Visualization and Concept Drift Detection Using Explanations of Incremental Models

Jaka Demšar, Zoran Bosnić and Igor Kononenko
University of Ljubljana, Faculty of Computer and Information Science
Večna pot 113, 1000 Ljubljana, Slovenia
E-mail: jaka.demsar0@gmail.com, {zoran.bosnic, igor.kononenko}@fri.uni-lj.si

Keywords: data stream mining, concept drift detection, visual perception

Received: October 10, 2014

The temporal dimension that is ever more prevalent in data makes data stream mining (incremental learning) an important field of machine learning. In addition to accurate predictions, explanations of the models and examples are a crucial component as they provide insight into model's decision and lessen its black box nature, thus increasing the user's trust. Proper visual representation of data is also very relevant to user's understanding – visualization is often utilised in machine learning since it shifts the balance between perception and cognition to take fuller advantage of the brain's abilities. In this paper we review visualisation in incremental setting and devise an improved version of an existing visualisation of explanations of incremental models. Additionally, we discuss the detection of concept drift in data streams and experiment with a novel detection method that uses the stream of model's explanations to determine the places of change in the data domain.

Povzetek: V članku predstavimo novo vizualizacijo razlage inkrementalnih klasifikacijskih modelov in posameznih napovedi. Predlagamo tudi metodo zaznave spremembe koncepta, temelječo na nadzorovanju toka razlag.

1 Introduction

Data streams are becoming ubiquitous. This is a consequence of the increasing number of automatic data feeds, sensoric networks and internet of things. The defining characteristics of data streams are their transient dynamic nature and temporal component. In contrast with static (tabular) datasets (used in *batch* learning), data streams (used in *incremental* learning) can be large, semi-structured, incomplete, irregular, distributed and possibly unlimited. This poses a challenge for storage and processing which should be done in constant time – for incremental models, operations of *increment* (fast update of the model with the new example) and *decrement* (forgetting old examples) are vital. Concepts and patterns in data domain can change (*concept drift*) – we need to adapt to this phenomenon or the quality of our predictions deteriorates. We discuss data streams and concept drift more in Section 2.1.

Bare prediction quality is not a sufficient property of a good machine learning algorithm. *Explanation* of individual predictions and model as a whole is needed to increase the user's trust in the decision and provide insight in the workings of the model, which can significantly increase the model's credibility. The model independent methods of explanation have been developed for batch [12, 13] and incremental learning [2] (Section 2.2).

Data visualisation is a versatile tool in machine learning that serves for sense-making and communication as it conveys abstract concepts in a form, understandable to humans. In Section 2.3 we discuss visual display of data in an incremental setting and describe the improper visualisation of explanation of incremental models. The main goal of the article is its improvement, which is presented in Section 3. An additional goal (Section 4) was to devise a method of concept drift detection which monitors the stream of explanations. Finally, we test the improved visualization and the novel concept drift detection method on two datasets and evaluate the results (Section 5).

2 Related work

2.1 Data stream mining

In incremental learning, we can observe possibly infinite data stream of pairs (x_i, C_i), where x_i is the $i-th$ instance and C_i is its true label. After the model makes a prediction $p_i = \hat{C}_i$, the environment reacts with a feedback which can be used to assess the model's performance (in the case of classifiers, the instance's true label becomes available). According to *PAC* (*Probably approximately correct*) learning model, if the distribution, generating the instances is stationary, the error rate will, at least for sound machine learning algorithms, decline towards the Bayes error rate as the number of processed instances increases [10]. Consequently, when a statistically significant rise in error rate

is detected, we can suggest that there has been a change in the generating distribution – concept drift. The nature of change is described with two dimensions: the cause of change (changes in data, hidden variables, changes in the context of learning...) and the rate of change. The change detection can be also completely left out (using windows and blindly forcing the operations of increment and decrement). The other approach is to actively detect change either by monitoring the evolution of performance measures or comparing distributions over two time windows. In the explanation methodology (Sections 2.2 and 4) we use two methods from the former group.

The basis of monitoring the learning process using *statistical process control* (*SPC*) [4] is detecting significant error rate using central limit theorem. Each processed instance is in one of the three possible states - *in control*, *out of control* and *warning* when the system is in between the former states. When *in control*, the current model is incrementally updated with the current instance, since the error rate is stable. In *warning* state, a buffer is filled with incoming instances – it serves as an optimally-sized container for data that is relevant for the new model if the drift occurs (*out of control* state) – the buffer is then used to construct a new learning model form the instances in it. If the system goes from *warning* back to *in control*, the buffer is emptied, since we deemed the error rise to be a false alarm.

The other method, *Page-Hinkley test* [8], is simpler in its nature. It was devised to detect the change of a Gaussian signal and is commonly used in signal processing. The method's behaviour can be controlled with parameters λ (threshold for alarm) and δ, which corresponds to the allowed magnitude of changes.

2.2 Explanation of models and individual predictions

Explanation of individual predictions and prediction models is used to gain additional insight into the model's decision and to lessen the black box nature of most predictions. The adequately explained prediction increases the user's trust and understanding of the model and extracts additional information.

Although some models are already transparent (e.g. decision trees, Bayesian networks) and model-dependent methods of explanation exist for most others, these methods do not meet the requirement for *consistency of explanation*, which enables comparison of explanations across different models. *IME* (*Interactions-based Method for Explanation*) [13] with its efficient adaptation [12] is a model independent method of explanation, which also addresses interactions of features and therefore successfully tackles the problem of redundant and disjunctive concepts in data.

The explanation of the prediction p_i for instance x_i is defined as a vector of *contributions of individual feature values* $(\varphi_1, \varphi_2, \ldots, \varphi_n)$ where φ_j is the contribution of the value of the j-th feature. Positive φ_j implies that the feature value positively contributed to the prediction (and vice versa) while the absolute value $|\varphi_j|$ is proportional to the magnitude of influence on the decision. The sum of all contributions is equal to the difference between the prediction using all feature values and a prediction using no features (*prediction difference*). The explanation quality in highly correlated with prediction quality [13]. Consequently, for very good models, the explanation gives us an insight not only into the working of the built model, but also into concepts behind the domain itself.

In order to discover dependencies among feature values, 2^n subsets of features are considered. An efficient sampling-based approximation method exists [12], where the explanation of a prediction is modeled with game theory as a cooperative game (N, Δ) where players N are features and Δ is the prediction difference. The payoff vector divides the prediction difference among feature values as contributions that correctly explain the prediction[1]. The explanation of a single prediction can be expanded to the whole model by iterating thorough all possible feature values and using contributions of randomly generated instance pairs to compute the contributions of each attribute value.

An adaptation of existing explanation methodology to incremental setting is proposed in [2]. In addition to the efficiency requirements, the key feature of incremental learning to consider is the possibility of a concept drift. The explanation in incremental learning should therefore itself be a data stream. An adaptation is developed by considering the existing model-independent method in batch learning [12] and equipping it with drift detection and adaptation methods.

In the incremental explanation methodology, the basic concept is slightly modified by introducing the parameter max_window. It acts as a limiter for maximum size of the learning model, by narrowing down the model by FIFO principle, if necessary. The batch explanation is integrated at key points in SPC[2] (changes of states), resulting in a granular stream of explanations which reflect local areas of static distributions and intermediate areas of concept drift. An optional parameter ω determines triggering of periodic explanations independent of change indicators. Aside from periodic explanations, the granularity of the stream is completely correlated with change detection – the explanation of the model occurs only when error rate significantly increases, indicating a change in the distribution of generating the instances[3]. Explaining an individual prediction follows the same process as in batch explanation, only that the local learning model is used.

[1] The concept of Shapley value is used as a solution.

[2] SPC meets the requirement of model-independence – it works as a wrapper for an arbitrary classifier. In addition to the occurrence of a concept drift, it also detects its rate (the smaller the buffer is between *warning* and *out of control*, the faster the drift occurs)

[3] The resulting stream of explanations itself can be a subject of analysis – if the model is good, patterns found in the explanation stream may reflect the patterns behind the domain.

2.3 Data visualisation

Data visualization is often utilised in machine learning since it shifts the balance between perception and cognition to take fuller advantage of the brain's abilities [3]. A data stream can be seen as a collection of observations made sequentially in time. Sampling based research shows that the majority of visualizations depict data that has a temporal component [9]. In this context, visualization acts as a form of *summarization*. The challenge lies in representing the temporal component, especially if we are limited to two-dimensional non-interactive visualisations. Concept drift is the other property in data streams that makes creating good visualization a hard task. Even if we manage to effectively summarize and display patterns in data at some point, we are still left with the task of displaying the change in them.

The main goal of this paper is improving the existing methodology for visualising explanations of incremental models [2]. The feature value contributions are represented with customised bar charts. When explaining a static model, all possible values are listed along the axis with mean negative and mean positive contributions of each feature. When plotting multiple such visualisations (as is the case in locally explaining incremental models based on change detection) they become very difficult to read as a whole because of the large number of visual elements that we have to compare (we sacrifice macro view completely in favour of micro view). To consolidate these images and address the *change blindness* [7] phenomenon, charts are stacked into a single plot, where the age and size of the explanation are represented with transparency (older and "smaller" explanations fade out). The resulting visualisation is not tainted by first impressions (as it is only one image) and is adequately dense and graphically rich. However, the major flaw of this approach lies in the situations when columns, representing newer explanations override older ones and thus obfuscate the true flow of changing explanations, for example, when the concept drift precipitates the attribute value contributions to increase in size without changing the sign. Concepts can therefore become not only hidden; what's more, the visualization can be deceiving, which we consider to be worse than just being too sparse. Therefore, we need to clarify the presentation of the concept drift along with an accurate depiction of each explanation's contributions while maintaining the macro visual value, that enables us to detect patterns and get a sense of true concepts and flow of changes behind the model.

3 Improved visualisation

When visualising explanations of individual predictions, horizontal bar charts are a fitting method also in the incremental setting – we plot the mean positive and negative contribution of each attribute value and the mean of each attribute as a whole. Individual examples are always explained according to the current model which, in our case, can change. This is not an obstacle, since the snapshot of the current model is in fact the model that classified the example, so we can proceed with the same explanation methodology as in the static environment[4].

This approach fails with explanations of incremental models as we need a new figure for each local explanation (Subsection 2.3). To successfully represent the temporal component of incremental models, we use two variations of a line plot where the x axis contains time stamps of examples and the splines plotted are various representations of contributions (y axis).

The first type of visualization (examples in Figures 2 and 4) has one line plot for each attribute. Contributions of values of the individual attribute are represented with line styles. The mean positive and mean negative contribution of the attribute as a whole are represented with two thick faded lines. Solid vertical lines indicate the spots where explanation of the model was triggered (and therefore become the joints for the plotted splines), while dashed vertical lines mark the places where the actual concept drift occurs in data. The second type is an aggregated version (examples in Figures 3 and 5) where the mean positive and mean negative contributions of all attributes are visualized in one figure. In these two ways we condense the visualization of incremental models without a significant loss in information while still providing a quality insight into the model. Exact values of contributions along with timestamps of changes can be read out (micro view), while general patterns and trends can be recognised in the shapes of lines that are intuitive representations of flowing time (macro view). The resulting visualisations are dense with information, easily understandable (conventional plotting of independent variable, time, on x-axis) and presented in gray-scale palette, making them more suitable for print.

4 Detecting concept drift using the stream of explanations

When explaining incremental models, the resulting explanations are, in themselves, a data stream. This gives us the option to process them with all the methods used in incremental learning. In our case, we'll devise a method to detect outliers in the stream of explanations and declare such points as places of concept drift. The reasoning behind this is the notion that if the model does not change, then also the explanation of the whole model will not change. When an outlier is detected, we consider this to be an indicator of a significant change in the model and thus also in the underlying data. In addition to this, the method provides us with a stream of explanations that is continuous to a certain degree of granularity and so enables us to overview the concepts behind the data at more frequent intervals than the existing explanation methodology.

[4]That means that, without major modifications, we can only explain instances at the time of their classification – the model changes with the arrival of the next instance.

We use a standard incremental learning algorithm [4] (we learn by incrementally updating the model with each new example, decrement the model if it becomes too big or rebuild the model if we detect change [5]) and introduce the $granularity$ parameter which determines how often the explanation of the current model will be triggered. This will generate a stream of explanations (vectors of feature value contributions) that will be compared using cosine distance. For each new explanation, the average cosine distance from all other explanations that are in the current model, is calculated. These values are monitored using the Page Hinkley test. When an alarm is triggered, it means that the current average cosine distance from other explanations has risen significantly, which we interpret as a change in data domain – concept drift. The last $granulation$ examples are then used to rebuild the model, the Page Hinkley statistic and the local explanation storage are reset (to monitor the new model).

The cosine distance is chosen because, in the case of explanations, we consider the direction of the vector of contributions to be more important than its size, which is very influential in the traditional Minkowski distances. It also carries an intuitive meaning. The Page Hinkley test is used in favour of SPC because of its superior drift detection times [10] and the lack of need for a buffer – examples are already buffered according to the granularity. The method is therefore model independent. Algorithm 1 describes the process in a high level pseudocode.

Algorithm 1 Detecting concept drift using the stream of explanations

Require: h {classifier}
Require: $(\vec{x_i}, y_i)_t$ {data stream}
Require: m {number of samples for *IME*}
Require: g {explanation granulation}
Require: max_window {maximum size of the model}
Require: λ {Page Hinkley threshold}
  $local\_explanations \leftarrow []$
  $\Phi = h$ {Incremental classifier}
  $buf = []$ {buffer}
  for $(\vec{x_i}, y_i) \in (\vec{x_i}, y_i)_t$ **do**
    $\Phi = \Phi.increment(\vec{x_i})$
    if $len(\Phi) > max\_window$ **then**
      $\Phi = \Phi.decrement()$
    end if
    $buf.append((\vec{x_i}, y_i))$
    if $len(buf) > g$ **then**
      $buf = buf[1:]$ {Maintain buffer size}
    end if
    if $i\%g == 0$ **then**
      $\phi_i = IME(\Phi)$ {Explain the current model}
      $dist\_\phi_i \leftarrow mean(cos\_dist(\phi_i, \phi')$
        $\forall \phi' \in local\_explanations)$
      if $PageHiknley(dist\_\phi_i) = ALERT$ **then**
        $\Phi = h(buf)$ {Rebuild}
        $local\_explanations \leftarrow []$
      end if
      $local\_explanations \leftarrow local\_explanations \cup \phi$
    end if
  end for

5 Results

5.1 Testing methodology and datasets

We test the novel visualisation method and the concept drift detection method on two synthetic datasets, both containing multiple concepts with various degrees of drift between them. These datasets are also used in previous work [2], so a direct assessment of visualization quality and drift detection performance can be made. From the results of this testing, we may conclude whether the new methods are the improvements. The naive Bayes classifier and the k nearest neighbour classifier are used. Their usage yields very similar results in all tests, so only results obtained by testing with Naive Bayes are presented.

SEA concepts [11] is a data stream comprising 60000 instances with continuous numeric features $x_i \in [0, 10]$, where $i = 1, 2, 3$. x_1 and x_2 are relevant features that determine the target concept with $x_2 + x_3 \leq \beta$ where threshold $\beta \in \{7, 8, 9, 9.5\}$. Points are arranged into four blocks: $\beta = 8, \beta = 9, \beta = 7$ and $\beta = 9.5$, consecutively. Although the changes between the generated concepts are abrupt, 10% class noise is inserted into each block.

The second dataset, *STAGGER*, is generated with *MOA (Massive Online Analysis)* data mining software [1]. The instances represent geometrical shapes which are in the feature space described by discrete features *size*, *color* and *shape*. The binary class variable is determined by one of the three target concepts ($(size = small) \wedge (color = red)$, $(color = green) \vee (shape = square)$ and $(size = medium) \vee (size = large)$). We generate 4500 instances which are divided into blocks belonging to the particular concept (presented in Table 5.1). The concept drift is applied by specifying an interval of certain length between blocks – there, the target concepts of the instances mix according to the sigmoid function. Therefore, the dataset includes gradual drift, disjunction in concepts and redundant features.

Interval	Concept	Width of drift
$[0, 749]$	1	50
$[750, 1799]$	2	50
$[1800, 3599]$	3	150
$[3600, 4500]$	1	/

Table 1: STAGGER dataset used for evaluation

5.2 Improved visualizations

Concept drifts in STAGGER dataset (we use $max\_length = \omega = 500$) are correctly detected and adapted to as reflected in Figure 2. The defined concepts can be easily recognized from explanations triggered by

the SPC algorithm – the change in explanation follows the change in concept. Windows generated by the vertical lines give us insight in local explanations of the model (where the concept is deemed to be constant). Disjunct concepts (2 and 3) and redundant feature values are all explained correctly (e.g. reduncacy of "shape" and disjunction of "size" values in concept 3). Figure 1 demonstrates how classifications of two instances with same feature values can be explained completely differently at different times – adapting to change is crucial in incremental setting. This is also evident in the aggregated visualization (Figure 3), which can be used to quickly determine the importance of each attribute.

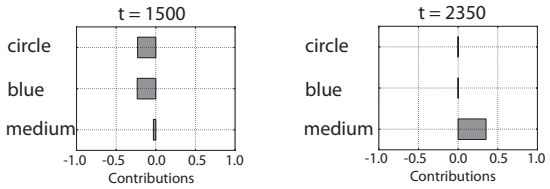

Figure 1: Explanations of individual predictions.

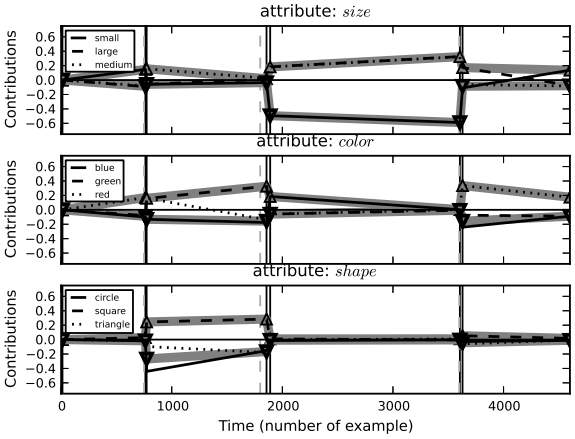

Figure 2: Visualization of explanations triggered at change detection (STAGGER).

For SEA dataset, feature values were obtained by equidistant discretization in 10 intervals for each feature. Since the blocks containing each concept are rather long and all concept drifts were successfully detected by SPC, the explanations obtained by change detection and those that were triggered periodically, do not differ greatly (we use $max\_length = 4000, \omega = 5000$). Explanations of individual instances are tightly corresponding to explanations of the model.

As can be seen in figure 4, the shape of contributions of features x_2 and x_3 reflects the target concept $x_2 + x_3 \leq \beta$; lower values increase the likelihood of positive classification and vice versa. Feature x_1 is correctly explained as

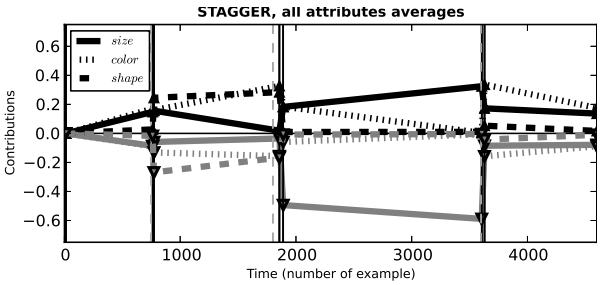

Figure 3: Aggregated visualization of explanations triggered at change detection (STAGGER).

irrelevant with its only contributions being the result of noise. The succession of target concepts $\beta \in \{8, 9, 7, 9.5\}$ is even more recognizable in the aggregated visualization (Figure 5). Changes in data are not as significant as those in STAGGER dataset, although the drift can still be observed, the dip around $t = 40000$ being a notable example (the concept drifts from $\beta = 7$ to $\beta = 9.5$).

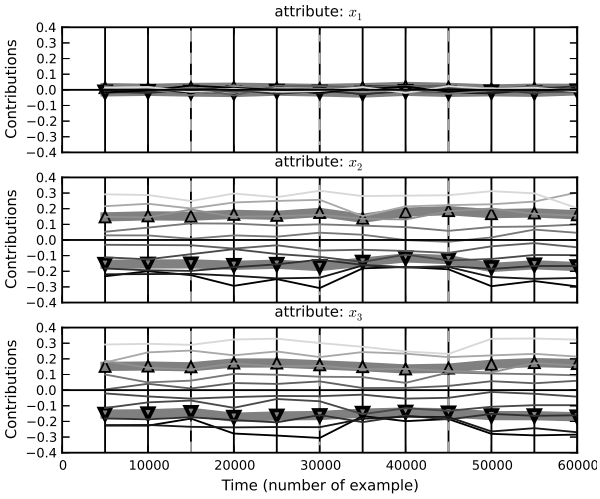

Figure 4: Visualization of periodic explanations (SEA).

5.3 Concept drift detection

Evaluating the concept drift detection using the stream of explanations on the STAGGER dataset yielded positive results. As depicted in Figure 6, the method correctly detects concept drifts without false alarms and is in that regard similar to SPC method. The stream of explanations itself fits patterns seen in testing with other successful drift detection methods (figure 2). Choices of larger granulations yielded similar results, but the change detection was obviously delayed. The concept drift was however never missed, provided that the granulation was smaller that the spacing between sequential changes in data. The delays of concept drift detection are correlated with the magnitude of change, e.g. the last concept drift ($t = 3600$) was detected

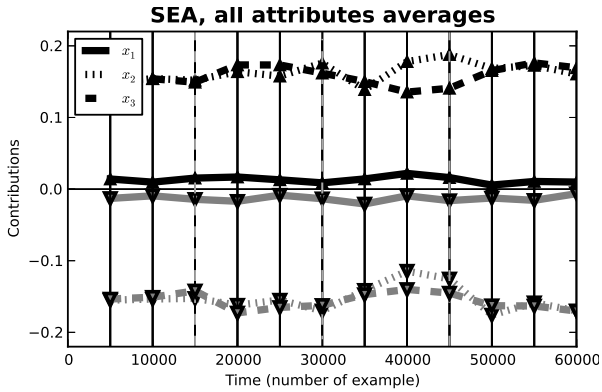

Figure 5: Aggregated visualization of periodic explanations (SEA).

with significant delay. In this regard the proposed method is inferior to SPC algorithm – the concept drift detection in noticeably delayed and we're also dependant on two parameters – granulation and alert threshold, so the generality of the method is diminished.

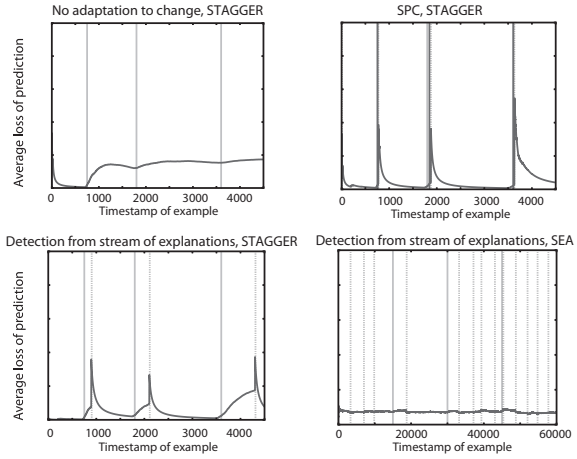

Figure 6: Loss of prediction and concept drift detection with several methods – no drift detection (static classifier), drift detection with SPC [2] and drift detection using the stream of explanations. The thick vertical lines indicate occurrence of true change in data and the dashed vertical lines mark drift detection.

When testing with SEA datasets, the concept drift was not correctly detected. Changing the granulation, Page Hinkley alert threshold and max_window parameter resulted in varying degrees of false alarms or non reaction to change (see Figure 6). This behaviour can be attributed to a small magnitude of change that occurs in data – the difference between concepts in data is quite small and continuous. However, when explaining this (incorrectly adapted) model we still recognise true underlying concepts due to the max_window property which causes the model to automatically decrement when it becomes too big[5].

We conclude that, in this form, the presented method is not a viable alternative to the existing concept drift detection methods. Its downsides include high level of parametrization (maximum size of the model, granularity and alert threshold level) which require a significant amount of prior knowledge and can also become improper if the model changes drastically. Consequently, another assessment of data is needed – the required manual supervision and lack of adaptability in this regard can be very costly and against the requirements of a good incremental model.

The concept drift detection is also not satisfactory – it is delayed in the best case or concepts can be missed or falsely alerted in the worst case. Another downside is the time complexity – the higher the granularity the more frequent explanations will be, which will provide us with a good stream of explanations but be very costly time-wise. The method is therefore not feasible in environments where quick incremental operations are vital. However, if we can afford such delays, we get a granular stream of explanations which gives us insight into the model for roughly any given time.

A note at the end: we should always remember that we are explaining the models and not the concepts behind the model. Only if the model performs well, we can claim that our explanations truly reflect the data domain. This can be tricky in incremental learning, as at the time of a concept drift, the quality of the model deteriorates.

6 Conclusion

The new visualization of explanation of incremental model is indeed an improvement compared to the old one. The overriding nature of the old visualisation was replaced with an easy to understand timeline, while the general concepts (macro view) can still be read out from the shape of the lines. Micro view is also improved as we can determine contributions of attribute values for any given time.

The detection of concept drift using the stream of explanations did not prove to be suitable for general use based on the initial experiments. It has shown to be hindered by delayed detection times, missed concept drift occurrences, false alarms, high level of parametrization and potential high time complexity. This provides motivation for further experiments in this field, especially because the stream of explanations provides good insight into the model with accordance to the chosen granulation.

The main goal of future research is finding a true adaptation of *IME* explanation methodology to incremental setting, i.e. efficient incremental updates of explanation at the arrival of each new example. Truly incremental explanation methodology would provide us with a stream of explanations of finest granularity. In addition to this result (for each timestamp we get an accurate explanation of the

[5]Considering prior domain knowledge.

current model), a number of new possibilities for visualisation would emerge, particularly those that rely on finely granular data, such as ThemeRiver [6].

References

[1] A. Bifet, G. Holmes, R. Kirkby, B. Pfahringer, and Braun M. Moa: Massive online analysis.

[2] J. Demšar. Explanation of predictive models and individual predictions in incremental learning (in slovene). B.S. Thesis, Univerza v Ljubljani, 2012.

[3] S. Few. *Now You See It: Simple Visualization Techniques for Quantitative Analysis*. Analytics Press, USA, 1st edition, 2009.

[4] J. Gama. *Knowledge Discovery from Data Streams*. Chapman & Hall/CRC, 1st edition, 2010.

[5] D. Haussler. Overview of the probably approximately correct (pac) learning framework, 1995.

[6] S. Havre, B. Hetzler, and L. Nowell. Themeriver: Visualizing theme changes over time. In *Proc. IEEE Symposium on Information Visualization*, pages 115–123, 2000.

[7] L. Nowell, E. Hetzler, and T. Tanasse. Change blindness in information visualization: a case study. In *Information Visualization, 2001. INFOVIS 2001. IEEE Symposium on*, pages 15–22, 2001.

[8] E. S. Page. Continuous inspection schemes. *Biometrika*, 41(1):100–115, 1954.

[9] Gunopulos D. Keogh E. Vlachos M. Das G. Ratanamahatana C.A., Lin J. Mining time series data. In *Data Mining and Knowledge Discovery Handbook*, pages 1049–1077. Springer US, 2010.

[10] R. Sebastião and J. Gama. A study on change detection methods. In *Progress in Artificial Intelligence, 14th Portuguese Conference on Artificial Intelligence, EPIA 2009, Aveiro, Portugal, October 12-15, 2009. Proceedings*, pages 353–264. Springer, 2009.

[11] W. Nick Street and Y. Kim. A streaming ensemble algorithm (sea) for large-scale classification. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '01, pages 377–382, New York, NY, USA, 2001. ACM.

[12] E. Štrumbelj and I. Kononenko. An efficient explanation of individual classifications using game theory. *The Journal of Machine Learning Research*, 11:1–18, 2010.

[13] E. Štrumbelj, I. Kononenko, and M. Robnik Šikonja. Explaining instance classifications with interactions of subsets of feature values. *Data Knowl. Eng.*, 68(10):886–904, October 2009.