A Self-adaptive 30-day Diabetic Readmission Prediction Model based on Incremental Learning

Peng Zhao
Informatics Institute
University of Missouri
Columbia, MO, 65211, United States
peng.zhao@mail.missouri.edu

Illhoi Yoo*
Department of Health Management and Informatics, School of Medicine
Informatics Institute
University of Missouri
Columbia, MO, 65211, United States
yooil@health.missouri.edu

***Abstract*—Hospital readmissions within 30 days after discharge are costly and it has been a prior for researchers to identify patients at risk of early readmission. Most of the reported hospital readmission prediction models have been built with historical data and thus can outdate over time. In this work, a self-adaptive 30-day diabetic hospital readmission prediction model has been developed. A diabetic inpatient encounter data stream was used to train the self-adaptive models based on incremental learning algorithms. The result indicated that the model can automatically adapt to the newly arrived data. The best model achieved an average AUC of 0.655 ± 0.078, which is consistent with static models built with the same dataset.**

Keywords—hospital readmission; incremental learning; self-adaptive; electronic health record; prediction model; data stream

I. Introduction

A hospital readmission means a patient is hospitalized again after being discharged from a hospital within a short time window (normally 30 days). It can reduce patients' quality of life and significantly contribute to the medical cost. In the United States, it has been estimated that unplanned readmissions account for $17.4 billion in Medicare expenditures annually [1]. Since 2012, readmission rates for selected conditions have become a health care quality measure and hospitals with high readmission rates will face financial penalties from the Hospital Readmissions Reduction Program (HRRP) of the Affordable Care Act [2]. Timely and accurate knowledge of readmission risk can help in decision making to adjust interventions to reduce potentially avoidable readmissions.

Predicting hospital readmissions has been a popular research topic and many hospital readmission prediction models have been reported in the literature [3], [4]. However, most of the reported models are static and cannot update themselves once new data become available. It has been argued that training with historical data is one of the reasons why most readmission prediction models fail in practice [5]. Models built with historical data are prone to getting outdated because the distribution of the patient population can evolve over time. To keep up-to-date, the static readmission models need to be re-trained periodically. It can be expensive to constantly discard existing models and build new models from scratch, especially in real-time applications. More importantly, it is hard to determine and discard outdated knowledge manually. Therefore, it is meaningful to build a hospital readmission prediction model that can automatically update itself when new data become available.

The majority of the current machine learning applications are batch learning, which uses all the instances in the dataset at the same time to train models without considering their temporal order [6]. In reality, most data arrive over time in a streaming fashion. A data stream is a continuous sequence of instances with temporal order. As the volume of health care data keeps growing, it is getting challenging to mine potentially unbounded data with bounded computing resources under the classical batch learning setting. For the readmission model to achieve the self-adaptiveness, the underlying machine learning algorithm should be incremental, which constantly extracts knowledge from the data stream and does not assume the availability of the training dataset prior to modeling. The previously learned pattern of the data will be retained and updated on-the-fly when new data become available in the data stream [6].

There are three main categories of incremental learning algorithms, including incremental decision tree, incremental Bayesian, and incremental support vector machine [7]. They are either naturally incremental or modifications of non-incremental algorithms. Unlike the time series analysis, which tries to infer the temporal pattern of the data, incremental learning aims to learn from the data stream by processing the instances in temporal order without random access.

In this work, we have developed a self-adaptive diabetic 30-day readmission prediction model, which can update itself after a new instance of encounter becomes available in the diabetic inpatient encounter data stream. We chose to target 30-day diabetic readmissions because the 30-day readmission rate of diabetic patients has been higher than the overall 30-day readmission rate of all patients (14.4-22.7% vs. 8.5-13.5%) [8]–[14] but has drawn relatively less attention to the research community compared to HRRP applicable conditions, such as acute myocardial infarction, heart failure, pneumonia, etc [15]. To the best of our knowledge, this is the first self-adaptive 30-day hospital readmission predictive model based on incremental learning. The remaining part of this work is organized as follows: part II details the data source, preprocessing, algorithms, evluation metic, and tools; part III shows the results and discussions; part IV summarizes the work and provides future plans.

II. Methods

A. Data Source and Preprocessing

A de-identified inpatient encounter-level dataset of diabetic patients from 130 US hospitals during 1999-2008 has been collected from a study [16]. There are 101,766 inpatient encounters in total with 11.2% being readmitted within 30 days

978-1-5090-3050-7/17/$31.00 ©2017 IEEE

after discharge and 50 attributes. Instances with missing values were removed and the classes were balanced by under-sampling encounters not readmitted within 30 days. By measuring information gain [17], 22 attributes were selected, including A1C test result, admission source, admission type, age, change of medications, diabetes medications, diagnoses 1-3, discharge disposition, glucose serum test result, insulin, medical specialty, metformin, number of diagnoses, number of emergency visits, number of inpatient visits, number of lab procedures, number of medications, number of outpatient visits, number of procedures, time in hospital.

B. Algorithms

Naïve Bayes, Hoeffding Tree, and Hoeffding Adaptive Tree algorithms were selected to build the candidate self-adaptive models because they are incremental and can work with both numeric and categorical variables. In addition, they can be used to build static models too so that the global performances of the self-adaptive models and static models can be directly compared.

1) Naïve Bayes

Naïve Bayes is a probabilistic classification algorithm based on Bayes' theorem [18] by assuming the independence of predictors [19]. It calculates the posterior probability of being one class given the predictors. The Naïve Bayes algorithm is naturally incremental and can keep the model up-to-date by constantly updating the prior probability according to the incoming training instances.

2) Hoeffding Tree (VFDT)

The Hoeffding Tree (Very Fast Decision Tree, VFDT) is an incremental decision tree algorithm, which uses the Hoeffding bound [20] to quantify the number of samples required to choose a split attribute and can produce the output asymptotically nearly identical to that of batch-based decision tree algorithms [21]. The Hoeffding bound guarantees with probability $1 - \delta$ that for n independent observations of a random variable r with range R, the true mean of r is at least $\bar{r} - \epsilon$, with $\bar{r}$ being the mean of the observations and $\epsilon = \sqrt{R^2 ln(1/\delta)/2n}$ [21]. When selecting a split attribute X_i, the information gain $G(X_i)$ is measured on attributes with $G(X_a)$ being the highest information gain and $G(X_b)$ being the second highest information gain. If the difference $\Delta G = G(X_a) - G(X_b) > \epsilon$, the X_a attribute will be selected as the split attribute [21].

3) Hoeffding Adaptive Tree

The Hoeffding Adaptive Tree algorithm [22] has been built based on the Hoeffding Tree algorithm. The Hoeffding Tree algorithm assumes that the distribution of the data stream remains unchanged. However, in reality the distribution generating the data stream may change over time (concept drift) [23] and this will make the Hoeffding Tree algorithm not applicable. The Hoeffding Adaptive Tree algorithm overcomes this limitation by monitoring the performance of all tree branches and replace them with more accurate new branches if their accuracy decreases [22].

C. Evaluation Metric

The model's performance metric used in this work is the prequential area under the receiver operating characteristic curve (AUC) [24], which only evaluates the classifications of the most recent instances within a sliding window. Prequential means when a new instance is available in the data stream, it is first used to test the model and then given back to the algorithm for additional training. The receiver operating characteristic curve is a graphical representation of a binary classifier's performance as the discrimination threshold is varied [25]. The AUC measures the model's ability of discrimination and can be interpreted as the probability that the model will rank a randomly selected positive sample higher than a randomly selected negative sample [26]. The prequential AUC is insensitive to class imbalance issue and can work with data streams with concept drift. Compared to the global AUC commonly used in evaluating machine learning models, the prequential AUC can measure the incremental model's ability to adapt to the new data.

D. Tools

Pandas [27] and scikit-learn [28] packages in Python were used to preprocess the data. Weka [29] was used for attribute selection, data format conversion, and development of static models. MOA [30] was used to generate the data stream from the static data and build the incremental models.

III. Results and Discussions

The data stream was directly generated from the static dataset. The cumulative class ratio versus the number of instances in the data stream is shown in Fig. 1. The cumulative class ratio is defined as the ratio of the total number of encounters not readmitted within 30 days to the total number of encounters readmitted within 30 days. In the ideal situation that

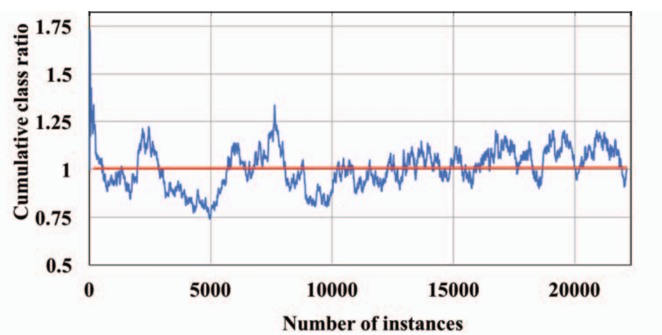

Fig. 1. The cumulative class ratio versus the number of instances in the data stream.

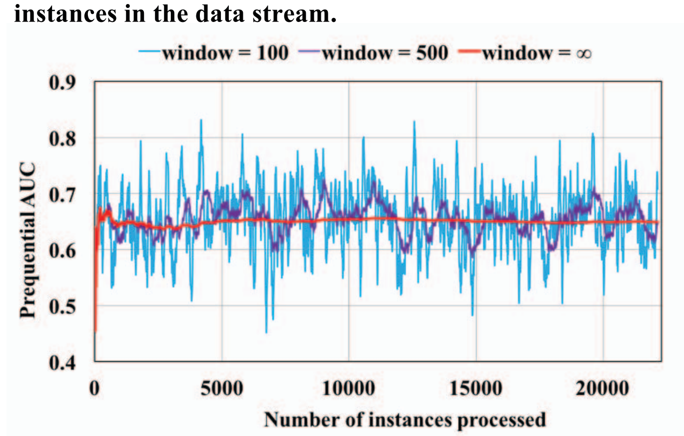

Fig. 2. Effects of the sliding-window size on the prequential AUC curves of the Naïve Bayes model.

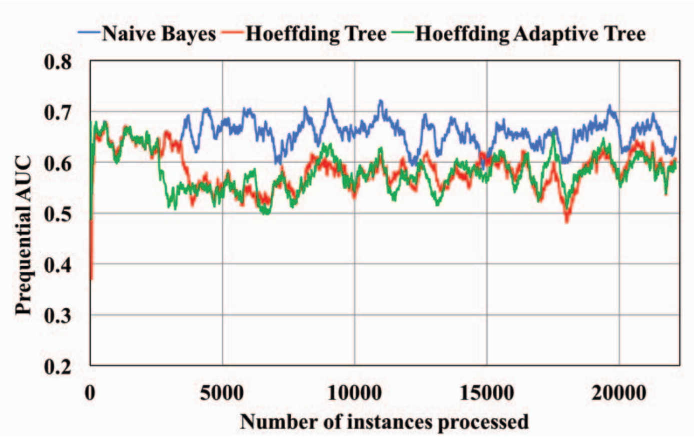

Fig. 3. The prequential AUC curves of the three models. The sliding window size is 500.

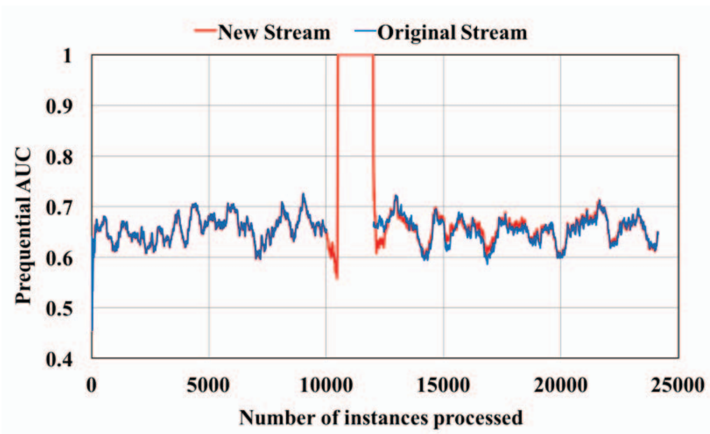

Fig. 4. The prequential AUC curves of the Naïve Bayes models for the new modified data stream and the original data stream. The sliding window size is 500.

the two classes keep arriving alternatively, the ratio should be one (the red line). Obvious fluctuations can be observed in the first 10,000 instances of the data stream (the blue curve). The class labels tend to be more balanced after 10,000 instances. This indicates the class's distribution varies over the whole stream.

Fig. 2 shows the effects of the sliding-window size on the prequential AUC of the Naïve Bayes model. It can be seen that if the window size goes to infinity, the prequential AUC will be indifferent from the global AUC and the model's adaption to the local trend of the data stream is not observable. When the window size is small (~ 100), the prequential AUC curve will become too sensitive to local variations. The size of 500 is appropriate for observing the model's self-adaptive ability.

Three incremental learning algorithms Naïve Bayes, Hoeffding Tree, and Hoeffding Adaptive Tree were used to build candidate models. Fig. 3 shows the prequential AUC curves of the three candidate models. As the evaluation window (size = 500) moves along the data stream, the classification results within the window will be used to calculate the prequential AUC and build the curves. The curves fluctuate because the classification results in each snapshot of the window can be heterogeneous.

The Naïve Bayes model's curve fluctuates between 0.6 to 0.7. The curves of Hoeffding Tree and the Hoeffding Adaptive Tree models have high degree of agreement and both decrease to about 0.55 at around 3,000 instances in the data stream. One possible reason is that the class label's distribution undergoes a sudden flip in this region as shown in Fig. 1. The incremental decision trees require a certain number of samples to determine a new splitting attribute to update the trees and thus it takes longer time to adapt to the new trend of data. As a result, the prequential AUC of the Hoeffding Tree and the Hoeffding Adaptive Tree models is lower than the Naïve Bayes models. The Naïve Bayes model was selected as the final model.

An experiment was conducted to further explore the Naïve Bayes model's self-adaptive ability. The first 2,000 of the randomly deleted encounters not readmitted within 30 days in the data balancing step were inserted back into the data stream after the $10{,}000^{th}$ instance. These 2,000 instances had never been seen by the model and caused a sudden change in the class label distribution in the middle of the data stream. The prequential AUC curve of the original data stream was shifted by 2,000 in the horizontal axis after the $10{,}000^{th}$ instance to align with the modified data stream's prequential AUC curve.

Fig. 4 shows the result. The reason for the spike in the middle is that the evaluation sliding window's size (500) is smaller than the number of the inserted "otherwise" instances (2,000). As the window moves along the data stream, it will enter a region where all the instances are of the same class ("not readmitted within 30 days"). In this case, the prequential AUC will be equal to one. It can be seen that after the window has moved out the single class region, the prequential AUC becomes less than 1 and the two curves have very high degree of agreement. The minor disagreements between 12,000 and 19,000 indicate the model retains the memory of the inserted 2,000 instances in the modified data stream. After 19,000, the model gradually adapts to the new trend and forgets the outdated knowledge of the inserted 2,000 instances.

To compare the global performance of the incremental model with the static model, a static Naïve Bayes model was built using the same dataset with 10-fold cross-validation. Another study has developed a static Naïve Bayes model with the same original dataset (training set : test set = 4:1) [31]. The AUC of the incremental model and the two static models is summarized in Table I. The average prequential AUC of the incremental Naïve Bayes model is consistent with the AUC of the static Naïve Bayes models developed in this study and study [31]. This indicates the self-adaptive incremental Naïve Bayes model has the global discriminative performance nearly identical to that of the static Naïve Bayes models. At the same level of predictive performance, the self-adaptive model is more economical because it can update itself and thus requires less interventions (e.g., re-training) from human and save computing resources. It also provides another dimension to better understand the change of the data over time.

TABLE I. THE AUC OF THE INCREMENTAL AND STATIC MODELS

Models	AUC
Naïve Bayes (incremental) (this study)	0.655 ± 0.078 (average)
Naïve Bayes (static) (this study)	0.660
Naïve Bayes (static) (study [31])	0.657

A limitation of this study is that the incremental learning algorithms used to build the model are not good at handling highly skewed data. To reduce the potential bias of the model, the majority class in the original dataset was under-sampled to make the cumulative class ratio of the data stream between 0.75 to 1.25 as shown in Fig. 1. In practice, however, there is no guarantee that the data will arrive in a balanced manner. Medical data are usually imbalanced and the minorities, such as readmissions are often of interest. For this self-adaptive model to be applied in a clinical setting, there has to be a mechanism to keep monitoring the balance of the class and remove excess instances of the majority class in the data stream when necessary. Another limitation is this study assumes the feature space remains unchanged in the data stream. However, in reality, it may change over time, too. For example, the transition from ICD-9 to ICD-10 in the United States can outdate prediction models relying on the old coding system. To handle feature space change, a feasible way is to chop the data stream into blocks with each block having the same feature space and build predictive models for each block of data stream. The separate models will be eventually combined with ensemble learning methods.

IV. Conclusion

In this work, we have developed a self-adaptive 30-day diabetic hospital readmission prediction model based on incremental learning. Compared to the static models that can get outdated over time, the presented incremental model can constantly update itself when new data becomes available. The model has been proven to be able to retain the memory of previous data for a period of time and gradually forget the outdated knowledge and adapt to the latest trend. The global performance of the incremental model is nearly identical to that of the static models built with the same dataset. The model has a moderate discriminative performance (average AUC = 0.655 ± 0.078) and is not good for highly imbalanced data streams. Therefore, we purpose the future work to be improving the algorithms' ability to handle imbalanced data streams.

References

[1] S. F. Jencks, M. V. Williams, and E. A. Coleman, "Rehospitalizations among Patients in the Medicare Fee-for-Service Program," *New England Journal of Medicine*, vol. 360, no. 14, pp. 1418–1428, 2009.

[2] "Readmissions Reduction Program (HRRP)." [Online]. Available: https://www.cms.gov/medicare/medicare-fee-for-service-payment/acuteinpatientpps/readmissions-reduction-program.html.

[3] D. Kansagara *et al.*, "Risk prediction models for hospital readmission: a systematic review.," *JAMA*, vol. 306, no. 15, pp. 1688–98, Oct. 2011.

[4] H. Zhou, P. R. Della, P. Roberts, L. Goh, and S. S. Dhaliwal, "Utility of models to predict 28-day or 30-day unplanned hospital readmissions: an updated systematic review," *BMJ Open*, vol. 6, no. 6, p. e011060, Jun. 2016.

[5] M. W. Tanzer and E. M. Heil, "Why majority of readmission risk assessment tools fail in practice," *Proceedings - 2013 IEEE International Conference on Healthcare Informatics, ICHI 2013*, pp. 567–569, 2013.

[6] A. Gepperth and B. Hammer, "Incremental learning algorithms and applications," no. Exc 277, pp. 27–29, 2016.

[7] W. Zang, P. Zhang, C. Zhou, and L. Guo, "Comparative study between incremental and ensemble learning on data streams: Case study," *Journal Of Big Data*, vol. 1, no. 1, pp. 1–16, 2014.

[8] "Economic Costs of Diabetes in the U.S. in 2012," *Diabetes Care*, vol. 36, no. 4, pp. 1033–1046, Apr. 2013.

[9] "Hospital Readmissions in Pennsylvania 2010," 2010. [Online]. Available: http://www.phc4.org/reports/readmissions/10/.

[10] J. M. Robbins and D. A. Webb, "Diagnosing diabetes and preventing rehospitalizations: the urban diabetes study.," *Medical care*, vol. 44, no. 3, pp. 292–6, Mar. 2006.

[11] R. E. Burke and E. A. Coleman, "Interventions to decrease hospital readmissions: keys for cost-effectiveness.," *JAMA internal medicine*, vol. 173, no. 8, pp. 695–8, Apr. 2013.

[12] H. J. Jiang, D. Stryer, B. Friedman, and R. Andrews, "Multiple hospitalizations for patients with diabetes.," *Diabetes care*, vol. 26, no. 5, pp. 1421–6, May 2003.

[13] H. Kim, J. S. Ross, G. D. Melkus, Z. Zhao, and K. Boockvar, "Scheduled and unscheduled hospital readmissions among patients with diabetes.," *The American journal of managed care*, vol. 16, no. 10, pp. 760–7, Oct. 2010.

[14] B. Friedman, H. J. Jiang, and A. Elixhauser, "Costly Hospital Readmissions and Complex Chronic Illness," *Inquiry*, vol. 45, no. 4, pp. 408–421, Dec. 2008.

[15] D. J. Rubin, "Hospital Readmission of Patients with Diabetes," *Current Diabetes Reports*, vol. 15, no. 4, 2015.

[16] B. Strack *et al.*, "Impact of HbA1c Measurement on Hospital Readmission Rates: Analysis of 70,000 Clinical Database Patient Records," *BioMed Research International*, vol. 2014, pp. 1–11, 2014.

[17] S. Kullback and R. A. Leibler, "On Information and Sufficiency," *The Annals of Mathematical Statistics*, vol. 22, no. 1, pp. 79–86, Mar. 1951.

[18] T. Bayes and R. Price, "An Essay towards Solving a Problem in the Doctrine of Chances. By the Late Rev. Mr. Bayes, F. R. S. Communicated by Mr. Price, in a Letter to John Canton, A. M. F. R. S.," *Philosophical Transactions of the Royal Society of London*, vol. 53, pp. 370–418, Jan. 1763.

[19] P. Domingos and M. Pazzani, "On the Optimality of the Simple Bayesian Classifier under Zero-One Loss," *Machine Learning*, vol. 29, pp. 103–130, 1997.

[20] W. Hoeffding, "Probability Inequalities for Sums of Bounded Random Variables," *Journal of the American Statistical Association*, vol. 58, no. 301, p. 13, Mar. 1963.

[21] 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 - KDD '01*, 2001, pp. 97–106.

[22] A. Bifet and R. Gavaldà, "Adaptive Learning from Evolving Data Streams," in *IDA 2009: Advances in Intelligent Data Analysis VIII*, 2009, pp. 249–260.

[23] A. Tsymbal, "The problem of concept drift: definitions and related work," *Computer Science Department, Trinity College Dublin*, vol. 4, no. C, pp. 2004–15, 2004.

[24] D. Brzezinski and J. Stefanowski, "Prequential AUC for Classifier Evaluation and Drift Detection in Evolving Data Streams," in *Proceedings of the 3rd International Workshop on New Frontiers in Mining Complex Patterns, Nancy, France*, 2015, pp. 87–101.

[25] D. M. Green and J. A. Swets, *Signal detection theory and psychophysics*. New York, New York, USA: John Wiley & Sons, Inc, 1966.

[26] J. A. Hanley and B. J. McNeil, "A method of comparing the areas under receiver operating characteristic curves derived from the same cases.," *Radiology*, vol. 148, no. 3, pp. 839–843, Sep. 1983.

[27] W. Mckinney, *pandas: a Foundational Python Library for DataAnalysis and Statistics*. San Francisco: O'Reilly Media, Inc, 2012.

[28] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, and B. Thirion, "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[29] E. Frank, M. A. Hall, and I. H. Witten, *The WEKA Workbench. Online Appendix for "Data Mining: Practical Machine Learning Tools and Techniques,"* 4th ed. Morgan Kaufmann, 2016.

[30] A. Bifet, G. Holmes, R. Kirkby, and B. Pfahringer, "MOA: Massive Online Analysis," *Journal of Machine Learning Research*, vol. 11, pp. 1601–1604, 2010.

[31] T. M. Ahmed, "Using Data Mining To Develop Model for Classifying Diabetic Patient Control Level Based on Historical Medical Records," *Journal of Theoretical and Applied Inforamtion Technology*, vol. 87, no. 2, pp. 316–323, 2016.