Entropy-based Concept Shift Detection

Peter Vorburger, Abraham Bernstein
University of Zurich
Department of Informatics
Binzmühlestrasse 14, 8050 Zurich, Switzerland
{vorburger, bernstein}@ifi.unizh.ch

Abstract

When monitoring sensory data (e.g., from a wearable device) the context oftentimes changes abruptly: people move from one situation (e.g., working quietly in their office) to another (e.g., being interrupted by one's manager). These context changes can be treated like concept shifts, since the underlying data generator (the concept) changes while moving from one context situation to another. We present an entropy based measure for data streams that is suitable to detect concept shifts in a reliable, noise-resistant, fast, and computationally efficient way. We assess the entropy measure under different concept shift conditions. To support our claims we illustrate the concept shift behavior of the stream entropy. We also present a simple algorithm control approach to show how useful and reliable the information obtained by the entropy measure is compared to a ensemble learner as well as an experimentally inferred upper limit. Our analysis is based on three large synthetic data sets representing real, virtual, and a combination of both concept drifts under different noise conditions (up to 50%). Last but not least, we demonstrate the usefulness of the entropy based measure context switch indication in a real world application in the context-awareness/wearable computing domain.

1 Introduction

In real-world applications the mining of data streams, rather than time independent data, is increasingly important. In many applications data (e.g., from the financial industry, sensor data, multimedia content) is gathered over time, which raises the problem that the concepts to be learned may drift (i.e., change) over time [5]. Also, the increasing amount of data (e.g., multimedia content, data warehouses) and the limitation of computing power due to miniaturization (e.g., wearable computing) call for faster and more resource friendly algorithms. The motivation for this paper is a real-world problem which stands exemplary for the problems mentioned above – the analysis of sensor data on wearable devices. In our research on context-awareness [1], where we learned classifiers predicting peoples' anticipated behavior based on sensory input, we found that contexts (or contextual situations) *switch* rather than gradually change. We also found, that contextual information could be reused, even for new, not yet encountered situations. Therefore, an ongoing monitoring of the sensor stream is needed. An online pattern matching mechanism comparing the sensor stream to the entire library of already known contexts is, however, computational complex and not yet suitable for today's wearable devices. One solution is to indicate possible candidates (or hot spots) for context changes limiting the computationally intensive context (re-)determination on those candidates. Thus, a computationally "cheap" technique to find such context-switch candidates would be very helpful. From the machine learning point of view the context generating the sensor data can be viewed as the underlying concept generating the data stream and the context switches can be viewed as "abrupt concept drifts" also referred to as *concept shifts*. This paper introduces an *entropy-based measure to detect concept shifts*. In the following we will show that this measure is very sensitive to concept shifts while remaining noise-tolerant. Additionally, it allows to distinguish between different shift intensities. In order to be able to assess this measure, we introduce a *coarse concept shift adapting algorithm*, which we show to (1) provide mostly a better prediction quality than conventional approaches, (2) require limited computational power, (3) exhibit quick reaction time, and (4) show good performance under noisy conditions. After the assessment of the algorithm on synthetic data sets we apply our approach to sensor data obtained by a context-aware wearable computing setup [1], where the entropy measure clearly indicates context switches on the basis of audio and accelerometer recordings.

The next section provides a short review on the related work relates our contributions to other projects in the field.

Proceedings of the Sixth International Conference on Data Mining (ICDM'06)
0-7695-2701-9/06 $20.00 © 2006 IEEE

Section 3 introduces our novel concept shift measure and algorithm. To evaluate our proposed measure and algorithm, section 4 presents the experimental setup, synthetic data sets, and benchmarks including an (experimental) upper limit for the learning algorithms used in this study. The following sections present/discuss the results and are followed by a presentation of our approach's performance on the real-world data set. We close with the limitations, future work and a final conclusion section.

2 Related Work

In his survey paper Tsymbal defines concept drifts as follows: *"In the real world concepts are often not stable but change with time. ... Often these changes make the model built on old data inconsistent with the new data, and regular updating of the model is necessary. This problem is known as concept drift,..."* [5]. Obviously, drifts can occur suddenly (abruptly, instantaneously) or gradually. Since this paper is motivated by the problem of indicating switching contexts from sensor data, it focuses solely on sudden concept drifts, which we call concept shifts[1]. Widmer and Kubat [8] differentiate between changes in the actual target concept called *real concept drifts* and changes in the distribution called *virtual concept drifts*. Our work distinguishes itself from previous studies in the following ways. First, we introduce novel synthetic data sets based on the idea of a rotating hyperplane [7]. This setup allows us to investigate real and virtual drifts independent of each other, which supports a comprehensive assessment of concept drift approaches. Second, we are the first to benchmark our approach to an experimentally determined upper limit. Third, our main contribution is the introduction of an entropy-based measure as concept shift indicator, which is able to quantify the intensity of the shift. To be able to assess the power of the measure, we introduce a simple window-based algorithm using the entropy measure. This algorithm shows it's strength compared to ensemble classifiers both with regards to quality and computational performance on the synthetic data set. Last but not least, we show the usefulness of the measure in the context of a real-world wearable computing data set.

3 Entropy and Concept Shift Adaption

In this section we motivate and introduce the entropy measure applied on data streams. Our approach bases on the following assumptions: 1) As long as the distribution of older instances (features and target values) is similar to the distribution of new instances no concept drift occurred. 2) A distribution difference between older and more recent instances indicates a change in the target concept. Hence, the current model may be outdated and needs to be adjusted. To measure the distribution inequality we make use of the entropy to compare old and new instances of a data stream. If two distributions are equal, the entropy measure results in a value of 1, if they are absolutely different the measure will result in an value of 0. Although entropy is well known from information theory as a measure for information content - and its application, thus, is self-evident - we make use of it mainly because of its symmetry and additive properties. This section first specifies how to tailor the entropy measure for data-streams and and we introduce a simple coarse instance-selection algorithm, which allows us the evaluation of the measure in the next sections.

[1] We have strong indication that our approach also holds for gradual drifts, but such an investigation goes beyond the scope of this paper.

3.1 Calculating Entropy on Data Streams

To use Shannon's entropy in the context of data streams we have to adapt it. To that end we chose the sliding window technique, which compares two windows, one representing older and the other representing more recent instances in the stream. Essentially, we compare the two windows by counting and comparing all instances with respect to their class and stream membership. Additionally, we discretize the range of instance values to a fixed number of bins to take the approximate value distribution into account.

We define a data stream as a sequence consisting of sequentially ordered tuples $\vec{d_i}$ in time t_i, where $i \in (1, 2, 3, ...)$. Each tuple $\vec{d_i}$ consists of S feature streams s and one label stream l, formally $\vec{d_i} := (\vec{s_i}, l_i)$, where $\vec{s_i}$ is the vector of all feature stream instances s_{n_i} at time t_i. The domain of the label stream l is discrete and contains all class values $c \in C$. In the following evaluation on the synthetic data sets, for example, we will limit all experiments to 2 class problems. Let H_i be the resulting entropy at time t_i. H_i is defined as the mean of all data stream entropies H_{is} at time t_i

$$H_i = \frac{1}{S}\sum_{s=1}^{S} H_{is} \quad \text{, where} \quad H_{is} = \sum_{c=1}^{C}\sum_{b=1}^{B} H_{iscb}\ .$$

S is is the number of feature-streams and H_{is} is calculated from the entropies H_{iscb}, that represent the entropy of each class ($c \in C$) and bin ($b \in B$) given the stream s at time t_i. We introduced the bins as discrete aggregation of the values of each feature stream s. To simplify the presentations we will use 2 bins for all calculations. H_{iscb} is generated by calculating the entropy of the two sliding windows

$$H_{iscb} = -\underbrace{w_{iscb}}_{\text{weight}}[\underbrace{\varrho_{iscb_{old}}\log_2(\varrho_{iscb_{old}})}_{H_{iscb}[\text{"old window"}]} + \underbrace{\varrho_{iscb_{new}}\log_2(\varrho_{iscb_{new}})}_{H_{iscb}[\text{"new window"}]}]$$

Proceedings of the Sixth International Conference on Data Mining (ICDM'06)
0-7695-2701-9/06 $20.00 © 2006 IEEE

where $\varrho_{iscb_{old}}$ is the probability that an instance occurs in the old window at time t_i, belonging to class c, with the feature domain of stream s in bin b. Obviously, $\varrho_{iscb_{new}}$ is calculated analogously. The weight w_{iscb} can depend on i, s, c, and b. The border condition $\sum_{s=1}^{S}\sum_{c=1}^{C}\sum_{b=1}^{B} w_{iscb} \stackrel{!}{=} 1$, $\forall i$ must be fulfilled in order to keep the entropy in the range [0,1]. To simplify the calculations we will choose $w_{iscb} = 1$ for $\forall i, s, c, b$ and keep both sliding windows of all streams at the same length. For a more detailed explanation see [6].

3.2 Algorithm Control Strategy using Entropy Measure

This subsection focuses on developing a simple, coarse algorithm that automatically adapts to concept shifts based on the entropy measure. This allows us to benchmark our algorithm to other approaches in terms of prediction power. Thus, we can draw conclusions from these comparisons for our entropy-based measure as concept shift indicator. Our approach is an instance selection style algorithm that adapts the window size whenever the entropy measure detects a shift. The window size control strategy is based on the very simple rule depicted in Figure 1. Let us assume that we start before a shift and the entropy measure value is at (or near) 1 and the window of the algorithm is of some given size ξ. When a shift occurs the entropy measure reacts. If it intersects an arbitrary chosen threshold τ we collapse the window size ξ of the algorithm to a minimal size and let it grow again by the newly arriving instances to an upper threshold, resulting in a linear recovery of the window size after the drift. Thus, every time the entropy intersects with the threshold (with a negative slope) the algorithm "forgets" its current model and starts to relearn on the most recent instances. In the remainder of this study we have chosen

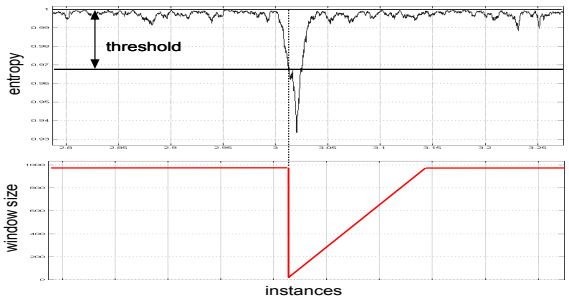

Figure 1. Illustration of the algorithm control strategy.

a fixed threshold $\tau = 0.95$ and set the lower bound window size $\xi_{\text{lower\_bound}}$ to 20 and the upper bound window size $\xi_{\text{upper\_bound}}$ to 1000 instances (the only reason to introduce an upper bound was to allow a fair comparison with the benchmark algorithms presented in the next section, which have a maximum window size of 1000).

4 Experimental setup

For a comprehensive analysis of concept drift algorithms the first requirement to a benchmark data set is that it needs to differentiate between virtual and real drifts. Furthermore, we need to ensure that the data sets don't contain any artifacts from their generation such as asymmetrical features or other hidden dependencies. As we did not find any benchmark data set in the literature conforming to these requirements we adapted the method of [7] to generate our own **synthetic data set**. Our data set domain consists of a sphere containing all instances and a plane intersecting this sphere through its origin. The orientation Θ of the plane is defined by a three dimensional vector $\vec{n}$ standing perpendicular of the plane's surface. Instances above the plane belong to class A and instances below the plane belong to class B. Hence, this mechanism defines a two class problem. We obtain the overall data set by combining three random and independent data streams with a fourth data stream generated by the rule above. So, we created a *"real drift"* data set by rotating the plane, a *"virtual drift"* data set by leaving the plane untouched and altering the class distribution ψ_A, and a *"mixed"* data set by overlaying the two data sets. To be able to assess all algorithms under noisy conditions we added 0%, 1%, 2%, 5%, 10%, 20%, and 50% of *noise* to all of the data sets by switching the lables at random. For a more detailed explanation see [6].

As a **performance measure** we chose both accuracy and the area under the ROC-curve. We used the *accuracy* as quality measure for classifier predictions because all the related literature makes use of it. Throughout this evaluation we have chosen to use a batch version of the *Naïve Bayes* ***algorithm*** as it is known for its robustness, does not require much computational power. As using a sliding window technique, we induced the model not on all instances available at time t_i but on a window w of size ξ. Thus, the window used was $w_{\xi,i} = [d_{i-\xi}, d_{i-\xi+1}, ..., d_i]$. In section 3.2 we presented a general rule to adapt an algorithm based on the outcome of the entropy measure.

To compare our solution against two accepted standards we calculated a representative set of **benchmarks** on the three data sets presented above. First, a so-called *perfect benchmark*, which assumes an oracle-given ideal window-size ξ for any point in time, and second, a selection of *ensemble classifiers* (based on 9 members), which the literature [3] so far showed to have the highest accuracy and robustness against noise. We limit the training set of all classifiers to a maximal window size of 1000 to keep the range in the order of magnitude of a single concept length as used in the synthetic data sets. For a more details see [6].

Proceedings of the Sixth International Conference on Data Mining (ICDM'06)
0-7695-2701-9/06 $20.00 © 2006 IEEE

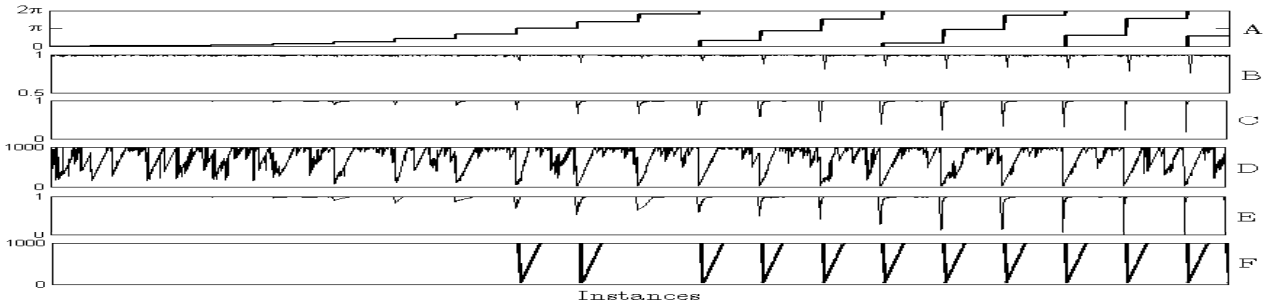

Figure 2. Overview on the real concept shift parameter (orientation angle Θ of $\vec{n}$).

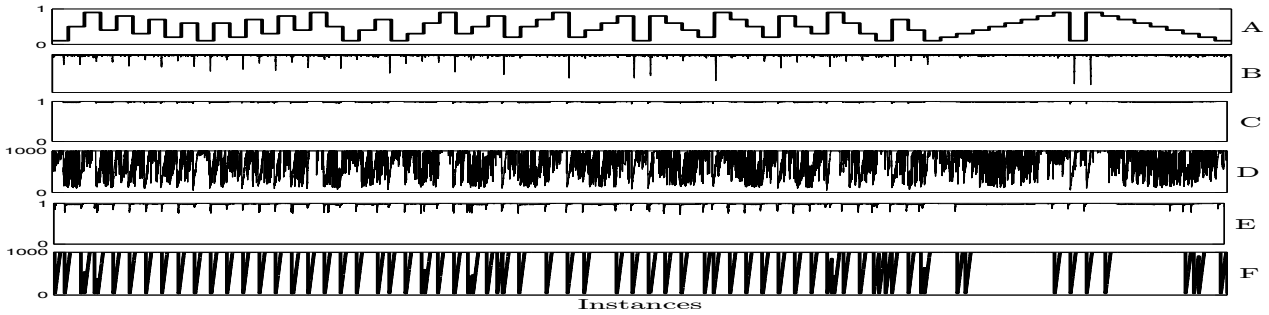

Figure 3. Overview on the virtual concept shift parameter (class distribution ψ_A).

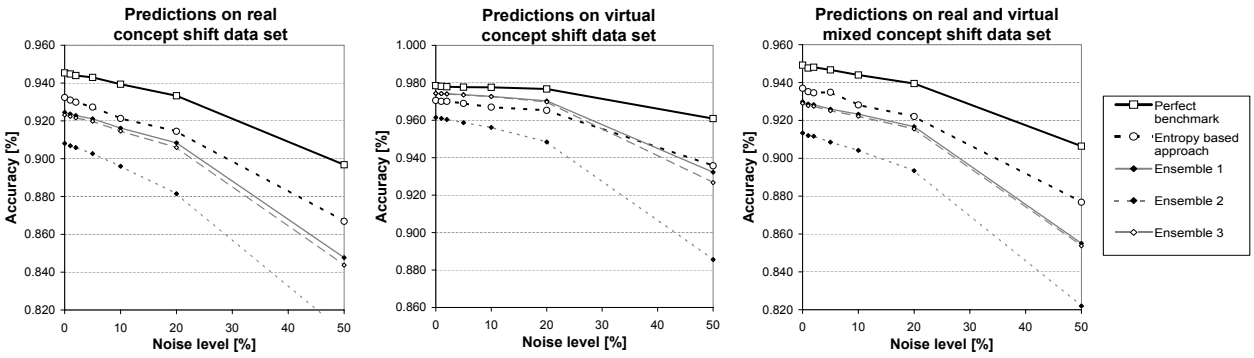

Figure 4. The results for the entropy based and all benchmark algorithm for the real, virtual, and the real and virtual mixed concept drift data sets.

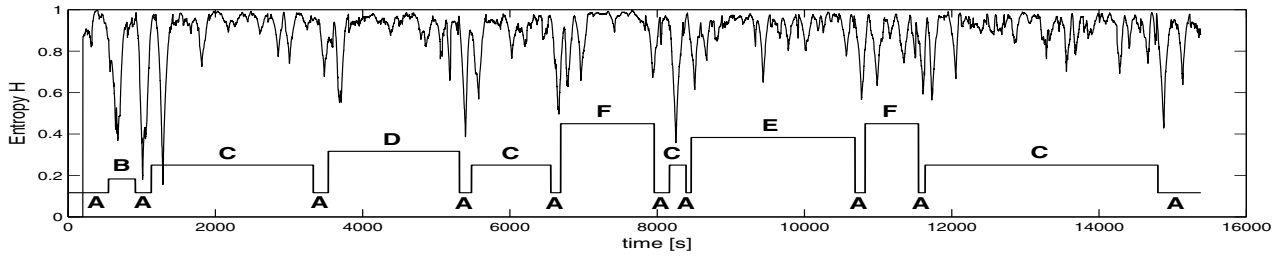

Figure 5. The upper line represents the entropy based measure applied on the real-world dataset. The lower line represents the sequence of different context situations.

5 Experimental Results

Figures 2 and 3 provide an *overview of the results* and an intuitive understanding of the entropy measure. These two figures are composed the same way. The curve (A) at the top represents the concept drift parameter. For real shifts this is the $\vec{n}$ vector orientation angle Θ. For virtual shifts the concept is represented by the class distribution ψ_A. We do not present the curves on the mixed data set because their behavior is consistent with curves presented on the real and the virtual shifts and thus, these curves do not provide any new information. Curve (B) is the derived entropy H. The curves (C) and (D) represent the perfect benchmark. (C) shows the highest reachable accuracy by a window based forgetting Naïve Bayes algorithm. (D) shows the corresponding window size in order to reach that prediction above. The two last curves (E) and (F) represent the accuracy and the window size of our entropy measure based algorithm. All curves are calculated under noise-absent conditions. The entropy based measure (B) reflects the concepts shifts (A) on both data sets and each amplitude corresponds to the shift intensity. The accuracy of the perfect benchmark (C) is a little bit biased by the prior class distribution as it follows the top line (A) in Figure 3. The window size behavior of the two algorithms ((D) and (F)) is very similar on real shift data except on the very small shifts that are overlooked by the entropy based algorithm. The perfect benchmark behavior on the virtual data is totally different. It shows vehement window size variations - even in non-drifting sequences. The entropy based approach is, again, synchronous to the virtual shift - except for the very small ones. Figure 4 shows the *prediction quality* on the three data sets *against increasing noise levels*. All three graphs show that our algorithm (dashed line) is as noise resistant as the benchmarks. The graphs shows also that our algorithm outperforms the ensemble classifiers - except for the virtual shift dataset. The fact that our algorithm cannot keep up with two of the ensemble algorithms for this setting indicates that our simple coarse approach is insufficient for this situation, although the entropy indicates almost every virtual concept shift (Fig. 3). The graphs on the right show the results on the mixed data set. These results are sound regarding the results of the real and the virtual shift results. Finally, we conducted experiments concerning *computational complexity*. We compared the ensemble classifiers and the entropy measure based algorithm. We first measured the elapsed time for all three committee classifiers for both of the quality measures. The elapsed time was about the same such we decided to report it as mean and standard deviation. The computation[2] of 10000 tuples taken from the synthetic data set required 2031.6 ± 15s. The entropy based algorithm required 148.6s, which is 13.7 times faster than the committee. The entropy calculation without following Naïve Bayes model building requires only 1.1 ± 0.1s. This indicates that the performance difference originates from the number and size of the used Naïve Bayes models. This emphasizes the computational advantage of our approach as expected.

[2] Using Matlab on a 3 GHz Pentium 4 machine with 1 GByte RAM.

6 Discussion of the Experiment

It is remarkable that the simple coarse algorithm based on the entropy measure outperforms the ensemble benchmark algorithm on real concept shifts. This confirms that the entropy measure is a very good indicator for detecting and controlling an algorithm adapting to real concept shifts. Also, our algorithm is one order of magnitude faster than the ensemble approach, because our approach calculates the Naïve Bayes algorithm only once, whereas the ensemble requires a Naïve Bayes calculation for each of its members. Hence, *our algorithm exhibits a greater predictive power while requiring less computational resources*. Note that the calculation of the entropy measure only accounted for less than 1% of the computational requirement of our algorithm. Furthermore, the *entropy measure based algorithm showed the nearly the same robustness towards noise as the perfect benchmark and the committee classifiers*. To reach this goal we invested the domain knowledge that the structure of the examined drifts is abrupt; i.e., that the domain exhibited concept shifts rather than concept drifts. But this assumptions holds for our initial real-world problem as we will show in the next section.

7 Application to a Real-World Problem: Context Switches in Sensor Data

As mentioned in the introduction section, the original motivation for the entropy based measure was the monitoring of sensor data streams for context switches. To demonstrate that functionality we use the exact same data set as presented in a prior study [1]. The data set consists of audio and accelerometer data recorded over a time of 15381 seconds. The wearable data acquisition setup included a microphone and three three-dimensional accelerometers attached on the subject's shoulder, wrist, and leg. To illustrate the applicability of the measure we focus on the audio stream and one single accelerometer (leg, would correspond to a mobile device's accelerometer carried in a pocket). The data was preprocessed in a very simple and fast way as it could be performed e.g. on a smart phone resulting in one feature vector for each second. The audio signal was decomposed into 10 features[3]. The accelerometer data was

[3] Spectral center of gravity, temporal fluctuations of spectral center of gravity, tonality, mean amplitude onsets, common onsets across frequency bands, histogram width, variance, mean level fluctuations strength, zero crossing rate, and total power.

merged in one single feature: the absolute value of the amplitude. To calculate the entropy based measure we applied the exact same parameters as used in the evaluation before. As input stream we have chosen the audio features and as target class we picked the accelerometer feature which has been discretized to represent a two class problem (large and small acceleration). We had 2 bins for each input stream, and chose a window size of 100 instances (=100 seconds)[4]. The upper line in Figure 5 shows the entropy H calculation on the sensor data and the lower line illustrates the subject's actual context sequence (scenario). The scenario consists of 6 context situations: (A) walking, (B) streetcar, (C) office work, (D) lecture, (E) cafeteria, and (F) meeting. The large peaks in the entropy measure look synchronous to the concept shifts. Based on this observation we can construct an algorithm that indicates a context switch every time the entropy crosses a given threshold (analogously to the algorithm introduced to adapt to the shifts, see Section 3.2). If we arbitrarily choose the threshold to be, e.g., 0.7 the algorithm would indicate 17 of total 18 context switches and six times cause a "false alarm". The one context switch at 8462 seconds is not detected because its signal overlaps with the signal of the context switch just before at 8391 seconds. Raising the threshold even further will result in increasingly fine-grained indications of context switches – not only concept switches between "walking" and other context situations. Additionally, the intensity of the peaks indicates the magnitude of the context switch. Hence, one can derive some degree of similarity between the context situations, which might be used to control the granularity of the segmentation.

8 Limitations, Future Work, and Conclusion

This projects focused on concept shifts. We are, therefore, planning to investigate more sophisticated control strategies for *gradual concept drifts* in future work. The chosen window size of our algorithm was experimentally chosen to cope with the signal-to-noise ratio. Alternatively, one could try to find boundary conditions such as lower and upper bounds for the window size as presented in [2] and [4]. Some algorithms *recognize recurring concepts and exploit this information* [8]. While this has not been the focus of this project, any algorithm based on our entropy measure could be enhanced by comparing stored models with new data as soon as the entropy indicates the appearance of a new concept. In this paper we illustrated our technique on a 2-class problem, but it is *generalizable to n-class problems*, since the entropy formula and the classifiers generalize accordingly. Last but not least, we provided a real-world example to show the usefulness of this approach. In future we would like to investigate the generalizability both to other subjects and different applications. Also the choice of the suitable parameters could be optimized.

In this paper we set out to find a measure for detecting and measuring concept shifts as an analogon for context switches. Our experimental findings show that the formulation of entropy on data streams presented in section 3 is indeed capable to detect and measure concept shifts. A simple and coarse algorithm with an entropy based instance selection strategy outperformed ensemble based algorithms on real concept shift data sets. Given our algorithms robustness towards noise, its sensitivity towards concept shifts, its computational efficiency, and predictive power on real concept shift data sets it addresses two central trade offs of current data streams mining approaches: predictive power versus computational complexity and noise versus sensitivity. As such we believe that our entropy based measure is a very promising basis to gain further insight into the problem of concept shifts, ultimately resulting in better induction algorithms for this increasingly important application domain.

We would like to thank Iwan Stierli and Martin Constam for their substantial support in the initial stage of this project. We also like to thank Haym Hirsh and Patrice Egger.

[4]We are well aware that there is a huge potential of improving our results by fine tuning the parameter settings, but we only want to show that satisfying results can be achieved - even with the most simple settings.

References

[1] A. Bernstein and P. Vorburger. A scenario-based approach for direct interruptablity prediction on wearable devices. *Journal of Pervasive Computing and Communications*, 2006.

[2] D. P. Helmbold and P. M. Long. Tracking drifting concepts by minimizing disagreements. *Mach. Learn.*, 14:27–45, 1994.

[3] J. Z. Kolter and M. A. Maloof. Dynamic weighted majority: A new ensemble method for tracking concept drift. In *ICDM*, pages 123–130, 2003.

[4] A. Kuh, T. Petsche, and R. L. Rivest. Learning time-varying concepts. In *NIPS-3: Proceedings of the 1990 conference on Advances in neural information processing systems 3*, pages 183–189. Morgan Kaufmann Publishers Inc., 1990.

[5] A. Tsymbal. The problem of concept drift: definitions and related work. 2004.

[6] P. Vorburger and A. Bernstein. Entropy-based detection of real and virtual concept shifts. *Working Paper - University of Zurich, Department of Informatics*, 2006.

[7] H. Wang, W. Fan, P. S. Yu, and J. Han. Mining concept-drifting data streams using ensemble classifiers. In *KDD '03: Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 226–235. ACM Press, 2003.

[8] G. Widmer and M. Kubat. Effective learning in dynamic environments by explicit context tracking. In *Machine Learning: ECML-93 - Proc. of the European Conference on Machine Learning*, pages 227–243. Springer, 1993.