available around classification of imbalanced data in a streaming environment. First, we introduce the particular problems faced within streaming environments and with imbalanced classes. Through examining current research in the field, we explore the facets of classification in this environment: sampling techniques; approaches to detecting concept drift; base learners for building models to classify unseen data; and methods of utilising ensembles to improve classification accuracy. Finally, we examine evaluation of these methods, and the pertinent measures used.

1 Introduction

In recent year, as computing power and hardware technologies become more powerful and robust, its possible to store and process large volume of data. Such data sets which continuously and rapidly grow over time are referred to as data streams. Mining Data streams brings challenges to traditional data mining techniques, as most of the traditional data mining techniques cannot simply apply to data streams due to: 1. Traditional data mining techniques need process the data multiple times, usually works only on static dataset. 2. Data streams temporal locality or concept drift nature, which makes it more difficult for traditional data mining techniques to adapt, because the class distribution changes as the data stream evolving over time in the context of classification. Furthermore, its more challenging to improve the overall accuracy of classification if the classes distribution are imbalanced, which is especially important for many applications in some fields, such as fraud detection and network intrusion detection.

This literature survey gives an overall picture on handling concept drift in imbalanced data streams from different angles such as sampling techniques, ensemble techniques, change detection algorithms and adaptive learning algorithms,and evaluation methodologies of adaptive learning algorithms. The survey is organized as follows. In Section 1, we introduce the problem of concept drift and imbalance problem in data stream. Section 2 presents several sampling methods

that are commonly used in data stream mining and imbalanced problem. Section 3 presents several common drift detection algorithms. Section 4 discusses several popular classification learning algorithms that are commonly used for skewed data streams. Section 5 discusses ensemble techniques. Section 6 discusses the evaluation methodologies that are close to skewed concept-drifted data stream mining area. Section 7 concludes the survey.

When data comes in the form of streams, it is impossible to fit the entire data into machines memory for processing, hence only online processing is feasible for mining data streams. Online processing refers to the predictive models are trained incrementally. However, this only solves the computational issue. In a dynamically changing environment, concept drift refers to the underlying class distribution of data stream can change over time. For example, online shopping customers interest can change over time. To solve the problem of concept drift, predictive models needs to be updated online, which is also called online adaptive learning, this approach explicitly employs extra techniques to handle concept drift without changing the base learning algorithms, however, it is also possible to modify the base learning algorithms to handle concept drift, such as Hellinger Distance Decision Tree(HDDT[1].) There are two strategies for updating the learning model: 1. update the model at regular intervals without considering whether there is a change occurred; 2. detect a concept change before updating the model. In this survey, we mainly discuss the second approach. Well known change detection algorithms are: Statistical Process Control (SPC[2]), ADaptative WINDdowing (ADWIN[3]) , Fixed Cumulative Windows Model (FCWM[4]), Page Hinkley Test (PHT[5]) and FLORA[6]. Detailed review of these algorithms are in in section 4.

Imbalanced data set has a skewed class distribution, usually one majority class distribution, one or more minority class distribution. The problem has been well researched since 2000, but more recent researches started applying some approaches to data streams. Three different approaches are: 1. data level approach. This approach mostly applies sampling technique, which simply modifies the data set in order to rebalance the class distribution; 2. algorithm level approach, which adapted to deal with minority class to improve performance; 3. cost sensitive approach has a cost matrix to describe the cost of misclassifying a class instance. Among them, data level approach is mostly used one in the context of data streams. Sample Ensemble Framework is a representative one in this approach. Most of the data level techniques for handling concept drift combined with ensemble techniques to improve accuracy. Data level approach has advantages of not changing the algorithm. However, the data level approach suffers from the problems from resource and computation. On the other hand, algorithm level approach doesnt need sampling as a pre-processing step, thus doesnt have the problem from resource and computation.

2 Sampling methods

There are two categories of sampling techniques that can be used for concept-drifted stream mining, one category aims to keep the training set small for fast processing, representative ones are Sliding Windows and Reservoir Sampling[7]. Another category is to improve performance on the imbalanced data set, SMOTE[8](Synthetic Minority Over-sampling Technique), SERA[9](Selectively Recursive Approach) framework, MuSeRA[10](Multiple Selectively Recursive Approach), REA[11](Recursive Ensemble Approach) are among this category. Balancing training data is the most straight-forward approach for imbalanced concept-drifting data stream mining. Many sampling algorithms improved accuracy by under-sampling the majority class or over-sampling the minority class. In "A General Framework for Mining Concept-Drifting Data Streams with Skewed Distributions", Gao proved that the sampling technique can reduce error by collecting positive instances and keeping them in training set. Synthetic Minority Over-sampling Technique (SMOTE), the minority class is over-sampled by taking each minority class sample and introducing synthetic examples along the line segments joining any/all of the k minority class nearest neighbours, rather than by over-sampling with replacement. SMOTE forces decision region of minority class to become more general. SMOTE gives better classification accuracy in terms of ROC. SERA, MuSeRA, REA took the similar approach either over-sampling, under-sampling or combined both. Neighbourhood Cleaning Rule (NCL[12]) uses the Wilsons Edited Nearest Neighbor Rule (ENN[13]) to remove majority class instances. In [14], the authors suggested using Classifier Ensemble together with Sampling techniques can improve performance. Sampling techniques is used at data level, which may keep the learning algorithm unchanged, however it could bring computational and memory issues when applying to data stream mining.

3 Concept drift detection

Drift detection methods provide feedback to the learner by detecting when concept drifts have occurred. These methods may be independent of the learning model, or structurally embedded inside a classifying algorithm such as RCD (Goncalves & Barros, 2013). When a concept drift has been detected by the detector, the learner is modified or retrained on some set of instances. This allows the learner to adapt and respond to changes which may lead to a higher classification accuracy compared to blind or non-adaptive approaches. Gama et al suggest one advantage of this approach is that it provides additional information about the way data was generated (Gama, 2014). However this may create more false positives - particularly problematic for noisy data. A variety of change detection methods have been proposed - earlier work focused on sequential analysis techniques, but adaptive windowing and statistical methods are also popular approaches. I will summarise some techniques below and highlight recent work on detection methods that addresses the class imbalance problem.

The CUSUM method is a sequential analysis technique that detects changes based on evaluating the cumulative residue of the learner and testing if there is a significant departure from zero. When the cumulative value is greater than some threshold (lambda) then it indicates a drift has occurred. The Page-Hinkley test is a variation on the CUSUM method that is used to detect sudden shifts in Gaussian signals. An important application of this method is in signal processing. It records a minimum cumulative value in addition to the current cumulative value. Both The Page-Hinkley and the CUSUM methods only require a small amount of memory to process instances O(1).

The drift detection method (DDM) proposed by Gama et al records the current mean error rate (p_i), and error standard deviation (s_i) as well as the minimum values (p_{min}, s_{min}) as each new instance is seen (Gama et al, 2004). A normal distribution is used to set the thresholds as the number of errors approximates a Bernoulli distribution for samples sizes greater than 30. This means that there is at least a delay of 30 errors until a drift is detected (Wang et al). The warning threshold is defined as $p_{min} + 2 \times s_{min}$ and is updated with the arrival of new instances. A warning is raised when the current value of pi + si is greater or equal to the warning threshold and subsequent instances are stored in a warning window. If the error rate surpasses the drift threshold of $p_{min} + 3 \times s_{min}$, then the learner is retrained on instances stored in the warning window and the minimum values ($p_{min} and s_{min}$) are reset. An analysis done by Goncalves et al on real and synthetic datasets shows that DDM performed most favourably on datasets affected by gradual drifts based on accuracies and the average ranking. The Early Drift Detection method (EDDM) is a variation on the drift detection method proposed by Gama et al (2004) and was designed to handle gradual concept drifts (Baena-Garcia et al 2006). It differs in that it uses the distance of the errors rather than the error rate as a measure of change. The main strength of this algorithm is the ability to detect slow gradual shifts. On the other hand it may be less resistant to noise, and is much less effective for detecting sudden shifts. Hence it may have a high false alarm and miss detection rate for these datasets (as shown in Goncalves et als? experiments using datasets with abrupt drifts). Similar to DDM, this method may also have a delay in drift detection due to the assumptions of the normal distribution approximation. Other algorithms like FLORA use the overall accuracy of the learner as the measure of change, a large decrease in accuracy indicating the occurrence of a concept drift.

ADWIN is a popular change detection method based on comparing data distributions from two different windows via the use of an adaptive sliding window. The detection method works by finding two sub-windows of the window W that have significantly different means. When two significantly different sub-windows are found the algorithm indicates a concept drift has occurred, and the window size is decreased by dropping the older sub-window. Otherwise the window size is enlarged when no drift is detected. As the window grows when no shift is detected, there is no upper limit to the window size during long periods of stability

? this leads to an unbounded memory requirement (this problem is remedied in a later version of the algorithm). One limitation of the ADWIN approach is that it may require more memory than sequential or statistical approaches as it has a memory complexity of O(logW) - O(W) compared to constant time O(1) for the other approaches (Gama et al, 2014). However it may give more precise information about the location of the concept drift (Gama et al, 2014). In addition to this, it has strict guarantees on the rate of false positives and false negatives (Goncalves et al). The time complexity of the algorithm is also higher than the other approaches but both experiments by Goncalves and Gama show that the runtime is lower than all other algorithms used in the studies. The entropy-based approach also uses sliding windows and is based on a modified version of Shannon's entropy. It uses a sliding 2 window approach, adapting the window size when a shift is detected. When a shift occurs window size is set to the minimum (forgets old instances), and grows with each new instance until the upper limit is reached. Although there has been much work done on drift detection techniques there are few studies that attempt to address the class imbalance problem by modification of the drift detection method. Some of the more common approaches for dealing with imbalanced classes are the use of ensembles, and sampling methods such as bagging or boosting.

The PerfSim algorithm first trains the learner on a dataset, and produces a confusion matrix that summarises the performance of the learner. After receiving a second dataset (batch of instances) it tests the learner on the new dataset and also produces a summary matrix. The summary statistics are converted into vectors and are compared using a similarity measure (cosine similarity). This may decrease the effect of class imbalance in comparison to methods that only rely on the overall accuracy where the contributions from the majority classes outweigh the minority classes. This method is independent of the learning algorithm and may be paired with any learner. Antwi et al show that this method does not detect false drifts and performs well when evaluated against other current methods (VC and MMD). The DDM-OCI method is based on the DDM method proposed by Gama, but specifically attempts to deal with imbalanced classes. This technique uses the recall of the minority class as the measure of change. The warning and drift thresholds are set in a similar manor to DDM and EDDM.

4 Base learners

Supervised learning requires models to be trained from labelled data. When model-building, we need an approach that can use explanatory variables and classes in a training set of data to create a set of rules for classifying unseen instances. This approach is our base learner: our tool that will build our classifier to apply to unseen instances of our data [15].

In a streaming environment, however, we can only make one pass through our

data. Concept drift requires a learner that can be run at regular intervals, so as to understand the current distribution that our instances fall into. This requires a trade-off between time, accuracy and memory. A successful learner will be sufficiently accurate while running quickly and with manageable memory overhead. These requirements are much more important than in a static environment, where we have a finite amount of data and time to process it.

Decision trees have particular features which make them a good fit for classifying streams with concept drift. The C4.5 decision tree, and its predecessor, the ID3 tree, are very popular within the research community [16]. Their appeal lies in their simplicity at every node, a binary decision is made related to one of the explanatory variables, resulting in a divide-and-conquer approach which results in fast classification and low overhead for the model. This leads to some clear drawbacks only rectangular regions can be used to discriminate between classes in the feature space for instance. Splits are made through information gain measures, seeking to maximise the change in impurity of the nodes with each split. These trees will naturally overfit, if fully grown, but can be pruned to achieve a desired level of fit to the data. Their adaptability and speed make them a natural fit in classification of streams.

Domingos and Hulten [17] highlight a major concern with C4.5, ID3 and CART they assume that all training examples can be held in memory, and are limited in the examples they can consider. Other trees such as SPRINT and SLIQ have attempted to resolve this issue by using a window to scan large datasets sequentially, but this is very slow when building complex trees, and still requires all data to reside in disk-space. They propose the Very Fast Decision Tree (VFDT), which requires that data items are only read once, in a small and constant time. They utilise Hoeffding Bounds [18] to guarantee that the statistically best attribute to divide a node on can very commonly be found using a small number of examples of instances as could be with infinite. They propose the Hoeffding tree, which regularly checks whether splitting a node on the best attribute will lead to an improvement in a selected measure (information gain or Gini) beyond a set threshold. This allows it to continue to adapt, even with infinite data, without sacrificing significant quality nor having burdensome memory requirements. Parameters need to be set by the user: the minimum number of examples before reviewing whether to split a node; the threshold to allow a split; and whether the algorithm can rescan prior examples. Even if these parameters are well selected by the user, there is a danger that they will lead to poor performance in environments featuring significant concept drift.

Recognising the issues posed by concept drift, Hulten et al. [19] proposed the Concept-Adapting Very Fast Decision Tree (CVFDT), an adaptation on the VFDT to give it the adaptability to handle changing data. Through adopting a sliding window approach, the CVFDT increments counts of recently seen data while reducing counts of older data. This should have no effect when there is no concept drift, but where there is drift, prior splits will no longer pass the Ho-

effding tests that deemed them worthwhile. At this point, an alternate subtree is grown by the algorithm from that node. If it accurately classifies new data better than the existing subtree, the existing subtree is replaced. Overall, this requires additional memory to keep alternate subtrees in memory, and summary statistics of split quality at each node. However, these memory requirements are still constant with the data items received. CVFDTs cannot utilise old sub-trees which are discarded, which could provide potential performance improvements where concepts drifts can revert back to prior probability distributions.

Decision trees have trouble accurately classifying rare classes. Consider a binary class problem: if a very large majority belongs to one class, often a classifier achieves best performance by classifying all instances as that class. Single classifier often needs to combine in a ensemble to achieve good performance for imbalanced data set, however recent researches has made it possible for some classifiers perform as good as ensemble. Lyon et al [20] propose an improvement on the VFDT by introducing a new criterion for splitting: Hellinger distance. This measure seeks the attributes which are most disjoint in the data, and seeks to create tree splits based on these features. Instead of prioritising information gain, this tree prioritises class discrimination. This will often lead to a smaller portion correctly classified, but should amplify the number of the rare class successfully classified. Calculating Hellinger distance can be costly, as it scales to the number of classes and needs to discretize continuous classes into bins, leading to further multiplicative scaling. Lyon et al propose the HD-VFDT which uses the splitting criterion above, and also the Gaussian Hellinger Very Fast Decision Tree, which assumes normality of the distribution to simplify calculation of the Hellinger Distance. However, instances in a data stream are not independent, as time affects their distribution, and so this could lead to flawed results in some circumstances. Experimental analysis of this approach showed statistically significant improvements using these methods over simple Hoeffding Trees where the minority class makes up 1% or less of the total data. They do not evaluate the approach thoroughly with regard to time nor memory use. Specifically, it would be interesting to measure the difference in accuracy, speed and time required between the two variants of the Hellinger tree proposed.

Liechtenwalter and Chawla [21] have found potential in using Hellinger trees in environments with concept drift. By weighting the Hellinger distance with Information Gain to measure distance between datasets, a learner can decide whether model learnt on previous data is valid on current data. Pazzolo et al [22] showed, using the HDDT proposed by Cieslak and Chawla [23] and C4.5 tree, that trees with this weighting generally outperformed trees without the weighting in class-imbalanced environments with drift. These papers do not address the multi-class problem, which would be a natural extension of this research.

5 Ensemble learners

6 Evaluating approaches

Both UCI datasets and synthetic datasets are often chosen to evaluate classifiers' performance. In order to form a skewed distribution problem and simulate data stream, a minority class and a majority class have to be chosen from a dataset to form the skewed distribution between two classes, the data then will need to be partitioned into some chunks with skewed distribution.

Efficiency and accuracy are two major factors for evaluating learning algorithms' performance on data stream. In terms of efficiency, it often refers to time overhead, sampling techniques perform not as efficient as other techniques. In terms of accuracy, there are two measures can be used: (1) probability estimation accuracy (2) classification accuracy. In [24], Gao suggested that using Mean Squared Error to measure the quality of probability estimation, for rare class, low Mean Squared Error is desired. Common measurement for classification accuracy is classification error rate, but this measurement is undesirable for imbalanced data streams because the rare minority class doesn't have an significant impact on classification error rate. Instead, three other measurement are typically used: Precision, Recall and False Alarm Rate. Gao suggested ROC curve can show the trade-off between Precision and False Alarm Rate. A good ROC has big Area Under ROC Curve (AUC), and the closer to the left-top corner the better. Another similar method is recall-precision plot. Gao showed that by employing Sampling and Ensemble techniques (SE[25]), it has significantly improved both probability accuracy (MSE) and ROC measurement. In [26], Gama discussed several other performance evaluation metrics can be used: Sensitivity and Specificity, Kappa-statistic[26]. Kappa-statistics is useful for imbalanced dataset. All these performance metrics should be taken look at when considering the basic reference point or baseline.

In [26], Gama discussed except above evaluation metrics for learning algorithm, the change detection's accuracy can be evaluated for those who equipped explicit drift detection technique. (1) Probability of true change detection. (2) Probability of false alarms. (3) Delay of detection.

In [26], Gama discussed that traditional cross-validation method is not applicable to data stream because it will not keep the temporal nature. He suggested two procedures instead: (1) Holdout. This is wildly used and most useful method for validation, but it is not always possible, because of the temporal nature of data stream. (2) Interleaved Test-Then-Train or Prequential. This method makes full use of every instance, also has a smooth accuracy plot. (3) Controlled Permutations. This method is useful for sudden drift data stream by randomization.

7 Future Work

more that two class (multi-class) problem.

8 Conclusion

References

[1] D. Cieslak, T. Hoens, N. Chawla, and W. Kegelmeyer, "Hellinger distance decision trees are robust and skew-insensitive," *Data Mining and Knowledge Discovery*, vol. 24, no. 1, pp. 136–158, 2012. [Online]. Available: http://dx.doi.org/10.1007/s10618-011-0222-1

[2] J. Gama, R. Sebastio, and P. Rodrigues, "On evaluating stream learning algorithms," *Machine Learning*, vol. 90, no. 3, pp. 317–346, 2013. [Online]. Available: http://dx.doi.org/10.1007/s10994-012-5320-9

[3] A. Bifet and R. Gavald, *Learning from Time-Changing Data with Adaptive Windowing*, ch. 42, pp. 443–448. [Online]. Available: http://epubs.siam.org/doi/abs/10.1137/1.9781611972771.42

[4] R. Sebastio, J. Gama, P. Rodrigues, and J. Bernardes, "Monitoring incremental histogram distribution for change detection in data streams," in *Knowledge Discovery from Sensor Data*, ser. Lecture Notes in Computer Science, M. Gaber, R. Vatsavai, O. Omitaomu, J. Gama, N. Chawla, and A. Ganguly, Eds. Springer Berlin Heidelberg, 2010, vol. 5840, pp. 25–42. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-12519-5_2

[5] E. Page, "Continuous inspection schemes," *Biometrika*, pp. 100–115, 1954.

[6] G. Widmer and M. Kubat, "Learning in the presence of concept drift and hidden contexts," *Machine learning*, vol. 23, no. 1, pp. 69–101, 1996.

[7] P. Vorburger and A. Bernstein, "Entropy-based concept shift detection," in *Data Mining, 2006. ICDM'06. Sixth International Conference on*. IEEE, 2006, pp. 1113–1118.

[8] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *arXiv preprint arXiv:1106.1813*, 2011.

[9] S. Chen and H. He, "Sera: selectively recursive approach towards nonstationary imbalanced stream data mining," in *Neural Networks, 2009. IJCNN 2009. International Joint Conference on*. IEEE, 2009, pp. 522–529.

[10] S. Chen, H. He, K. Li, and S. Desai, "Musera: multiple selectively recursive approach towards imbalanced stream data mining," in *Neural Networks (IJCNN), The 2010 International Joint Conference on*. IEEE, 2010, pp. 1–8.

[11] S. Chen and H. He, "Towards incremental learning of nonstationary imbalanced data stream: a multiple selectively recursive approach," *Evolving Systems*, vol. 2, no. 1, pp. 35–50, 2011.

[12] G. E. Batista, R. C. Prati, and M. C. Monard, "A study of the behavior of several methods for balancing machine learning training data," *ACM Sigkdd Explorations Newsletter*, vol. 6, no. 1, pp. 20–29, 2004.

[13] ——, "A study of the behavior of several methods for balancing machine learning training data," *ACM Sigkdd Explorations Newsletter*, vol. 6, no. 1, pp. 20–29, 2004.

[14] A. Godase and V. Attar, "Classifier ensemble for imbalanced data stream classification," in *Proceedings of the CUBE International Information Technology Conference*. ACM, 2012, pp. 284–289.

[15] T. Hoens, R. Polikar, and N. Chawla, "Learning from streaming data with concept drift and imbalance: an overview," *Progress in Artificial Intelligence*, vol. 1, no. 1, pp. 89–101, 2012. [Online]. Available: http://dx.doi.org/10.1007/s13748-011-0008-0

[16] S. L. Salzberg, "Book review: C4.5: Programs for machine learning by J. Ross Quinlan. Morgan Kaufmann Publishers, Inc., 1993," *Mach. Learn.*, vol. 16, no. 3, pp. 235–240, Sep. 1994. [Online]. Available: http://dx.doi.org/10.1023/A:1022645310020

[17] P. Domingos and G. Hulten, "Mining high-speed data streams," in *Proceedings of the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '00. New York, NY, USA: ACM, 2000, pp. 71–80. [Online]. Available: http://doi.acm.org/10.1145/347090.347107

[18] W. Hoeffding, "Probability inequalities for sums of bounded random variables," *Journal of the American Statistical Association*, vol. 58, no. 301, pp. pp. 13–30, 1963. [Online]. Available: http://www.jstor.org/stable/2282952

[19] G. Hulten, L. Spencer, and P. Domingos, "Mining time-changing data streams," in *Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ser. KDD '01. New York, NY, USA: ACM, 2001, pp. 97–106. [Online]. Available: http://doi.acm.org/10.1145/502512.502529

[20] R. Lyon, J. Brooke, J. Knowles, and B. Stappers, "Hellinger distance trees for imbalanced streams," *arXiv preprint arXiv:1405.2278*, 2014.

[21] R. N. Lichtenwalter and N. V. Chawla, "Adaptive methods for classification in arbitrarily imbalanced and drifting data streams," in *New Frontiers in Applied Data Mining*. Springer, 2010, pp. 53–75.

[22] A. Dal Pozzolo, R. Johnson, O. Caelen, S. Waterschoot, N. V. Chawla, and G. Bontempi, "Using HDDT to avoid instances propagation in unbalanced and evolving data streams," 2014, in 2014 IEEE World Congress on Computational Intelligence.

[23] D. Cieslak and N. Chawla, "Learning decision trees for unbalanced data," in *Machine Learning and Knowledge Discovery in Databases*, ser. Lecture

Notes in Computer Science, W. Daelemans, B. Goethals, and K. Morik, Eds. Springer Berlin Heidelberg, 2008, vol. 5211, pp. 241–256. [Online]. Available: http://dx.doi.org/10.1007/978-3-540-87479-9_34

[24] J. Gao, W. Fan, J. Han, and S. Y. Philip, "A general framework for mining concept-drifting data streams with skewed distributions." in *SDM*. SIAM, 2007, pp. 3–14.

[25] ——, "A general framework for mining concept-drifting data streams with skewed distributions." in *SDM*. SIAM, 2007, pp. 3–14.

[26] J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia, "A survey on concept drift adaptation," *ACM Computing Surveys (CSUR)*, vol. 46, no. 4, p. 44, 2014.