Handling Local Concept Drift with Dynamic Integration of Classifiers: Domain of Antibiotic Resistance in Nosocomial Infections

Alexey Tsymbal
Dept. of Computer Science, Trinity College Dublin, Ireland
tsymbalo@cs.tcd.ie

Mykola Pechenizkiy
Dept. of Math. IT, Univ. of Jyväskylä, Finland
mpechen@cs.jyu.fi

Pádraig Cunningham
Dept. of Computer Science, Trinity College Dublin, Ireland
cnnnghmp@cs.tcd.ie

Seppo Puuronen
Dept. of CS&ISs, Univ. of Jyväskylä, Finland
sepi@cs.jyu.fi

Abstract

In the real world concepts and data distributions are often not stable but change with time. This problem, known as concept drift, complicates the task of learning a model from data and requires special approaches, different from commonly used techniques, which treat arriving instances as equally important contributors to the target concept. Among the most popular and effective approaches to handle concept drift is ensemble learning, where a set of models built over different time periods is maintained and the best model is selected or the predictions of models are combined. In this paper we consider the use of an ensemble integration technique that helps to better handle concept drift at the instance level. Our experiments with real-world antibiotic resistance data demonstrate that dynamic integration of classifiers built over small time intervals can be more effective than globally weighted voting which is currently the most commonly used integration approach for handling concept drift with ensembles.

1. Introduction

The problem of concept drift is of increasing importance to machine learning and data mining as more and more data is organized in the form of data streams rather than static databases, and it is rather unusual that concepts and data distributions stay stable over a long period of time [13, 17]. There are many known cases of concept drift in biomedical data, and among the most important examples in the biomedical context is antibiotic resistance, where pathogen sensitivity may change over time as new pathogen strains develop resistance to antibiotics which were previously effective [5, 14].

Ensemble learning is among the most popular and effective approaches to handle concept drift, in which a set of concept descriptions built over different time intervals is maintained, predictions of which are combined using a form of voting, or the most relevant description is selected [8, 10, 11, 16]. However, there is a problem with current ensemble approaches in that they are not able to deal with local concept drift, which is a common case with real-world data.

The most common integration approach with ensembles for handling concept drift is weighted voting, where each base classifier receives a weight proportional to its relevance to the current concept [8, 10, 11, 16]. With weighted voting, lower weights can be assigned to predictions from base classifiers simply because their global accuracy on the current block of data falls, even if they are still good experts in the stable regions of the data.

In this paper, we suggest a solution to this problem. To improve the treatment of local concept drifts, dynamic integration of classifiers can be used, which integrates base classifiers at an instance level. In dynamic integration, each base classifier receives a weight proportional to its local accuracy in the neighbourhood of the current test instance, instead of the global classification accuracy.

We apply dynamic integration of classifiers to ensembles of classifiers built in the domain of antibiotic resistance in nosocomial infections in order to better handle concept drift. Antibiotic resistance is an especially difficult problem for nosocomial infections in hospitals because pathogens attack critically ill patients who are more vulnerable to infections than the general population and therefore require more antibiotics.

This paper is organized as follows: in Section 2 we consider the general problem of concept drift, and in Section 3 we briefly review approaches to ensemble

Proceedings of the 19th IEEE Symposium on Computer-Based Medical Systems (CBMS'06)
0-7695-2517-1/06 $20.00 © 2006 IEEE

integration with a focus on dynamic integration. In Section 4 we present the application domain of antibiotic resistance in nosocomial infections. In Section 5 we consider the basic characteristics of our real-world dataset used for analysis; in Section 6 we present the results of our experiments on the use of different ensemble integration techniques, and in Section 7 we conclude with a brief summary and further research directions.

2. The problem of concept drift

A difficult problem with learning in many real-world domains is that the concept of interest may depend on some hidden context, not given explicitly in the form of predictive features. Changes in the hidden context over time can induce more or less radical changes in the target concept, which is generally known as *concept drift* [17]. An effective learner should be able to track such changes and to quickly adapt to them.

Changes in hidden context may not only result in a change of the target concept, but may also cause a change of the underlying data distribution. Even if the target concept remains the same, and it is only the data distribution that changes, this may often lead to the necessity of revising the current model, as the model's error may no longer be acceptable with the new data distribution. The necessity in the change of current model due to the change of data distribution is called *virtual concept drift* [13]. Virtual concept drift and real concept drift often occur together. From the practical point of view it is not important, what kind of concept drift occurs, real or virtual, or both. In all cases the current model needs to be changed.

Three basic approaches to handling concept drift can be distinguished in the literature [13]: (1) instance selection; (2) instance weighting; and (3) ensemble learning [13]. In instance selection, the goal is to select instances relevant to the current concept. The most common concept drift handling technique is based on instance selection and consists in generalizing from a window that moves over recently arrived instances and uses the learnt concepts for prediction only in the immediate future [17]. Many case-base editing strategies in case-based reasoning that delete noisy, irrelevant and redundant cases are also a form of instance selection [3].

Instance weighting uses the ability of some learning algorithms such as Support Vector Machines (SVMs) to process weighted instances [7]. Instances can be weighted according to their "age", and their competence with regard to the current concept. Klinkenberg [7] demonstrates in his experiments that instance weighting techniques handle concept drift worse than analogous instance selection techniques, which is probably due to overfitting the data.

Ensemble learning maintains a set of concept descriptions, predictions of which are combined using a form of voting, or the most relevant description is selected. Street and Kim [11] and Wang et al. [16] suggest that simply dividing the data into sequential blocks of fixed size and building an ensemble on them may be effective for handling concept drift. Stanley [10] and Kolter and Maloof [8] build ensembles of incremental learners in an online setting, starting to learn new base classifiers after fixed intervals while continuing to update the existing ones. All incremental ensemble approaches use some criteria to dynamically create, delete, or reactivate new ensemble members, which are normally based on the base models' consistency with the current data.

Often changes in the concept or data distribution occur in some regions of instance space only, and besides, the type and severity of changes may depend on the location in the instance space. We call this phenomenon *local concept drift*. It may be demonstrated that the most popular approach to handling concept drift, windowing, will suffer from the presence of local concept drifts. Commonly used ensemble learning approaches are not able to deal with local concept drift neither.

3. Ensemble learning and dynamic integration of classifiers

Brodley and Lane [2] have shown that simply increasing diversity of an ensemble is not enough to ensure increased prediction accuracy. If an integration method does not utilize diversity, then no benefit arises from the integration. The challenging problem of integration is to decide which of the classifiers to select or how to combine the results produced by the base classifiers. A number of selection and combination approaches have been proposed.

One of the most popular and simplest techniques used to combine the results of base classifiers, is simple voting (also called majority voting) [1]. In voting, the output of each base classifier is considered as a vote for that particular class value. The class value that receives the biggest number of votes is selected as the final classification. Weighted Voting (WV), where each vote has a weight proportional to the estimated generalization performance of the corresponding classifier, usually works better than the simple voting [1].

A number of selection techniques have also been proposed to solve the integration problem. One of the most popular and simplest selection techniques is Cross-Validation Majority (CVM, also called Single Best) [9]. In CVM, cross-validation accuracy for each base classifier is estimated, and then the classifier with the highest accuracy is selected.

The approaches described above are static. They select one model for the whole instance space or combine the models uniformly. In dynamic integration each new instance to be classified is taken into account. Usually, better results can be achieved if integration is dynamic.

We consider in our experiments three dynamic techniques based on the same local error estimates: Dynamic Selection (DS), Dynamic Voting (DV), and Dynamic Voting with Selection (DVS) [15]. They contain two main phases. First, at the learning phase, the local classification errors of each base classifier for each instance of the training set are estimated according to the 1/0 loss function using cross validation. The learning phase finishes with training the base classifiers on the whole training set. The application phase begins with determining *k* nearest neighbours for a new instance. Then, weighted nearest neighbour regression is used to predict the local classification errors of each base classifier for the new instance.

After that, in DS a classifier with the least predicted local classification error is selected. In DV, each base classifier receives a weight that is proportional to its estimated local accuracy, and the final classification is produced as in WV. In DVS, the base classifiers with the highest local classification errors are discarded (the classifiers with errors that fall into the upper half of the error interval) and locally weighted voting (DV) is applied to the remaining classifiers.

The dynamic integration techniques that we consider were constructed and previously applied to stationary problems only (‘dynamic’ in the name refers to locality, or being different for each particular instance). In this paper, we consider their application to data streams with changing concept and data distributions, so that ‘dynamic’ in the name is now more justified, meaning being local as well as different in time. Clearly, the cross-validation procedure used for getting performance estimates for the base classifiers becomes irrelevant in the new context, where base classifiers are built on data corresponding to different time intervals.

Currently, in ensembles used for handling concept drift, the most common integration techniques are voting and weighted voting [8, 10, 11, 16], although as we demonstrate in this paper they perhaps are not the most appropriate techniques especially when local concept drift is present.

4. Application domain: antibiotic resistance in nosocomial infections

Infections acquired during a hospital stay are called nosocomial infections. Formally, they are defined as infections arising after 48 hours of hospital admission. For earlier periods it is assumed that the infection arose prior to admission, though this is not always true [6].

Nosocomial infections are the inevitable consequence of long treatment, especially in Intensive Care Units (ICUs). The first step of this process is the colonization of skin and mucous tunic by hospital microorganism cultures. The peculiarity of these cultures is the acquisition of unpredictable antibiotic resistance according to the policy of the use of antimicrobial medications in the present department or institution.

Factors, contributing to nosocomial infections, include the defects of mucous tunics and skin, long lasting artificial ventilation of lungs, long catheterization of vessels and urinary tracts, implantation of foreign bodies and prosthetic devices, insufficient feeding, decrease in the resistance of organism etc.

Multiple investigations, conducted in different institutions, have shown the possibility of reduction of the number of nosocomial infections by about one third at maximum, even when optimal organization of the treatment process is used. The use of antibiotics with the objective of prophylaxis of nosocomial infections has proven to be ineffective, as pathogens become resistant to the antibiotics used. Normally, antibiotics are used strictly according to indications only.

To treat nosocomial infections, at first a microbiological investigation is normally conducted. In this investigation pathogens are isolated and, for each isolated bacterium, an antibiogram is built (which represents bacterium’ s resistance to a series of antibiotics). The user of the test system can define the set of antibiotics used to test bacterial resistance. The result of the test is presented as an antibiogram, which is a vector of couples “antibiotic/resistance”. The information included in the antibiogram is used to prescribe an antibiotic with a desired level of resistance for the isolated pathogen.

An antibiogram is not uniquely identified given bacterium species, but it can sometimes vary for the bacteria of the same species. This is due to the fact that bacteria of the same species may have evolved differently and have developed different resistances to antibiotics. However, quite often groups of antibiotics still have similar sensitivity when tested on a given bacterium species, despite its strains [6].

Antibiotics, also known as antimicrobial drugs, are drugs that are used to fight against infections caused by bacteria. After their discovery in the 1940' s they transformed medical care and dramatically reduced illness and death from infectious diseases. However, over the decades the bacteria that were controlled by antibiotics have developed resistance to these drugs. Today, virtually all important bacterial infections throughout the world are becoming resistant. Infectious microorganisms are developing resistance faster than scientists can create new drugs. This problem is known as *antibiotic resistance*, also known as antimicrobial resistance or drug resistance [12].

Antibiotic resistance is an especially difficult problem for nosocomial infections in hospitals because they attack critically ill patients who are more vulnerable to infections than the general population and therefore require more antibiotics. Heavy use of antibiotics in these patients hastens the mutations in bacteria that bring about drug resistance [12]. Persons infected with drug-resistant organisms are more likely to have longer hospital stays and require treatment with second or third choice drugs that may be less effective, more toxic, and more expensive [12]. In short, antimicrobial resistance is driving up health care costs, increasing the severity of disease, and increasing the death rates of some infections.

5. Data collection and organization

An important problem with most real-world datasets in known experimental investigations with concept drift is that there is little concept drift in them, or the drift is introduced artificially [13]. In contrast to that, in our experiments we use a real-world data set with a significant amount of inherent concept drift [14].

The data for our analysis were collected in the N.N. Burdenko Institute of Neurosurgery, Russia, over the years 2002-2004, using a bacterial analyzer Vitek-60 (developed by *bioMérieux,* www.biomerieux-vitek.com). Each instance of the data used in analysis represents one sensitivity test and contains the following features: pathogen that is isolated during the bacterial identification analysis, antibiotic that is used in the sensitivity test and the result of the sensitivity test itself (sensitive S, resistant R or intermediate I), obtained by the Vitek analyzer according to the guidelines of National Committee for Clinical Laboratory Standards (NCCLS) [5]. The information about sensitivity analysis is related to a patient, his/her demographical data (sex, age) and hospitalization in the institute (main department, days spent in ICU, days spent in the hospital before test, etc.). Each instance included 34 features altogether.

Each instance of microbiological test in the database corresponds to a single specimen that may be blood, cerebrospinal fluid (liquor), urine, etc. In this study we focus on the analysis of meningitis cases only, and the specimen is liquor. For the purposes of our analysis we picked up 4430 instances of sensitivity tests including the meningitis cases of the years 2002-2004.

This dataset was considered earlier in [14] where it was experimented with using different inductive learning and dimensionality reduction algorithms and the principle of natural clustering. Besides, it was shown that there is a significant level of concept drift pertinent to this domain. The dataset was divided into sets of sequential blocks of different length, and these blocks were used as train and test sets sequentially in order to analyze underlying concept and data distribution changes. A few interesting findings were discovered, including seasonal context recurring with winter and spring models, which corresponded to yearly spring and winter infection outbreaks.

6. Experimental studies

In our experimental studies we used an implementation based on the machine learning library WEKA 3.4.2 (available at http://www.cs.waikato.ac.nz/~ml/weka/), which is currently perhaps the most popular library of machine learning algorithms [18]. Default settings were used in the WEKA learning algorithms used in our experiments. With all ensembles considered here we use the simple so-called "replace the loser" ensemble pruning strategy. With this strategy, if the ensemble size is greater than or equal to 25, the worst classifier, according to the current validation estimates, is replaced with a new one trained on the most recent data.

So-called progressive evaluation was used in order to evaluate different integration techniques. The most recent window before the current test block was used in order to get the validation estimates for dynamic integration and weighted voting. In order to avoid overly optimistic validation estimates for the last classifier in the ensemble, which is built on data from the current validation block, 10-fold cross validation was used.

As the basis of our analysis we consider the classification problem aimed at predicting the sensitivity of a pathogen to an antibiotic based on data about the antibiotic, the isolated pathogen, and the demographic and clinical features of the patient.

To build ensembles, we divide the data into blocks corresponding to a certain time interval. We use a sliding window approach, and thus, when the window shift is less than the size of the window, the data blocks are not mutually exclusive. We use the last (current) data block as the test set, and the current ensemble includes only those base classifiers that are built on data blocks including different instances only with regard to the test set in order to avoid an overly optimistic error estimate for the ensemble. After the block was considered as a test set and accuracy characteristics were calculated for it, a model built on it is included into the ensemble, and the next block is selected for testing, according to the given values of window size and window shift.

We evaluate five different integration strategies in our experiments: Voting (V), Weighted Voting (WV), Dynamic Selection (DS), Dynamic Voting (DV), and Dynamic Voting with Selection (DVS). Beside the accuracies for integration strategies, we collect also the minimum, average and maximum accuracies of the base classifiers in the ensemble (min, aver, max).

We have tried a number of learning algorithms available in WEKA (such as Naïve Bayes, C4.5 decision tree, k-NN etc.) for the base classifiers in ensembles. The accuracy of base models is usually radically different for different learning algorithms with this dataset. For example, for one set of data blocks, the maximum base classifier accuracy with Naïve Bayes can be 0.6 only, but 0.75 with decision trees and instance-based learning. However, the final ensemble accuracy was not so different with different learning algorithms, and it always achieved 0.80 on average with the dynamic approaches (weighted average accuracy according to the number of instances in each block).

Weighted voting was always a little better than plain majority voting, supporting previous experiments [1]. Dynamic integration was always better than the best base classifier and weighted voting regardless of the learning algorithm, window size and window shift in this domain. The maximum gain by dynamic integration was achieved with Naïve Bayes: for some blocks where the maximum base classifier accuracy was 0.60 only, the accuracy of dynamic integration was 0.85. On average with Naïve Bayes, the accuracy of WV was 0.65, best base classifier accuracy 0.69, and the accuracy of DVS (best dynamic strategy) 0.81. The accuracies of three dynamic strategies (DS, DV, and DVS) are always close to each other, and the relative efficacy of them depends on the selected learning algorithm, window size and window shift, however DVS and DS seem to be a little superior to DV.

In Figures 2 and 3 example experimental results are shown for C4.5 decision tree, window size of 3 months, and window shift of 1 month. With this configuration, the best weighted average ensemble accuracy was achieved with this dataset (0.86 with DS, Figure 3).

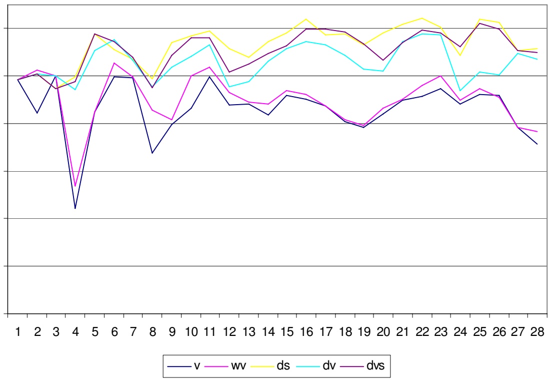

Figure 2. Classification accuracy over sequential data blocks (C4.5 ensembles)

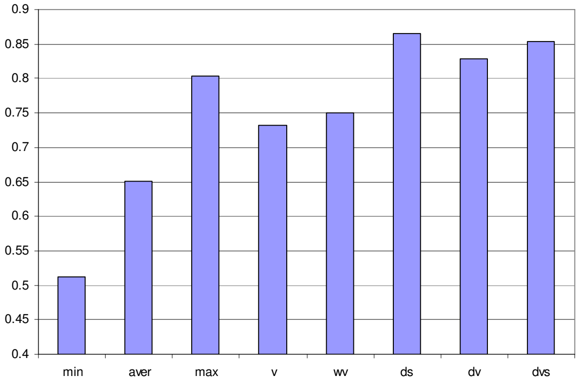

Figure 3. Weighted average of classification accuracy (C4.5 ensembles)

It is important to note that this accuracy is high for this dataset, and it beats the best stationary batch learning algorithm for this domain (k-NN in combination with wrapper feature selection gives 0.82) on the whole dataset using 10-fold cross-validation, where the training sets are substantially larger (3987 instances). Besides, with our ensemble construction, the first ensembles including a few members only, naturally, are very weak. If we consider ensembles including 7 classifiers and more (omit the first 6 points in Figure 2), the average accuracy with dynamic integration is more than 0.88. From Figure 3 one can see that the dynamic integration techniques improve ensemble accuracy by more than 10% on average in this domain.

7. Conclusions

In this paper we suggest a dynamic integration approach for ensembles used in handling concept drift, which integrates the base classifiers at an instance level, assigning to them weights proportional to their local accuracy on each instance considered.

Our experiments demonstrated that dynamic integration results in better accuracy on the antibiotic resistance dataset than the most commonly used weighted voting. This supports our hypothesis that dynamic integration can be a more appropriate integration technique, and that it might be especially useful in the presence of local concept drift.

We plan to apply this approach to other datasets with concept drift, including commonly used benchmark synthetic data with different characteristics of concept drift and other real-world data including biomedical and financial data streams, where concept drift is a common phenomenon, in order to better study the effect of dynamic integration on handling various concept drifts.

Acknowledgements. This material is based upon work supported by the Science Foundation Ireland under Grant No. S.F.I.-02/N.1/111. This research was partly supported by the Academy of Finland, and the Graduate School COMAS of the University of Jyväskylä, Finland. We are thankful to Prof. Michael Shifrin and Dr. Irina Alexandrova from N.N.Burdenko Institute of Neurosurgery, Moscow, Russia for the database used in our study.

References

[1] Bauer E., Kohavi R. An empirical comparison of voting classification algorithms: bagging, boosting, and variants, *Machine Learning*, 36 (1,2), 1999, 105-139.

[2] Brodley C., Lane T. Creating and exploiting coverage and diversity. In: Proc. AAAI-96 Workshop on Integrating Multiple Learned Models, Portland, OR, 1996, 8-14.

[3] Delaney S.J., Cunningham P., Tsymbal A., Coyle L. A case-based technique for tracking concept drift in spam filtering. In: Proc. 24th SGAI Int. Conf. on Innovative Techniques and Applications of Artificial Intelligence, Cambridge, UK, Springer, LNCS, 2004, 3-16.

[4] Fan W. Systematic data selection to mine concept-drifting data streams. In: Proc. 10th Int. Conf. on Knowledge Discovery and Data Mining KDD'04, 2004.

[5] Ferraro M.J., et al. Methods for dilution antimicrobial susceptibility tests for bacteria that grow aerobically: approved standard; Performance standards for antimicrobial susceptibility testing. Wayne, PA: National Committee for Clinical Laboratory Standarts, NCCLS, 2005. (Documents M7-A6 and M100-S14, available at www.nccls.org).

[6] Gaynes R.P. Surveillance of nosocomial infections: a fundamental ingredient for quality. Infect Control Hosp Epidemiol, 18(7), 1997, 475– 478.

[7] Klinkenberg R. Learning drifting concepts: example selection vs. example weighting, *Intelligent Data Analysis, Special Issue on Incremental Learning Systems Capable of Dealing with Concept Drift*, 8 (3), 2004.

[8] Kolter J.Z., Maloof M.A. Dynamic weighted majority: a new ensemble method for tracking concept drift. In: 3rd IEEE Int. Conf. on Data Mining ICDM'03, IEEE CS Press, 2003, 123-130.

[9] Schaffer C. Selecting a classification method by cross-validation, *Machine Learning*, 13, 1993, 135-143.

[10] Stanley K.O. Learning concept drift with a committee of decision trees, Tech. Report UT-AI-TR-03-302, Dept of Computer Science, University of Texas at Austin, USA, 2003.

[11] Street W., Kim Y. A streaming ensemble algorithm (SEA) for large-scale classification. In: Proc. 7th Int. Conf. on Knowledge Discovery and Data Mining KDD-2001, 2001, 377-382.

[12] The Problem of Antibiotic Resistance, NIAID Fact Sheet. National Institute of Allergy and Infectious Diseases (NIAID), National Institutes of Health, U.S. Department of Health and Human Services, USA, 2004 (available at www.niaid.nih.gov/factsheets/antimicro.htm).

[13] Tsymbal A. The problem of concept drift: definitions and related work, Tech. Report TCD-CS-2004-15, Dept. of Computer Science, Trinity College Dublin, Ireland, 2004.

[14] Pechenizkiy M., Tsymbal A., Puuronen S., Shifrin M., Alexandrova I. Knowledge discovery from microbiology data: many-sided analysis of antibiotic resistance in nosocomial infections. In: 3rd Int. Conf. on Professional Knowledge Management: Experience and Visions (WM05), Springer, LNAI 3782, 2005, 360-372.

[15] Tsymbal A., Puuronen S. Bagging and boosting with dynamic integration of classifiers. In: D.A. Zighed, J. Komorowski, J. Żytkow (eds.), Principles of Data Mining and Knowledge Discovery, Proceedings of PKDD 2000, Springer, LNAI 1910, 2000, 116-125.

[16] Wang H., Fan W., Yu P.S., Han J. Mining concept-drifting data streams using ensemble classifiers. In: Proc. 9th Int. Conf. on Knowledge Discovery and Data Mining KDD'03, 2003, 226-235.

[17] Widmer G., Kubat M. Learning in the presence of concept drift and hidden contexts, *Machine Learning*, 23 (1), 1996, 69-101.

[18] Witten I., Frank E. Data Mining: Practical Machine Learning Tools With Java Implementations, San Francisco: Morgan Kaufmann, 2000.