

Dynamic Integration of Classifiers for Tracking Concept Drift in Antibiotic Resistance Data

Alexey Tsymbal¹, Mykola Pechenizkiy², Pádraig Cunningham¹, Seppo Puuronen²

¹Department of Computer Science, Trinity College Dublin, Ireland,
{Alexey.Tsymbal, Pdraig.Cunningham}@cs.tcd.ie

²Department of Computer Science and Information Systems, University of Jyväskylä, Finland,
{mpechen, sepi}@cs.jyu.fi

Abstract. In the real world concepts are often not stable but change with time. A typical example of this in the medical context is antibiotic resistance, where pathogen sensitivity may change over time as new pathogen strains develop resistance to antibiotics which were previously effective. This problem, known as concept drift, complicates the task of learning a model from medical data and requires special approaches, different from commonly used techniques, which treat arriving instances as equally important contributors to the final concept. The underlying data distribution may change as well, making previously built models useless, which is known as virtual concept drift. These changes make regular updates of the model necessary. 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 according to their expertise level regarding the current concept. In this paper we propose a new ensemble integration technique that helps to better track concept drift at the instance level. Our experiments with the antibiotic resistance data show that dynamic integration of classifiers built over small time intervals can be more effective than the best single learning algorithm applied in combination with feature selection, which gives the best known accuracy for the considered problem domain. Besides, dynamic integration is significantly better than weighted voting which is currently the most commonly used integration approach for tracking 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 long period of time [13, 18]. There are many known cases of concept drift in medical data, and among the most important examples in the medical context is antibiotic resistance, where pathogen sensitivity may change over time as new pathogen strains develop resistance to antibiotics which were previously effective [12, 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 [7, 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. For example, only particular bacteria may develop their resistance to certain antibiotics, while resistance to the others can remain the same; or data distribution can change for particular bacteria only.

The most common integration approach with ensembles for tracking concept drift is weighted voting, where each base classifier receives a weight proportional to its relevance to the current concept [7, 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 parts 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 instance level. In dynamic integration, each base classifier receives a weight proportional to its local accuracy in the neighbourhood of the current 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 they attack critically ill patients who are more vulnerable to infections than the general population and therefore require more antibiotics.

Our experiments show that dynamic integration achieves much better classification accuracy than commonly used integration techniques such as voting and weighted voting on the problem of antibiotic resistance prediction, supporting our hypothesis that it is a better technique for tracking concept drift.

This paper is organized as follows: in Section 2 we consider the general problem of concept drift, in Section 3 we review approaches to ensemble integration with a focus on dynamic integration, in Section 4 we present the domain of antibiotic resistance in nosocomial infections, in Section 5 we consider the basic characteristics of our dataset used for analysis, in Section 6 we present the results of our experiments with the use of different ensemble integration techniques with the given dataset, 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 can induce more or less radical changes in the target concept, which is generally known as *concept drift* [18]. A typical example of this in the medical context is antibiotic resistance, where pathogen sensitivity may change over time as new pathogen strains develop resistance to antibiotics which were previously effective. An effective learner should be able to track such changes and to quickly adapt to them. Kukar [8] states that even in most strictly controlled environments some unexpected changes may happen due to fail and replacement of some medical equipment, or due to changes in personnel, causing the necessity to change the model.

Changes in hidden context may not only be a cause of a change of 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* [17]. 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 approaches to handling concept drift can be distinguished: (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 [18]. 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 [6]. Instances can be weighted according to their "age", and their competence with regard to the current concept. Klinkenberg [6] shows 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 [7] build ensembles of incremental learners in an online setting, starting to learn new base classifiers after fixed intervals while continuing to update existing ones. All incremental ensemble approaches use some criteria to dynamically delete, reactivate, or create new ensemble members, which are normally based on the base models' consistency with the current data.

In the real world, concept drift may often be local, e.g. only particular bacteria may develop their resistance to certain antibiotics, while resistance to the others could remain the same. In the case of local concept drift, models of an ensemble should not be discarded (or lower weights assigned to them) simply because their global accuracy on the current block of data falls, even if they are still good experts in the stable parts of the data. This is an important problem with the existing ensemble approaches for tracking concept drift. One solution to this problem is the use of dynamic integration of classifiers, in which the models are integrated at instance level according to their local accuracies.

3 Ensembles 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 the 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 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, the cross-validation accuracy for each base classifier is estimated, and then the classifier with the highest accuracy is selected.

The described above approaches are *static*. They select one model for the whole data 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, DS simply selects a classifier with the least predicted local classification error. 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.

In the context of ensembles for tracking concept drift the most commonly used integration techniques are voting and weighted voting [7, 10, 11, 16], although as we show in this paper they are not the most appropriate techniques especially when local concept drift is present.

4 Domain Area: 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 [5].

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 [5].

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.

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 from Vitek according to the guidelines of National Committee for Clinical Laboratory Standards (NCCLS) [4]. The information about sensitivity analysis is connected with *patient*, his or her demographical data (*sex*, *age*) and hospitalization in the institute (*main department*, *days spent in ICU*, *days spent in the hospital before test*, etc.). This dataset was also considered in [14] where a subset of the data from the year 2002 was studied using various machine learning techniques with the purpose of exploratory analysis.

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.

After a discussion with medical experts, we have formed new binary features for antibiotics and pathogens corresponding to the tree-like categorization of them. For antibiotics the 35 different classes of them were grouped into one feature with 4 major categories and for pathogens the 16 different classes were grouped into one feature with 7 major groups.

Each instance included 34 features that contained information corresponding to a single sensitivity test augmented with data concerning the antibiotic used, the isolated pathogen, clinical characteristics of the patient and his/her demographics. These features are summarized in Table 1.

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 [19].

As the base 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 (cur-

rent) 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 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.

Table 1. Features used in the dataset

Name	Type
<u>Patient and hospitalization related</u>	
sex	{Male, Female}
age	Integer
recurring stay	{True, False}
days of stay in NSI	Integer
days of stay in ICU	Integer
days of stay in NSI before specimen was received	Integer
bacterium is isolated when patient is in ICU	{True, False}
main department	{1, ..., 10}
department of stay (departments + ICU)	{1, ..., 11}
<u>Pathogen and pathogen groups</u>	
pathogen name	{Pat_name1, ..., Pat_name17}
gram(+)	{True, False}
staphylococcus	{True, False}
enterococcus	{True, False}
enterobacteria	{True, False}
nonfermenters	{True, False}
<u>Antibiotic and antibiotic groups</u>	
antibiotic name	{Ant_name1, ..., Ant_name39}
group1	{True, False}
...	...
group15	{True, False}
Sensitivity	{Sensitive, Intermediate, Resistant}

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). 7 nearest neighbours were used in the experiments with dynamic integration, which was supported by a set of preliminary cross-validation experiments. In dynamic integration, local accuracies were estimated using instances from the current block, except the instance being tested. Thus, a form of leave-one-out cross validation was used. Beside the 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 dynamic approaches (weighted average 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. The maximum gain by dynamic integration was achieved with Naïve Bayes: for some ensembles 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.

In Figures 1 and 2 example experimental results are shown for C4.5 decision tree, window size of 90 days, and window shift of 30 days. With this configuration, the best average ensemble accuracy was achieved with this dataset (0.82 with DS, Figure 2). It is important to note that this accuracy is very high for this dataset, and it can be achieved by the best batch learning algorithm for this domain (k -NN in combination with wrapper feature selection) on the whole dataset using 10-fold cross-validation, where the training sets are substantially larger (3987 instances). 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 1), the average accuracy with dynamic integration is more than 0.85, which is significantly better than all batch learning techniques.

From Figure 2 one can see that the dynamic integration techniques improve ensemble accuracy by more than 10% on average.

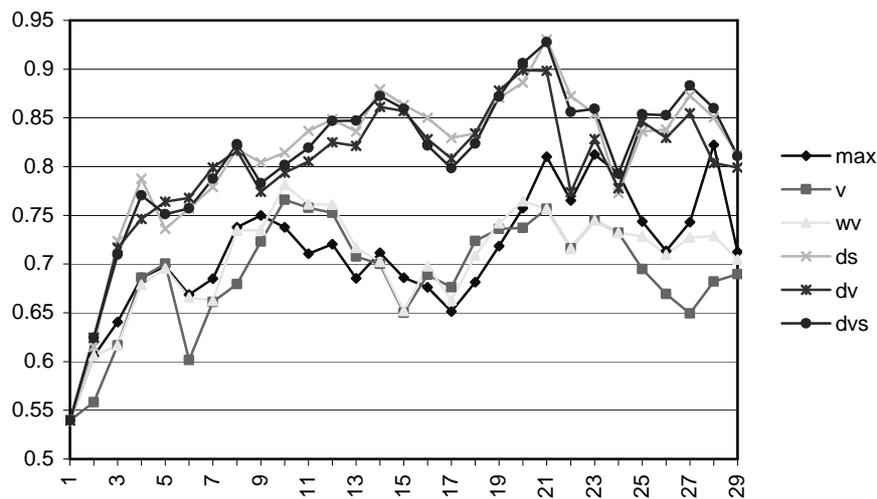


Fig. 1 – Classification accuracy over sequential data blocks (ensembles of C4.5 decision trees)

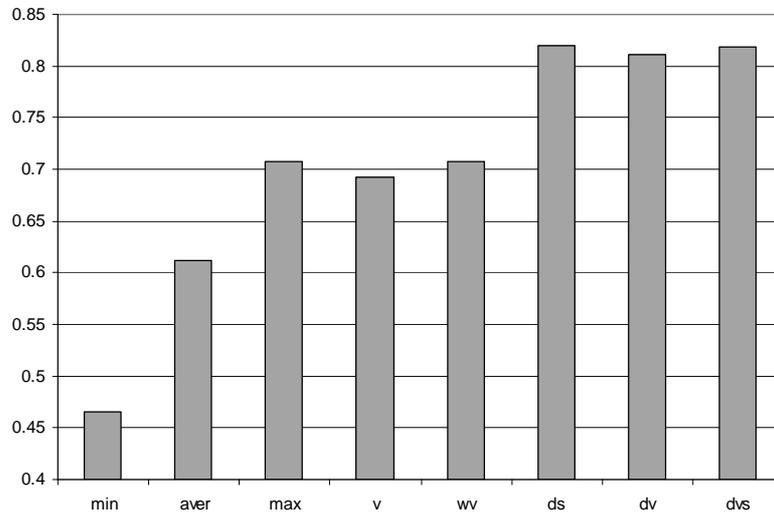


Fig. 2 – Weighted average of classification accuracy (ensembles of C4.5 decision trees)

7 Conclusion

In the real world concepts are often not stable but change with time, which is known as the problem of concept drift. Concept drifts complicate the task of learning and require unusual solutions, different from commonly used batch learning techniques. In this paper we consider an example of concept drift from the area of antibiotic resistance, where pathogen sensitivity may change over time as new pathogen strains develop resistance to antibiotics which were previously effective. Among the most popular and effective approaches to tracking concept drift is ensemble learning, where a set of concept descriptions built on data blocks corresponding to different time intervals is maintained, and the final prediction is the aggregated prediction of ensemble members.

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

Our experiments have shown that dynamic integration results in much better accuracy with this antibiotic resistance dataset than the most commonly used weighted voting. This supports our hypothesis that dynamic integration is a more appropriate integration technique for tracking concept drift, and that it is especially useful in the presence of local concept drift.

We believe that dynamic integration of classifiers can become a reliable tool to fight concept drift for data streams in many subject areas. As a direction for the future research we plan to consider its application to other datasets with inherent concept drift, including medical data streams, where concept drift is a common phenomenon.

Acknowledgements: This material is based upon work supported by the Science Foundation Ireland under Grant No. S.F.I.-02IN.1I111. 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, 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.
4. 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 Standards, NCCLS, 2005. (Documents M7-A6 and M100-S14, www.nccls.org).
5. Gaynes R.P. Surveillance of nosocomial infections: a fundamental ingredient for quality. *Infect Control Hosp Epidemiol*, 18(7), 1997, 475– 478.
6. 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.
7. 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-2003*, IEEE CS Press, 2003, 123-130.
8. Kukar M. Drifting concepts as hidden factors in clinical studies. In: *Proc. 9th Conf. on Artificial Intelligence in Medicine in Europe, AIME 2003*, Springer, LNCS, 2003, 355-364.
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, *Technical Report TCD-CS-2004-15*, Department of Computer Science, Trinity College Dublin, Ireland, 2004.
14. Tsymbal A., Pechenizkiy M., Puuronen S., Shifrin M., Alexandrova I. Knowledge discovery in microbiology data: antibiotic resistance analysis. In: *Proc. Current Aspects of Knowledge Management in Medicine (KMM05)*, Springer, LNCS, 2005.
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-2003, 2003, 226-235.
17. Widmer G., Kubat M. Effective learning in dynamic environments by explicit context tracking. In: Proc. 6th European Conf. on Machine Learning ECML-1993, Springer, LNCS 667, 1993, 227-243.
18. Widmer G., Kubat M. Learning in the presence of concept drift and hidden contexts, *Machine Learning*, 23 (1), 1996, 69-101.
19. Witten I., Frank E. *Data Mining: Practical Machine Learning Tools With Java Implementations*, San Francisco: Morgan Kaufmann, 2000.