

# Entropy-based, Semi-dynamic Regulation of Incremental Algorithms in the case of Instantaneous Concept Drifts

---

Diploma Thesis in Business Information Technology

submitted by Iwan Stierli  
Zurich, Switzerland

student number 98-915-192

written at the Department of Informatics  
University of Zurich  
Prof. Abraham Bernstein, Ph.D.

supervised by Peter Vorburger

submitted on 15-02-2005

## **Abstract**

Incremental classifiers build their prediction rules according to the known instances of a continuous data stream. As these algorithms learn from correct and incorrect predictions, their performance improves the more instances their rules are based on. In the case of an instantaneous concept drift, this assumption is no longer valid as the old concepts' instances falsify the rules which are to be built. Therefore, it would be ideal to forget the instances. In this thesis, it is tried to regulate this forgetting rate accurately by using an adapted form of the entropy term. First, a simple, linear correlation between the entropy and the forgetting rate will be excluded. Furthermore, a second, semi-dynamic and noise-resistant switching strategy will be pursued. It will be tested on a synthetic data set and compared with the applicable benchmarks according to two different quality measures.

## **Zusammenfassung**

Inkrementelle Klassifizierer bilden ihre Vorhersageregeln anhand den ihnen bereits bekannten Instanzen eines kontinuierlichen Datenstromes. Da diese Algorithmen aus richtigen und falschen Voraussagen lernen, werden sie je besser, desto mehr Instanzen sie betrachten. Im Falle einer plötzlichen Veränderung der Voraussetzungen ändert sich diese Regel, Instanzen des alten Datenmodells sollten jetzt vergessen werden, da sie falsche Informationen hinsichtlich des neuen Modells liefern. In dieser Diplomarbeit wird versucht, die Vergessensrate mittels des Informationsgehaltes des Datenstromes möglichst genau zu regeln. Als erstes wird eine einfache, lineare Korrelation zwischen Informationsgehalt und Vergessensrate ausgeschlossen. Dann wird eine zweite, quasi-dynamische und noise-resistente Strategie verfolgt. Diese wird mittels zwei verschiedenen Qualitätsmassen auf einem künstlichen Datensatz getestet und gegen verschiedene Benchmarks verglichen.

## Thanks

First of all, I would like to thank my ravishingly beautiful girlfriend b. for her lovely moral and linguistic support. Then I would like to thank my tutor and man in black (👤!) Peter Vorburger who worked with me on this thesis. This diploma thesis is based on his idea and preparatory work. Last but not least, I would like to thank all of my friends which had to suffer from my sulkiness during the last six months. The special ingratitude-price goes to the Matlab Distributed Toolbox which cost me too much time and too much nerves.

Hail Eris! All hail Discordia!

# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Motivation . . . . .	1
1.2	Annotation . . . . .	2
<b>2</b>	<b>Data Sets</b>	<b>3</b>
2.1	Structure Of The Data Set . . . . .	3
2.2	Continuous Versus Discrete Parameter Values . . . . .	5
<b>3</b>	<b>Concept Drifts</b>	<b>6</b>
3.1	Virtual Drifts . . . . .	6
3.2	Real Drifts . . . . .	9
<b>4</b>	<b>Algorithms</b>	<b>11</b>
4.1	Naïve Bayes . . . . .	11
4.2	K-Nearest Neighbour . . . . .	12
<b>5</b>	<b>Benchmarks</b>	<b>15</b>
5.1	Ultimate Benchmark . . . . .	17
5.1.1	Conclusion . . . . .	19
5.2	Optimal Window Sizes . . . . .	19
5.3	Monitoring of the Results . . . . .	24
5.4	Error caused by Smoothing the Curve . . . . .	25
5.5	Smoothed Benchmark . . . . .	26
5.6	Committee Benchmark . . . . .	27
<b>6</b>	<b>Entropy</b>	<b>29</b>
6.1	Basics . . . . .	29
6.2	Primal Form Of Entropy Term . . . . .	31
6.2.1	Breaking Down the Data Stream into Pieces . . . . .	31
6.2.2	Arranging of the Instances According to Label/Time . . . . .	31
6.2.3	Compilation of the Conditional Histograms . . . . .	32
6.2.4	Computing the Entropy . . . . .	33
6.3	Further Development of the Entropy Term . . . . .	35
6.3.1	Structural Weakness of Constam's Approach . . . . .	35
6.3.2	Avoiding the Need of Future Information . . . . .	35
6.4	Normalisation . . . . .	43
6.5	Entropy Limit Definitioin . . . . .	43

<b>7</b>	<b>Static Analysis</b>	<b>49</b>
7.1	Introduction . . . . .	49
7.2	Limitations . . . . .	52
7.2.1	Finding a More Significant Display Format . . . . .	52
7.2.2	Foreclosing the Need to Manually Edit the Entropy Curve . . . . .	52
7.2.3	Handling of Unequal Reactions on Concept Drifts . . . . .	53
<b>8</b>	<b>Semi-dynamic approach</b>	<b>57</b>
8.1	Motivation . . . . .	57
8.2	Implementing a Threshold . . . . .	57
8.3	First Form of Switching and Linear Learning . . . . .	58
8.4	Fixing of the Variables . . . . .	60
8.4.1	Fixing $\psi$ , Regulating $ws$ of $H$ and $\theta$ . . . . .	61
8.4.2	Fixing $ws$ of $H$ , Regulating $\psi$ and $\theta$ . . . . .	61
8.5	Second Form of Switching and Linear Learning . . . . .	61
8.6	Third Form of Switching and Linear Learning . . . . .	63
8.7	Performance of the Regulation . . . . .	64
8.7.1	Determining Entropy $H$ and Threshold $\theta$ . . . . .	65
8.8	Noise Resistance of the System . . . . .	66
8.9	Overfitting the Solution to the Problem? . . . . .	67
<b>9</b>	<b>Discussion</b>	<b>70</b>
9.1	Introduction . . . . .	70
9.2	Main Discussion . . . . .	71
9.3	Transferability to Other Scenarios . . . . .	73
<b>10</b>	<b>Future prospects</b>	<b>74</b>
<b>11</b>	<b>Conclusion</b>	<b>76</b>
<b>A</b>	<b>Appendix</b>	<b>78</b>
	List of Tables	80
	List of Figures	81
	Bibliography	83

# Chapter 1

## Introduction

### 1.1 Motivation

“One of the basic tasks of Machine Learning is to provide methods for deriving descriptions of abstract concepts from their positive and negative examples. So far, many powerful algorithms have been suggested for various types of data, background knowledge, description languages, and some special “complications” such as noise or incompleteness. Nevertheless, relatively little attention has been devoted to the influence of varying context. Daily experience shows that in the real world, the meaning of many concepts can heavily depend on some given context, such as season, weather, geographic coordinates, or simply the personality of the teacher. “Ideal family” or “affordable transportation” have different interpretations in poor countries than in the North, the meaning of “nice weather” varies with season and “appropriate dress” depends on time of the day, event, age, weather, and sex, among other things. So time-dependent changes in the context can include changes in the meaning or definition of the concepts to be learned. Such changes in concept meaning are sometimes called concept drift.” [Widmer 93].

This citation of Widmer/Kubat dates back to 1993 which is 13 years ago. Since then, the amount of the data which have to be processed increased exceedingly. There is hardly a business area which does not collect and store shoals of data. Basically, in recent years, Machine Learning progressed and a lot of development work has been made in the domain of Data Mining. But of all areas, in the one Widmer/Kubat have addressed, research is still in its infancy. The behaviour of incremental algorithms in case of concept drifts is neglected due to the fact that useful means are still not available. The main reason for this phenomenon may be the fact that all approaches made so far need a lot of computing power. Even modern processors are too slow to run the known algorithms real-time. Committee classifiers, decision trees or ensemble learners [Kuncheva 04] provide good classification results indeed, but the

required computing time therefor inhibits a real-time application.

In this thesis, based on the entropy-term known from the domain of thermodynamics and its meaning in modern information theory, a new approach is taken. It will be shown that the entropy is suitable to detect concept drifts in a reliable and noise-resistant way. The fundamental idea behind this approach is to regulate the incremental algorithm's forgetting rate by the information content of the data stream. As a consequence, the algorithm becomes adaptive and the real-time classification of a continuous data stream will be made possible.

## 1.2 Annotation

This thesis is not completely self-contained, it stands in a series of related works. Therefore, the introduction chapter of is kept short and only a quick overview over the fundamental ideas has been given. For further basic information, it is referred to the thesis of M. Constam, "Dynamische Regelung inkrementeller Algorithmen unter dem Einfluss von Concept Drifts"<sup>1</sup> [Constam 05]. In principle, this thesis is a continuation of Constam's work. In comparison therewith, the scope of this thesis is defined more accurately, while Constam's thesis is the more widespread. For example, this thesis only focuses on instantaneous concept drifts (the definition thereof can be found in chapter 3). Nevertheless, it is tried to appreciate preparatory work by keeping other forms of concept drifts in mind and by setting up the definitions pursuantly. E.g. the entropy-function is defined in order to detect also continuous drifts (details in chapter 6), even though this kind of drift is not dealt with in this thesis. Therefore, with this work covers a predefined<sup>2</sup> part of the entire domain. In this part, possible solution approaches will be developed, be rated against each other and, finally, their numerically proved power will be presented. As the final results are absolutely promising, in chapter 10, future prospects will be denoted as possible link for further research.

---

<sup>1</sup> In English: "Dynamic control of incremental algorithms influenced by concept drifts"

<sup>2</sup> Details are to be found in the further part of the thesis

## Chapter 2

# Data Sets

In this thesis, a continuous system of data which changes over time, will be looked at. All information about the past is known, but none about the future. A data stream which provides a new instance  $i$  at every single point in time  $t$  will be examined. The final aim is to detect the nature of instance  $i_t$  at point in time  $t$  based on the information obtained by  $i_{t-1}$ ,  $i_{t-2}$ ,  $\dots$ ,  $i_{t-n+1}$ ,  $i_{t-n}$  as precisely as possible by using an incremental algorithm. The following paragraphs accurately describe the general layout of the data stream and the exact nature of the single instances.

### 2.1 Structure Of The Data Set

The data set consists of a continuous stream of instances<sup>1</sup>. For every point in time  $t$  an instance  $i$  is presented to the algorithm. A single instance  $i$  is composed of three parameter-values,  $x$ ,  $y$ ,  $z$ , and a label-value<sup>2</sup>  $\in [1, 2]$ . The label-value itself is not known *a priori*, but has to be determined by the algorithm on the basis of the parameter-values. To avoid results which are influenced by the set up of the underlying data set, the latter is held as simple as possible. Furthermore, a simple set up of the data set reduces the influence of possible side effects.

For the set up of the data set, instances will be used which are simply defined by their location in space within a sphere. The sphere itself is split into two hemispheres by a plane. This splitting enables a partitioning of the instances into two classes. The specified design of the instances is constructed as follows: A sphere with its center in the origin and radius  $r$  contains accidentally distributed points. Every point is defined by its three Cartesian coordinates  $x$ ,  $y$

---

<sup>1</sup> See figure 2.1.

<sup>2</sup> In the further part of the thesis, “label” and “class” are used synonymously.



point in time		1	(...)	t-4	t-3	t-2	t-1	t	t+1	t+2	t+3	t+4	(...)	n
instance		1	(...)	i-4	i-3	i-2	i-1	i	i+1	i+2	i+3	i+4	(...)	n
parameterstream	x	-0.16	(...)	-0.33	0.43	-0.21	0.30	0.25	-0.17	-0.19	0.35	0.05	(...)	0.35
	y	-0.23	(...)	0.29	0.14	-0.39	-0.23	-0.10	0.41	0.31	0.21	-0.36	(...)	-0.17
	z	0.15	(...)	0.06	-0.40	-0.19	0.00	0.35	-0.11	0.42	-0.38	0.40	(...)	0.24
labelstream		2	(...)	2	1	2	1	1	2	2	1	1	(...)	1

Figure 2.1: Data stream with parameter- and label-values

and  $z \in [-0.5 \dots 0.5]$ . As already mentioned, the coordinates will be defined as the so called parameter values of the instance. Thus all parameter values of an instance  $i$  comply with

$$r_i \leq \sqrt{x_i^2 + y_i^2 + z_i^2} \leq r = 0.5 \quad (2.1)$$

according to the spherical equation

$$V_{sphere} = \frac{4\pi}{3}r^3. \quad (2.2)$$

The sphere itself is divided into two halves by a plane passing through its centre. This plane is always parallel to the z-axis and therefore satisfies the plane equation

$$\vec{s} = \vec{s}_0 + \lambda \vec{e}_z + \mu \vec{v} \quad \text{with} \quad \vec{s}_0 = \vec{0} = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \quad \text{and} \quad \vec{e}_z = \text{unit vector } z = \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix}. \quad (2.3)$$

Independent from the hade of the plane defined by  $\vec{v}$ , the two hemispheres are exactly commensurate. Every point, respectively every instance, is located in only one of the two partitions. Thus its label is defined, e.g. the instance belongs either to class 1 or to class 2. The mathematical expression for determining which class an instance belongs to can be found in section 3.2. Figure 3.4 and the corresponding formulas 3.3 and 3.4 clarify the relation. The above described data stream is depicted in figure 2.1.

## 2.2 Continuous Versus Discrete Parameter Values

Finally, two different types of parameter values are introduced, continuous and discrete ones. The former reach values of up to a accuracy equal to the calculation depth of the value-generating mathprogram, thus approximative infinite<sup>3</sup>. The latter are limited to a resolution of 0.1. In the predefined range of  $[-0.5 \dots 0.5]$ , only 10 possible values can occur. Extrapolated to the three dimensions of the parameter stream, overall, only  $\approx 10^3$  different combinations<sup>4</sup> are possible. The main purpose for this discretisation is to effect an economy of computing time. Compared with the continuous ones, the results based on the discrete data set do not show significant differences which will be demonstrated in the following part of the thesis.

---

<sup>3</sup> Purely mathematical considered certainly not infinite, but in practical approach this simplification is absolutely feasible

<sup>4</sup> Note that the approximation sign results from the fact that “extreme” combinations of (x,y,z) (like (0.5, 0.5, 0.5)) violate the constraint  $\sqrt{x_i^2 + y_i^2 + z_i^2} \leq 0.5$ .

## Chapter 3

# Concept Drifts

Generally, a concept drift is a mutation in the structure of the data set. It involves a changing target concept. Two different target concepts,  $A$  and  $B$ , are considered. A sequence of instances  $i_1$  to  $i_n$  is presented in order to the concept drift algorithm. Before some instance  $i_d$ , the target concept  $A$  is stable and does not change. After a number of instances  $\Delta x$  beyond  $i_d$ , the concept is once again stable, this time at concept  $B$ . Between instance  $i_d$  and  $i_d + \Delta x$  the concept is drifting between targets  $A$  and  $B$  according to a distribution. If  $\Delta x = 1$ , the concept shifts instantaneously between  $A$  and  $B$ . Unless otherwise noted, this kind of drift is meant when it is simply spoken about a concept drifts. When  $\Delta x > 1$ , the concept is changing over a number of instances. This is called a continuous drift and will not be dealt with in this thesis. In our data set, two different kinds of drifts can be distinguished from each other. This is described in the following two paragraphs.

### 3.1 Virtual Drifts

So called virtual drifts are based on a simple variation of the distribution of the instances. Primarily, the single classes are uniformly distributed, which means that there are exactly 50 percent of the instances in each class. In case of a virtual drift, this distribution changes, whereas the concept itself levels off. The prior, which indicates the larger ratio of both classes, is the characterising measure to specify such a drift. E.g. a prior  $p$  of 0.8 represents a distribution

$$p = 0.8 \quad \hat{=} \quad \frac{\text{number of instances in class 1}}{\text{number of instances in class 2}} = 80\% \text{ to } 20\%. \quad (3.1)$$

Figures 3.1 and 3.2 display such a virtual drift.

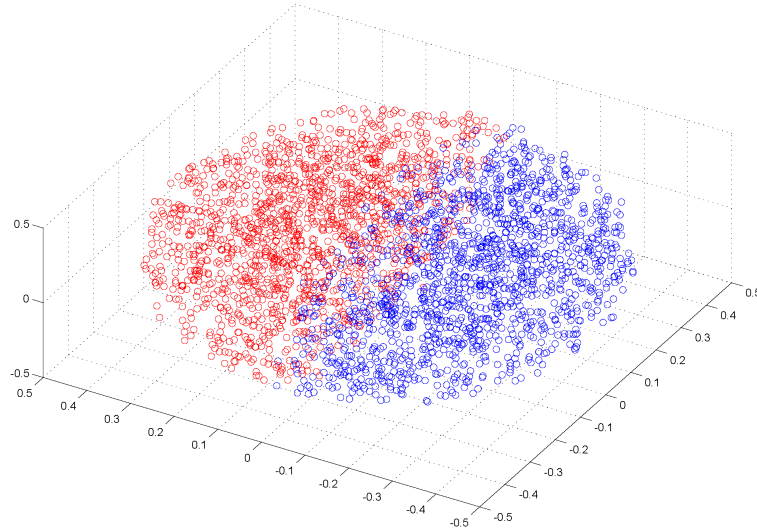


Figure 3.1: Uniformly distributed instances (graphic by P. Vorburger).

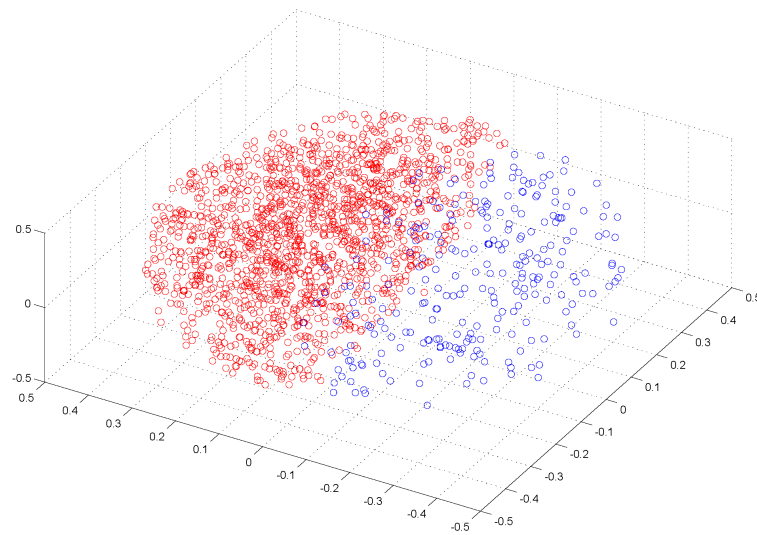


Figure 3.2: Distribution after prior has changed (graphic by P. Vorburger).

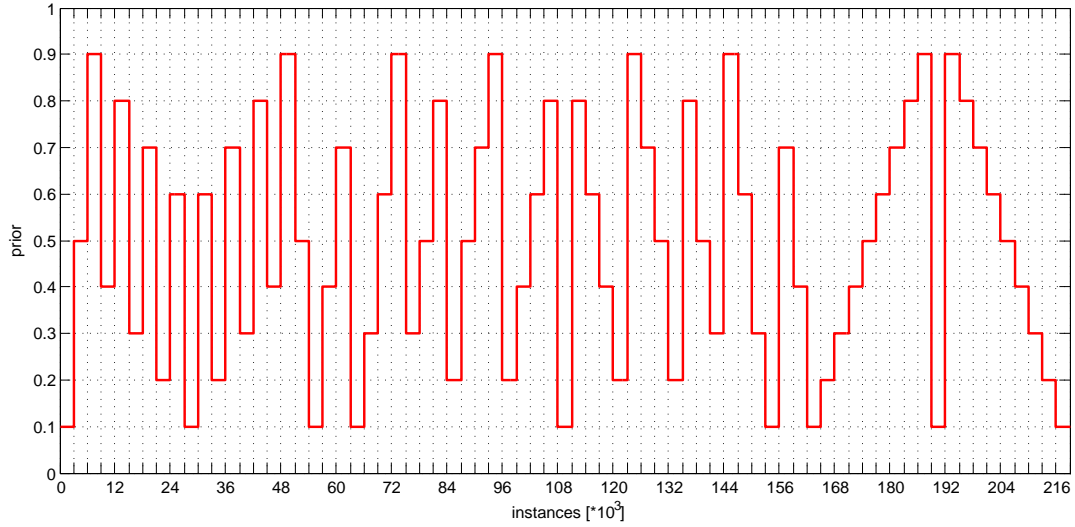


Figure 3.3: Prior distribution

The data set simulating virtual drifts consists in 73 concepts per 3000 instances. This number of instances between two drifts has been chosen to achieve a reasonable average between drifting and stable phases of the data set. Would the number be smaller, drifts would occur unrealistically often. Would the number be greater, the contrary case would occur. Therefore, the entire data set contains  $73 * 3000 = 219000$  instances. This value of 73 is a consequence of the decision to model all possible drifts from one prior to another. Needless to say, the prior-resolution is limited to a fineness of 0.1. Altogether, there are  $n = 8$  possible class distributions (prior = 0.1 up to prior = 0.9 in steps of 0.1). If every possible drift shall be modeled, the number of drifts come across

$$2 * \sum_{k=1}^n k = \frac{2n(n+1)}{2} = 72 \text{ drifts} \quad (3.2)$$

between 73 concepts. The factor 2 in front of the sigma sign results from the fact that a difference is made between drifts and their converse. Therefore drift  $(\text{prior}_i \rightarrow \text{prior}_j) \neq (\text{prior}_j \rightarrow \text{prior}_i)$ . Although we have defined the prior as the larger ratio of both classes, e.g.  $\text{prior}_l = (1 - \text{prior}_l) \forall (\text{prior}_l < 0.5)$ , for the sake of simplicity, in the further part of the thesis, we will not strictly follow. So if e.g. a “prior of 20%” is mentioned, in fact a prior of 80% is meant. Finally, figure 3.3 shows the performance of the prior over the whole data set. The numerical values of the prior performance are to be found in table A.2 in the appendix.

### 3.2 Real Drifts

Unlike virtual drifts, real drifts are effectively based on a shifting of the concept. In exchange the prior levels off, due to the design of the data set it remains constant at a value of 0.5. For the sake of simplicity, the test arrangement is broken down into only two dimensions. Thus, there is a circle (originally a sphere), divided into two halves by a straight line  $\Theta$  (originally a plane) passing through the center, as shown in figure 3.4. The single instances (the points inside the sphere) are defined by two dependent variables  $x$  and  $y$  and one independent<sup>1</sup> variable  $z$  (whereas  $z$  is not to be seen in the two-dimensional figure 3.4). Plane  $\Theta$  divides the instances into two classes. In order to determine to which class an instance belongs, a simple but accurate and mathematical proper way is used. The normal  $\vec{n}$  of the plane  $\Theta$  is defined with  $|\vec{n}| = 1$  and  $\omega = 90^\circ$ . This gives rise to

$$\vec{n} * \vec{v}_P > 0 \longrightarrow P \in \text{Class 1} \quad (3.3)$$

$$\vec{n} * \vec{v}_Q < 0 \longrightarrow Q \in \text{Class 2} \quad (3.4)$$

So every instance will be assigned one-to-one to a class. The concept drift itself is defined by rotating the plane by a rotating angle  $\varphi$ . The rotating axis is passing through the center of the sphere and is parallel to the  $z$ -axis. Thus the parameter  $z$  is independent, a variation of it has no influence on the class membership of the instance. In figure 3.5, a real drift is shown.

The whole data set consists of 20 concepts per 3000 instances. Consequently, it contains 60000 instances and 19 drifts between the single concepts. The drifts themselves become the more intense the bigger the number of the instance is. Table A.1 in the appendix summarises all drifts. It must be pointed out that the given angle  $\varphi$  always corresponds to the preceding concept, not to a global reference. Angle  $\varphi$  is altered according to the following sequence: it starts at value  $\frac{\pi}{128}$  [radians] and will be duplicated at each step until the value of  $\frac{\pi}{16}$ . Thenceforward  $\text{drift}_i = \text{drift}_{i-1} + \frac{\pi}{16}$  up to  $\pi$ , which is equivalent to  $180^\circ$  degrees. A further rotating of the plane would not make any sense as the period of the systems is equal to  $\pi$ . A drift with  $\varphi + \pi$  corresponds exactly to a drift with  $\varphi$ .

---

<sup>1</sup> The reason for this independence is explained later on.

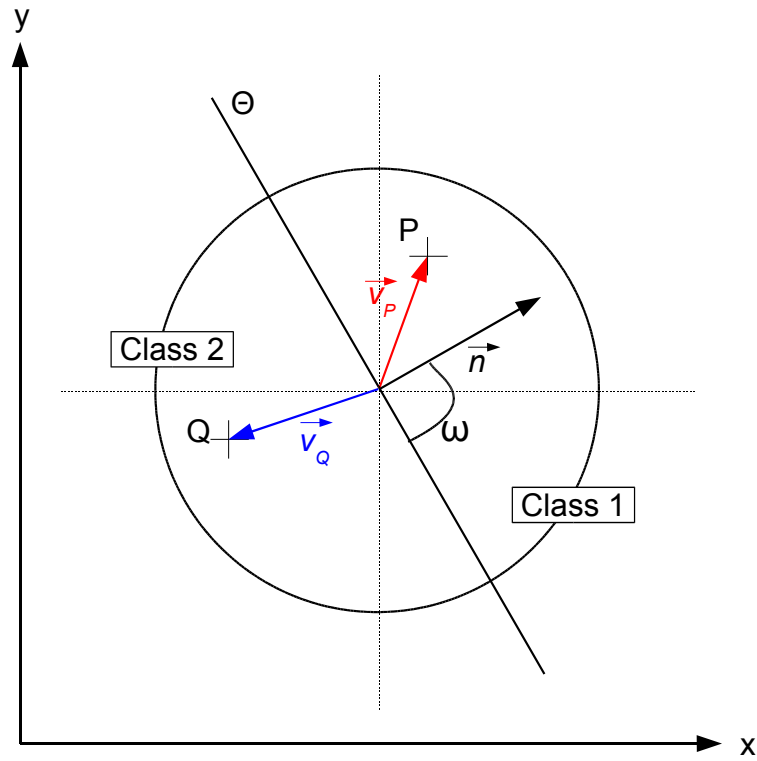


Figure 3.4: Mathematical background of a class determination (own graphic).

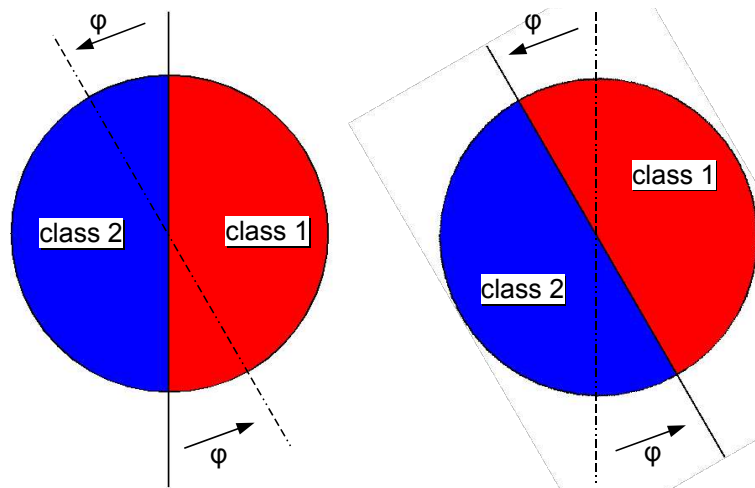


Figure 3.5: Illustration of a real drift (own graphic).

## Chapter 4

# Algorithms

### 4.1 Naïve Bayes

The naïve Bayes algorithm is known for its robustness. It does not require much computational power and produces prediction probabilities. The naïve Bayes algorithm is a simple probabilistic classifier. It is based on models which incorporate strong independence assumptions between the single parameter values<sup>1</sup>, hence are (deliberately) naïve. The basic assumption is that every attribute depends on the corresponding class only. The mathematical background is the Bayes theorem which describes the handling of conditional probabilities:

$$P(A|B) = \frac{P(A \cap B)}{P(B)} = \frac{\frac{P(A \cap B)}{P(A)} \cdot P(A)}{P(B)} = \frac{P(A) \cdot P(B|A)}{P(B)} \quad (4.1)$$

in which  $P(A)$  is the a priori probability and  $P(B|A)$  the probability for an occurrence of  $B$  on the condition that  $A$  occurs. If the number of occurrences is finite, the Bayes theorem is incidental in the following way: If  $A_j, j = 1, \dots, N$  represents a decomposition of the event space in disjoint occurrences, for the a posteriori probability  $P(A_i|B)$  applies

$$P(A_i|B) = \frac{P(B|A_i) \cdot P(A_i)}{P(B)} = \frac{P(B|A_i) \cdot P(A_i)}{\sum_{j=1}^N P(B|A_j) \cdot P(A_j)} \quad (4.2)$$

in which the relation

---

<sup>1</sup> In the following, probabilities must be multiplied. So the strong independence is a necessary constraint.



$$P(B) = \sum_{j=1}^N P(A_j \cap B) = \sum_{j=1}^N P(B|A_j) \cdot P(A_j) \quad (4.3)$$

is called the law of total probability. By combining the probability model with a decision rule, the naïve Bayes classifier itself is constructed. The common decision rule is to pick the hypothesis which is most probable. This is known as the maximum a posteriori decision rule. The corresponding classifier is the function  $\Omega$ , which is defined as follows:

$$\Omega(b_1, \dots, b_N) = \operatorname{argmax}_a P(A = a) \prod_{j=1}^N P(B_j = b_j | A = a). \quad (4.4)$$

Additionally, the naïve Bayes algorithm used in this thesis is combined with the Laplace estimation<sup>2</sup>[Duda 00]. Thus probabilities  $P(X) = 0$  (and therefore divisions by zero) will be prevented. Even though the parameter values of our data set are not independent, the algorithm provides quite good results. Furthermore, its simple calculation rule allows a fast computation. Basically, in spite of their naïve design and apparently over-simplified assumptions, naïve Bayes classifiers often work much better in many complex real-world situations than it might be expected because of their very simple design- as long as the parameter values are not too intercorrelated among themselves.

## 4.2 K-Nearest Neighbour

The nearest neighbour rule, a typical non-parametric decisions rule, is quite attractive because no prior knowledge of the distributions is required. In order to make decisions on the membership of unknown objects, these rules rely, instead on the training set, on objects with known class membership. The nearest neighbour rule, as its name suggests, classifies an unknown object according to the most prevalent class of its nearest neighbours in the measurement space using, most commonly, Euclidean metrics. Thus the algorithm calculates the distance to every known neighbour  $n_i$

$$\Delta n_i = |\vec{v}_i| = \sqrt{x_i^2 + y_i^2 + z_i^2} \quad \text{for } i = 1, \dots, t \quad (4.5)$$

---

<sup>2</sup> Laplace estimation inserts a single entry to every occurrence

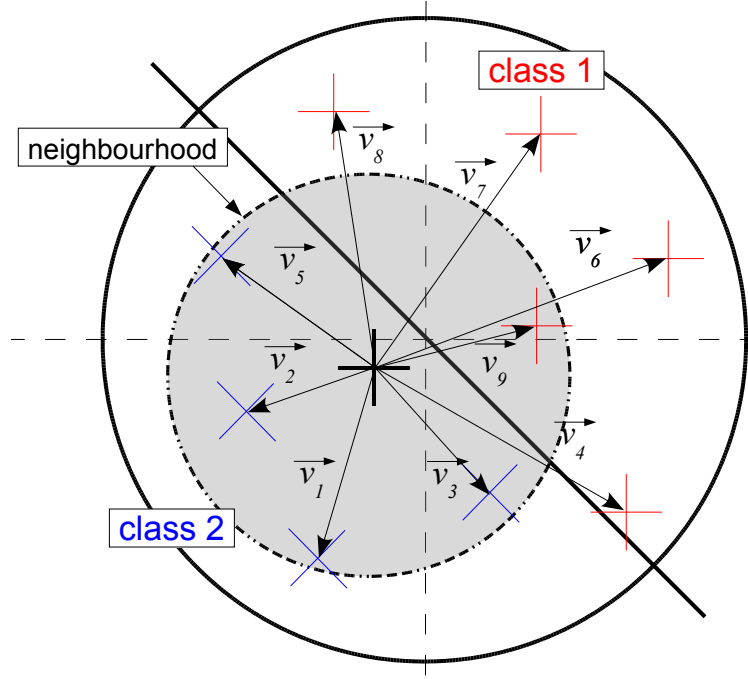


Figure 4.1: Diagram k-nearest neighbours

with  $t$  = total number of neighbours. In a second step, it collects the  $k$  (in our case  $k = 21$ ) nearest ones  $n_{near\_j}$ .

$$\begin{aligned}
 n_{near\_01} &= \min([\Delta n_1, \Delta n_2, \dots, \Delta n_{t-1}, \Delta n_t]) \\
 n_{near\_02} &= \min([\Delta n_1, \Delta n_2, \dots, \Delta n_{t-1}, \Delta n_t] \setminus [n_{near\_01}]) \\
 n_{near\_03} &= \min([\Delta n_1, \Delta n_2, \dots, \Delta n_{t-1}, \Delta n_t] \setminus [n_{near\_01}, n_{near\_02}]) \\
 &\dots \\
 n_{near\_ (k-1)} &= \min([\Delta n_1, \Delta n_2, \dots, \Delta n_{t-1}, \Delta n_t] \setminus [n_{near\_01}, n_{near\_02}, \dots, n_{near\_ (k-2)}]) \\
 n_{near\_ k} &= \min([\Delta n_1, \Delta n_2, \dots, \Delta n_{t-1}, \Delta n_t] \setminus [n_{near\_01}, n_{near\_02}, \dots, n_{near\_ (k-1)}]).
 \end{aligned}$$

Then the number of all instances which belong to the same class<sup>3</sup> are counted<sup>4</sup>

<sup>3</sup> As we exclusive deal with a 2-class problem only this case is mentioned. Basically, n-class problems are handled the same way.

<sup>4</sup> Under the assumption, that one instance  $n_{near\_j}$  counts as 1.

$$\text{Number of class 1} = m_1 = \sum_{j=1}^k n_{near\_j} \quad \forall \quad n_{near\_j} \in \text{class 1}$$

$$\text{Number of class 2} = m_2 = \sum_{j=1}^k n_{near\_j} \quad \forall \quad n_{near\_j} \in \text{class 2}$$

whereas  $m_1 + m_2 = k = 21$ . Finally, the class of instance  $n_{i+1}$  is equivalent to the class of  $\max(m_1, m_2)$ . E.g. in figure 4.1 with  $k = 5$ ,  $|\vec{v}_1|, |\vec{v}_2|, |\vec{v}_3|, |\vec{v}_5|$  and  $|\vec{v}_9|$  are the shortest connections to the nearest neighbours. As four of the neighbours belong to class 2 but only one to class 1, the new instance will be (correctly) classified as class 2 by the algorithm. Generally, it is a simple and evident algorithm. However, in practise it needs a lot of computing time as for every single instance all  $\Delta n_i$  have to be calculated and stored.

## Chapter 5

# Benchmarks

In order to know how well the controlled algorithm works, an applicable benchmark is needed. Without such a possibility of comparison, it would be useless to consider the power of the controlling instrument, or rather the performance of the incremental algorithm. For this reason so called benchmarks are calculated which represent either the highest reachable limit or the performance of other approaches than the regulation by the entropy.

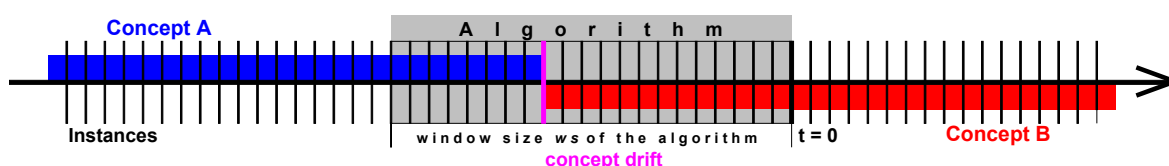


Figure 5.1: Window size  $ws$  of an algorithm

In the present case, the efficiency of the algorithm which predicts the label of an instance  $i$  is exclusively dependent on its forgetting rate. The forgetting rate is strongly coupled to the window size  $ws$  of the incremental algorithm. The window size  $ws$  determines on how many past instances the classifier (the algorithm) builds its rules. E.g.  $ws = 600$  means that the classification rule is built on the basis of the last 600 instances. This is schematically shown in figure 5.1. Therefore, the forgetting rate should be as small as possible as long as the concept beneath the data stream does not change. As the algorithm “learns” from wrong predicted labels, the more instances the algorithm classifies, the more precise the algorithm becomes. In the case of a concept drift, the suppositions are completely different. It would be optimal for the algorithm to totally forget the old concept and to concentrate only on the instances of the new concept. In other words, in case of a concept drift, the forgetting rate should be very high and the instances of the old concept should not be implicated as they provide (now) wrong

information about the actual situation. A high forgetting rate is equal to a small window size which contains only a few points which the algorithm is based on. Taking figure 5.1 as an example, as a sizable part of the old concept is involved in building the classifier rule, the window size of the algorithm is too large.

Until this point, vague terms like “efficiency” or “precision” in connexion with the algorithm have been discussed. In the next two short paragraphs, two measured values which allow us to express the efficiency of the algorithm (in relation to the forgetting rate) in numerical values will be specified.

### **Accuracy $\eta$**

The accuracy-term quotes the percentage rate of the correct predicted labels,

$$\text{accuracy } \eta = \frac{\text{correct predicted instances}}{\text{all instances}}. \quad (5.1)$$

A value of e.g. 430 correct classified instances of totally 500 instances leads to an accuracy of  $\eta = \frac{430}{500} = 0.86$  or 86%. The accuracy is a very lucid measure, simple to comprehend and very easy to calculate. It directly indicates information about the power of the algorithm. Since it does not include cost information, it is possible that a less accurate model is more cost-effective. This disadvantage will be compensated by calling in a second measure, the area under curve.

### **Area Under Curve $AUC$**

The so called area under curve  $AUC$  is a measure based on receiver operation characteristics  $ROC$  [Provost 01]. The  $ROC$  can be represented equivalently by plotting the fraction of true positives vs. the fraction of false positives<sup>1</sup>. This is equivalent to the including of cost information. In short, the area under curve is a measure which class-wise determines the prediction quality of a classifier. The value of the  $AUC$  for a certain class is equal to the area under the  $ROC$ -curve of the corresponding class<sup>2</sup>. A special case are two-class problems, where the  $AUC$ s of both classes have the same values. Therefore, in the subsequent discussion, only one value for the area under curve is considered. Although it is mostly used on two-class problems, it can be applied to multi-class problems [Ferri 03].

The main point of  $ROC$ -curves is that the power of a classifier can be determined per class without considering the class distributions [Fawcett 05]. On the one hand, this is very useful

---

<sup>1</sup> A false positive, also called a Type I error, exists when a test incorrectly reports that it has found a result where none really exists.

<sup>2</sup> Which is 0.5 for random and 1 for perfect classifiers

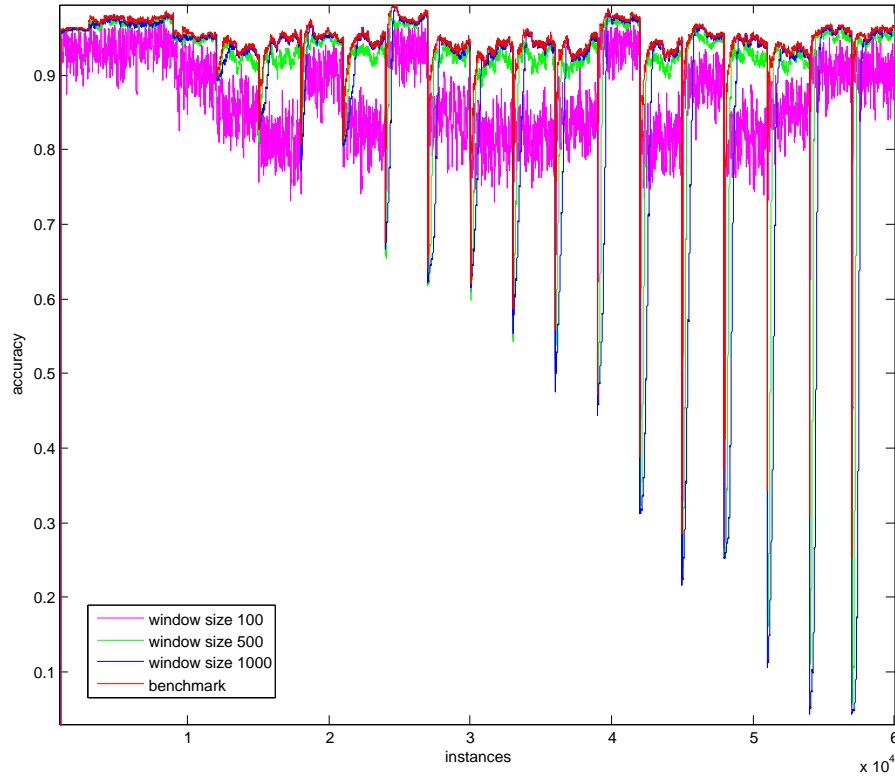


Figure 5.2: Accuracy benchmark sphere real discrete naïve Bayes

in the case of very different misclassification costs (e.g. credit fraud) or, on the other hand, if it is intended to assess of a classifier without regarding the class distributions is intended.

## 5.1 Ultimate Benchmark

The idea of the calculation of a benchmark as mentioned above is very simple. It is assume that each instance belongs to an optimal window size which results in the highest possible accuracy. Therefore, the algorithm is trained with all possible window sizes. In a next step, the accuracies at every single instance are compared and the window size which leads to the highest accuracy is defined as optimal. Consequently, the optimal window size  $ws_{opt}$  at instance  $i$  is defined by:

$$ws_{opt}(i) = ws \text{ of } \max(\eta(ws(i))) \quad \forall \quad ws(i) \in [1, \dots, 1000]. \quad (5.2)$$

In figure 5.2, three exemplary accuracies to corresponding windowsizes are shown.

Figure 5.3 points out a detail of figure 5.2 and allows to draw some interesting conclusions:

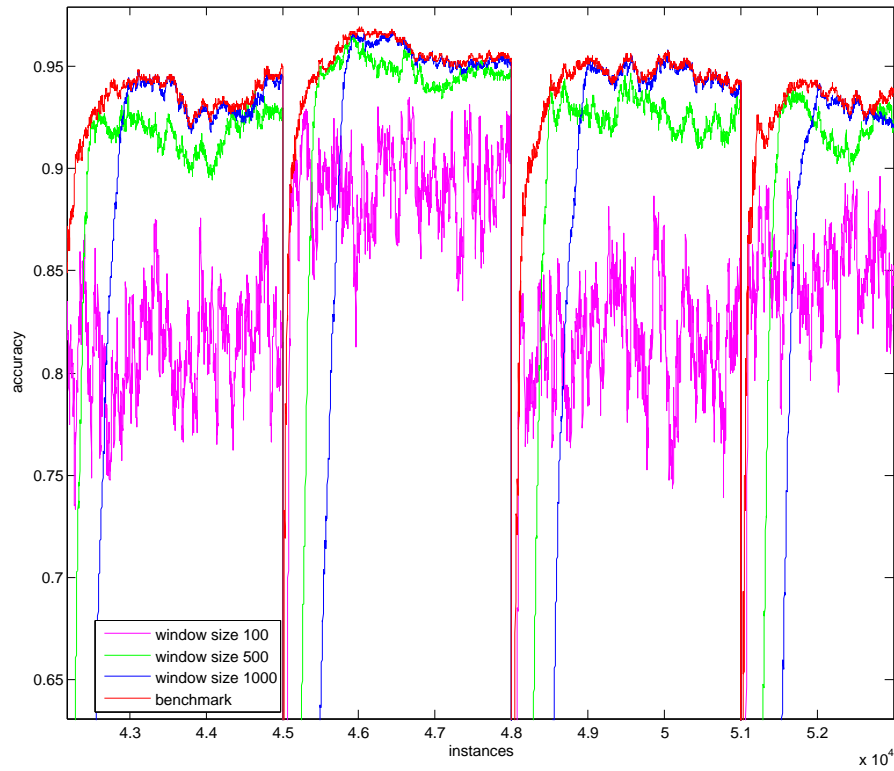


Figure 5.3: Accuracy benchmark sphere real discrete naïve Bayes, detail

1. Obviously, it can not be avoided that the accuracy breaks in after a concept drift, no matter which forgetting rate is chosen.
2. Shortly after a concept drift, small window sizes (e.g.  $ws = 100$ , the magenta colored line) perform better. Even though small window sizes are above-average reactive and deflect strongly, they lead to a higher accuracy than large window sizes. Due to this fact, the short window sizes are obviously inapt for the “normal state” of the system, e.g. if there are no concept drifts.
3. Large window sizes (e.g.  $ws = 1000$ , the dark blue colored line) behave “sedate”. Single outlier-instances do not influence the behaviour of the accuracy as much as if the algorithm would have been based on a small window size. On the other hand, large window sizes behave inertial. They need a lot more time to adapt to the new situation. After about 1000 instances after the concept drift, they perform plus-minus equal to the benchmark.

4. An average window size (e.g.  $ws = 500$ , the green colored line) combines the two extrema (see points 2. and 3.). It reacts slower than the small window sizes, but is more robust. Also, it performs good results earlier than large window sizes but never reaches their high average accuracy  $\bar{\eta}$

$$\bar{\eta} = \frac{1}{n} \sum_{i=1}^n \eta(i). \quad (5.3)$$

### 5.1.1 Conclusion

First of all, it should be noted that, due to the limited available CPU-performance for this project, the effective resolution of the above mentioned benchmark is set to a value of 10  $ws$ . With the given infrastructure, it was not possible to calculate every single window size  $ws$  as denoted in equation 5.2. Basically, a resolution of 1 %<sup>3</sup> is a close enough approximation. Missing values will be interpolated. A higher resolution would flatten the benchmark-curve, but would definitely not offer new information. Therefore, the defined benchmark (red line in figures 5.2 and 5.3) represents the most optimal effectiveness the algorithm is able to perform. If this mark is reached by the way of controlling the algorithm presented in this thesis, the best possible aim will be achieved. In the further part of the thesis, this algorithm's upper limit will be called the ultimate benchmark.

## 5.2 Optimal Window Sizes

If the above mentioned path is followed backwards, it should be able to draw a function which - hypothetically - assigns an optimal window size  $ws$  to every single instance  $i$ . Basically, this proceeding should lead to a regular function  $ws(i)$  over all instances. In reality it has to be dealt with outliers, which adulterate this function  $ws(i)$ . There are two reasons for this interfering behaviour. On the one hand, there are for example small window sizes which accidentally produce (independent from a concept drift) isolated good results (figure 5.4). On the other hand, other large window sizes perform similarly good values of accuracy; while the accuracy itself fluctuates within half of a percent, the window size fluctuates within a range of 250 (figure 5.5).

If this two phenomena are disregarded and just the graph without any adjustment factors is generated, the result looks rather disrupted (figures 5.6 and 5.7). The correlation between

---

<sup>3</sup> The maximum window size is set to 1000, 10 equals 1% of this value



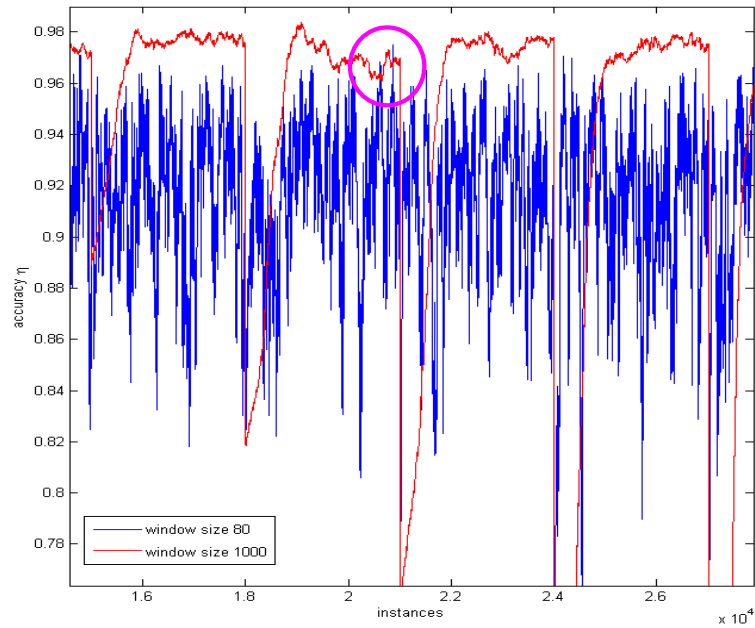


Figure 5.4: Small window sizes performs good accuracy by accident (plane sphere real continuous knn)(own graphic).

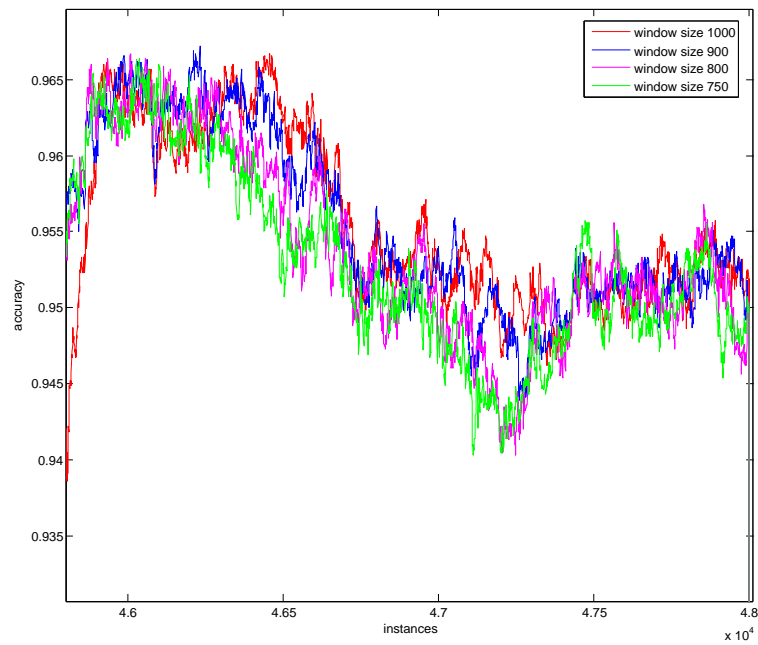


Figure 5.5: Big window sizes perform similar good results (plane sphere real continuous knn)(own graphic).

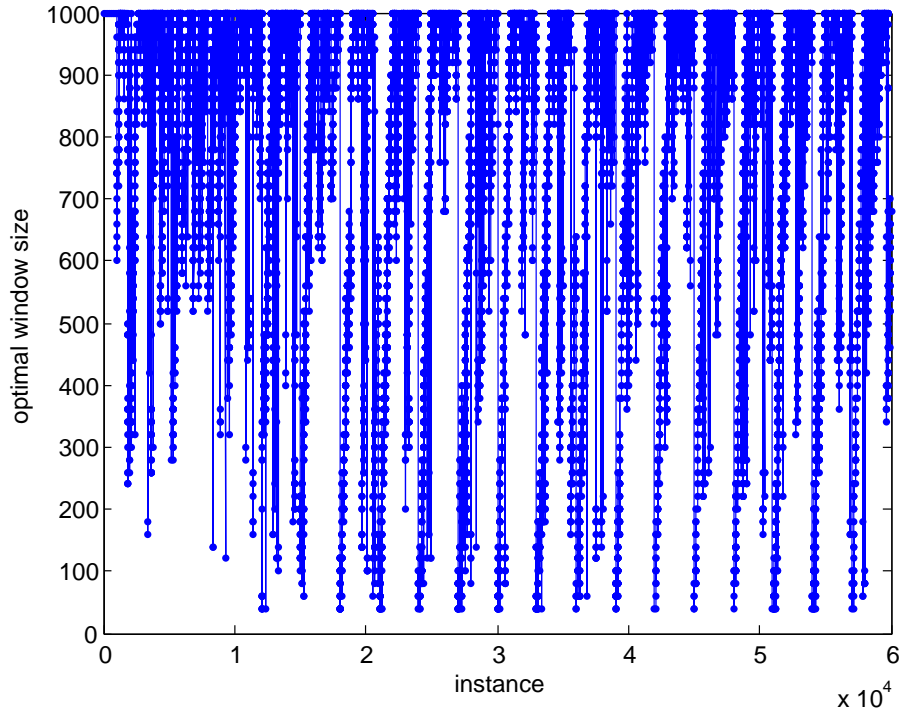


Figure 5.6: Optimal window sizes per instance without adjustment (plane sphere real continuous knn)(own graphic).

concept drifts, instances and optimal window sizes can be imagined, but it is not very obvious at a first glance. The reason therefore might be the above characterised facts. With slight modifications in the mapping formula from best accuracy  $\eta(i)$  to window size  $ws(i)$ , the correlation shall be clarified by avoiding the “noise” using different means. Mentioned below are four approaches, their effects and the consequences for the further part of this thesis. To clarify the theoretical explanations, in figure 5.8 three exemplary instancens are mentioned.

### Arithmetic Mean $\bar{x}$

The arithmetic mean  $\bar{x}$  is defined as

$$\bar{x} = \frac{1}{n} \sum_{i=1}^n x_i. \quad (5.4)$$

It is arguably the most common mean. However, it is not robust against single outliers. E.g.<sup>4</sup> for  $n = 3$  the arithmetic mean would result in  $ws = \bar{x} = 670$  at point in time  $t$ , what is a pretty bad indication of the real distribution. Outliers disrupt intense, therefore the arithmetic mean is an inapplicable measure.

---

<sup>4</sup> Please refer to figure 5.8.

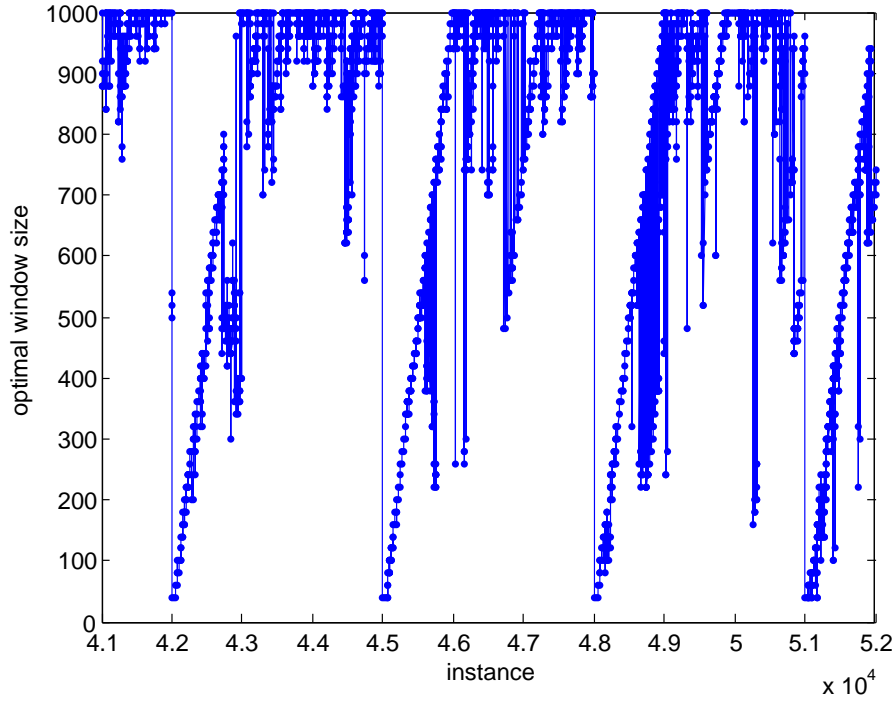


Figure 5.7: Optimal window sizes per instance without adjustment, detail (plane sphere real continuous knn)(own graphic).

	acc	ws	acc	ws	acc	ws
1	0.96	990	0.96	60	0.97	1000
2	0.95	960	0.95	1000	0.97	990
3	0.95	60	0.95	990	0.96	500
4	0.94	1000	0.95	970	0.96	490
5	0.93	980	0.95	980	0.58	550
6	0.90	950	0.95	960	0.57	950
7	0.88	890	0.94	920	0.56	870
8	0.87	880	0.94	930	0.55	940
9	0.85	910	0.94	890	0.55	960
10	0.80	930	0.94	900	0.54	780
	instance t		instance (t +1)		instance (t+2)	

Figure 5.8: Three exemplary instances with window sizes  $ws$  and corresponding accuracies  $\eta$  (own graphic).

**Median  $\tilde{x}$** 

The median  $\tilde{x}$  is the value below which 50% of the scores fall, or the middle score. For an odd number of scores the median is

$$\tilde{x} = x_{(n+1)/2}, \quad (5.5)$$

for an even number

$$\tilde{x} = \frac{1}{2} * (x_{n/2} + x_{(n+1)/2}). \quad (5.6)$$

If the median is applied to the best accuracies, it would result in the value  $ws = \tilde{x} = 60$  at point in time  $(t + 1)$ . Only the outlier is counted. This is exactly the opposite of what was aimed for. Applying it to the best window sizes (e.g.  $n = 5$ ), it would result in  $ws = \tilde{x} = 550$  at point in time  $(t + 2)$  (with a low accuracy  $\eta$  of 0.58) which would be mistaken likewise.

**Best of  $n$   $x_n$** 

Typically, the arithmetic mean  $\bar{x}$  and the median  $\tilde{x}$  result in window sizes which are too small. One might consider whether it would make sense to pick only the largest window size in a range of the best  $n$  window sizes. This approach would result in reasonable values for large window sizes. In case in which small window sizes (e.g. after a concept drift) would be expected,  $x_n$  would falsify the result towards too large window sizes.

 **$X$  Percent Neighbourhood**

Each of the first three examples given in figure 5.8 shows weaknesses in special situations of distributions. In the case of a relatively large data set with lots of instances, these special situations often occur. In any case, the final conclusion will be strongly influenced by the average measure. Therefore, a subtle measure which smoothes the curve “intelligently” is needed. According to the fact described in interrelation with figure 5.5, accuracies often appear in close quarters, i.e. in a nearby “neighbourhood”. Using this feature, the best accuracy is determined and all accuracies located in a  $X$  percent neighbourhood are assigned. Thereafter, the largest of the corresponding window sizes will be defined as optimal window size  $ws_{opt}$ .

This proceeding prevents the window size curve from outliers, delivers a comprehensible measure and does not adulterate the results. E.g. in case of the three exemplary instances in table 5.8, the three 1.5 percent neighbourhoods would reach from  $\eta = 0.96$  to 0.9456 at instance  $t$ , including window sizes  $ws$  990, 960 and 60 which results in an optimal window

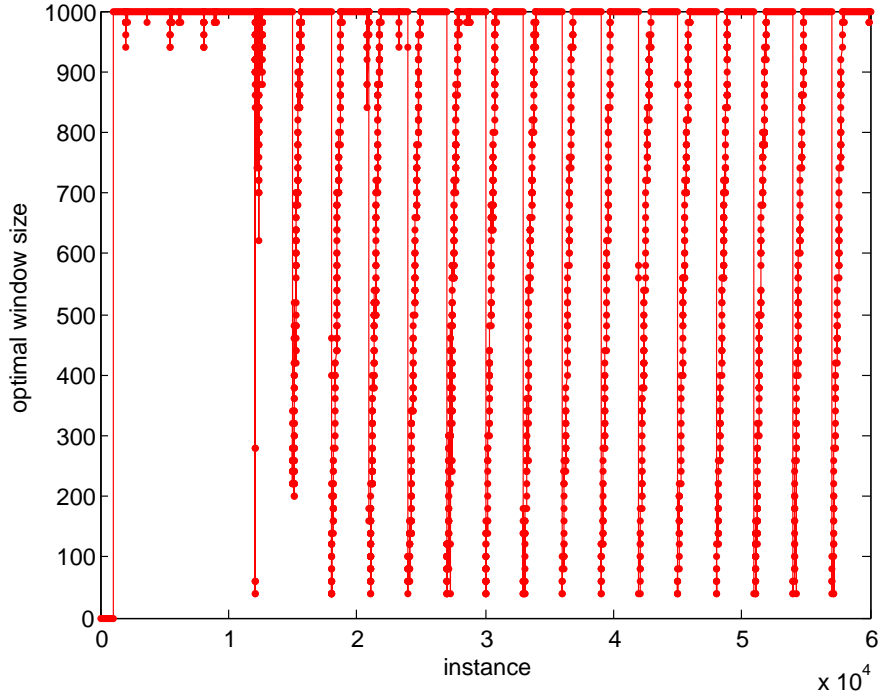


Figure 5.9: Optimal window sizes per instance with adjustment (plane sphere real continuous knn)

size  $ws_{opt}$  of 990. At instance  $t + 1$ , the same values appear for the neighbourhood, including window sizes  $ws$  1000, 990, 980, 970, 960 and 60, which results in  $ws_{opt}$  of 1000. At the third instance,  $t + 2$ , the neighbourhood reaches from  $\eta = 0.97$  to 0.9554, including the window sizes 1000, 990, 500, 490, also resulting in an optimal window size  $ws_{opt}$  of 1000.

Although the constructed instances at  $t$ ,  $t + 1$  and  $t + 2$  are not especially fitted to this example compared to the other means, doubtlessly this result makes more sense from a statistical point of view. If this so defined measure (maintaining the value of 1.5% for the range of neighbourhood) is applied to one of the data sets, the result looks like expected in theory (Figure 5.9 and 5.10).

### 5.3 Monitoring of the Results

In the further part of the thesis, basic considerations will be based on the benchmark and its optimal window size. For this reason, a short look into the noticeable specifics of these curves is taken in this paragraph. At this stage these specifics are only noted. Later on, the phenomenons will be explained in depth. First of all, it is pointed out that the window size  $ws$

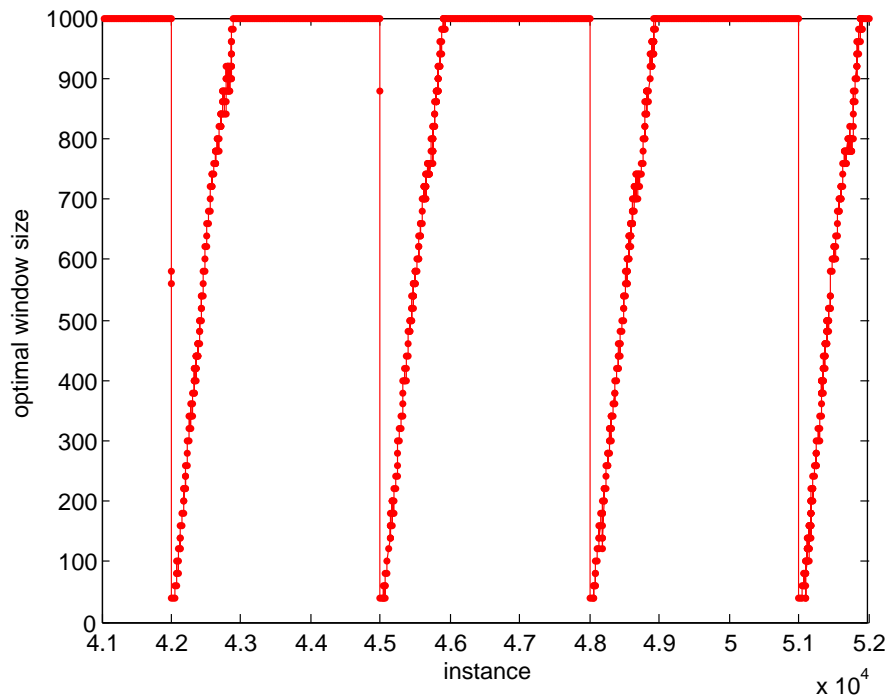


Figure 5.10: Optimal window sizes per instance with adjustment, detail (plane sphere real continuous knn)

reacts in exactly the same way after every concept drift<sup>5</sup> which can be seen in figure 5.9. Right after the drift, the window size “breaks totally in” and arrives at a minimal value. Afterwards, the window size recuperates and rises linearly (with a constant gradient) up to the predefined maximum value of 1000. This happens after exactly 1000 instances.

## 5.4 Error caused by Smoothing the Curve

The advantages of smoothing the window size-curve on theoretical grounds were discussed above. At this point, a short look at the numerical impact of this decision is taken and its consequences are shown. For this purpose the corresponding accuracy-curves are compared and the loss of precision is calculated. In Figure 5.11, the accuracies of the original benchmark are confronted with the accuracies of the smoothed benchmark. It is noticeable that after the first couple of instances after the concept drift the smoothed benchmark provides not as many as good accuracies as the ultimate benchmark, whereas after about 1000 instances, the two curves are nearly congruent. The error in the first part roughly spans an interval of 0 to maximal 1.5 percent. This maximum value seems to be rather large. However, it occurred accidentally.

<sup>5</sup> At the present, the small drifts at the beginning of the data set are disregarded

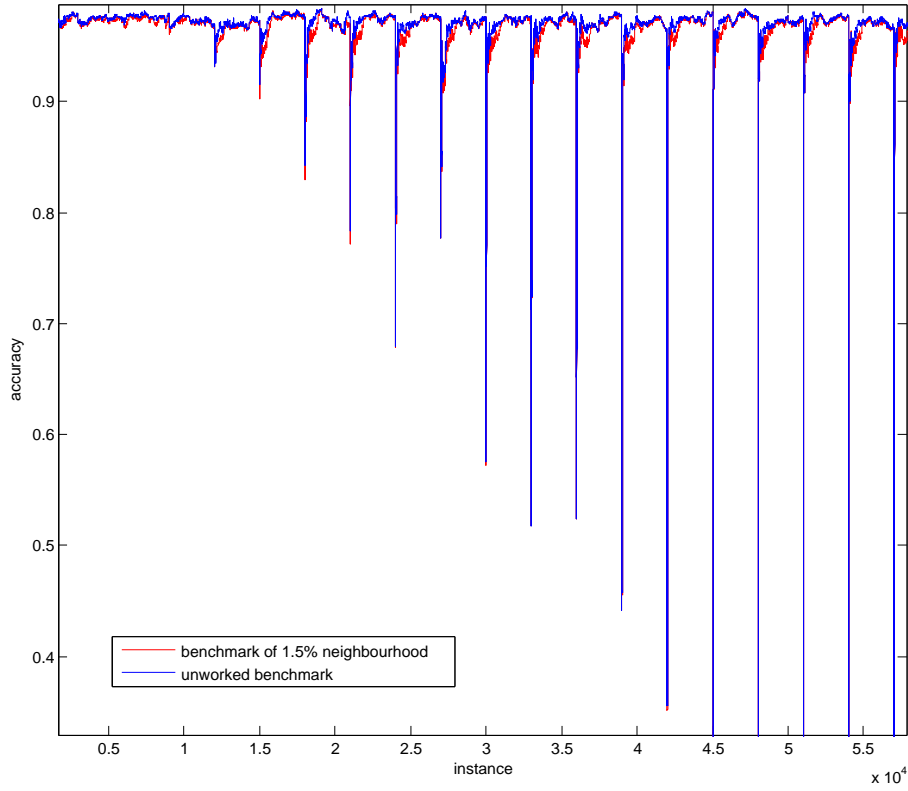


Figure 5.11: Accuracy without/with 1.5 percent neighbourhood smoothing (plane sphere real continuous knn)

Due to the fact that small window sizes react in a quite fragile way on outliers in the dataset, they sometimes produce “lucky punches”, i.e. good values. However, this is infrequent and not reproducible. Furthermore, the mean of all errors is only 0.3% which relativates the single outliers very much. Concluding, it can be retained that the demonstrated smoothing (and the curve itself) produces a reasonable result and makes sense from a mathematical point of view. Therefore, the further research is based on the assumption that the behaviour of the window size as shown in Figure 5.10 is ideal and leads to really very results.

## 5.5 Smoothed Benchmark

By eliminating outliers and therefore smoothing the curve of the optimal window sizes, a further benchmark was unconsciously generated. If the slightly smaller average accuracy  $\bar{\eta}$  of the smoothed curve is not considered as an error but as a more realistic assumption compared to the ultimate benchmark, a more reasonable benchmark is obtained. If the experiment would be performed several times and only the average result would be looked at, it would definitely look like the smoothed curve and not like the ultimate benchmark. Therefore, also from a

statistical point of view, the smoothed benchmark has its right to exist. In the further part of the thesis, each result will be tested on both benchmarks for the purpose of comparison.

## 5.6 Committee Benchmark

As previously mentioned, for real-time uses, committee-based algorithms are applicable only in a restricted way. This is due to the fact that they require a lot of calculating capitivity. Nevertheless, for the sole purpose of having a third possibility of comparison for our approach, based on such a committee, a benchmark is generated. Our committee consists of maximal 9 members  $m$ . The single members are distinguished by their different underlying forgetting rates<sup>6</sup>. In the present case, the window sizes 1000, 750, 500, 250, 100, 75, 50, 25 and 10 are chosen. As this selection contains both minima, maxima, mean window sizes and a sufficient number of small window sizes which achieve good results shortly after a drift, it is balanced and representative.

At every given point in time  $t$ , every committee-member  $m(ws)$  is trained on the trainingset by ten fold cross validation. Generally, cross validation is the practice of partitioning a sample of data into subsamples such that the analysis is initially performed on a single subsample, while further subsamples are retained “blind” in order for subsequent use for confirming and validating the initial analysis. In the domain of Data Mining,  $k$ -fold cross validation means a partitioning of the data into  $k$  subsamples, whereas  $(k-1)$  subsamples are used as (sub-)trainingsset and one as (sub-)testset. After a cycle, the roles are exchanged and the testing starts over until every subsample has been tested. The performance of the algorithm is defined as the average of all passes and gives an idea of the algorithm’s power.

In a second step, the achieved accuracies  $\eta$  of the members  $m(ws)$  are ranked. Deduced from this ranking, the committee of the  $n$ , whereas  $n \in [1, \dots, 9]$ , is composed. Now every member  $m(ws)$  is tested on the testset and, thereafter, predicts the label value of the instance  $i$  at point in time  $t$ . Finally, the class of instance  $i(t)$  is determined according to the chosen weighting of the single committee-members.

In sum, 14 different committee-benchmarks have been calculated. Numbers # 1 to # 9 represent unweighted committee-decisions. Finally, the class-prediction of the member  $m(ws)$  with the best accuracy  $\eta$  is chosen. By contrast, numbers # 10 to # 14 stand for weighted decisions. The following list displays the different weights of the particular 9 members, commencing with the member with the best accuracy<sup>7</sup>.

---

<sup>6</sup> Remember: Forgetting rate and window size  $ws$  are strongly coupled. A high forgetting rate stands for a small window size  $ws$  and a low one for a large window size.

<sup>7</sup> The mathematic expression of how committees # 12 and # 13 are weighted may be misunderstood. Simply



committe # 10 $\longrightarrow$	$\frac{1}{1},$	$\frac{1}{2},$	$\frac{1}{3},$	$\frac{1}{4},$	$\frac{1}{5},$	$\frac{1}{6},$	$\frac{1}{7},$	$\frac{1}{8},$	$\frac{1}{9}$
committe # 11 $\longrightarrow$	$\frac{9}{10},$	$\frac{8}{9},$	$\frac{7}{8},$	$\frac{6}{7},$	$\frac{5}{6},$	$\frac{4}{5},$	$\frac{3}{4},$	$\frac{2}{3},$	$\frac{1}{2}$
committe # 12 $\longrightarrow$	$\max \eta(m_{class1}) + (\max-1) \eta(m_{class1}) + \dots$ $\max \eta(m_{class2}) + (\max-1) \eta(m_{class2}) + \dots$								
committe # 13 $\longrightarrow$	$\max (\eta(m_{class1}) - 0.5) + (\max-1) (\eta(m_{class1}) - 0.5) + \dots$ $\max (\eta(m_{class2}) - 0.5) + (\max-1) (\eta(m_{class2}) - 0.5) + \dots$								
committe # 14 $\longrightarrow$	9,	8,	7,	6,	5,	4,	3,	2,	1

Not all committee benchmarks perform equally good results, numerical values<sup>8</sup> can be found in table 5.1. In addition thereto, the handling of such a large number of different benchmarks is tedious. On this account, for the furter part of the thesis, it is restricted to two significant ones, committees # 3 (greatest  $\bar{\eta}$ ) and # 13 (members with the best accuracy determine the class).

committe #	1	2	3	4	5	6	7
$\bar{\eta}$	0.9127	0.9158	0.9245	0.9218	0.9224	0.9182	0.9172
committe #	8	9	10	11	12	13	14
$\bar{\eta}$	0.9055	0.8999	0.9240	0.8999	0.9081	0.9184	0.9231

Table 5.1: Values of committee benchmarks

put into words, in committee # 12, all accuracies  $\eta$  of the members which predict the same class are added. The greater sum then determines the class-prediction of the whole committee. Committee # 13 is built the same way as # 12. The slight difference is that, in each case, 0.5 is subtracted from  $\eta$ . This shifts the weight towards members with high accuracies.

<sup>8</sup> The values refer to the naïve Bayes algorithm on the real drift data set.

## Chapter 6

# Entropy

### 6.1 Basics

The basic idea of this approach is to control the forgetting rate of the incremental algorithm by using the underlying information content of the data stream. In other words, if new information in the data stream is available, to provide good results in the future, the algorithm must react thereupon. The challenge was to find an instrument which delivers exact details about changes in the data stream. This latter instrument shall be found in the basic information theory, known as entropy.

The entropy-term itself was originally defined in thermodynamics and statistical mechanics where entropy is a key physical variable in describing a thermodynamic system. There is an important connection between entropy and the amount of internal energy in a system which is not available to perform simple work. Without going very deep into the theory of thermodynamics, it can be stated that entropy is a mightful measure to determine the order/disorder or rather the information content of a system. Claude E. Shannon detected in his fundamental paper of 1948 “*A Mathematical Theory of Communication*” [Shannon 48] that entropy is also a self-evident measured value to describe the information content in information theory. He defined entropy  $H$  in terms of a discrete random event  $x$ , with possible states  $1 \dots n$  as:

$$H(x) = \sum_{i=1}^n p(i) * \log_2 \left( \frac{1}{p(i)} \right) = - \sum_{i=1}^n p(i) * \log_2 (p(i)) \quad (6.1)$$

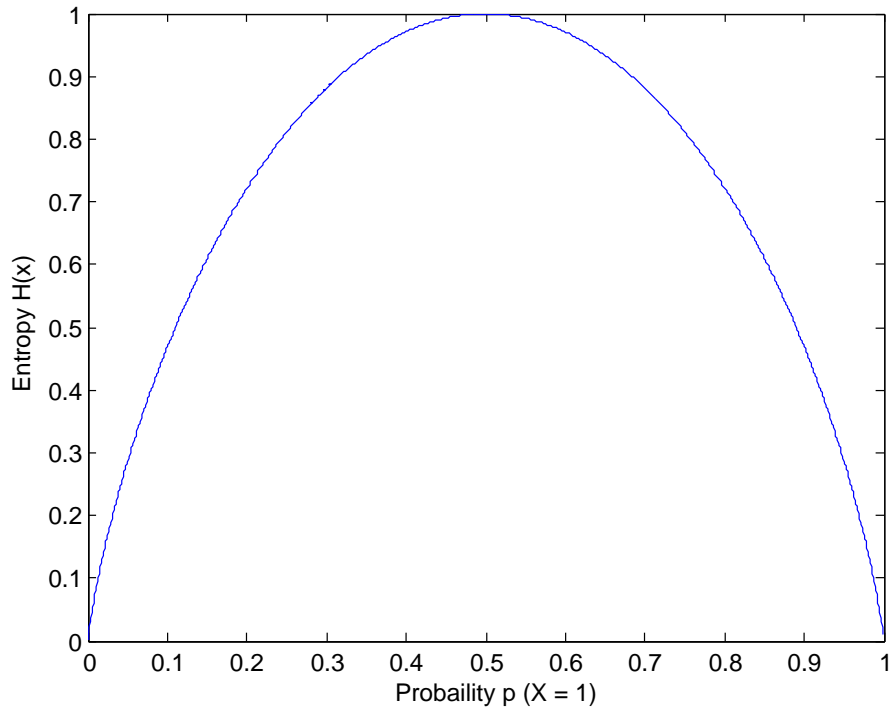


Figure 6.1: Entropy of a Bernoulli Trial

It is obvious that, in a two class problem (e.g. Bernoulli trial<sup>1</sup>, Figure 6.1), this formula leads to the following conclusions:

- $H(x) \in [0 \dots 1]$
- $H(x)$  reaches its maximum when the probabilities are equal,  $p_1(x) = p_2(x) = 0.5$
- If the result is already predefined (probability  $\in$  of  $[0, 1]$ )  $\rightarrow H(x) = 0$
- The entropy function is symmetric relative to the straight line  $p(x) = 0.5$

If it is tried to measure the information content of a continuous data stream by means of the entropy, the entropy-term has to be modified slightly and adapted to the present problem statement. Based on M. Constam's entropy term<sup>2</sup> in his master thesis [Contam 05], the term shall be adapted, its problematic parts<sup>3</sup> shall be eliminated, it shall be normalised and made adaptive - and therefore dynamically adjustable.

---

<sup>1</sup> In the theory of probability and statistics, a Bernoulli trial is an experiment of which the outcome is random and can be either of two possible outcomes. Therefore, the sum of all probabilities is always equal to 1.

<sup>2</sup> A brief subsumption is to be found in the following section.

<sup>3</sup> E.g. in the primal form the entropy term has to look far into the future

## 6.2 Primal Form Of Entropy Term

Based on the original formula for the entropy  $H(x)$ , M. Constam has developed a slightly modified form. The foundation pillars of this modified form consist in I. a sliding window over the instances which identifies the corresponding instances, II. a distribution of the values according to a bin and III. the taking of the (simple) average of all calculated data. In order to avoid confusion, the following example is considered. For further information it is referred to [Contam 05].

### 6.2.1 Breaking Down the Data Stream into Pieces

Our data stream is composed of  $n$  “sub-streams”;  $(n - 1)$ , so-called parameter streams, and one dependent label stream, whereas the label stream depends on the parameter streams. Only the parameter streams, all of them equally weighted, are finally implicated in the entropy.

$$H_{total} = \left( \frac{1}{n - 1} \right) * (H_1 + H_2 + \dots + H_{n-2} + H_{n-1}) \quad (6.2)$$

In the following, one of these parameter streams  $H_i$  which, in principle, are all the same, is exclusively considered. In order to become familiar with the proceeding of calculating the entropy of a continuous data stream, it will be illustrated by using an example.


### 6.2.2 Arranging of the Instances According to Label/Time

The exemplary sample stream  $H_i$  looks as follows:

				past window						future window								
Point of time t	...	-8	-7	-6	-5	-4	-3	-2	-1	0	1	2	3	4	5	6	7	...
Parameter stream	...	3	5	5	2	10	4	5	6	1	1	2	5	6	7	1	9	...
Label stream	...	1	0	0	1	1	0	1	0	1	0	1	0	1	1	0	0	...

For every point in time  $t$ , there are  $n - 1$  (equals 1 in the example) parameter streams. The parameters reach values in a predefined range, e.g. they are  $\in \mathbb{N}^+$  and  $< 11$ . Needless to say, in synthetic generated data sets, this range is known. Compared therewith, natural ones are not stringent. Commencing with a readout of all instances within the time frame from  $(t - n)$  to  $(t + n)$  (e.g.  $n = 6$ ), these instances are arranged according to their label and corresponding to the time window. Furthermore, the minima and maxima shall be determined.

	Label 0	Label 1
Past window	5, 4, 6	2, 10, 5
Future window	1, 5	1, 2, 6, 7



	Label 0	Label 1
Minimum	1	1
Maximum	6	10

### 6.2.3 Compilation of the Conditional Histograms

The determination of minima and maxima allows to calculate the borders for the bin separation. Generally, this segmentation of the instances into multiple bins is necessary for detecting a successive change<sup>4</sup> in the data stream. As initially mentioned, in principle, continuous concept drifts are not discussed in this thesis. However, in order to enable future research in this area, the entropy term will be calculated in a way that continuous concept drifts are recognisable. Without this separation, such a continuous concept drift would not be noticed by the entropy. The reason therefor is that all the instances would shift in the exact same way, the difference  $\Delta$  would also remain the same and the entropy would miss out the drift. For this reason, two bins are defined which are separated by the following borders:

$$\text{bin}_n\text{border} = \frac{1}{2} * (\text{Minimum} + \text{Maximum}) \quad \text{for } n \in [1, 2], \quad (6.3)$$

so  $\text{bin}_1\text{border} = \frac{1+6}{2} = 3.5$  and  $\text{bin}_2\text{border} = \frac{1+10}{2} = 5.5$  in the example. Starting from these borders, all values are divided into a so called “conditional histogram” which provides a basis for comparing the instances and calculating the entropy. In this histogram, not the values of the instances itself, but the number of values is important. The next step is to scale these two conditional histograms. The scaling is performed in order to keep the entropy values in the range range  $[0 \dots 1]$ .

$$\text{scaled value} = \frac{\text{number of instances}(\text{bin } n \text{ (past and future window)})}{\text{number of all instances}} \quad \text{for } n \in [1, 2] \quad (6.4)$$

---

<sup>4</sup> Continuous concept drift, described in section 3

Histogram Label 0		Bin 1 (1 <= x <= 3.5)	Bin 2 (3.5 < x <= 6)
Past window	Items itself	∅	4, 5, 6
	Number of items	0	3
Future window	Items itself	1	5
	Number of items	1	1

Histogram Label 1		Bin 1 (1 <= x <= 5.5)	Bin 2 (5.5 < x <= 10)
Past window	Items itself	2, 5	10
	Number of items	2	1
Future window	Items itself	1, 2	6, 7
	Number of items	2	2

Scaled histogram Label 0		Bin 1	Bin 2
Past window	Scaled number of items	0.00	0.75
	Scaled number of items	1.00	0.25

Scaled histogram Label 1		Bin 1	Bin 2
Past window	Scaled number of items	0.50	0.33
	Scaled number of items	0.50	0.67

### 6.2.4 Computing the Entropy

According to the entropy formula for a two class problem,  $H(x) = -(p_1 * \log_2(p_1) + p_2 * \log_2(p_2))$  the following four entropies are calculated

$$\text{Entropy label 0, bin 1} = H_{01} = -(0 * \log_2(0) + 1 * \log_2(1)) = 0$$

$$\text{Entropy label 0, bin 2} = H_{02} = -(0.75 * \log_2(0.75) + 0.25 * \log_2(0.25)) = 0.8113$$

$$\text{Entropy label 1, bin 1} = H_{11} = -(0.5 * \log_2(0.5) + 0.5 * \log_2(0.5)) = 1$$

$$\text{Entropy label 1, bin 2} = H_{12} = -(\frac{1}{3} * \log_2(\frac{1}{3}) + \frac{2}{3} * \log_2(\frac{2}{3})) = 0.9183.$$

As the distribution in the scaled conditional histogram can vary highly, the entropies are weighed according to the numbers of instances on which they are based. This procedure of weighting forecloses that a few instances warp the result. Also, it keeps  $H_{total}$  in the range  $[0 \dots 1]$ . The weights are calculated using the following equation

$$w_{ij} = \frac{\sum_{\text{windows}}^{\text{both}} \text{instances} \in [\text{same label}]}{\text{total number of instances}} \quad (6.5)$$

$$\text{weight label 0, bin 1} = w_{01} = \frac{0+1}{12} = \frac{1}{12}$$

$$\text{weight label 0, bin 2} = w_{02} = \frac{3+1}{12} = \frac{1}{3}$$

$$\text{weight label 1, bin 1} = w_{11} = \frac{1+2}{12} = \frac{1}{4}$$

$$\text{weight label 1, bin 2} = w_{12} = \frac{2+2}{12} = \frac{1}{3}.$$

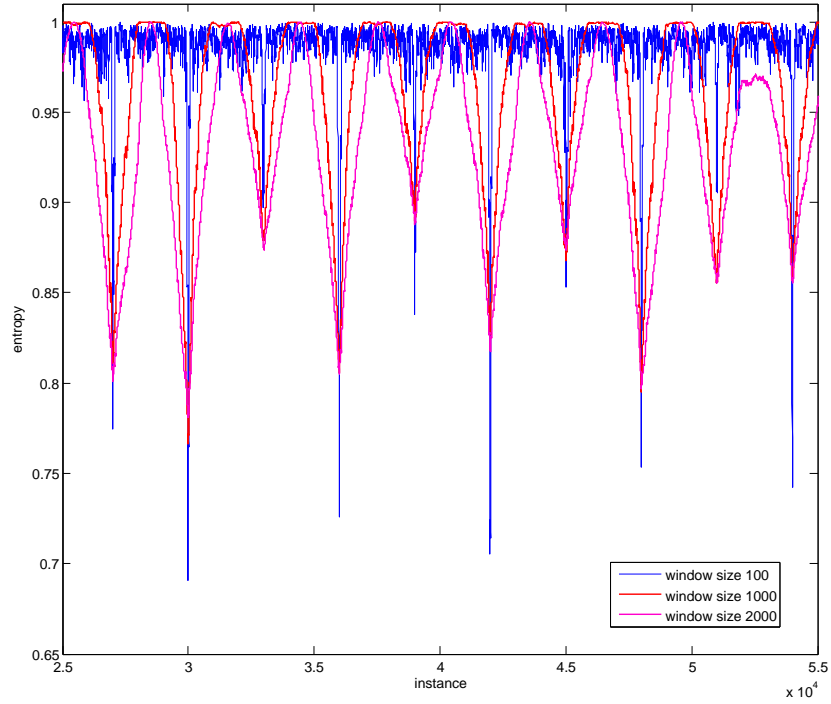


Figure 6.2: Entropys  $[-n; 0; 0; n] \in \text{of } [100, 1000, 2000]$  (prior discrete nb)

Finally, the total entropy  $H_{total}$  is calculated by multiplying the single entropies  $H_{ij}$  with their corresponding weights  $w_{ij}$ .

$$H_{total} = \sum_{i=0}^1 \sum_{j=1}^2 w_{ij} * H_{ij} = \frac{1}{12} * 0 + \frac{1}{3} * 0.8113 + \frac{1}{4} * 1 + \frac{1}{3} * 0.9183 = 0.8265 \quad (6.6)$$

If this procedure is repeated for every single instance, it leads to a one-to-one assignment between instances and entropy. At this state of our study, only the question of the length of the time frame of the sliding window remains to be defined. In the above example,  $n = 6$  was taken, M. Constam opted for  $n = 1000$  in his thesis. The reason for his choice was a threshold argumentation. Below this threshold, the curve is too discontinuous and craggy, above it, the decreasing edge (which indicates a forthcoming drift) is too flat. In figure 6.2, zoomed in figure 6.3, the two extremas are calculated and shown in comparison to Constam's choice. In chapter 6.3, the result of this calculation will be discussed. Also, a closer look shall be taken at the assets and drawbacks of the so defined entropy and develop a form with better applicability.

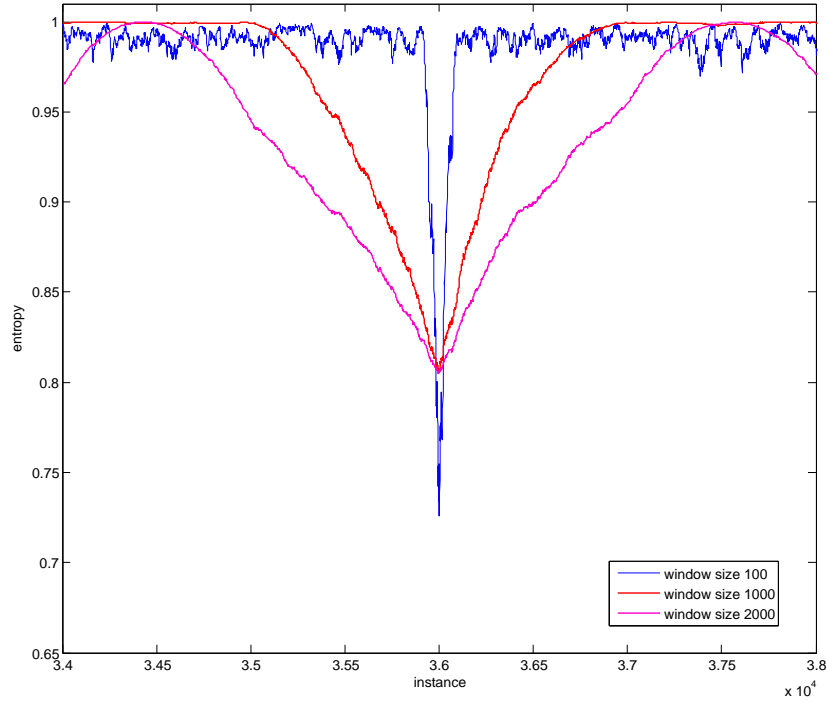


Figure 6.3: Entropies  $[-n; 0; 0; n] \in [100, 1000, 2000]$ , detail (prior discrete nb)

## 6.3 Further Development of the Entropy Term

### 6.3.1 Structural Weakness of Constam's Approach

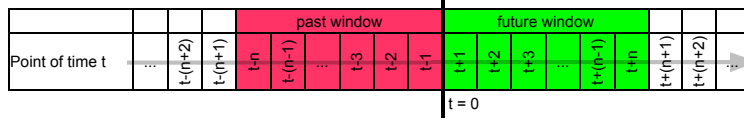
Basically, Constam's approach of calculating the entropy of a continuous data stream is well-founded and coherent. However, it possesses certain weaknesses which shall be tried to be eliminated. Some further elaborations shall be made regarding his approach. First of all, attention is paid to the fact that the formula deals with future information.

### 6.3.2 Avoiding the Need of Future Information

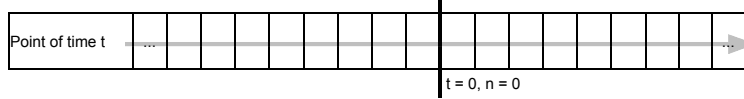
As it is impossible to unerringly predict incidents in the remote future it must be avoided that the term  $(t + n)$  for  $(t = 0)$  becomes too large and points too far into the future. Regardless of this latter fact, the ability to predict upcoming concept drifts must remain intact, even if an "effort" must be made in order to meet this demand. Subsequently, a couple of approaches are exemplified.



1. Starting at the initial situation with  $n = 1000$  ( $[-1000; 0; 0; 1000]$ ), the time frames<sup>5</sup> are tried to be optimised.

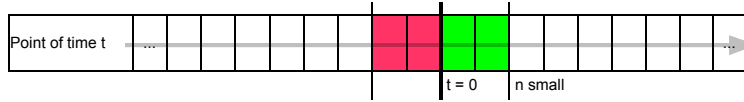


2. Choosing  $n = 0$



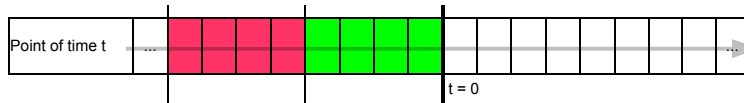
A simple, but inapt solution.  $n = 0$  ( $[0; 0; 0; 0]$ ) leads to window size  $ws = 0$  which again leads to entropy  $H = 0$  over all instances  $i$ . Thereby, the approach would be frustrated from the outset.

3. Choosing  $n$  small



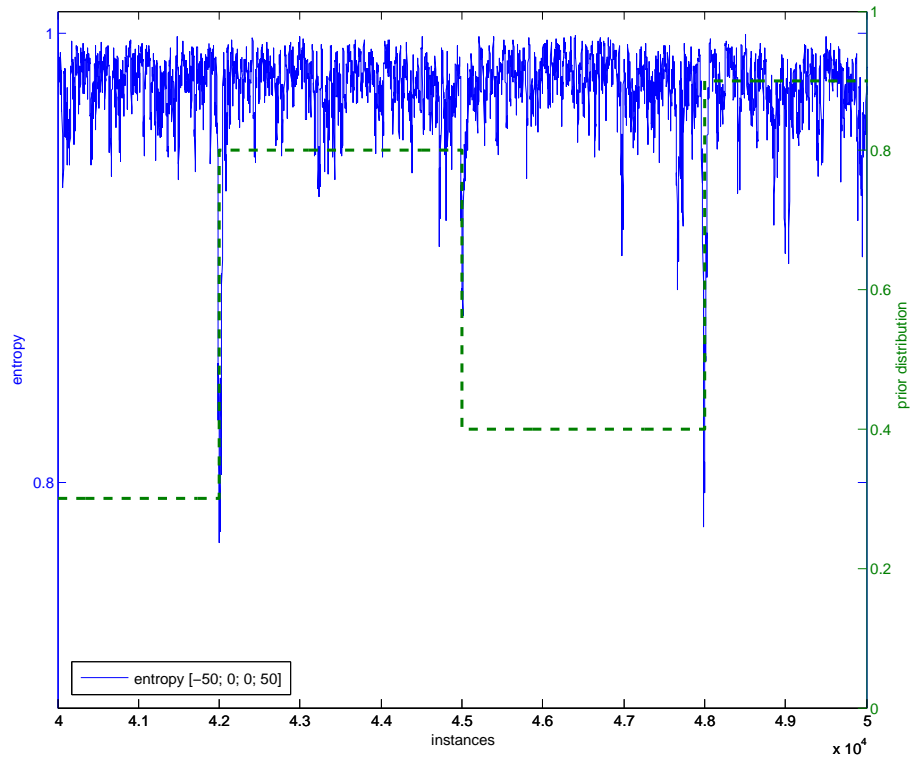
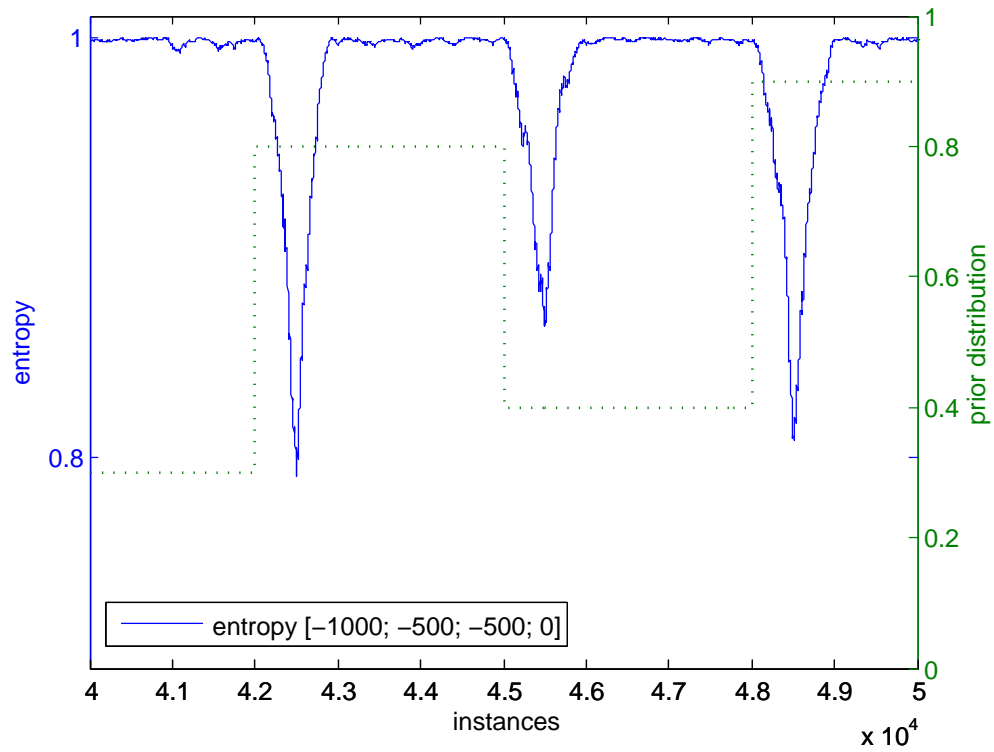
Looking “a bit” in the future is not a bad as looking “far” in the future. If  $n$  is chosen small (e.g.  $n = 50$ ,  $[-50; 0; 0; 50]$ , figure 6.4), on the one hand, the handicap of the need to know the future is relativated. On the other hand, the entropy becomes discontinuous and craggy and is therefore useless.

4. Shifting the windows to the past

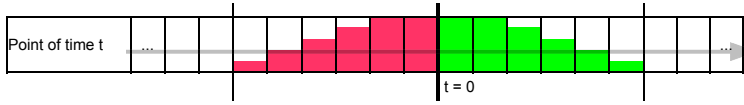


If the windows are shifted to the past, a look into the future is no longer necessary. Unfortunately, the entropy term lags behind the actual incidents (e.g. Figure 6.5, window sizes  $ws$  500 each,  $[-1000; -500; -500; 0]$ ). Literally spoken, the loss of up-to-dateness is the price for the compliance with axiomatic physical laws. As it emanates from figure 6.5, it takes 500 instances in order to discover the upcoming drift. Therefore, this approach does not meet the set demand either.

<sup>5</sup> To avoid misunderstandings and to unambiguously indicate the time frames, they are labeled in the following way:  $[spol; epol; spfw; epfw]$  whereas  $spol$  = start point old window,  $epol$  = end point old window,  $spfw$  = start point future window,  $epfw$  = end point future window

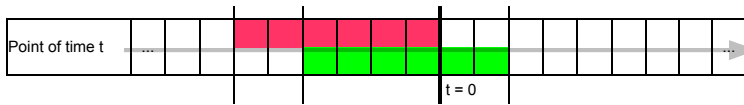
Figure 6.4: Entropy window size 50( $[-50; 0; 0; 50]$ ) (prior discrete nb)Figure 6.5: Entropy window size 500( $[-1000; -500; -500; 0]$ ) (prior discrete nb)

## 5. Weighting the windows



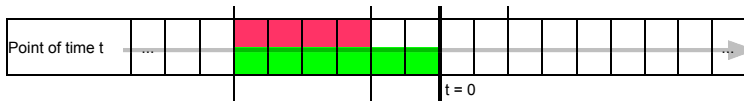
The initial idea of this approach was to privilege the importance of the close-by instances and to neglect the importance of the distant ones. This approach does not directly deal with the problem explained earlier on. It came up when working on the ideal window size  $ws$ , this is the reason why this approach is dealt with at this point. While the figure suggest, at the given resolution of course, a linear correlation between the distance from  $t = 0$  and the weight of the instance, experiments with logarithmic  $weight(t_n) \sim \log_i(t-n)$  (especial  $\log_e, \ln$ ) approaches were made at the same time. In a nutshell, smooth weighting entailed almost only disadvantages. The indications of a forthcoming drift, a precipitous gradient flank, are “washy” and unclear. For this reason, this approach was rejected. The mathematically interested reader’s attention is drawn to the fact, that the examples given up to now are applications of weighted windows. E.g. if the the weighting function is convolved, the result would look exactly like the first examples.

## 6. Overlapping the windows

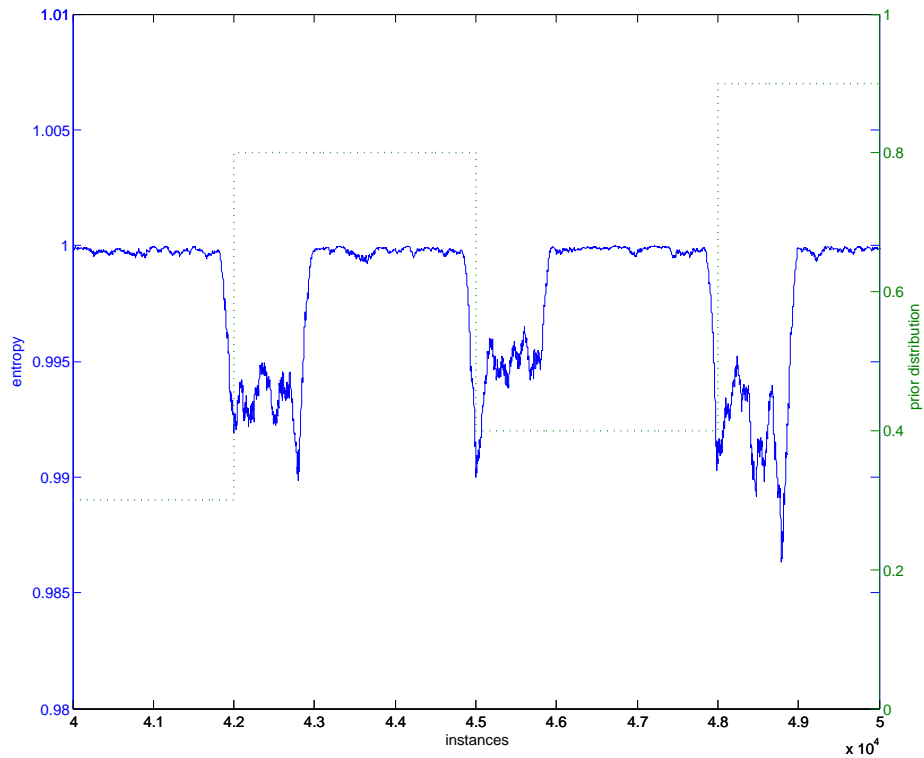


In this approach, the basic idea of shifting the window into the past is reassumed. The main and obvious difference to the previous approaches is that the past window adheres to  $t = 0$  and only the future window is shifted backwards. The effect is that parts of the windows overlap. The effect is seen in figure 6.6. Without explaining the mathematical background of the entropy’s behaviour, this approach can be rejected. The recovery phase which, in a second step or a further thesis is intended for regulating the increasing window size, is hardly existent. Therefore, also this approach is rejected.

## 7. Overlapping the windows but maintain the back border



Even though the results of the last attempt have not been promising, for this approach most of the the arrangement’s parts are kept. Just the back borders of the two windows are matched. This formation features several advantages, as to be retraced considering figure 6.8:

Figure 6.6: Entropy  $([-1000; 0; -800; 200])$  (prior discrete nb)

- A foresight into the future is no longer necessary
- The entropy reacts comparatively fast on concept drifts.
- The overlapping part can be understood as some kind of a smoothing factor which levels out disturbing factors in the curve.
- The overlaying part of the future window processes the incoming information.
- In a “ideal world” the future window would overlay the past one with only one instance. So the algorithm would react instantaneously on changings of the concept and therefore drifts.

According to the above mentioned facts, it has been invested a lot of time in the above described form of the entropy. Finally, it has been rejected all the same. There are two reasons therefor. First, the numerical entropy-values provided by this form range in the region of  $10^{-3}$  down to  $10^{-7}$ , dependend on the ratio  $\Delta$  between the windows. A simple correction factor to transfer the values back in the range of  $[0 \dots 1]$  simply does not exist. This conclusion is based on mathematical considerations. Besides, in these considerations the second reason for rejecting the approach is to be found. Overlapping

the windows ends<sup>6</sup>, after some substitutions, in the following entropy formula derived from the original one. Basically, the fraction  $\frac{\zeta}{\zeta+\gamma}$  represents the overlap of parts of the windows.

$$\begin{aligned}
 H_{orig} &= -(p * \log_2 p + (1 - p) * \log_2 (1 - p)) \\
 H_{overlap} &= - \left( \frac{\zeta}{\zeta + \gamma} * \log_2 \left( \frac{\zeta}{\zeta + \gamma} \right) + \left( 1 - \frac{\zeta}{\zeta + \gamma} \right) * \log_2 \left( 1 - \frac{\zeta}{\zeta + \gamma} \right) \right) \\
 &= - \left( \frac{\zeta}{\zeta + \gamma} * \log_2 \left( \frac{\zeta}{\zeta + \gamma} \right) + \frac{\gamma}{\zeta + \gamma} * \log_2 \left( \frac{\gamma}{\zeta + \gamma} \right) \right) \\
 &= - \left( \frac{\zeta}{\zeta + \gamma} * (\log_2 \zeta - \log_2 (\zeta + \gamma)) + \frac{\gamma}{\zeta + \gamma} * (\log_2 \gamma - \log_2 (\zeta + \gamma)) \right) \\
 &= \frac{-\zeta}{\zeta + \gamma} * \log_2 \zeta + \frac{\zeta}{\zeta + \gamma} * \log_2 (\zeta + \gamma) + \frac{-\gamma}{\zeta + \gamma} * \log_2 \gamma + \frac{\gamma}{\zeta + \gamma} * \log_2 (\zeta + \gamma) \\
 &= \frac{\zeta}{\zeta + \gamma} * (\log_2 (\zeta + \gamma) - \log_2 (\zeta)) + \frac{\gamma}{\zeta + \gamma} * (\log_2 (\zeta + \gamma) - \log_2 (\gamma))
 \end{aligned}$$

$$H_{orig} \nleftrightarrow H_{overlap}$$

At this point the transformation is stopped. Obviously, the term  $\log_2(\zeta + \gamma)$  inhibits a conversion of  $H_{overlap}$  to  $H_{orig}$ . This is not a simple question of missing numeracy skills,

$$\log(a + b) \nleftrightarrow x * \log(y) \quad (6.7)$$

is an generally insolvable problem, as the sum  $a + b$  into the brackets can not be eliminated. To illustrate this problem, a synthetic data set with a maximal drift is generated. Then

---

<sup>6</sup> The time-consuming derivation is omitted due to legibility. Interested readers are welcome to write out the comprehensive equation in full by their own.

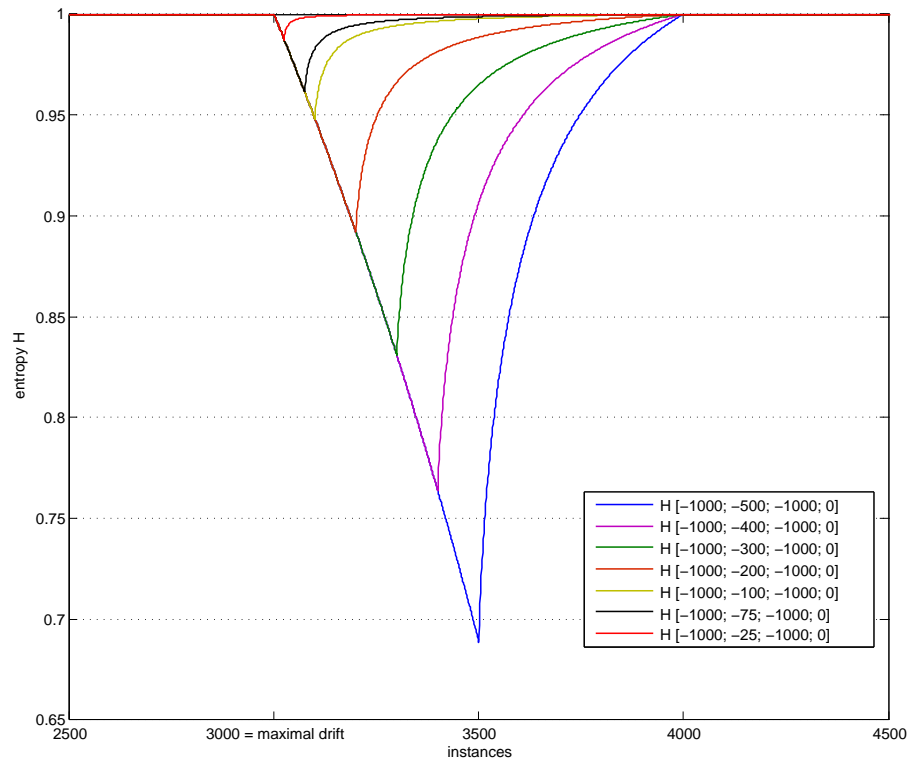
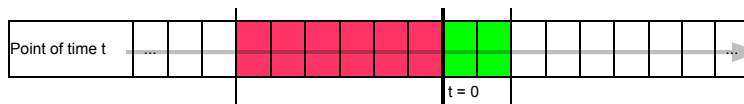


Figure 6.7: Behaviour the entropy with overlapping windows in case of a maximal drift at instance 3000

the entropy with different window sides slides over this instances. According to the theory, the entropy should reach a value of 0 in case of a maximal drift (cp. equation 6.1 and figure 6.1). But as to be seen in figure 6.7 the entropys' minima depends on the window size. Introducing a universal correcting factor is, according to equation 6.7, impossible. Recapitulating, this approach is rejected according to theoretical and practical reasons.

#### 8. Surrender the congruence of the time frames



As the entropy-form of overlapping the windows is unfit for regulation the algorithm, a further approach is tracked. By trying to make the most of the previous insight, the “normal” past window is combined with a small future one. The large past window is supposed to smooth the curve. The future window is held short to keep the algorithm’s up-to-dateness. With this arrangement useful results can be reached. Besides, the

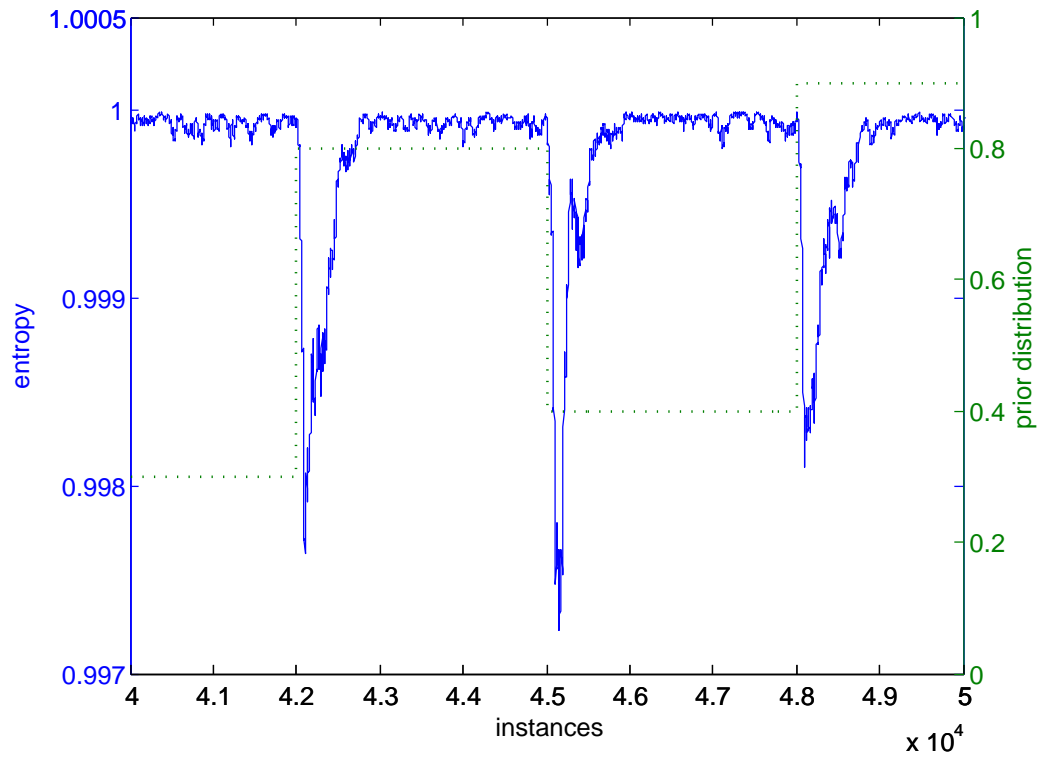
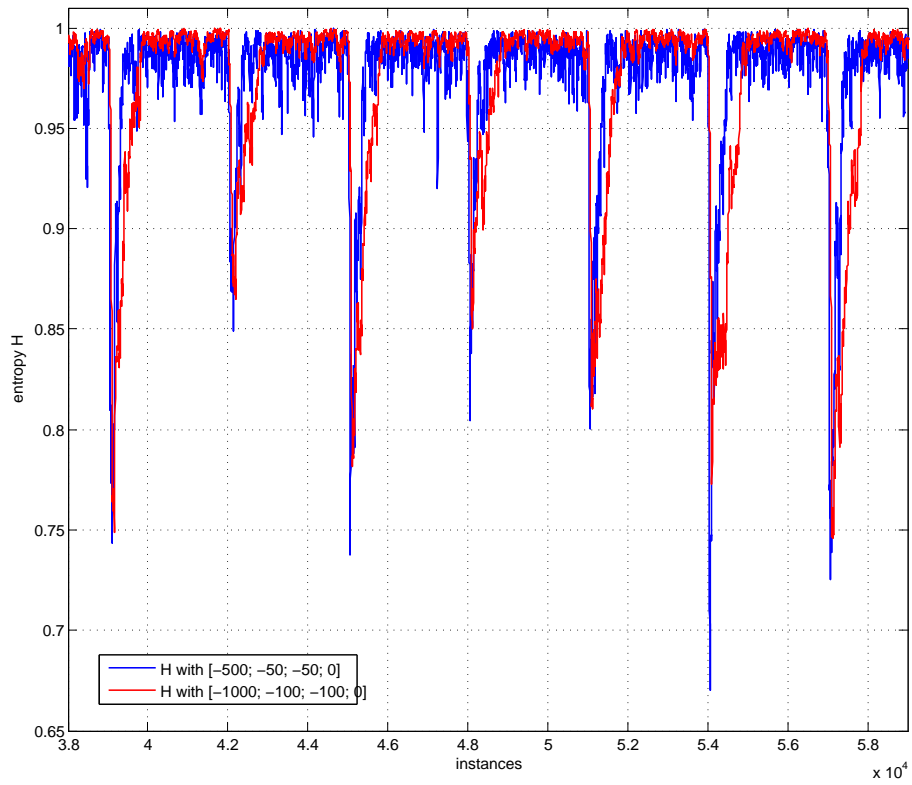
Figure 6.8: Entropy  $([-1000; -100; -1000; 0])$  (prior discrete nb)

Figure 6.9: Entropies side by side (real discrete nb)

mathematical background is kept correct, further adjustments of the entropy-term must not be made. Figure 6.9 shows two possible entropy curves. In the further part of the thesis this arrangement of the windows will be called “side-by-side”.

## 6.4 Normalisation

In the original formula 6.1 only the parameter streams of the data set are included. In contrast, the label stream itself is irrelevant. For the purpose of normalisation, this is changed. Basically, the label entropy will be calculated the same way as a single parameter stream. Per definition, only one label stream is existing. Therefore, a weighting according to equation 6.2 does not apply. The normalised entropy is defined as follows:

$$H_{normalised}(x) = \frac{H_{parameter}(x)}{H_{label}(x)} \quad \forall \text{ instances.} \quad (6.8)$$

The data sets react different on this normalisation. In the case of real drifts,  $H_{label} \approx 1$ , and therefore  $H_{normalised} \approx H_{parameter}$ . In contrast, in the case of prior drifts,  $H_{label} \approx H_{parameter}$ ,  $H_{normalised} \approx 1$ . Consequence there of is that  $H_{normalised}$  distinguishes between real and prior drifts. Therewith, the problem of handling the drifts’ different natures has been solved. See figures 6.10 and 6.11.

## 6.5 Entropy Limit Definition

Above, the ideal arrangement of the entropy’s window sizes has been established. As the arrangement itself is determined, its specific parameters will be deduced by limit observations. The available parameters are the window sizes and their ratio. These parameters are described by the variables  $ws$  which define the total window size ( $ws = ws_{old} + ws_{new}$ ), and the ratio  $\Delta$  between them.

$$\Delta_{relative} = \frac{ws_{new}}{ws_{old} + ws_{new}} = \frac{ws_{new}}{ws}, \quad (6.9)$$

$$\Delta_{absolute} = ws - ws_{old} = ws_{new}. \quad (6.10)$$



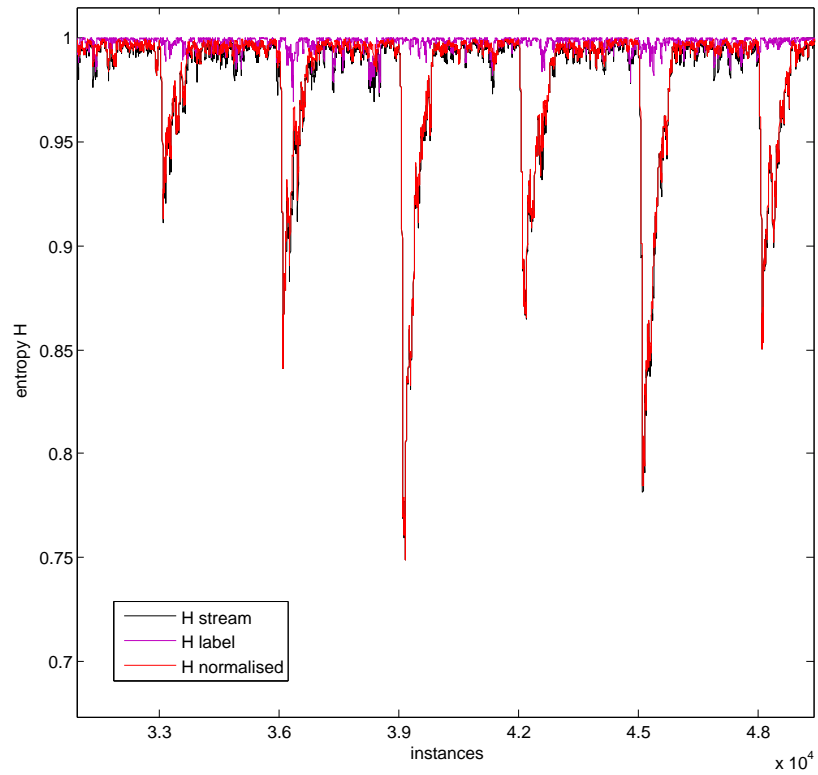


Figure 6.10:  $H_{normalised}$ ,  $H_{parameter}$  and  $H_{label}$  in the case of real drifts

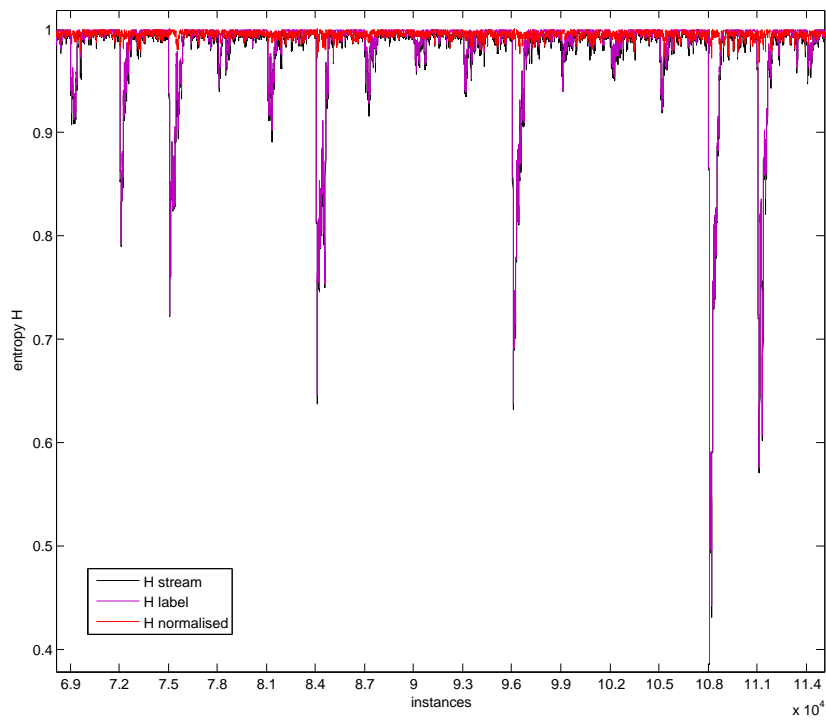


Figure 6.11:  $H_{normalised}$ ,  $H_{parameter}$  and  $H_{label}$  in the case of prior drifts

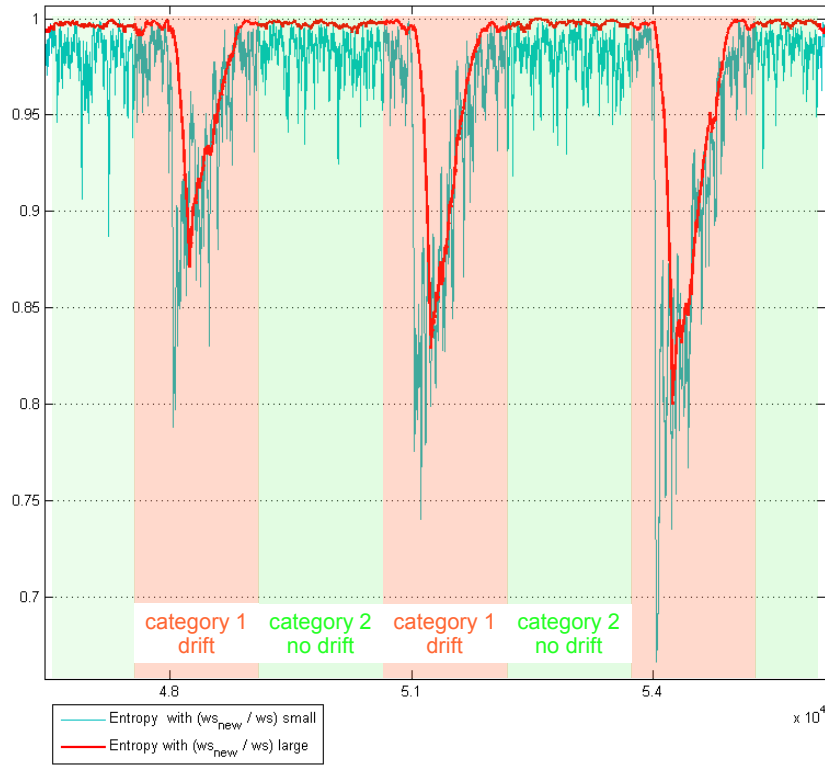


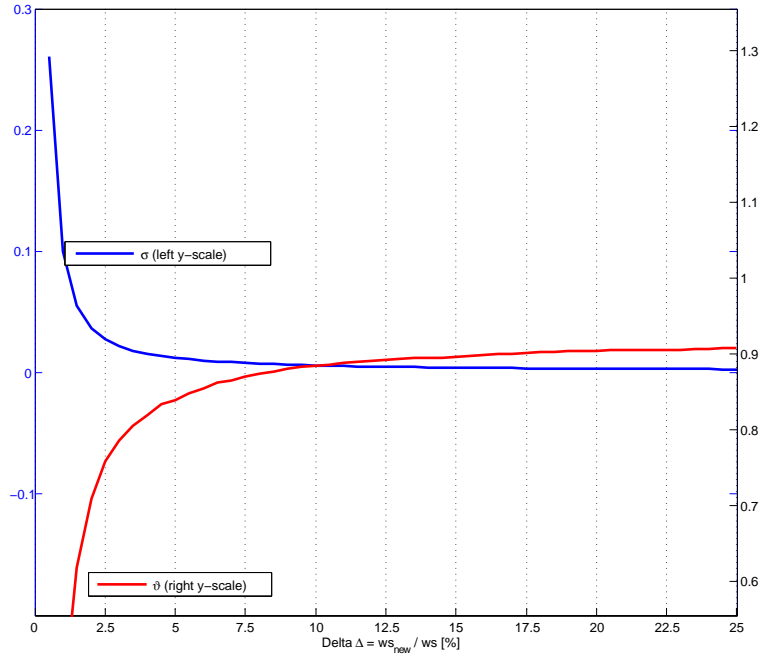
Figure 6.12: Divide the dataset into drift/ no drift-categories

The first restriction that has to be followed is that  $\Delta$  must be prevented from getting too large, upper limit of  $\Delta_{absolute} = 200$ . If  $\Delta_{absolute}$  exceeds this limit, the information extracted from the entropy calculation is too obsolete and therefore not usable. Accepting a  $\Delta_{absolute} > 200$  would mean a substantial time delay. As explained in chapter 1, the whole approach is aimed for real-time applications, therefore, such a limit must be set.

First, the window size  $ws$  of the entropy is defined to be 1000. This derives from the fact that the algorithm's maximal size equals 1000. Later on,  $ws$  can be changed if necessary. However, the value of 1000 represents a reasonable set up value. With a given  $ws$ , the last variable to be defined is  $\Delta_{relative}$ . It is restricted to range between 0 and 20 percent<sup>7</sup>,  $0\% < \Delta_{relative} < 20\%$ .

At first sight, it is not obvious whether  $\Delta_i$  or  $\Delta_j$  suits the purpose better. Therefore, an accurately defined criteria for the evaluation of the different Deltas is needed. For this purpose, all instances of the whole data set are divided into two categories. As the first category's instances do not take part of any drifts, they can be distinguished from the second category. In contrast thereto, the instances of the second category are directly affected by the drift. Figure 6.12 explains the problem. Category 1 represents the drift and category 2 the data

<sup>7</sup> The value of 20% results from the condition  $\Delta_{absolute} < 200$  and the chosen  $ws = 1000$

Figure 6.13: Determinate an ideal  $\Delta$ ;  $\sigma$  vs  $\vartheta$ 

between the drifts. In view of the search for an optimal  $\Delta$ , the following statements can be made:

1. The entropy should indicate the drifts as explicitly as possible. Therefore, the minimum of the instances of category 1 has to be as small as possible.
2. If there is no drift, the entropy-curve should remain flat and should not look noisy.

In figure 6.12, two different curves are displayed which point out an upcoming problem. Needless to say, the curves show “extrem” window sizes, but they are ideal to understand the basic behaviour of this particular curves. The blue line represents a small  $\Delta$ . During the drift, the curve gets relatively small; a positive characteristic. Between the drifts, the single instances (of category 2) show a discontinuous behaviour. In turn, this is negative for our purposes. Finally, the red curve with a large Delta  $\Delta$  shows a contrary behaviour. Consequently, a trade-off problem has to be dealt with and, preferably, a middle way has to be found.

In order to put the described trade-off problem into measurable quantities, two measures are defined as follows:

$$\min(H(x_j) = \min(H(x_i)) \quad \forall \quad x_i \in \text{category 1 of a drift } j \quad (6.11)$$

$$\vartheta = \frac{1}{n} \sum_{j=1}^n \min(H(x_j)) \quad \text{with} \quad n = \text{number of drifts} \quad (6.12)$$

$$\sigma = \frac{1}{m} \sum_{k=1}^m (1 - H(x_k)) \quad (6.13)$$

with  $m$  = total number of instances in category 2. The measures  $\sigma$  and  $\vartheta$  are simple and useful to compare and evaluate the quality of different entropies. It is discussible whether the averaging of  $\vartheta$  is justified or not as the drifts' intensity varies over time. Figure 6.14 displays the minima  $\min H(x_j)$  for  $j = 1$  to 19, all drifts. In essence, the basic structure of the minima-curves among the different Deltas  $\Delta_i$  and  $\Delta_j$  remains the same. Accordingly, the average  $\sigma$  represents the whole curve meaningfully.

In figure 6.13  $\sigma$  versus  $\vartheta$  is plotted, whereas  $\vartheta$  declines and  $\sigma$  increases. Needless to say, the x-coordinate of the intersection point depends exclusively on the scaling of the y-axis – a useless graph in a purely mathematical sense. Nevertheless, the behaviour of the curves provides valuable information. Basically, their forms resemble a logarithmic function, as they flatten in positive x-direction. While

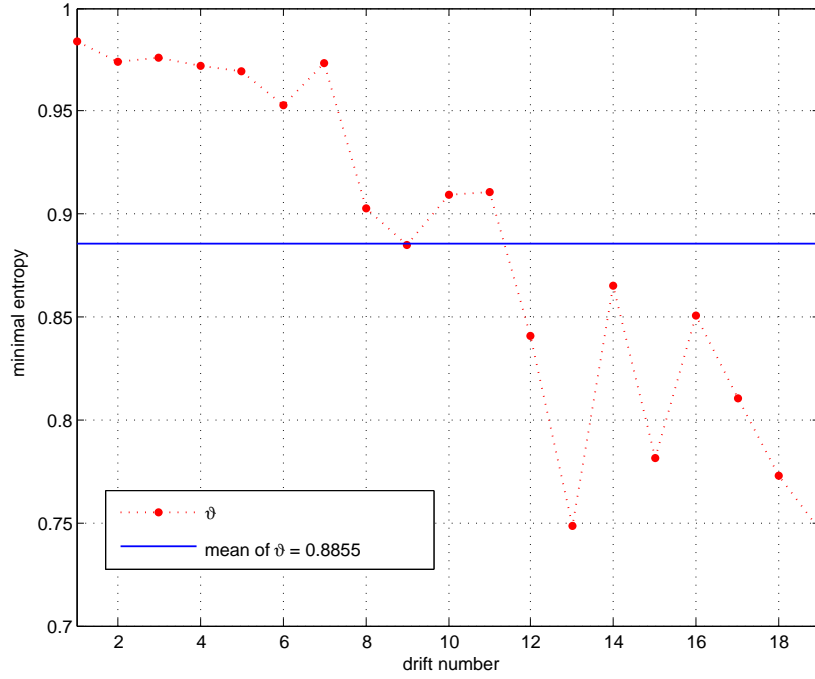
$$\sigma(x) \gg \sigma(x+1) \quad \text{for small } x, \quad (6.14)$$

$$\sigma(x) \approx \sigma(x+1) \quad \text{for large } x. \quad (6.15)$$

Just as well

$$\vartheta(x) \ll \vartheta(x+1) \quad \text{for small } x, \quad (6.16)$$

$$\vartheta(x) \approx \vartheta(x+1) \quad \text{for large } x. \quad (6.17)$$

Figure 6.14: Behaviour of  $\vartheta$  for  $\Delta = 0.1$ 

Hence for Delta  $\Delta > 7.5\%$   $\sigma$  and  $\vartheta$  remain plus-minus constant. As long as it is greater than the critical value of  $\approx 7.5\%$ , therefore it is justified to freely choose a Delta  $\Delta$ . If this conclusion is paired with the demand  $\Delta_{absolute} < 200$ , it results in the choice  $\Delta = 0.1$  with  $\sigma = 0.006$  and  $\vartheta = 0.8855$ . For information purposes, figure 6.14 illustrates the behaviour of  $\theta$  with  $\Delta = 0.1$ . An averaging of  $\theta$  is meaningful as the shape of the curve over all drifts does not change while varying  $\Delta$ .

## Chapter 7

# Static Analysis

In chapter 5, a correlation between accuracy, area under curve and optimal window size at a point in time  $t$  has been derived. In chapter 6, the term of entropy has been adapted to the given problem. At this point, is tried to make a further step and to define a correlation between entropy and optimal window size. Ideally, this would allow to control the window size according to an exactly defined mathematical rule. Consequentially, results would be achieved at the best possible rate. This has already been the keynote of M. Constam's work. He made the first move but did not reach the focused aim entirely. The target correlation does not seem to be as simple as expected, therefore, the involved parts have to be modified by trying to approach the solution. First, Constam's results are recapitulated in section 7.1. His ideas are then further developed in the following chapters.

### 7.1 Introduction

Constam always calculated the entropy with a "simple" form of window size in which the past window reached from instance -1000 up to 0 and the future window from 0 to 1000 (c.p. 6.2). The resulting entropy curve is axially symmetric concerning a vertical straight lines passing through the instance at which the concept drift occurs. So

$$H(d_i - x) = H(d_i + x) \quad \text{in interval} \quad \left[ \frac{d_{i-1} + d_i}{2}, \frac{d_i + d_{i+1}}{2} \right] \quad (7.1)$$

for  $d_i$  = instance at which drift number  $i$  occurs. Unlike this symmetry of the entropy, the curve of the optimal window size does not display such a behaviour. Starting at the minimum after a drift, it increases (almost) linearly during  $ws_{entropy}$  (window size over which the entropy

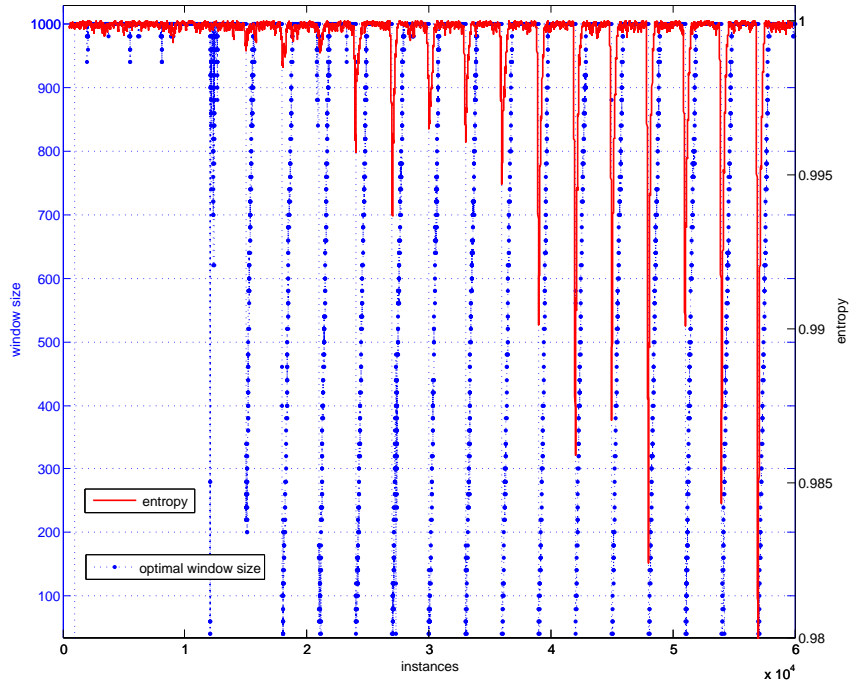


Figure 7.1: Entropy vs optimal window size (plane sphere real continuous knn)(own graphic).

is calculated) instances until the maximum is reached. Thenceforward, it remains at this level until the next drift occurs.

$$ws(d_i - x) = ws_{max} \quad \text{in interval} \quad \left[ \frac{d_{i-1} + d_i}{2}, d_i \right] \quad (7.2)$$

and

$$ws(d_i + x) \approx \frac{ws_{max}}{ws_{entropy}} * (d_i + x) \quad \text{in interval} \quad \left[ d_i, \frac{d_{i+1} + d_i}{2} \right]. \quad (7.3)$$

As this difference in symmetry disrupts the correlation between the two curves, it has to be eliminated. Constam simply did this manually. Knowing the structure of the dataset and the moments of drifts he defined

$$H(d_i - x) = H_{max} \approx 1 \quad \text{in interval} \quad \left[ \frac{d_{i-1} + d_i}{2}, d_i \right]. \quad (7.4)$$

Basically, this is not a acceptable procedure with respect to the final solution. However, it is a warrantable first workaround which will be dealt with in a couple of paragraphs. Figure 7.2

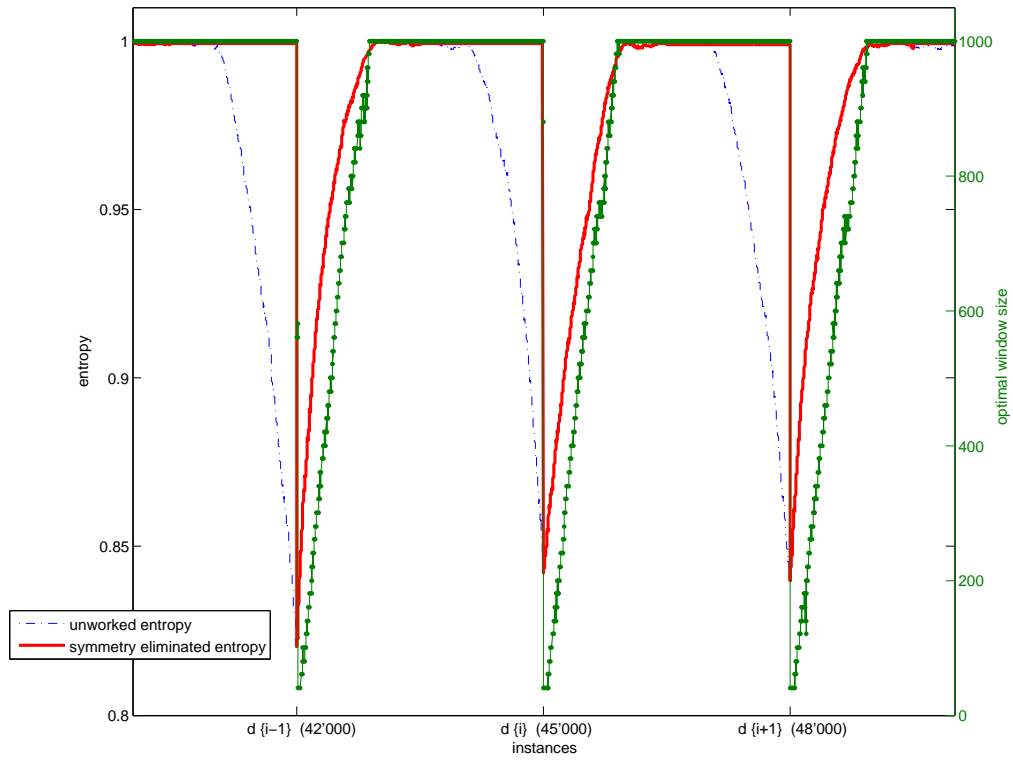


Figure 7.2: Entropy vs optimal window size ( $[-1000; 0; 0; 1000]$ ) (plane sphere real continuous km)

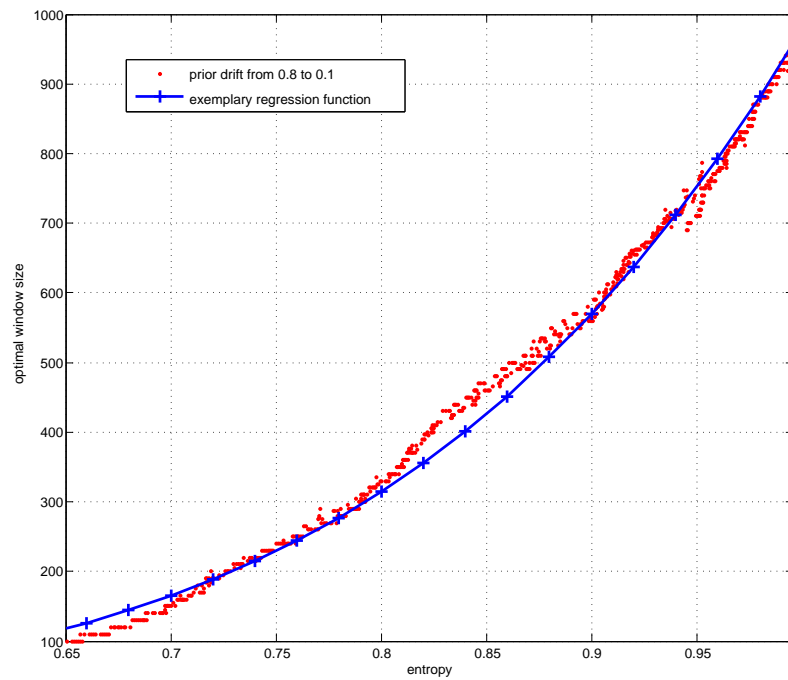


Figure 7.3: Entropy vs optimal window size, one single drift (plane sphere prior discrete nb)



illustrates a first impression of the comparison of the symmetry-eliminated entropy versus the optimal window size. Without looking closely, it can be stated that the two curves in this particular part of the whole data set seem to show quite similar behaviours. Based on this first example, a deeper look into the properties of the correlation will be taken in the further part of the thesis.

## 7.2 Limitations

In the last section, illustrated by figure 7.2, a quite promising picture of the correlation between the entropy and the optimal window size has been drawn. Unfortunately, this optimistic view of the given facts must be revised: Considering a few examples, the deficiencies of the simple “one-to-one” matching model are pointed out. Simultaneously, an attempt to remedy the model’s weak points (if possible at the current store of knowledge) and to evolve a sense for the delicate issues is undertaken.

### 7.2.1 Finding a More Significant Display Format

Using a two dimensional display format which charts instances on the abscissa versus entropy-/window size on one ordinate does not allow a strict mathematical comparison<sup>1</sup>. In order to provide for the implementation of a tool which describes the correlation, the display format is changed. Now, on the abscissa, the entropy  $H$  is plotted versus the window size  $ws$  on the ordinate. Thus a regression line with its coefficient of correlation gives a clean definition of the performance available. Figure 7.3 illustrates the train of thoughts. If it is possible to fit the curve of a regression function into the figure, it is also possible to control the forgetting rate of the incremental algorithm.

### 7.2.2 Foreclosing the Need to Manually Edit the Entropy Curve

This issue is not entirely new. It is, from another point of view, already dealt with in section 6.3. Generally, in section 6.3 it is mentioned in connection with the physical impossibility to look into the future. Now, the same situation is approached from different angle. In chapter 6, a new shape of the entropy curve has been derived from different requirements. Finally, a so called side-by-side arrangement of the entropy’s windows has been chosen. Due to the choice of a large past window and a small future window, the shape of the entropy curve presents itself as illustrated in figure 6.9. In turn, this shape of the curve looks like the one manually

---

<sup>1</sup> Needless to say, in theory this would work. But the regression function  $f_{reg}(x)$  would be extremely difficult to calculate – impossible in a manner of speaking.

worked out which is displayed in figure 7.3. As a consequence thereof, it is no longer necessary to manually edit the entropy curve.

### 7.2.3 Handling of Unequal Reactions on Concept Drifts

This subject has an enormous impact on the entire thesis. In fact, it contains the main reason why Constam did not reach his initial goal and, ultimately, why this thesis has been seized. Recapitulating, the impossibility of combining the entropy – and the optimal window size – curve in one single formula over I. all instances and II. over all different strengths of concept drifts will be discussed. In figure 7.1 the afore mentioned, entire curves are illustrated.

The curves' quite different reaction on varied concept drifts is conspicuous. As already seen in chapter 3, the drifts are getting more intense the greater the number of instances becomes. This fact can easily be read off from the entropy curve. At this stage, it is concentrated on the ratio between the single deflexions and the absolute values of the entropy are neglected. Simplifying, the more intense the drift, the smaller is the entropy. The entropy emerges as a precise instrument to indicate the intensity of a drift. Starting at drift number five or six (located at instance 15'000 and 18'000), the entropy reacts proportionally to the distinctness of a drift. Compared therewith, the window size-curve looks quite different. The first couple of drifts do not seem to have a large impact on it. But from drift number four or five (instances 12'000 and 15'000) onwards, the window size totally collapses every time. No matter how intense the drift actually is, the algorithm reaches the best results by regulating the window size to its minimum and by increasing it from this point on.

At this point, not the cause but the effect of these different reactions on drifts are shown. Therefore, figure 7.3 is upgraded. In its primary form, it only shows one single drift. Enlarging the graph with a few more drifts, a totally different image is presented. Figure 7.4 complies with figure 7.3, up to three additional curves. It must be noticed that the added curves represents the same range of drift, e.g. a drift range of 0.7. The shapes of all curves are similar, even though they are not congruent. As the curves are located within a relatively narrow band of about 100 window sizes, a regression function would be possible<sup>2</sup>.

If the precondition that only drifts of the same strength are considered is cancelled and a look at a “natural” situation, in which all possible ranges of drifts appear, is taken, the correlation vanishes nearly entirely. Figure 7.5 provides a first impression of the correlation's problematic nature. At first glance it resembles a modern painting rather than a mathematical

---

<sup>2</sup> even though the regression line would evince a relatively large coefficient of correlation...

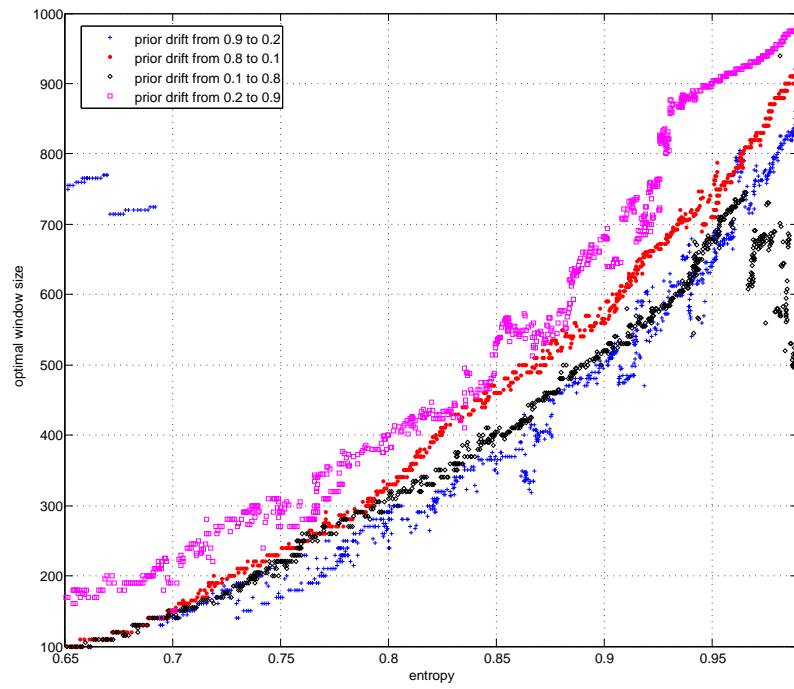


Figure 7.4: Entropy vs optimal window size, several drifts, all the same range (prior discrete nb)

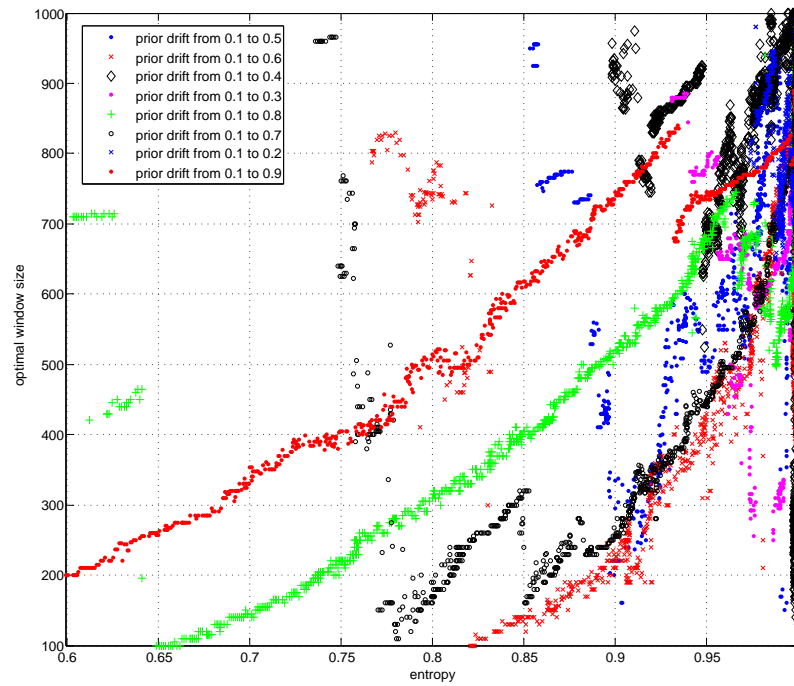


Figure 7.5: Entropy vs optimal window size, several drifts, different ranges (prior discrete nb)

function. It must be pointed out that the graph looks more disrupted than it actually is; the first impression is a bit deceiving. As the basic data is nearly unworked<sup>3</sup>, the ensemble comes across as disjointed and incoherent. Indeed, this fact disturbs the image. However, this is not further problematic, as it can be straightened out. The handling of the curves illustrated by prior drifts 0.1 to 0.9 (red points), 0.1 to 0.8 (green plus signs) and 0.1 to 0.6 (red crosses) is much more challenging. All these curves trend towards window size  $ws_{drift}$  1000 when the entropy  $H(x)$  converges towards 1,

$$ws_{drift}(H(x)) \rightarrow 1000 \quad \text{for} \quad H(x) \rightarrow 1, \quad (7.5)$$

which is comprehensible and serves the initial purpose. The behaviour of the curves for  $H(x) < 1$  is rather different. As the gradients  $\frac{dH(x)}{dx}$  and the shapes of the curves are different, they fan out. The consequence thereof is that a single value of the entropy  $H(x)$  leads to several values of  $ws_{drift}$ .

$$H(i) \rightarrow ws_n(H(i)) \quad \wedge \quad H(i) \rightarrow ws_m(H(i)) \quad (7.6)$$

with  $ws_n(H(i)) \neq ws_m(H(i))$

e.g.  $H(x) = 0.9 \rightarrow ws(0.9) = 220$  for prior drift 0.1 to 0.6

$ws(0.9) = 260$  for prior drift 0.1 to 0.7

$ws(0.9) = 515$  for prior drift 0.1 to 0.8

$ws(0.9) = 730$  for prior drift 0.1 to 0.9

$ws(0.9) = 920$  for prior drift 0.1 to 0.4

This contradicts the fundamental postulation of mathematical functions, as more than one element of the target quantity is assigned to one single element of the definition quantity. If this contradiction would be tried to be avoided by defining a regression line through the points, a massive error in choosing the correct window size would result. A value could be just as well chosen randomly.

Recapitulating, it must be pointed out that the above described phenomenon of the single drifts' fanning out is intrinsic and not solvable by a simple modification of the given parameters.

---

<sup>3</sup> for example outliers are not eliminated and disturb the overall impression rather intensely

---

Based on fundamental different behaviours of the entropy- and the window size-curve in case of concept drifts, the conclusion is that no statistic solution exists. In chapter 8, a simplified solution (by means of a threshold) is presented which works amazingly well.

## Chapter 8

# Semi-dynamic approach

### Annotation

As this chapter the efforts made so far are consolidated. It is the last chapter that actually deals with the semi-dynamic regulation of instantaneous drifts. The chapter after the following chapter already takes a look into future work. Therefore, in this chapter, mainly facts and graphs are presented, most of the corresponding explanations are to be found in the next chapter with the title “Discussion”. All figures in this chapter refer to the real data set and the naïve Bayes algorithm.

### 8.1 Motivation

As it has been proven in chapter 7, a linear, and in manners of speaking, simple control of the optimal window size by the entropy is impractical. The rather different behaviours of the entropy and of the optimal window sizes in the case of concept drifts make a static matching between them impossible. Independent of the nature of the concerned algorithm the entropy reacts too sensitive to changes of the concept. Even if the entropy deflects very little from its normal value of 1, the window size should be adjusted to its minimum. Consequentially, a simple static approach is abolished and, henceforth, it is tried to regulate the system semi-dynamically. As the structure of a totally dynamic connection is not obvious at this time, it is tried to approach the final solution by an intermediate step.

### 8.2 Implementing a Threshold

The basic idea of the semi-dynamic approach is the implementation of a threshold value. Figure 8.1 contains an overview. Behind the concept of introducing such a threshold lies the attempt to “stretch” the entropy, in order to reach its minimum faster and more often.

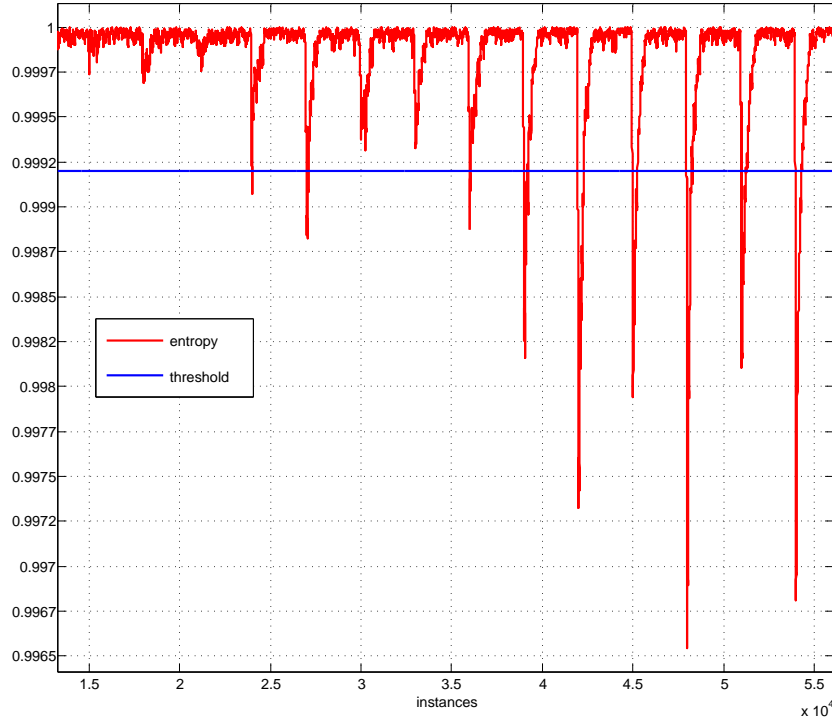


Figure 8.1: Implementing a threshold

Metaphorically speaking, the zero-level is scrolled up from the bottom up. Where exactly the threshold is fixed will be discussed below. In the beginning, the threshold line is treated like a natural zero level.

### 8.3 First Form of Switching and Linear Learning

Regarding the control system of the window size, a quite simple rule is established. Basically, the window size is maximal, namely 1000. If the entropy declines under the threshold  $\theta$  at the instance  $i$ , the window size will be adjusted to its minimum. Therefore, the window size will linearly increase by 1 at every following instance:

$$\begin{aligned}
 ws(j) &= ws(j-1) + 1 \quad \text{with} \\
 ws(j=i) &= 0 \quad \text{and} \\
 H(i) &\leq \theta < H(i-1).
 \end{aligned}$$

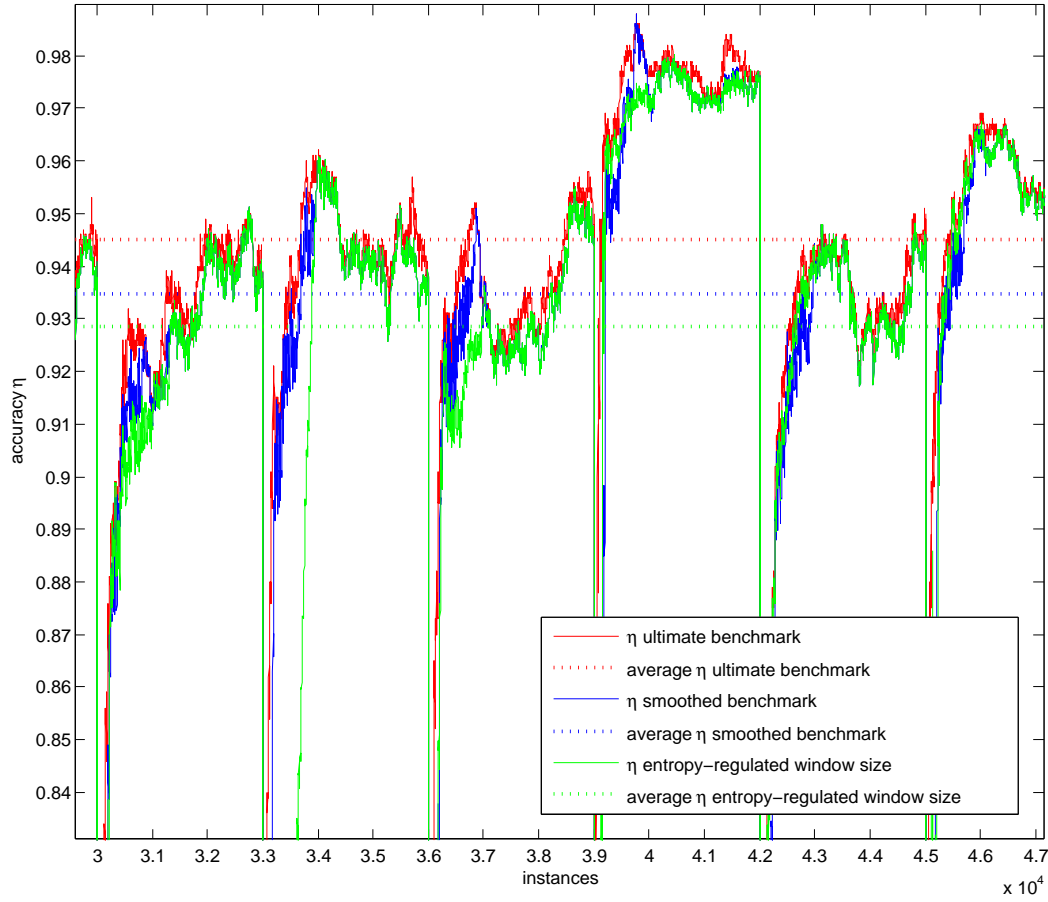


Figure 8.2: Entropy-regulated accuracy vs benchmarks (sphere real nb)

The title of the subsection has been chosen based on the above explanation. The threshold  $\theta$  switches the window sizes' behaviour and the classifier builds its rules on a linear increasing number of instances. From a geometrical point of view, the window size increases linearly with an angle of  $45^\circ$ . Even though a more simple approach can hardly be imagined, astonishingly good results can be achieved. Figure 8.2 displays the obtained accuracy  $\eta$  in comparison with the benchmarks introduced in chapter 5. The dashed lines in the graph correspond to the average accuracy  $\bar{\eta}$  and reveal the quality of the regulation. For the purpose of a better survey, only a section of the whole data set is displayed, showing mid-intense drifts. The regulation of the window sizes is based on the entropy of a  $[-200; -100; -100; 0]$ -window, a threshold  $\theta = 0.93$  and, as mentioned before, an angle of inclination  $\psi = 45^\circ$  (cp. figure 8.5). A closer look at the scaling of the variables will be taken in section 8.4.

Primarily, it can be noticed that the entropy-based regulation of the window sizes leads to a poorer performance than the benchmarks. Numerically,  $\bar{\eta}$  ultimate benchmark = 0.9452,  $\bar{\eta}$  smoothed benchmark = 0.9347 and  $\bar{\eta}$  entropy-regulated window size = 0.9286. So, on aver-



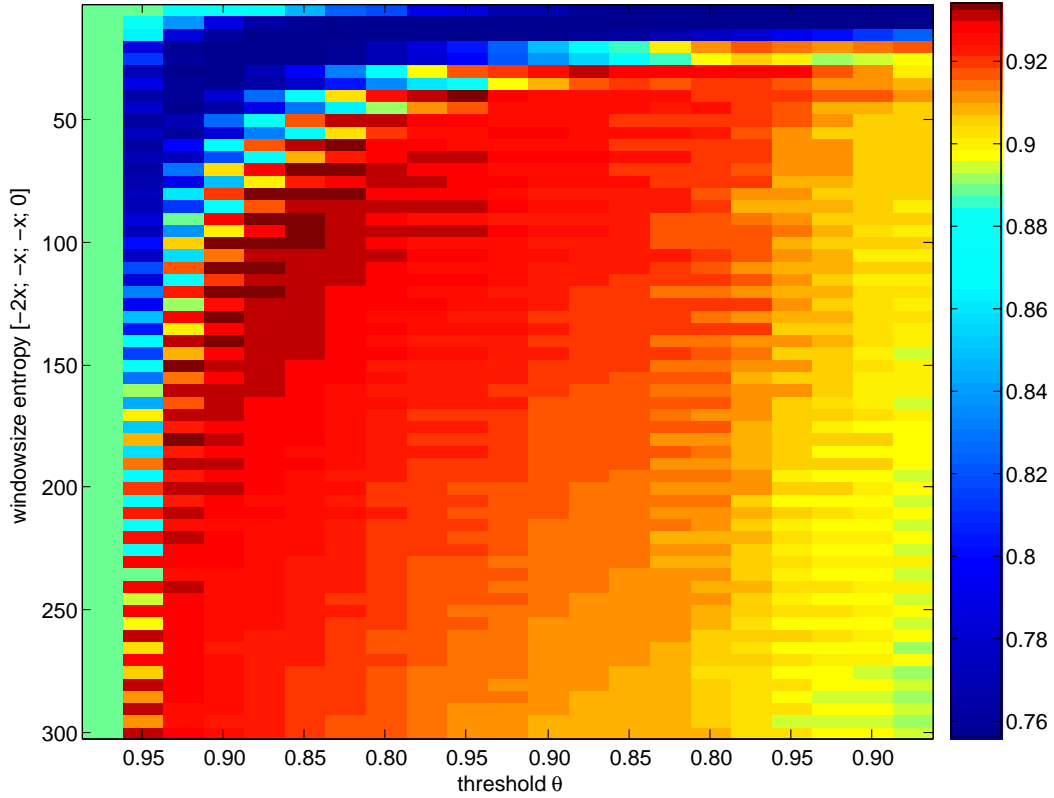


Figure 8.3: Heatmap  $\psi=45^\circ$ ,  $ws$  of  $H$  and  $\theta$  variable, colouring according to  $\bar{\eta}$

age, 6.1‰ on the smoothed benchmark are lost. In addition, by having a closer look at the drift at instance  $3.3 * 10^3$ , as the accuracy  $\eta$  lags behind the benchmarks<sup>1</sup>, the regulation rule seems to have missed out the drift. Obviously, the threshold  $\theta$  was chosen disadvantageously, a greater value would have performed better. This argument leads directly to the next section where the variables  $ws$ ,  $\psi$  and  $\theta$ , which have not been determined yet, will be discussed.

## 8.4 Fixing of the Variables

In section 8.2, relatively good results have been achieved by using a simple, but promising approach of regulating the window size  $ws$  semi-dynamically. This promising will be enhanced and further developed. First of all, for this purpose, all the variables values' which have been freely chosen up to this point must be fixed. A so called heatmap has been calculated therefore. It illustrates the performance of the variables by using an example. For this purpose, the data set simulating real drifts and the naïve Bayes algorithm have been chosen.

<sup>1</sup> As the window size have not been switched, the algorithm slid over the drift with window size 1000. As been explained in chapter 5 this leads to bad performance.

### 8.4.1 Fixing $\psi$ , Regulating $ws$ of $H$ and $\theta$

The heatmap in figure 8.3 illustrates the behaviour of the average accuracy

$$\bar{\eta} = \frac{1}{n} \sum_{i=1}^n \eta(i) \quad (8.1)$$

with  $n$  = total number of instances. Angle  $\psi$  is fixed at  $45^\circ$ . The colouring of the heatmap is according to  $\bar{\eta}$ .

Angle  $\psi=45^\circ$  has been introduced as a geometrical view of increasing the entropy's window size according to the number of instances after the threshold  $\theta$  has been reached. However,  $\psi$  could be treated uncoupled, as self-contained variable. In such a case,  $\psi$  defines how fast the window size will reach its maximum again. The extreme value  $\psi=90^\circ$  means that the window size will be regulated back to 1000 within one instance after the drift. The other extrema,  $\psi=0^\circ$ , will keep the window size on its minimum for the rest of the data set after the first drift. In the next subsection,  $\psi$  will be varied.

### 8.4.2 Fixing $ws$ of $H$ , Regulating $\psi$ and $\theta$

In order to obtain a comprehensive perspective, the variables and the fixed values are exchanged. Figure 8.4 illustrates a heatmap in which the entropy's window size is fixed at  $[-200; -100; -100; 0]$ . Angle  $\psi$  and threshold  $\theta$  are varied.

## 8.5 Second Form of Switching and Linear Learning

In section 8.3, a very simple switching strategy has been pursued. The basic idea was to regulate the window size as soon as the entropy  $H$  declines under the threshold  $\theta$ . It can be discussed, whether this kind of switching makes sense in view of the theory behind it. At the most, this strategy might be overfitted to the given set up. It seems possible that, it would be more plausible to regulate the window size after the entropy has exceeded the threshold  $\theta$ . Figure 8.5 illustrates the train of thoughts.

$$\text{window size } ws \text{ at instance } j \quad ws_j = |(\text{instance}_j - \text{instance}_i)| * \tan(\psi) \quad (8.2)$$

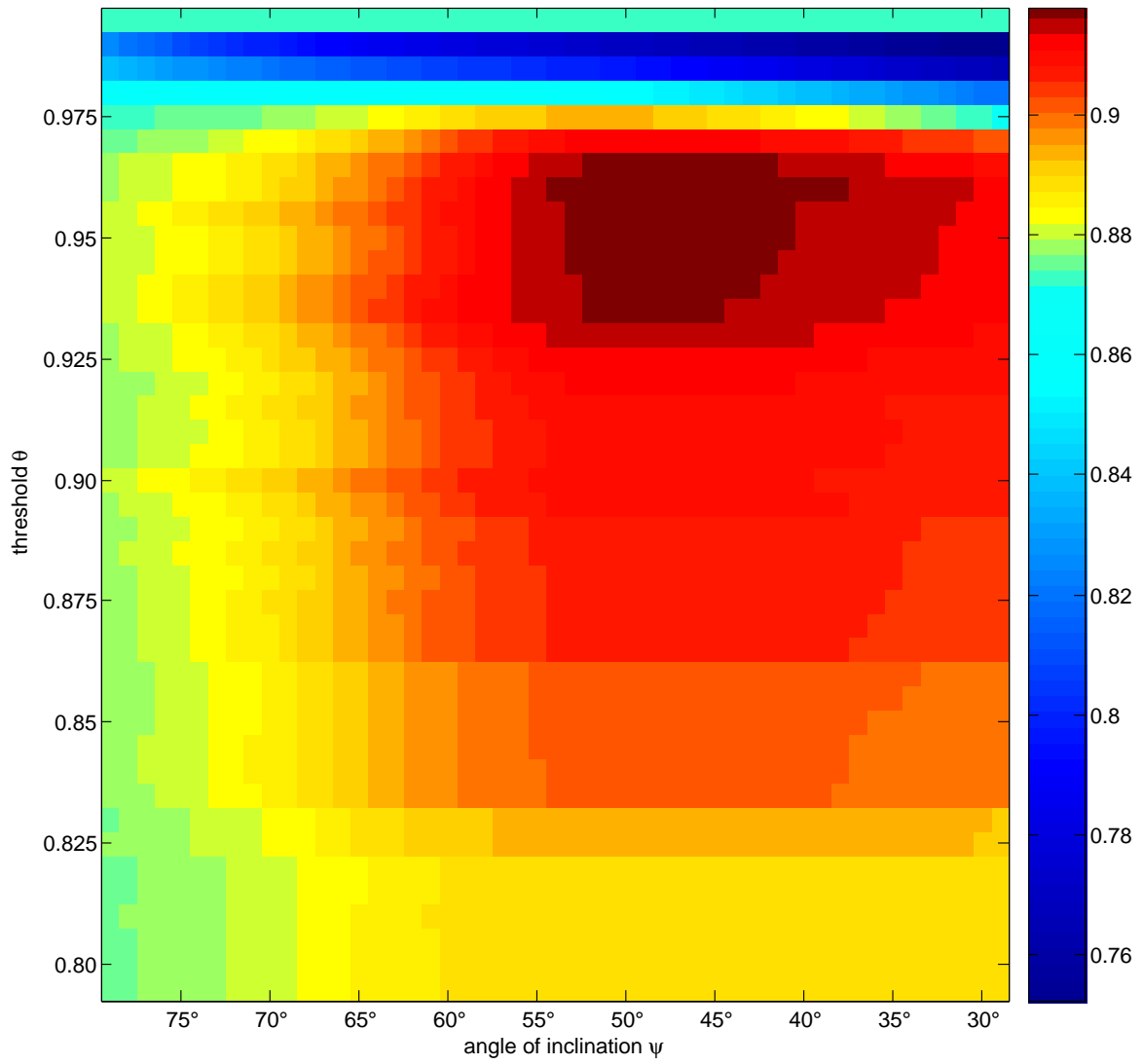


Figure 8.4: Heatmap  $ws$  of  $H$  with  $[-200; -100; -100; 0]$ ,  $\psi$  and  $\theta$  variable, colouring according to  $\bar{\eta}$

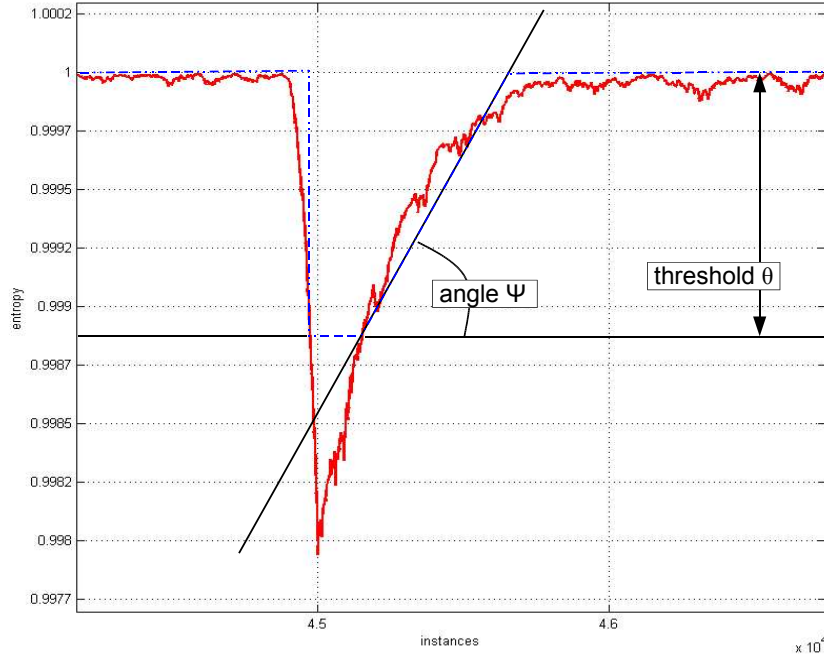


Figure 8.5: threshold based control of the window size

with  $\text{instance}_i$  = the instance where the entropy  $H$  exceeds  $\theta$  again. Chapter 9 contains the respective discussion.

## 8.6 Third Form of Switching and Linear Learning

The third form focuses on the word “linear” in the title of this section. Up to now, it described a linear recovery of the entropy’s window size, either from the point where the entropy falls below the threshold  $\theta$  or the point where  $\theta$  is exceeded. This linearity shall now be directly applied to the matching function entropy  $H$  – window size  $ws$ . Some definitions must be made therefore:

$$ws_{max} \doteq 1000, \quad (8.3)$$

$$ws_{min} \doteq 10, \quad (8.4)$$

$$H_{max} \doteq 1, \quad (8.5)$$

$$H_{min} \doteq \theta. \quad (8.6)$$

In a second step, a simple linear dependency is defined:

$$ws = (H - H_{min}) * \left( \frac{ws_{max} - ws_{min}}{H_{max} - H_{min}} \right) + ws_{min} \quad \text{for } H \geq H_{min} \quad \text{and} \quad (8.7)$$

$$ws = ws_{min} \quad \text{for } H \leq H_{min} \quad (8.8)$$

At a first glance, this third kind of switching and linear learning looks “the best”. The rule is not as inflexible as the rules of the first two approaches. Also, several crossings<sup>2</sup> of the entropy curve with the threshold line would not cause a back-regulation of the window size to its minimum and the therewith related slow recovery phase each time. At this point, no numerical values are presented yet. (See chapter 9.)

## 8.7 Performance of the Regulation

In section 8.3, the idea of a simple switching mechanism has been introduced. Considering an example with randomly chosen variables, the performance of the regulation has been compared with the benchmarks, see figure 8.2. In the course of the next steps, this variables have been varied, and the results have been displayed in form of heatmaps. These heatmaps clearly show how stable the regulation is. Therefore it is not very remarkable that the accuracy of the regulated window size performed very well compared to the benchmarks. As the regulation does not strongly depend on the choice of the variables, it can not be spoken of a “fitting” to the given problem. Therefore, it is possible to compare this solution – the regulation by switching and linear learning – with common ways of controlling incremental algorithms. In chapter 5, such an established solution, namely the committee, has been introduced. In figure 8.6, the result of this comparison is illustrated. The variables  $\psi$  and  $\theta$  have been chosen according to figure 8.4,  $\psi$  and  $\theta$  are located in the ranges of

$$35^\circ < \psi < 55^\circ$$

$$0.925 < \theta < 0.975.$$

With this set up, the smoothed benchmark has (almost) been reached. The difference of the average accuracies amounts to only  $\approx 1\%$ . Compared therewith, the average accuracy  $\bar{\eta}$  of the regulated window size is  $\approx 1\%$  greater than the committee #3 and  $\approx 1.5\%$  greater than the

---

<sup>2</sup> For example by accident, as the system is noisy or as the threshold  $\theta$  has been chosen at a disadvantageous value.

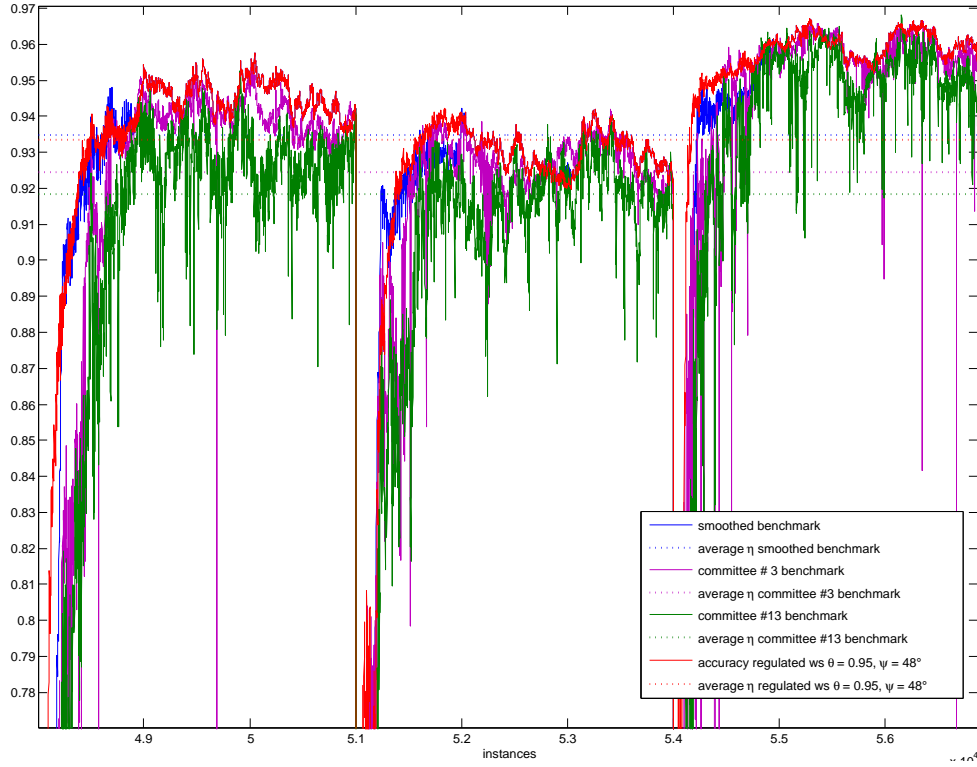


Figure 8.6: Accuracies regulated  $ws$  vs best committee benchmarks (prior discrete nb)

committee #13.

### 8.7.1 Determining Entropy $H$ and Threshold $\theta$

The switching and linear learning starts as soon as the entropy  $H$  falls below the threshold  $\theta$ . Therefore, only the declining edge of the entropy curve acts as switching instrument. The shape of the declining edge is determined only by the length of the entropy's future window. In chapter 6, the maximum length has been defined. As the future window  $ws$  of the entropy has to range in narrow band of

$$0 < ws < 200, \quad (8.9)$$

the shape of the declining edge remains very similar.

Threshold  $\theta$  is coupled to the entropy's declining edge. Their intersection point defines the beginning of the linear learning. On the one hand, these conditions prevent a free choice of  $H$  and  $\theta$ . On the other hand, there is no need to vary these variables and therefore no possibility

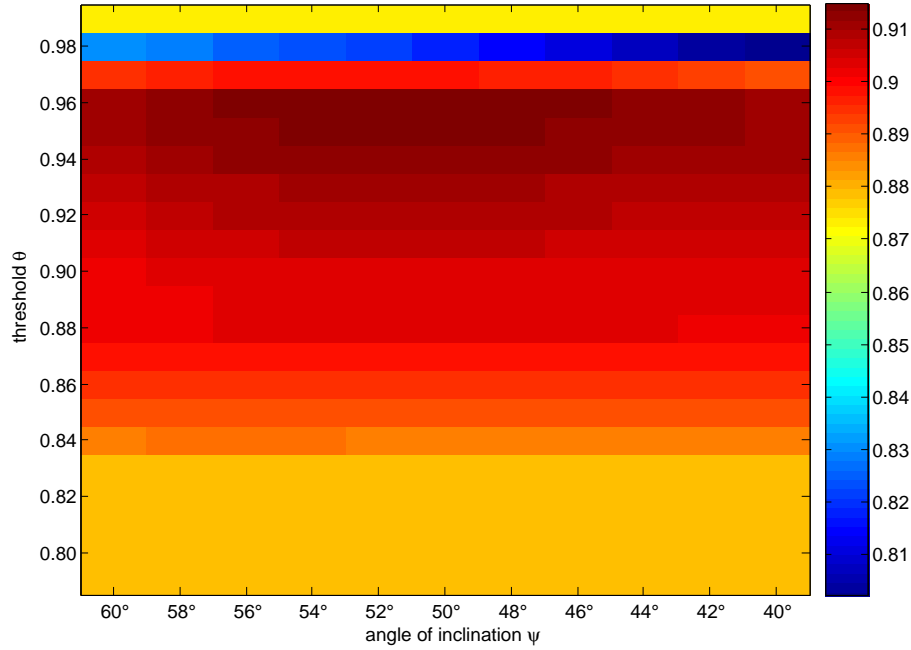


Figure 8.7: Heatmap 10% noise  $ws$  of  $H$  fixed  $[-300; -150; -150; 0]$ ,  $\psi$  and  $\theta$  variable, colouring according to  $\bar{\eta}$

to make mistakes. As long as  $\theta \approx 0.95\%$  of the length of the future window and the future window equals approximately 100, the system remains stable.

## 8.8 Noise Resistance of the System

Based on the information provided by the above presented heatmaps, a strong reaction of the system to noise is not to be expected. The semi-dynamic regulation provides equally good results in a wide range of the variables. This fact is an indicator for a stable system. In order to prove this assumption, noisy data sets have been generated. All in all, sets with 1%, 5%, 10%, 20%, 50% and 100% noise are existing. 100% means that the class of every other data set instance has been changed randomly. This is the maximal amount of noise that can be introduced to a system. Needless to say, such a high percentage of noise does not make any sense from a realistic point of view. Therefore, only a 10% heatmap has been calculated to give a basic idea of the system's behaviour. All calculations refer to the data set simulation real drifts and to the naïve Bayes algorithm. Figure 8.7 displays this heatmap. Due to the noise the region with high  $\bar{\eta}$  is smaller, but the system does not collapse at all.

Recapitulating, the approach of regulating the algorithm's window size which has been described in this chapter is strongly noise-resistant.

## 8.9 Overfitting the Solution to the Problem?

So far, all presented solutions have been based on the average accuracy  $\bar{\eta}$ , respectively, on the average area under curve  $\overline{AUC}$ . It can be argued that, averaging these values is not an ideal measure of performance as all possible states<sup>3</sup> of the system are summed up and divided. Therefore, so the criticism, as the presented approach shall be fitted exactly to the arrangement of the “drifting”/“normal” parts of the data set, the presented approach shall be overfitted to the given data set.

To invalidate this argument, the introduced heatmaps above are split up. Every part of the data set will be analysed individually. Therefore, three possible “regions” of the data set are determined.

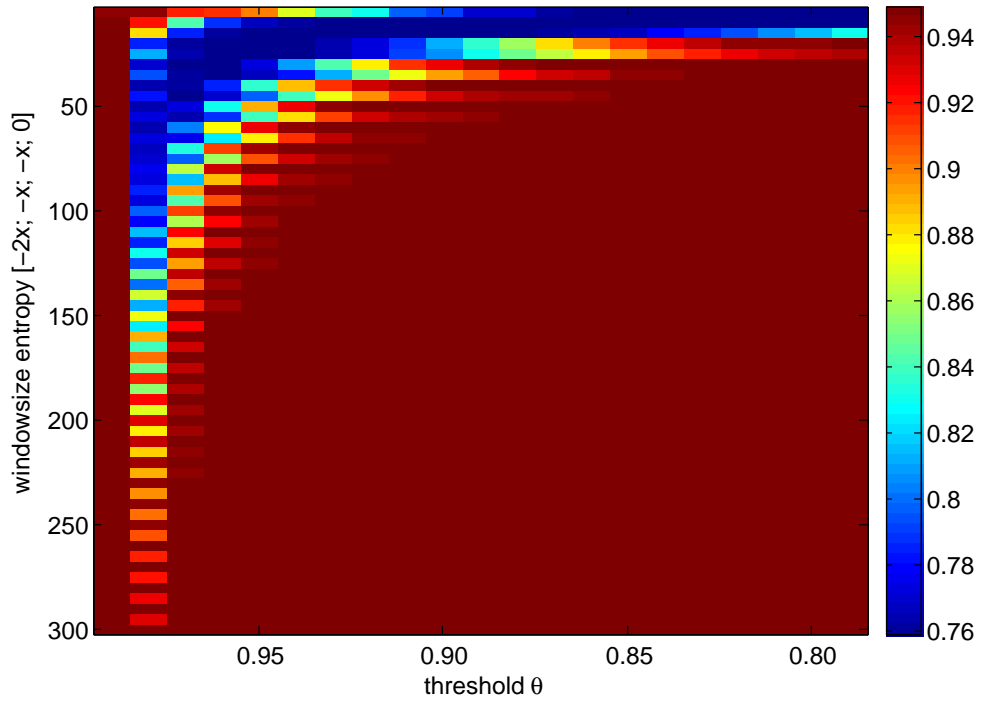
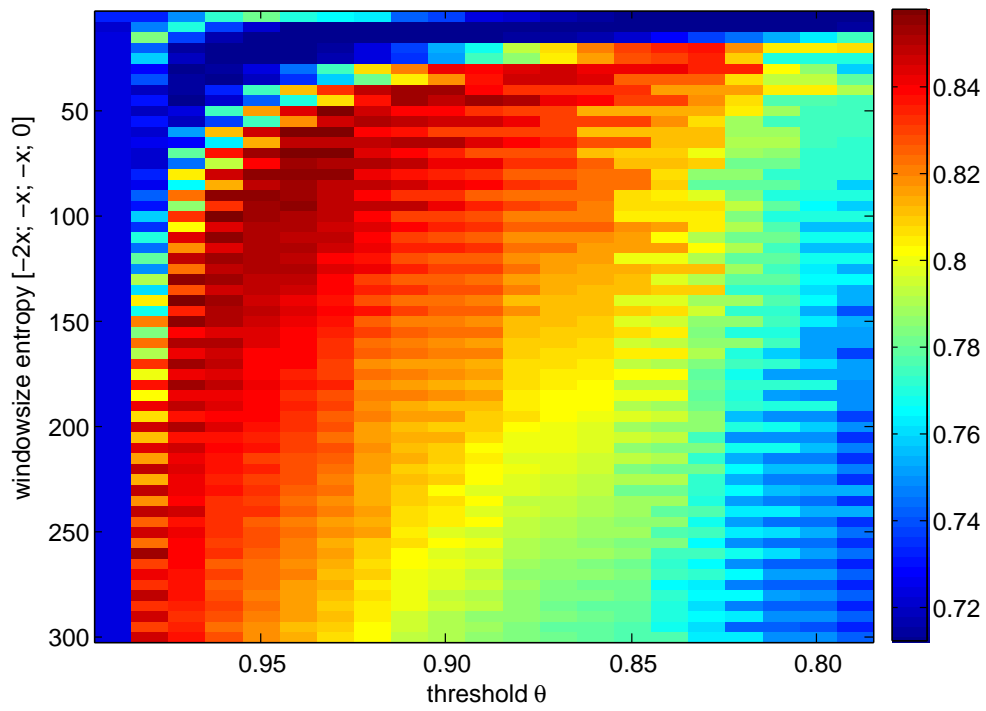
1. Regions of upcoming drifts  $\mathcal{A}$  which contains  
the instances  $i \in [(\text{drift } n + 2000), \dots, (\text{drift } n + 2999)]$ .
2. Drifting regions  $\mathcal{B}$  which contains  
the instances  $i \in [(\text{drift } n), \dots, (\text{drift } n + 999)]$ .
3. Regions between two drifts  $\mathcal{C}$  which contains  
the instances  $i \in [(\text{drift } n + 1000), \dots, (\text{drift } n + 1999)]$ .

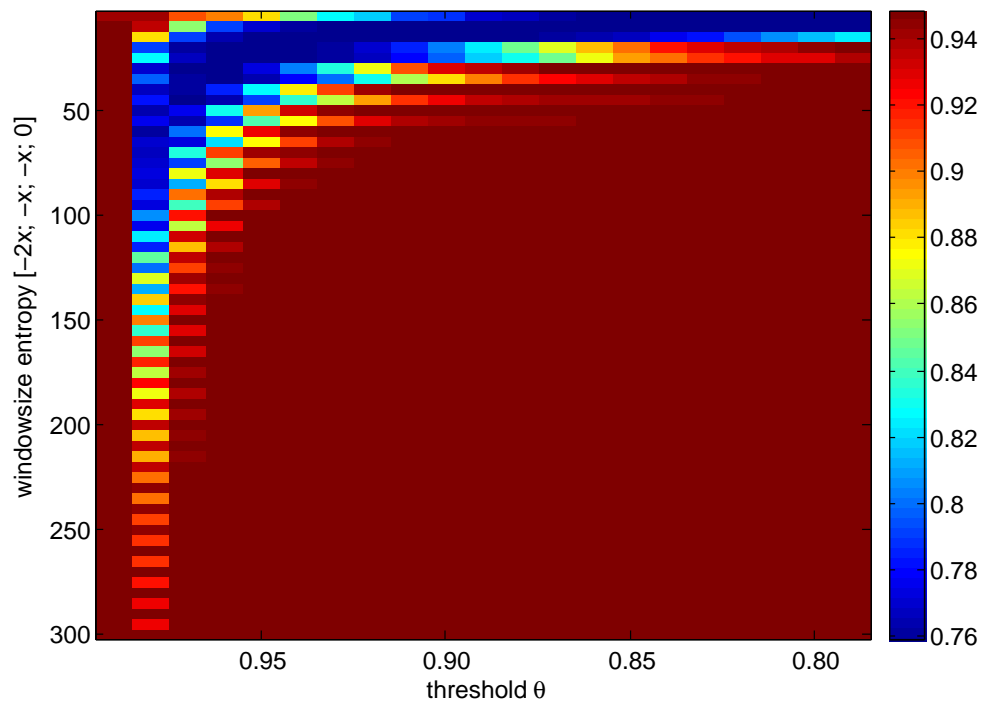
Consequently,  $|\mathcal{A}| + |\mathcal{B}| + |\mathcal{C}| = 3000$  and  $n = 1, \dots, 19$  in the case of real drifts. The average accuracy  $\bar{\eta}$  is calculated region-wise. Figures 8.8, 8.9 and 8.10 illustrate the results. It must be noted that due to the colour-resolution, figures 8.8 and 8.10 look exactly the same. In fact, their numerical values are slightly different. Basically, the structure of figure 8.9 is similar to the structure of figure 8.3. Both figures 8.9 and 8.10 display a very high average accuracy  $\bar{\eta}$ , as the regions  $\mathcal{A}$  and  $\mathcal{C}$  contain instances of stable parts of the data set. Therefore, only region  $\mathcal{B}$  is of note. The use of  $\bar{\eta}$  as a quality measure is justified as the distribution of  $\bar{\eta}(\mathcal{B})$  is similar to the distribution of  $\bar{\eta}(\text{entire data set})$ . Therefore, an overfitting does not take place.

---

<sup>3</sup> It is referred to the “normal” and the “drifting” part of the data set. See also section 6.5, especially figure 6.12 and the corresponding explanations



Figure 8.8: Heatmap region  $\mathcal{A}$   $\psi=45^\circ$ ,  $ws$  of  $H$  and  $\theta$  variableFigure 8.9: Heatmap region  $\mathcal{B}$   $\psi=45^\circ$ ,  $ws$  of  $H$  and  $\theta$  variable

Figure 8.10: Heatmap region  $\mathcal{C}$   $\psi=45^\circ$ ,  $ws$  of  $H$  and  $\theta$  variable

## Chapter 9

# Discussion

### 9.1 Introduction

In this chapter the results presented in chapter 8 will be discussed. Primarily, it is referred to the sections 8.3, 8.5 and 8.6, beginning with the latter. In this a linear relation between entropy  $H$  and window size  $ws$ , using a threshold value  $\theta$ , has been introduced. This section has been kept short deliberately and no numerical values have been presented. The reason therefor is the very close connection between this approach and the original idea of regulating the algorithm by a static matching. Introducing a threshold  $\theta$  does change the appearance of the situation, but it does not solve the approaches' inherent weakness. Assuming  $\theta = 0$ , a step back is taken to chapter 7. As already deduced and explained in this chapter, the approach does not lead to a usefull solution. Recapitulating, the entropies' and the window sizes' behaviours impede a simple one-to-one matching.

The motivation for section 8.5, see figure 8.5, is similarly to the one contained in explanations concerning the bin separation for the entropy calculation (section 6.2). The regulation of the entropy, beginning at the instance, where  $H$  exceeds  $\theta$  has been envisaged in order to handle continuous drifts. Unfortunately, no figures of the entropys' behaviour in case of continuous drifts are available, as they have never been simulated for the purpose of this thesis. Therefore, a hand-made sketch is presented in order to clarify the situation. The situation is illustrated in figure 9.1. In case of a continuous drift (red line), the entropy remains below the threshold  $\theta$  for a long period of time. If the window size would be regulated as explained in section 8.3, the window size would be increased from point A onwards. After a couple of instances, the maximum would be exceeded regardless of the fact that it should be held on its minimum. Needless to say, the accuracy  $\eta$  would be accordingly low. By regulating the window sizes according to section 8.5, this would not occur. Until point C, the window size would not have

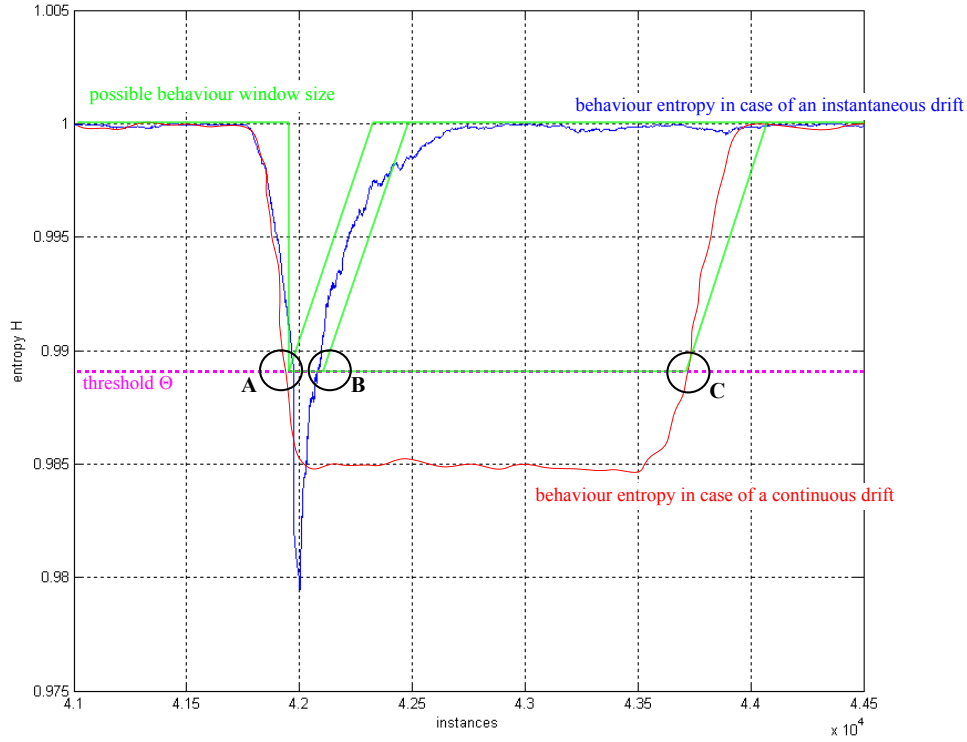


Figure 9.1: Overview entropy behaviour

increased in case of a continuous drift. In case of an instantaneous drift, the window size would have increased from point B on.

## 9.2 Main Discussion

In this section, mainly section 8.3 of chapter 8 will be discussed. Beginning with figure 8.6 in section 8.7, it can be stated that the initial aim has mainly been achieved. The smoothed benchmark has almost been reached and, the committee benchmarks have excelled. The envisaged kind of regulation is stable with respect to its variables and noise-resistant. Different heatmaps, as to be seen in chapter 8, document the stability of the regulation system. An overfitting to the given problem statement has not been undertaken as explained in 8.9 and, the choice of the values for the entropy  $H$  and the threshold  $\theta$  is unproblematic as described in section 8.7.1.

The main reason for the performance of this simple approach is illustrated in figure 9.3. For presentation purpose, only a part of the dataset is displayed. Figure 9.2 displays a wider part of the data set in order to demonstrate that the drift displayed in figure 9.3 is not a special case and only shows a common behaviour.

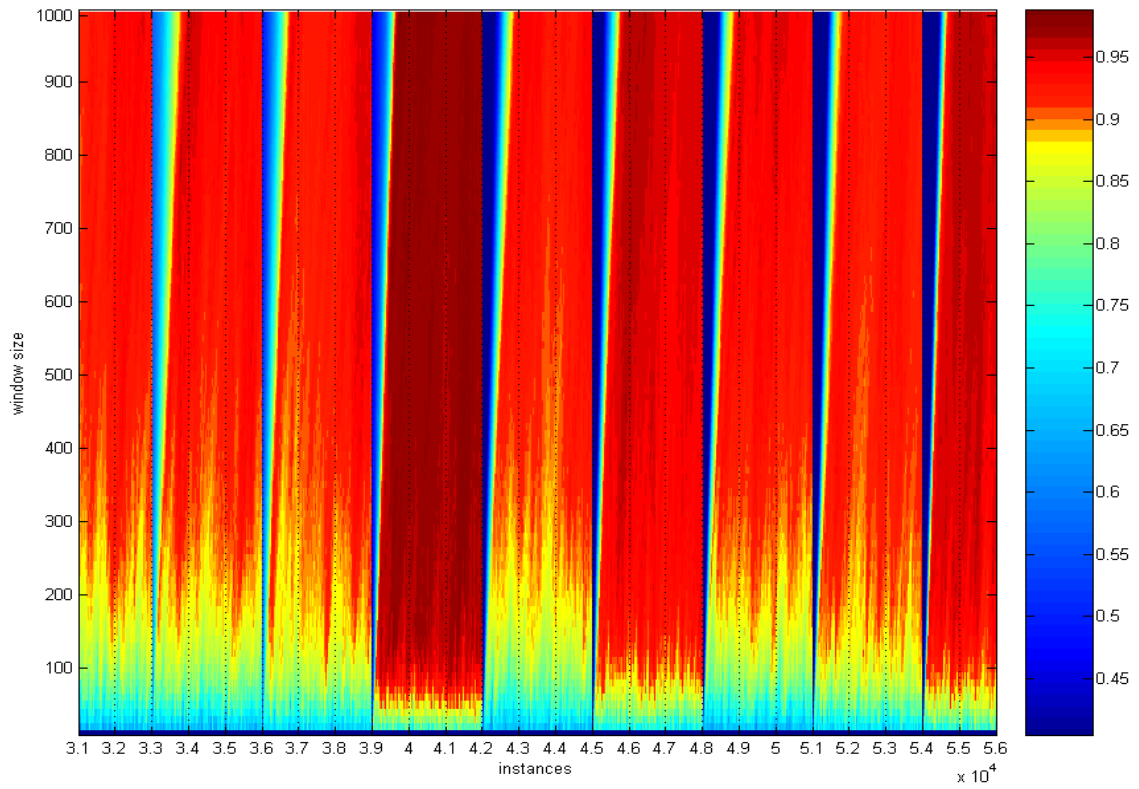


Figure 9.2: Heatmap accuracy, window sizes vs instances (sphere real nb), overview

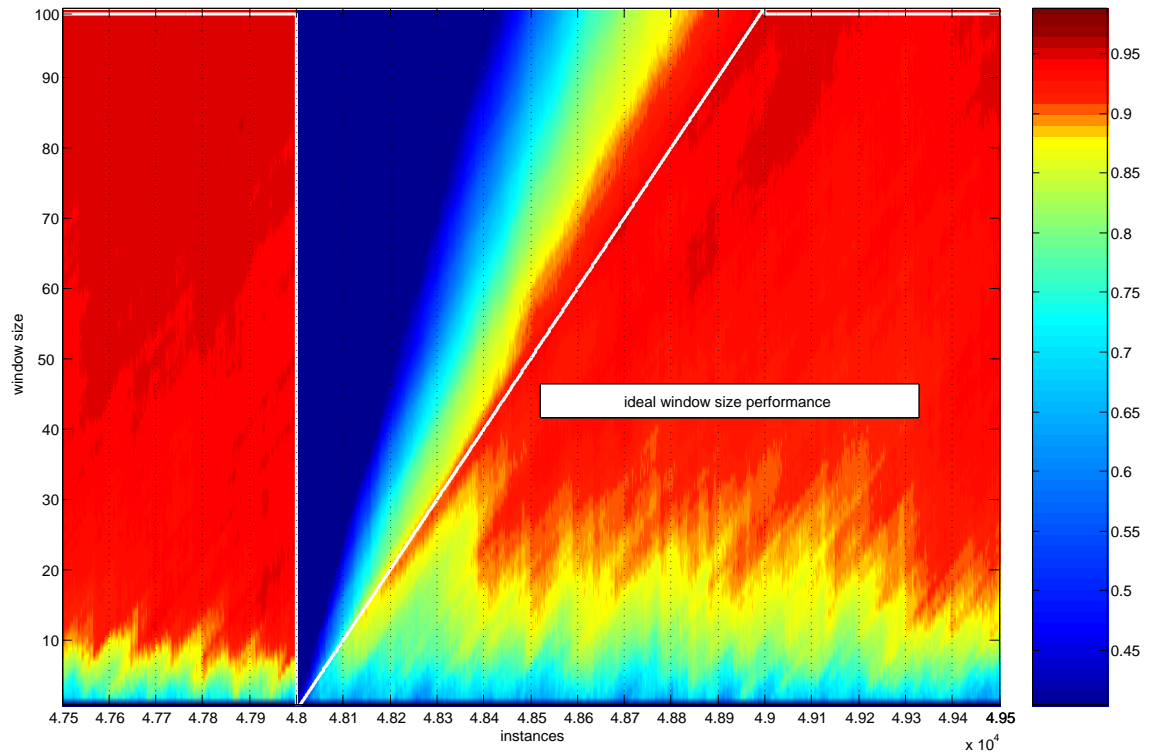


Figure 9.3: Heatmap accuracy, window sizes vs instances (sphere real nb)

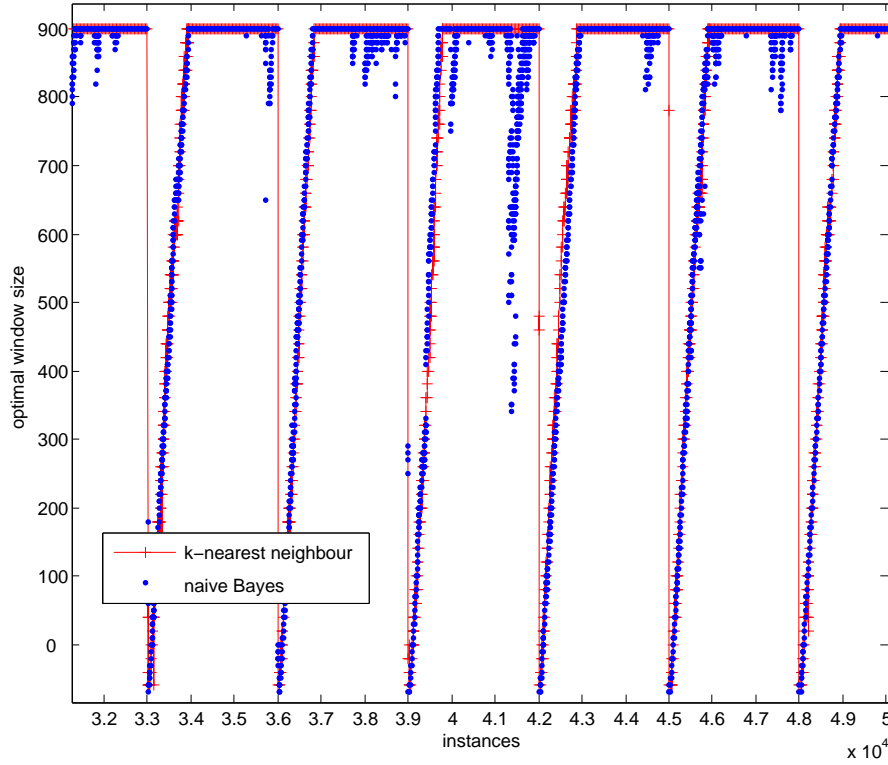


Figure 9.4: Optimal window sizes naïve Bayes and  $k$ -nearest neighbour algorithm

Due to the very linear behaviour of the system in the case of concept drifts, the regulation according to section 8.3 works well. It provides a high average  $\bar{\eta}$ , shortly after the drift as well as in stable phases of the continuous data stream.

### 9.3 Transferability to Other Scenarios

Due to limited computing capacity, in the last two chapters, only the real data set and the naïve Bayes algorithm have been used. In future work, the prior data set and the  $k$ -nearest neighbour algorithm will be involved more frequently in order to support the theory by more fundamentals. However, due to the algorithms' similar natures, no deviating results are expected. Figure 9.4 provides an illustration of the behaviour of the optimal window sizes<sup>1</sup> of the naïve Bayes and the  $k$ -nearest neighbour algorithm. As the  $k$ -nearest neighbour algorithm is less noisy, even better results can be expected from the transfer to this scenario.

<sup>1</sup> In a 1.5%-neighbourhood according to chapter 5.

## Chapter 10

# Future prospects

At the beginning of this thesis, the prospects of success were unpredictable. After a couple of months, the first promising results have been achieved and, therefore, the original strategy has been pursued. Finally, as presented in chapters 8 and 9, an important intermediate aim has been reached. As this “milestone” was reached, the decision was made to advance the approach. Therefore, the future prospect summed up in this chapter does not consist in theoretical thoughts, it is a but prearranged plan. The final aim is to accurately control the window size in a fully dynamic way in each possible case<sup>1</sup> and to write out the underlying theory. In order to achieve these aims, the following steps must be taken.

1. Testing the result of chapter 8 on a mixed data set. “Mixed” means a combination of real and virtual drifts.
2. Testing the result of chapter 8 with different algorithms. It is possible that other algorithms react differently on concept drifts than the naïve Bayes and  $k$ -nearest neighbour algorithms.
3. Finding a possibility of simulating continuous drifts and testing the approaches in case of such drifts.
4. Making the regulation fully dynamic. A possible approach is that the window size of the entropy is dynamically adjusted. First tests have been made which are not presented in this thesis. The reason is that this thesis focusses on the semi-dynamic regulation. The introduction of a half-finished, fully dynamic approach would have been confusing.

---

<sup>1</sup> E.g. in case of different algorithms, different kinds of drifts, different data sets and so forth.

5. Developing of a theory. The regulation by the entropy can be understood as a controlling by the change of the information content. In, e.g. the oscillation equation

$$m\ddot{x} + b\dot{x} + cx = F(t), \quad (10.1)$$

in nature, a lot of laws accord to differential equations. Potentially , a change of information content can be treated as  $H(\dot{x})$  and fitted to a similar differential equation.



## Chapter 11

# Conclusion

This thesis addresses the entropy-based, semi-dynamic regulation of incremental algorithms in case of instantaneous concept drifts. In order to achieve such a regulation, some prepository work was necessary. First of all, a synthetic data set has been generated. This data set belongs to a continuous stream of instances. These instances refer to points in the three-dimensional Cartesian coordinate-system which are accidentally distributed within a sphere. This sphere is divided into two hemispheres by a plane. Thus two classes are defined. By rotating the plane, so called real concept drifts are simulated. Real concept drifts describe the shifting between different concepts within a data set. A second form of concept drifts, virtual drifts, do not describe a real shifting between the concepts but a varying of the class distributions' prior.

In order to predict the instances' classes based on their parameter values, the concepts have to be learned by the predicting classifier, also known as incremental algorithm. In this thesis, the naïve Bayes and the  $k$ -nearest neighbour algorithms have been used. Learning, from the algorithms point of view, means to build classifying rules on a predefined number of instances. This number is determined by the algorithms' window size. In a normal state of the system, the algorithms' prediction is the better the greater the number of learned instances is. In case of concept drifts, this rule does no longer apply as the old instead of the new concept would be learned. Consequently, a shortening of the window size would be ideal. The main problem is to predict such drifts and therefore to regulate the algorithm's window size. In order to know which window size would be ideal for each instance, so called benchmarks have been introduced. For each instance, the benchmark represents the ideal window size. Every window size is directly coupled with a measure of performance. The more ideal the window size, the greater the measure. In this thesis, two measures are used, the accuracy and the area under curve.

Based on the entropy-term introduced by Shannon, a special kind of entropy has been

developed. It is able to measure the information content of the parameter stream and therefore to detect concept drifts. In a first step, it was tried to couple this entropy directly with the window size. It was then shown that a simple matching is not possible. The reason therefor are the different behaviours of the entropy and the optimal window size in the case of a concept drift. Therefore, this approach was rejected and a second approach has been introduced. This approach allows for the different behaviours. Using a simple, semi-dynamic switching and linear learning strategy, the regulation provides astonishingly good results close to the benchmark. Comparable algorithms such as committees, which need more computing capacity, provide worse results.

The entropy and therefore the regulation strategy is fast, noise-resistant and simple to implement. In future work, this strategy shall be I. tested on other data sets and algorithms, II. enhanced in order to detect continuous concept drifts and III. made fully dynamic so that it is as adaptive as possible.

## Appendix A

## Appendix

drift no.	1	2	3	4	5	6	7	8	9	10
$\varphi$ [rad]	$\frac{\pi}{128}$	$\frac{\pi}{64}$	$\frac{\pi}{32}$	$\frac{\pi}{16}$	$\frac{\pi}{8}$	$\frac{3\pi}{16}$	$\frac{\pi}{4}$	$\frac{5\pi}{16}$	$\frac{3\pi}{8}$	$\frac{7\pi}{16}$
$\varphi$ [deg]	1.41	2.81	5.62	11.25	22.50	33.75	45.00	56.25	67.50	78.75
drift no.	11	12	13	14	15	16	17	18	19	
$\varphi$ [rad]	$\frac{\pi}{2}$	$\frac{9\pi}{16}$	$\frac{5\pi}{8}$	$\frac{11\pi}{16}$	$\frac{3\pi}{4}$	$\frac{13\pi}{16}$	$\frac{7\pi}{8}$	$\frac{15\pi}{16}$	$\pi$	
$\varphi$ [deg]	90.00	101.25	112.50	123.75	135.00	146.25	157.50	168.75	180	

Table A.1: Values of real drifts

drift no	1	2	3	4	5	6	7	8	
instance prior $\Delta$ prior	3000	6000	9000	12000	15000	18000	21000	24000	
	$0.1 \rightarrow 0.5$	$0.5 \rightarrow 0.9$	$0.9 \rightarrow 0.4$	$0.4 \rightarrow 0.8$	$0.8 \rightarrow 0.3$	$0.3 \rightarrow 0.7$	$0.7 \rightarrow 0.2$	$0.2 \rightarrow 0.6$	
	0.4	0.4	0.5	0.4	0.5	0.4	0.5	0.4	
9	10	11	12	13	14	15	16	17	
27000	30000	33000	36000	39000	41000	45000	48000	51000	
$0.6 \rightarrow 0.1$	$0.1 \rightarrow 0.6$	$0.6 \rightarrow 0.2$	$0.2 \rightarrow 0.7$	$0.7 \rightarrow 0.3$	$0.3 \rightarrow 0.8$	$0.8 \rightarrow 0.4$	$0.4 \rightarrow 0.9$	$0.9 \rightarrow 0.5$	
0.5	0.5	0.4	0.5	0.4	0.5	0.4	0.5	0.4	
18	19	20	21	22	23	24	25	26	
54000	57000	60000	63000	66000	69000	72000	75000	78000	
$0.5 \rightarrow 0.1$	$0.1 \rightarrow 0.4$	$0.4 \rightarrow 0.7$	$0.7 \rightarrow 0.1$	$0.1 \rightarrow 0.3$	$0.3 \rightarrow 0.6$	$0.6 \rightarrow 0.9$	$0.9 \rightarrow 0.3$	$0.3 \rightarrow 0.5$	
0.4	0.3	0.3	0.6	0.2	0.3	0.3	0.6	0.2	
27	28	29	30	31	32	33	34	35	
81000	84000	87000	90000	93000	96000	99000	101000	104000	
$0.5 \rightarrow 0.8$	$0.8 \rightarrow 0.2$	$0.2 \rightarrow 0.5$	$0.5 \rightarrow 0.7$	$0.7 \rightarrow 0.9$	$0.9 \rightarrow 0.2$	$0.2 \rightarrow 0.4$	$0.4 \rightarrow 0.6$	$0.6 \rightarrow 0.8$	
0.3	0.6	0.3	0.2	0.2	0.7	0.2	0.2	0.2	
36	37	38	39	40	41	42	43	44	
107000	110000	113000	116000	119000	122000	125000	128000	131000	
$0.8 \rightarrow 0.1$	$0.1 \rightarrow 0.8$	$0.8 \rightarrow 0.6$	$0.6 \rightarrow 0.4$	$0.4 \rightarrow 0.2$	$0.2 \rightarrow 0.9$	$0.9 \rightarrow 0.7$	$0.7 \rightarrow 0.5$	$0.5 \rightarrow 0.2$	
0.7	0.7	0.2	0.2	0.2	0.5	0.2	0.2	0.3	
45	46	47	48	49	50	51	52	53	
135000	138000	141000	144000	147000	150000	153000	156000	159000	
$0.2 \rightarrow 0.8$	$0.8 \rightarrow 0.5$	$0.5 \rightarrow 0.3$	$0.3 \rightarrow 0.9$	$0.9 \rightarrow 0.6$	$0.6 \rightarrow 0.3$	$0.3 \rightarrow 0.1$	$0.1 \rightarrow 0.7$	$0.7 \rightarrow 0.4$	
0.6	0.3	0.2	0.6	0.3	0.3	0.2	0.5	0.3	
54	55	56	57	58	59	60	61	62	
162000	165000	168000	171000	174000	177000	180000	183000	186000	
$0.4 \rightarrow 0.1$	$0.1 \rightarrow 0.2$	$0.2 \rightarrow 0.3$	$0.3 \rightarrow 0.4$	$0.4 \rightarrow 0.5$	$0.5 \rightarrow 0.6$	$0.6 \rightarrow 0.7$	$0.7 \rightarrow 0.9$	$0.8 \rightarrow 0.9$	
0.3	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	
63	64	65	66	67	68	69	70	71	72
189000	192000	195000	198000	201000	204000	207000	210000	213000	216000
$0.9 \rightarrow 0.1$	$0.1 \rightarrow 0.9$	$0.9 \rightarrow 0.8$	$0.8 \rightarrow 0.7$	$0.7 \rightarrow 0.6$	$0.6 \rightarrow 0.5$	$0.5 \rightarrow 0.4$	$0.4 \rightarrow 0.3$	$0.3 \rightarrow 0.2$	$0.2 \rightarrow 0.1$
0.8	0.8	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1

Table A.2: Values of virtual drifts

# List of Tables

5.1	Values of committee benchmarks . . . . .	28
A.1	Values of real drifts . . . . .	78
A.2	Values of virtual drifts . . . . .	79

# List of Figures

2.1	Illustration data stream . . . . .	4
3.1	Distribution before drift . . . . .	7
3.2	Distribution after drift . . . . .	7
3.3	Prior distribution . . . . .	8
3.4	Real drift mathematical background . . . . .	10
3.5	Real drift illustration . . . . .	10
4.1	Diagram k-nearest neighbours . . . . .	13
5.1	Window size $ws$ of an algorithm . . . . .	15
5.2	Accuracy benchmark . . . . .	17
5.3	Accuracy benchmark detail . . . . .	18
5.4	Small window sizes performs good accuracy by accident . . . . .	20
5.5	Big window sizes perform similar good results . . . . .	20
5.6	Optimal window sizes per instance without adjustment . . . . .	21
5.7	Optimal window sizes per instance without adjustment, detail . . . . .	22
5.8	table with three exemplary instances . . . . .	22
5.9	Optimal window sizes per instance with adjustment . . . . .	24
5.10	Optimal window sizes per instance with adjustment, detail . . . . .	25
5.11	Accuracy without/with 1.5 percent neighbourhood smoothing . . . . .	26
6.1	Entropy Bernoulli Trial . . . . .	30
6.2	Entropys according constam . . . . .	34
6.3	Entropys according Constam, detail . . . . .	35
6.4	Entropy window size 50 . . . . .	37
6.5	Entropy window size 500, shift in the past . . . . .	37
6.6	Entropy, future window shift in the past . . . . .	39
6.7	Entropy, maximal drift . . . . .	41
6.8	Entropy, overlapping, maintain back border . . . . .	42
6.9	Entropy, side by side . . . . .	42
6.10	Entropy normalised, real drift . . . . .	44
6.11	Entropy normalised, prior drift . . . . .	44
6.12	Dataset, divided into categories . . . . .	45
6.13	Entropy, ideal $\Delta$ . . . . .	46
6.14	Entropy, minimal for $\Delta = 0.1$ . . . . .	48

7.1	Entropy vs. optimal window size . . . . .	50
7.2	symmetric entropy vs. optimal window size . . . . .	51
7.3	Entropy vs. optimal window size . . . . .	51
7.4	Entropy vs. optimal window size . . . . .	54
7.5	Entropy vs. optimal window size . . . . .	54
8.1	Overview of implementing a threshold . . . . .	58
8.2	Entropy-regulated accuracy vs benchmarks . . . . .	59
8.3	Heatmap $\psi=45^\circ$ , $ws$ of $H$ and $\theta$ variable . . . . .	60
8.4	Heatmap $ws$ of $H$ , $\psi$ and $\theta$ variabel . . . . .	62
8.5	Threshold based control of the window size . . . . .	63
8.6	Accuracies, regulated $ws$ vs committee benchmarks . . . . .	65
8.7	Heatmap 10% noise $ws$ of $H$ fixed, $\psi$ and $\theta$ variable . . . . .	66
8.8	Heatmap region-wise $\psi=45^\circ$ , $ws$ of $H$ and $\theta$ variable . . . . .	68
8.9	Heatmap region-wise $\psi=45^\circ$ , $ws$ of $H$ and $\theta$ variable . . . . .	68
8.10	Heatmap region-wise $\psi=45^\circ$ , $ws$ of $H$ and $\theta$ variable . . . . .	69
9.1	Entropy's ossible behaviour in case of a continuous drift . . . . .	71
9.2	Heatmap window sizes vs instances, overview . . . . .	72
9.3	Heatmap window sizes vs instances . . . . .	72
9.4	Optimal window sizes knn and nb . . . . .	73

# Bibliography

- [Contam 05] Martin Contam. Dynamische regelung inkrementeller algorithmen unter dem einfluss von concept drifts. Master's thesis, University Zurich, Zurich, 2005.
- [Duda 00] Richard O. Duda, Peter E. Hart & David G. Stork. Pattern classification (2nd edition). Wiley-Interscience, 2000.
- [Fawcett 05] Tom Fawcett & Peter A. Flach. *A response to Webb and Ting's on the application of ROC analysis to predict classification performance under varying class distributions*. Mach. Learn., vol. 58, no. 1, pages 33–38, 2005.
- [Ferri 03] C. Ferri, J. Hernández-Orallo & M.A. Salido. *Volume Under the ROC Surface for Multi-class Problems. Exact Computation and Evaluation of Approximations*. Proceeding of 14th European Conference on Machine Learning, pages 108–120, 2003.
- [Kuncheva 04] Ludmila I. Kuncheva. *Classifier Ensembles for Changing Environments*. In Multiple Classifier Systems, pages 1–15, 2004.
- [Provost 01] Foster Provost & Tom Fawcett. *Robust Classification for Imprecise Environments*. Mach. Learn., vol. 42, no. 3, pages 203–231, 2001.
- [Shannon 48] Claude E. Shannon. *A Mathematical Theory of Communication*. Bell System Technical Journal, vol. 27, pages 379–423, 1948.
- [Widmer 93] G. Widmer & M. Kubat. *Effective Learning in Dynamic Environments by Explicit Context Tracking*. In Machine Learning: ECML-93 - Proc. of the European Conference on Machine Learning. 1993.