Operative Assessment of Predicted Generalization Errors on Non-Stationary Distributions in Data-Intensive Applications

Sergio Decherchi, Paolo Gastaldo, Fabio Sangiacomo, Alessio Leoncini, and Rodolfo Zunino

Dept. Biophysical and Electronic Engineering (DIBE), University of Genova
Email: {sergio.decherchi, paolo.gastaldo, fabio.sangiacomo, alessio.leoncini , rodolfo.zunino}@unige.it

Abstract

Data-intensive applications use empirical methods to extract consistent information from huge samples. When applied to classification tasks, their aim is to optimize accuracy on unseen data hence a reliable prediction of the generalization error is of paramount importance. Theoretical models, such as Statistical Learning Theory, and empirical estimations, such as cross-validation, can both fit data-mining classification domains very well, provided some crucial assumptions are verified in advance. In particular, the stationary distribution of the observed data is critical, although it is sometimes overlooked in practice. The paper formulates an operative criterion to verify the stationary assumption; the method applies to both theoretical and practical predictions of generalization errors. The analysis addresses the specific case of clustering-based classifiers; the K-Winner Machine (KWM) model is used as a reference for its known theoretical bounds; cross-validation provides an empirical counterpart for practical comparison. The criterion, based on efficient unsupervised clustering-based probability distribution estimation, is tested experimentally on a set of different, data-intensive applications, including: intrusion detection for computer-network security, optical character recognition, text mining and pedestrian detection. Experimental results confirm the effectiveness of the proposed approach to efficiently detect non stationarity.

Keywords: Statistical Learning Theory, Data mining, non-stationary distribution, K-Winner Machine, Clustering

1. INTRODUCTION

In data-intensive applications, clustering methods arrange huge amounts of data into a structured representation and search for relevant information [1][2]. The vast datasets and the heterogeneous descriptions of patterns set stringent requirements on the algorithms adopted; when empirical classifiers aim to optimize prediction on unseen data [1], attaining an accurate estimate of the run-time generalization error is a critical issue. Several methods in the literature have tackled that problem from both a practical [3] and a theoretical viewpoint [4][5][6].

From a practical viewpoint, empirical estimates such as cross-validation methods often support the prediction of the run-time generalization performance in real applications [30]. The literature reports that these techniques are quite accurate and outperform theoretical models in complex

classifier design [29]. From a theoretical viewpoint, within the framework of Statistical Learning Theory [26], the formulation based on the Vapnik-Chervonenkis dimension (d_{VC}) [7] exhibits a theoretical foundation: given a classifier C, and its associated class of decision functions, the d_{VC} is the largest number of patterns that C can correctly classify; this makes d_{VC} a reliable measure of complexity of a classification algorithm. Unfortunately, the resulting bounds to the generalization error often prove impractical for a variety of reasons. First, Vapnik's theory stems from a worst-case analysis, hence it usually requires a huge number of patterns to tighten generalization bounds down to reasonable ranges. Secondly, many practical classifiers are so powerful that the crucial parameter, d_{VC}, measuring a model's complexity grows uncontrollably (e.g. SVM with Guassian kernel). Finally, few families of clustering-based classifiers have been fully characterized in compliance with Vapnik's theory. This is, incidentally, the case of the K-Winner Machine (KWM) model [8], which is used here as the clustering-based framework for pattern classification.

In spite of the above limitations, Statistical Learning Theory can yet be of practical significance when dealing with clustering and data-mining [1], since data mining applications are typically rich in patterns and can therefore offer the required large samples. Moreover, the complexity of clustering-based classifiers often proves much lower than that of other approaches [1].

A crucial prerequisite in applying both theoretical and empirical predictions, however, is that the probability distribution of data is stationary [26]; such a condition holds in many practical testbeds and is often assumed to hold implicitly. In some data-intensive domains, however, the stationary nature of data distributions may prove questionable, either because the data refer to a phenomenon whose time-varying nature is overlooked, or because the original sample is so large that new samples stem from unexplored areas of the probability distribution, hence test data are virtually uncorrelated from training ones. Retraining (either from scratch or as an update of existing learning results) is a typical solution to that problem, but in data-intensive applications it may prove very expensive. This may occur for a variety of reasons: for instance, because the actual process for updating training results is difficult to design or implement, or because the amount of data is excessive, or because older data might not be easily accessible.

In general, non stationarity can be classified according to two different categories: *covariate shift* [38] refers to those cases in which the non stationarity only affects the pattern probability distribution $P(\mathbf{x})$; *concept drift* [40] refers to those case in which non stationarity is confined to the target probability distribution $P(c|\mathbf{x})$. This paper addresses *covariate shift* and tackles the stationary-sampling issue from the conventional viewpoint of validation methods for classifier training [26]; the basic observation is that non-stationary distributions ultimately give rise to discrepancies between the data distributions in the training and test phases. To this aim, an efficient and reliable

method to estimate the pdf and assess the stationarity of input data is required. The proposed criterion asses the non-stationarity by casting the original multivariate problem to an univariate problem via a clustering procedure. This procedure avoids the curse of dimensionality and the possible numerical instabilities that can occur using traditional parametric methods such as Parzen Windows or Mixture of Gaussian Models [31][32][33][34]. Indeed, the class of problems covered by the proposed methodology is not strictly limited to *covariate shift*, as the procedure can effectively apply also to problems in which non stationarity affects both the pattern probability distribution and the target probability distribution.

The approach proposed in this paper adopts the KWM model [8] as classifier. The rationale behind such choice is twofold. KWM yields tight bounds to generalization performance [8] and inherently supports multi-class classification tasks [9]. These features make KWM profitably suitable for data-intensive applications and for evaluating the applicability of Vapnik's generalization predictions accordingly, together with conventional cross-validation methods.

The proposed methodology improves and generalizes the preliminary analysis discussed in [35]. Experiments first show the approach validity in a synthetic domain, mainly to provide an intuitive demonstration of the basic non stationarity detection principle; the method is then tested in a group of complex real-world problems: the detection of intrusions in computer networks, Optical Character Recognition for numerical patterns, Emails Spam detection and Pedestrian Detection. The "KDD Cup 1999" dataset [10], the Manuscript NIST (MNIST) OCR dataset [24], the Spam Assassin dataset [37] and the Daimler dataset [39] provided the related experimental domains. Experimental results show that the proposed criterion successfully detected the non-stationary/stationary nature of the proposed domains. Furthermore, an additional experiment it is reported to point out the differences between a problem characterized by *covariate shift* and a problem characterized by *concept drift*. Such experiment shows that the unsupervised nature of the proposed method cannot be applied to such kind of non stationarity where the *drift* is only limited to the target values.

The paper is organized as follows. Section 2 briefly illustrates error predictions methods and the K-Winner Machine model. Section 3, the core section, introduces the criterion designed to validate the applicability of generalization error estimation by stationarity assessment. Section 4 presents the experimental results obtained on reference, real-word testbeds, and deals with peculiar theoretical and practical emerging issues. Some concluding remarks are made in Section 5.

2. THE PREDICTION OF GENERALIZATION PERFORMANCE

2.1 Analytical and Empirical Prediction Methods

Generalization theory proves that a classifier's performance is upper-bounded by the empirical training error, ν, increased by a penalty term. In this term, the Growth Function [6], $GF(N_p)$, measures the complexity of the fact that the classifier has been trained with a set of N_p patterns. This theory derives a bound, π, to the generalization error of the considered classifier:

$$\pi < \nu + \frac{\varepsilon}{2}\left(1 + \sqrt{1 + \frac{4\nu}{\varepsilon}}\right) \qquad (\#1)$$

where

$$\varepsilon = \frac{4}{N_p}\left[\ln GF(N_p) - \ln\frac{\eta}{4}\right] \qquad (\#2)$$

and η is a confidence level.

Vapnik's theory adopts a worst-case analysis, hence the predicted error bound (#1) often falls in a very wide range that eventually lessens the practical impact of the overall approach. This is especially true when a limited sample of training patterns is available; data-mining environments, however, typically involve very large datasets, whose cardinality ($N_p >> 10^5$) can actually shrink the complexity penalty term down to reasonable values. Moreover, some models intrinsically prevent an uncontrolled increase in the classifier's $GF(N_p)$ (thus of the d_{VC} of the classifier [6]). Thus data-mining domains seem to comply with basic Statistical Learning Theory quite well [7][19].

From a different perspective, empirical approaches to estimate a classifier's generalization performance bypass worst-case theoretical analysis by using observed data themselves as a prediction support. In the popular method of *k-fold* Cross Validation (CV) [30], one partitions the training set into k non-overlapping subsets: the classifier is trained on the union of (k–1) subsets, and the remaining k-th subset provides a test set to measure the associated classification performance. The procedure encompasses all combinations of k test sets; the average classification error, π_{CV}, is the estimate of the classifier's generalization performance:

$$\pi_{CV} = \frac{1}{k}\sum_{j=1}^{k} \nu^{(j)} \qquad (\#3)$$

It is worth noting that the mutual correlation among the empirical tests on folded partitions brings about a statistical bias to the prediction (#3).

2.2 KWM Classifiers and Prediction Error Estimation

The approach proposed in this paper adopts a specific family of classifier belonging to the Structural Risk Minimization paradigm, namely, the K-Winner Machine model. Such a choice is justified by two main aspects: first, an established analysis has described the theoretical properties of the classifier model [8] and shown that the resulting generalization bounds can be profitably applied; secondly, the KWM approach relies on unsupervised Vector Quantization and therefore implicitly takes into account the problem of rendering the probabilistic distribution of samples, which is at the core of the present analysis.

The training strategy of the K-Winner Machine (KWM) model first develops a representation of the data distribution by means of an unsupervised process, then applies a calibration process to train a supervised classifier. A detailed outline of the KWM training algorithm [8] is given in Appendix A.

Among the wide variety of possible approaches, the research presented here adopts the Plastic Neural Gas (PGAS) algorithm [11], as it can adjust both the number and the positions of prototypes simultaneously [11]; moreover, PGAS prevents the occurrence of *dead vectors* (void prototypes covering empty partitions). After unsupervised training, a calibration process [8] labels the Voronoi tessellation of the data space induced by the positions of the prototypes. Each partition/prototype is labeled according to the predominant class. From a cognitive viewpoint, the latter step aims to reproduce the conditional distribution of classes.

At run time, each point in the data space is classified locally, under the cognitive assumption that the risk in the classification outcome for a given point decreases when more and more neighboring prototypes concur in the classification of that point. As opposed to conventional ensemble methods, a KWM requires a complete agreement among the set of best-matching prototypes and does not involve any majority counting; the smallest set will include the nearest prototype only.

The advantage of applying Statistical Learning Theory to the KWM model mainly lies in the computation of generalization bounds at the local level. The literature offers a variety of approaches to estimating the generalization error of a classifier [6] [7]. The analysis presented in [8] adopted the formulation based on the Vapnik-Chervonenkis dimension [7], and derived several analytical properties of KWMs, including the Vapnik-Chervonenkis dimension and the analytical expression of the Growth Function of the family of classifiers used in the KWM model.

The resulting theory [8] proves that one can compute an error bound, $\pi(k)$, for each agreement level, k, and more importantly, that such a bound is a non-increasing function when k increases.

This confirms the intuitive notion that the risk in a classification decision about a given point should be reduced by the concurrence of several neighboring prototypes.

A crucial feature of the KWM model is that, by using the prototype-agreement criterion at run time, any point in the data space is characterized by a local bound to the classification error. Such a bound, that is the instantiation of (#1) for the KWM case, has been derived analytically [8]: the main result states that with probability η holds:

$$\pi(k) \le \nu + \frac{2}{\sqrt{N_p}}\left(\left\lfloor \frac{N_h}{k} \right\rfloor \ln N_c - \ln\frac{\eta}{4}\right)\left(\frac{1}{\sqrt{N_p}} + \sqrt{\frac{1}{N_p} + \nu\left(\left\lfloor \frac{N_h}{k} \right\rfloor \ln N_c - \ln\frac{\eta}{4}\right)^{-1}}\right) \qquad (\#4)$$

As a consequence, unsupervised prototype positioning sharply reduces the bounding term in (#1). By contrast, the KWM training algorithm does not provide any a-priori control over the empirical training error, due to the unsupervised training mechanism. This brings about the problem of model selection, which is usually tackled by a tradeoff between accuracy (classification error in training) and complexity (number of prototypes).

The specific benefits of the KWM in the research presented here mainly consists in relating the VQ framework, which provides a powerful tool to render the statistical distribution of data, to an analytical, precise formulation of the generalization bounds that derive from the theoretical application of Statistical Learning Theory. The combination of these features allows one to verify the properties of generalization theory in a controlled scenario under various conditions of sample distributions.

3. NON STATIONARITY DETECTION FOR ASSESSING THE APPLICABILITY OF GENERALIZATION ERROR ESTIMATION

Data-intensive applications pose the crucial issue of the stationary nature of the pattern distribution. In fact, the stationary-distribution assumption [20] is a basic prerequisite to the applicability Statistical Learning Theory. Indeed, also empirical methods ultimately rely on the fact that the data distribution is consistently represented by the available sample [29], hence the assumption of a stationary distribution is critical in this case, as well. Non-stationary phenomena are in fact quite frequent in data mining, due to the time-varying nature of data or the huge size of the probability distribution that makes a complete sampling unfeasible. This in turn affects the reliability of the generalization error estimation associated to the trained classifier.

This research addresses such critical issue by introducing a general criterion that exploits the clustering-based paradigm to evaluate the applicability of generalization prediction approaches

when variations on $P(\mathbf{x})$ occur. For instance, a sufficient condition for the developed method to work is the presence of a *covariate shift* [38] on data: from a cognitive viewpoint, such a dynamic scenario can be formalized by noting that the pattern probability distribution, $P(\mathbf{x})$ is not stationary while the target probability distribution $P(c|\mathbf{x})$ satisfies the stationarity assumption. Such condition actually applies to most of the real world problems of practical interest. The proposed methodology, which is outlined in the following, in addition, can also successfully tackle problems in which non stationarity characterizes both the $P(\mathbf{x})$ and the $P(c|\mathbf{x})$ at the same time.

3.1 Non Stationarity Detection by Using Vector Quantization

In normal practice, one measures generalization performance by using a test set that is not involved in the training process. This is done for a variety of reasons: either because cross-validation drives model selection [3], or because the test set is partially labeled [21], or because the test set was not available at the time of training. Within that context, the assumption of a stationary distribution may be rephrased by asserting that, given a set $\mathrm{C} = \left\{c^{(h)}, h = 1,.., N_c\right\}$ of N_c possible pattern classes, the training set instance, including N_p patterns, $T = \left\{\left(\mathbf{x}_l^{(T)}, c_l\right), \mathbf{x}_l^{(T)} \in \Re^D, c_l \in C, l = 1,.., N_p\right\}$, and the test set instance, including N_u patterns, $S = \left\{\left(\mathbf{x}_j^{(\mathrm{S})}, c_j\right), \mathbf{x}_j^{(\mathrm{S})} \in \Re^D, c_j \in C, j = 1,.., N_u\right\}$, are identically and independently drawn from a common probability distribution, $P(\mathbf{x})$. If such an assumption does not hold, the training set is not representative of the entire population and expressions (#1) and (#3) may not provide the correct estimate of classification accuracy.

The present analysis derives a general, yet practical criterion to verify the stationarity assumption, and consequently to validate the associate generalization error estimation. The proposed methodology tackles stationarity sampling from the conventional viewpoint of validation methods for classifier training [26]. The method uses a paradigm based on Vector Quantization (VQ) to check on the stationary-distribution assumption. A VQ–based classifier positions a set of prototypes so as to minimize some (unsupervised) distortion criterion in representing training data and calibration process observes the distribution pattern classes to assign a class to each prototype and the final step is the proposed criterion . The following conventions will be adopted:

- $\mathrm{W}' = \left\{\left(\mathbf{w}_n, c_n\right), \mathbf{w}_n \in \Re^D, c_n \in C, n = 1,..., N_h\right\}$ is a set of N_h labeled prototypes;
- $\mathbf{w}^*(\mathbf{x}) = \arg\min_{\mathbf{w} \in \mathrm{W}'} \left\{\left\|\mathbf{x} - \mathbf{w}\right\|^2\right\}$ is the prototype that represents a pattern, $\mathbf{x}$. The operator $\left\| \bullet \right\|^2$ here denotes the standard Euclidean distance.

Within the above conventions, the VQ-based stationarity criterion is outlined as follows:

- First, one trains and calibrates a codebook, W', to classify training and test data.
- Secondly, one estimates the discrete probability distributions, $P^{(T)}(\mathbf{x})$ and $P^{(S)}(\mathbf{x})$, of the training set, T, and of the test set, S, respectively; this is easily attained by counting the number of training/test patterns that lie within the data-space partition spanned by each prototype. Then the number of patterns of each cluster divided by the total number of patterns, constitutes the normalized frequency or 'bin' of the distribution.
- Finally, one checks whether the data in S and T have been drawn from the same distribution.

In principle, several, different techniques may support the latter step. In the present approach and without loss of generality, $D\left(P^{(S)}(\mathbf{x}), P^{(T)}(\mathbf{x})\right)$ will denote a measure of divergence between the discrete probability distributions, $P^{(T)}(\mathbf{x})$ and $P^{(S)}(\mathbf{x})$. Any analytical measure of the discrepancy between two probability distributions is applicable for that purpose; in this regard, this paper analyzes the performance of the general class of f-divergences [22][27].

Let be $f(t)$, a convex function defined for $t>0$, with $f(1) = 0$. The f-divergence [27] of a distribution $P^{(S)}(\mathbf{x})$ from $P^{(T)}(\mathbf{x})$ is defined by:

$$D_f\left(P^{(S)}(\mathbf{x}), P^{(T)}(\mathbf{x})\right) = \sum_{n=1}^{N_h} t_n f\left(\frac{s_n}{t_n}\right) \qquad (\#5)$$

where s_n and t_n denote the normalized frequencies associated with $P^{(S)}(\mathbf{x})$ and $P^{(T)}(\mathbf{x})$, respectively. Different instances can be derived from the general class (#5) by exploiting different implementations of the function f. Table 1 lists the most common divergences, which have also been adopted in this work.

The minimum (zero) value of $D_f\left(P^{(S)}(\mathbf{x}), P^{(T)}(\mathbf{x})\right)$ marks the ideal situation and indicates perfect coincidence between the training and test distributions. Non-null values, however, typically occur in common practice, and it may be difficult to interpret from such results the significance of the numerical discrepancies measured between the two distributions. The present research adopts an empirical approach to overcoming this issue by building up a 'reference' experiment setting that constitutes the sample based threshold used to decide if a distribution is stationary or not.

The procedure can be outlined as it follows: first, one creates an artificial, stationary distribution, J, that joins training and test data: $J:=T\cup S$. Secondly, one uses the discrete distribution J to draw at random a new training set, T_J, and a new test set, S_J, such that $T_J \cap S_J = \varnothing$. Both these sets have the same relative proportions as the original samples. Third, using these sets for a session

of training and test yields a pair of discrete distributions, $P_J^{(S)}(\mathbf{x}), P_J^{(T)}(\mathbf{x})$; finally, one measures the divergence between the new pair of data sets by computing $D_f\left(P_J^{(S)}(\mathbf{x}), P_J^{(T)}(\mathbf{x})\right)$. This value provides the numerical reference threshold for assessing the significance of the actual discrepancy value $D_f\left(P^{(S)}(\mathbf{x}), P^{(T)}(\mathbf{x})\right)$ by comparison.

If the original sample had been drawn from a non stationary distribution, then the associate discrepancy value, $D_f\left(P^{(S)}(\mathbf{x}), P^{(T)}(\mathbf{x})\right)$ will be greater than the reference threshold $D_f\left(P_J^{(S)}(\mathbf{x}), P_J^{(T)}(\mathbf{x})\right)$, computed on the artificial distribution *J*.

The following pseudo-code works out the complete procedure, which is characterized by low computational cost and high numerical reliability, compared with other estimation methods [32][34].

Criterion for validating the applicability of generalization error estimates

---

0. **Input**: a training set including N_T labeled data, ($\mathbf{x}_i$, $c(\mathbf{x}_i)$) ; a test set including N_S labeled data, ($\mathbf{x}_j$, $c(\mathbf{x}_j)$)

1. (*VQ training*)
   Apply a VQ algorithm to the training set and position the set of prototypes:
   $W' = \left\{(\mathbf{w}_n, c_n); \mathbf{w}_n \in \Re^D, c_n \in C, n = 1,\ldots,N_h\right\}$

2. (*Probability distribution modeling*)
   2.1 Estimate the training discrete probability distribution, $P^{(T)}(\mathbf{x})$ as follows:
   $$P^{(T)}(\mathbf{x}) := \left\{P_n^{(T)};\quad n = 1,...,N_h\right\};\quad \text{where:}\quad P_n^{(T)} = \left\{\mathbf{x}_i^{(T)} \in \Re^D : \mathbf{w}^*\left(\mathbf{x}_i^{(T)}\right) = \mathbf{w}_n\right\};$$
   2.2 Estimate the test discrete probability distribution, $P^{(S)}(\mathbf{x})$ as follows:
   $$P^{(S)}(\mathbf{x}) := \left\{P_n^{(S)};\quad n = 1,...,N_h\right\};\quad \text{where:}\quad P_n^{(S)} = \left\{\mathbf{x}_i^{(S)} \in \Re^D : \mathbf{w}^*\left(\mathbf{x}_i^{(S)}\right) = \mathbf{w}_n\right\};$$

3. (*Measuring discrepancy*)
   3.1 Compute normalized frequencies: $t_n = \left|P_n^{(T)}\right| / N_T$; $s_n = \left|P_n^{(S)}\right| / N_S$; $n = 1,\ldots,N_h$
   3.2 Compute the divergence $D_f\left(P^{(S)}(\mathbf{x}), P^{(T)}(\mathbf{x})\right)$ between $P^{(T)}(\mathbf{x})$ and $P^{(S)}(\mathbf{x})$.

4. (*Applicability of generalization estimates*)
   4.1 **If** a reference validation set is not available, form an artificial discrete distribution by joining training and test data: $J := T \cup S$; Draw from *J* at random a training set, T_J, and a test set, S_J, having the same relative proportions as the original data sets;
   Else use the 'reference' as S_J and set $T_J = T$
   Repeat steps (1,2,3) by using the new pair of sets;
   If $D_f\left(P^{(S)}(\mathbf{x}), P^{(T)}(\mathbf{x})\right) > D_f\left(P_J^{(S)}(\mathbf{x}), P_J^{(T)}(\mathbf{x})\right)$
   Then: The stationary nature is not verified and generalization error estimates are not supported
   Else: The stationary nature is verified and generalization error estimates are validated empirically

---

3.2 Discussion

For the sake of completeness and clarity, in the following the actions executed by the proposed procedure in two opposite scenarios are analyzed:

1) *Stationary Case*. If *T* and *S* are drawn from the same distribution $P(\mathbf{x})$, then *T* and *S* alone can be used to estimate the underlying $P(\mathbf{x})$. In other words, the discrete probability distributions $P^{(T)}(\mathbf{x})$ and $P^{(S)}(\mathbf{x})$ are both reliable estimates of $P(\mathbf{x})$ (under the assumption of a data-intensive problem). Thus, when applying step 4 of the proposed procedure (see pseudo-code above), one obtains a pair of sets (a training set, T_J, and a test set, S_J) that leads to a consistent estimates of $P(\mathbf{x})$ as well. As a consequence, $D_f\left(P_J^{(\mathrm{S})}(\mathbf{x}), P_J^{(\mathrm{T})}(\mathbf{x})\right)$ must approximately coincide with $D_f\left(P^{(\mathrm{S})}(\mathbf{x}), P^{(\mathrm{T})}(\mathbf{x})\right)$.
2) *Non Stationary Case.* In this case *T* and *S* are drawn from two distinct distributions distribution $P_1(\mathbf{x})$ and $P_2(\mathbf{x})$ respectively. From *T* and *S*, one estimates $P^{(T)}(\mathbf{x})$ and $P^{(S)}(\mathbf{x})$ that are reliable estimates of $P_1(\mathbf{x})$ and $P_2(\mathbf{x})$ respectively; then, eventually, the reference value $D_f\left(P^{(\mathrm{S})}(\mathbf{x}), P^{(\mathrm{T})}(\mathbf{x})\right)$ is computed. If one merges-shuffles *T* with *S*, and split them with the same original proportions, as per step 4, one obtains the new pair of sets T_J and S_J. After working out the corresponding discrete probability distributions $P_J^{(\mathrm{S})}(\mathbf{x})$ and $P_J^{(\mathrm{T})}(\mathbf{x})$, the quantity $D_f\left(P_J^{(\mathrm{S})}(\mathbf{x}), P_J^{(\mathrm{T})}(\mathbf{x})\right)$ can finally be computed. This time, as the stationarity assumption does not hold, one expects $D_f\left(P_J^{(\mathrm{S})}(\mathbf{x}), P_J^{(\mathrm{T})}(\mathbf{x})\right)$ to be significantly larger than the reference value $D_f\left(P^{(\mathrm{S})}(\mathbf{x}), P^{(\mathrm{T})}(\mathbf{x})\right)$. Such discrepancy is caused by the fact that $P_J^{(\mathrm{S})}(\mathbf{x})$ and $P_J^{(\mathrm{T})}(\mathbf{x})$ cannot estimate consistently $P_1(\mathbf{x})$ and $P_2(\mathbf{x})$.

In the above scenarios the merge-shuffle and split operation has no effect on distributions in the stationary case; instead in the non stationary case the merge-shuffle and split operation induces a change in the discrepancy values; in both cases $D_f\left(P_J^{(\mathrm{S})}(\mathbf{x}), P_J^{(\mathrm{T})}(\mathbf{x})\right)$ is the adaptive threshold that allows to discriminate between stationarity or not.

If stationarity is verified, the theoretical assumptions underlying Statistical Learning Theory hold, and the bound formulation (#1) or cross-validation (#3) are valid. Otherwise, when $D_f\left(P^{(\mathrm{S})}(\mathbf{x}), P^{(\mathrm{T})}(\mathbf{x})\right) > D_f\left(P_J^{(\mathrm{S})}(\mathbf{x}), P_J^{(\mathrm{T})}(\mathbf{x})\right)$, one might infer that the original sampling process was not stationary, hence a direct application of theoretical results (#1) or empirical estimates (#3) is questionable. Clearly the artificial dataset (S_J, T_J) is built whenever a stationary 'reference' dataset is not provided. Indeed, the proposed procedure to validate the applicability of generalization-theory

bounds can eventually exploits, when needed, the implementation of a resampling strategy, which can of course be repeated several times to enhance the statistical robustness of numerical estimates.

Two aspects make the methodology presented above suitable for data-intensive application:

- in a single step one can obtain both the classification of data and the reliable estimation of the data distribution.
- the obtained estimation is based upon an univariate pdf, thus avoiding the usual issues that arises when classical methods such as Parzen Windows [32] and Mixture of Gaussians [34] when high dimensional spaces are involved.

4. EXPERIMENTAL RESULTS

The aim of this section is to operatively investigate the previously proposed method. In particular a synthetic 2-D dataset is studied to intuitively show the effectiveness of the approach; further four real domains are analyzed. A last artificial experiment, that violates the hypothesis of the method for which $P(\mathbf{x})$ must change, shows that, consistently, the methodology does not detect this kind of purely supervised non stationarity (i.e. concept drift).

In this section all references to generalization bounds are assumed to be computed by using (#4) in compliance with KWM theory. The parameter k present in (#4) in all experiments is locked to 1: this simply means applying a 1- nearest-neighbor policy to the bound estimation, that in other words denote that we are not interested in a local estimate for each pattern but in a global one [8].

4.1 Artificial 2-dimensional testbed

The experiments on an artificial testbed aimed at demonstrating the operational principles in a 2-D space that allowed visual inspection; in particular the following analysis, for simplicity, will only deal with the theoretical estimation method as per (#3). The dataset simulated a context in which the test distribution progressively diverged from the original training one. The overall experiment involved 5 sets of data as per Fig.1: the basic pair included the original training set (Xtg) and an associate test set (Xts), which was drawn from the same distribution, thus mimicking a stationary case. Three additional samples (Xts1, Xts2 and Xts3) emulated non-stationary cases. All datasets involved binary classification problems; the non-stationary phenomenon was emulated by generating data from a Normal distribution whose mean value drifted progressively. Thus the three sets Xts1, Xts2 and Xts3, could be interpreted as time-dependent variation laws of data

distributions. This artificial experiment clearly did not require any re-sampling strategy to get a stationary reference.

For each pair of sample, a range of clustering settings were tested, in particular by increasing the number of prototypes; then the performances of the resulting KWM were measured. The discrepancy values associated with clustering outcomes progressively followed the non-stationary nature of the phenomenon. At the same time, the error bounds predicted by generalization theory became more and more unreliable. Tables 2 a), b), c), and d) give the values obtained from the analysis on the test sets for the implemented discrepancy formulations. The discrepancy measures, computed as illustrated in Table 1, progressively increased from the stationary data set, Xts, up to the most 'distant' data set, Xts3. The relevant property in these results is that all measurements, albeit derived from different formulations, exhibit a common trend, thus supporting the method validity and robustness. Table 3 compares the error bounds predicted by generation theory for different numbers of prototypes with the actual errors measured on the various test sets. Empirical evidence confirmed that the bound values became more and more inaccurate and, as predicted in Section 2, followed the same progression marked by the discrepancy values.

The graphs in Fig.2 summarize the obtained results and clarify the divergence-based criterion in an intuitive way; in each graph, the *x* axis marks the different test sets, whereas the *y* axis gives each divergence formulation. The curves are plotted for various settings of the number, N_h, of VQ prototypes, and always witness a sharp increasing trend in discrepancy as long as the non-stationary test distribution diverges from the training sample.

Likewise, the graph in Fig.3 demonstrates that the validity of the theoretical error bound progressively weakens in the presence of increasingly non-stationary distributions. Indeed, the predicted error bound from Statistical Learning Theory holds for the stationary case (Xts) only, as expected from the analysis presented in the paper. As long as a non-stationary distribution takes place, discrepancy values increase and generalization performances diverge progressively from the theoretical bound.

4.2 – Intrusion detection in computer networks: the "KDD Cup 1999" dataset

The data set used for the network-intrusion testbed was originally created for the Third International Knowledge Discovery and Data Mining Tools Competition [10]. The KDD dataset [10] originated from the 1998 DARPA Intrusion Detection Evaluation Program managed by MIT Lincoln Labs [23], with the objective of surveying and evaluating research in intrusion detection.

The original data spanned a 41-dimensional feature space; crucial descriptors that took on categorical values, most notably "Protocol_type" and "Flag", were remapped into a mutually exclusive numerical representation, thereby leading a 52-dimensional feature vector.

Each pattern encompassed cumulative information about a connection session. In addition to "normal" traffic, attacks belonged to four main macro-classes. The complete training set contained about $5 \cdot 10^6$ patterns; normal traffic represented about 20% of the whole dataset, while attack types were quite unbalanced, as just two classes ('neptune' and 'smurf') spanned 78% of the entire dataset. The experimental session in this research involved a smaller training set, provided by the KDDCup'99 benchmark, which had been obtained by subsampling original training data at a 10% rate. The resulting "10% training set", T, included 494,021 patterns and preserved the original proportions among the five basic categories. The test set, S, provided by the KDD challenge held 311,029 patterns, and featured 'novel' attack schemes that were not covered by the training set.

To verify the stationary nature of the observed data distribution, the procedure described in Section 2 compares the original distribution (T,S) with the representation supported by the exhaustive distribution, $J=T \cup S$, that approximated a stationary situation. The artificial, reference training and test sets, T_J and S_J, were obtained by randomly resampling J. The measurement of the various divergences between the training and test coverages for both distributions (T,S) and (T_J,S_J) completed the validation process. Table 4 gives the empirical results obtained for increasing codebook sizes. The number of prototypes, N_h, varied significantly in the two situations: when training and test data were drawn from a common distribution, J, the probability support was wider, hence the VQ algorithm required a larger number of prototypes to cover the data space. Conversely, the original training data, T, were drawn from a limited sector of the actual support region, thus a smaller codebook was sufficient to represent the sample distribution. Numerical results pointed out that the divergence for the original distributions (T,S) always turned out to be larger than the divergence measured when training and test data were drawn from a stationary distribution (T_J,S_J). Such empirical evidence was mainly due to the marked discrepancies between training and test data sets, and clearly seemed to invalidate the applicability of the theoretical bounds from Statistical Learning Theory for the KDD99 dataset. In particular, this strong discrepancy, was characterized by the fact the real divergences are one or two order of magnitudes bigger than the reference ones. As a result, the validation criterion would predict that Vapnik's bound or cross validation error, would not hold for the original challenge data. For the sake of completeness, Table 5 compares the actual classification errors with the theoretical bounds for the original and the stationary distributions. These results are also given in a graphical fashion in Fig.4a, which also reports the cross-validation

predictions of generalization, as per expression (#3). Empirical evidence showed that theoretical or cross-validated predictions failed in bounding or predicting the generalization performance for the original data sets, whereas they provided good approximations as long as the sample distribution was artificially reduced to a stationary case. Such a conclusion gave both an empirical support and a numerical justification to a fact that has often been reported in the literature, namely, the notable discrepancy between the training and the test set in the KDD testbed. Such a critical issue had been hinted at by the proponents themselves of the competition [23], and explains the intrinsic difficulty of the challenge classification problem.

4.3 The Manuscript NIST dataset

The NIST handwritten digits database [24] provided an additional complex, real-world domain. This multiclass problem involves using three different data sets: a training set, Xtg, consisted of 60,000 samples, a test set, Xts, and a validation set, Xval, including 60,000 and 58,646 samples, respectively. The data patterns underwent the same set of pre-processing steps that had been adopted and described in [25],[28]; the resulting set of features describing each character spanned a data space having dimension 80. A far as the proposed approach is concerned, the relevant fact of the MNIST data set is that the validation set, Xval, is quite uncorrelated with respect to the previous ones [25]; such critical issue, which was known in the literature, made it possible to verify the proposed bound-validating criterion in a real case with known and documented properties.

When applying the validation procedure proposed in Section 2, empirical results highlighted the marked discrepancy that characterized the pair of training and test data, (Xtg, Xts), with respect to the pair including the training and validation set, (Xtg, Xval). Table 6 gives the experimental results obtained for the various divergence measures, for different settings of the codebook cardinality.

In all cases, divergence results clearly suggested that applying generalization theory or cross validation estimates to the validation set would yield unreliable bounds. Such a prediction was verified by testing the generalization performance of a KWM classifier (trained on the basic data set Xtg) on the MNIST validation set, Xval. Table 7 and Fig.5 report on the obtained error measures, showing that empirical generalization errors always exceeded both worst-case theoretical predictions and cross-validation estimates on the original test set. An interesting result concerns the total variation obtained divergence values: on Table 6 can note that when the divergence value $D_{TV}\left(P^{(S)}(\mathbf{x}), P^{(T)}(\mathbf{x})\right)$ is below $10 \cdot D_{TV}\left(P_J^{(S)}(\mathbf{x}), P_J^{(T)}(\mathbf{x})\right)$, then the corresponding actual error, is not so far from the theoretical bound. This observation empirically suggests that when

$D_{TV}\left(P^{(S)}(\mathbf{x}), P^{(T)}(\mathbf{x})\right)$ is over $10 \cdot D_{TV}\left(P_J^{(S)}(\mathbf{x}), P_J^{(T)}(\mathbf{x})\right)$ the non stationarity level is significant, conversely when the divergence values is below that threshold then the associated test error will be not so far from the generalization error bound.

4.4 The Email Spam Database

This dataset is from a text mining classification problem proposed in [36] derived from the SpamAssassin Apache project [37]. Raw texts of emails were transformed in a vector space model for texts; a vocabulary of terms spans the features space. The usual approach in building the vector space is counting the number of occurrences of each term [41] and then filling the data matrix with the corresponding term frequency for each text; conversely in this case only the presence or absence of a term is recorded; this leads to a sparse data matrix composed only by '0's and '1's.

Emails are about 20% of spam and the remaining are legitimate emails traffic. Emails (and so patterns) are stored in chronological order so that one can capture the drift in time. The total number of emails is 9324 that were split in 2000 emails for training and 7324 emails for test: the split was performed such that the training set is composed by the first 2000 emails and the test set by the remaining 7324 emails, thus maintaining the chronological order.

This domain has a time varying distribution on input data: this is understandable by considering that every time new spam arrives, correspondingly new words appear; this makes the data distribution non stationary in the input variable, e.g. the words of the emails and so consistent with the developed machinery.

Tables 8,9 reports the obtained results in divergences, actual error on test data, and theoretical bounds. This dataset is strongly non stationary; also in this case the divergences values are much higher than corresponding references thresholds. Correspondingly the actual error on test original data is much higher than the predicted worst case bound. As per Manuscript NIST dataset a similar observation can be carried for the total variation divergence: under the empirical threshold of $10 \cdot D_{TV}\left(P_J^{(S)}(\mathbf{x}), P_J^{(T)}(\mathbf{x})\right)$ the non stationarity makes the actual error not so far from the predicted bound.

4.5 The Daimler Pedestrian Detection Dataset

This dataset is for an automotive application: the detection problem consists in discriminating between pedestrians against background objects. This testbed is composed of 9800 8–bits grey-scale images; 4900 are pedestrian and 4900 are non-pedestrians. The dataset was split in 1225 training samples and 8575 test samples. Table 10,11 show the divergences values and the

corresponding generalization error bounds. In this case the divergence values are almost identical to the reference threshold values; some divergence values are higher than the reference ones, however these values are still much less then the empirical threshold $10 \cdot D_{TV}\left(P_J^{(S)}(\mathbf{x}), P_J^{(T)}(\mathbf{x})\right)$. This suggests that the Daimler dataset if fully stationary; this conclusion is supported and confirmed by the associated generalization bounds that are never violated by the actual error on test data.

4.6 SeaConcept Dataset

This last experiment aimed at showing what is the proper field of application of the proposed method. The SeaConcept dataset [40] is artificial, bi-class; the input space includes three randomly generated features in the range [0,1]. The dataset is composed of 60,000 samples, of which 50,000 are training samples. The training set is divided in four blocks, each block represents a *concept*. In each block a point belongs to a *concept* if, called the first feature f_1 and the second f_2 then, $f_1 + f_2 \leq \vartheta$ where ϑ represents a threshold to decide if the pattern belongs to +1, or -1 class. The *concept drift* happens when the value of ϑ changes. The values used for the threshold ϑ were 8, 9, 7, and 9.5 for the four data blocks of 12,500 patterns each as in the original work [40]. In this experiment only the training samples were used because it was already known, by definition, that the drift is present in the training set at well defined locations. The training set was then split in the four blocks, the *concepts*; then each block was considered as a training set and another block as test set. By this procedure all the 6 combinations of training-test data blocks were built and tested.

As expected when there is no drift on input data, as in this case, the proposed unsupervised method is not able to capture the non stationarity due to the changing ϑ. Table 12 show that divergences values are constant and does not signal any anomaly; conversely when looking at generalization bounds the anomaly clearly emerges because bounds are violated. Table 13 records the bound, actual errors values and the difference on ϑ between training and test concepts: interestingly when the difference on ϑ is maximal, bounds are violated almost all times; this nice feature is due the tight nature of KWM generalization error bounds.

4.7 Summarizing comments

Obtained results show some interesting common features; among the various tested divergences measures, Total Variation divergence, appeared to be the most interesting. Its values, along with the several performed experiment, are quite regular and allows to decide an empirical threshold able to distinguish among stationarity, modest non stationarity and severe non stationarity. In particular one can empirically assert the following rule of thumb:

- $D_{TV}\left(P^{(S)}(\mathbf{x}), P^{(T)}(\mathbf{x})\right) > 10 \cdot D_{TV}\left(P_J^{(S)}(\mathbf{x}), P_J^{(T)}(\mathbf{x})\right)$ means a severe non stationarity and highly probable bound violations.
- $D_{TV}\left(P_J^{(S)}(\mathbf{x}), P_J^{(T)}(\mathbf{x})\right) < D_{TV}\left(P^{(S)}(\mathbf{x}), P^{(T)}(\mathbf{x})\right) < 10 \cdot D_{TV}\left(P_J^{(S)}(\mathbf{x}), P_J^{(T)}(\mathbf{x})\right)$ indicates a modest non stationarity and possible bound violations
- $D_{TV}\left(P^{(S)}(\mathbf{x}), P^{(T)}(\mathbf{x})\right) \leq D_{TV}\left(P_J^{(S)}(\mathbf{x}), P_J^{(T)}(\mathbf{x})\right)$ means full stationarity.

The last situation is less likely to occur because in every real world dataset exist a residual "physiological" non stationarity level. As in the Daimler case or Manuscript Nist (only the test set) case when $D_{TV}\left(P^{(S)}(\mathbf{x}), P^{(T)}(\mathbf{x})\right) \cong D_{TV}\left(P_J^{(S)}(\mathbf{x}), P_J^{(T)}(\mathbf{x})\right)$ then the dataset can be reliably defined stationary.

5. CONCLUSIONS

The baseline of the presented research is that non stationarity detection is a notable practical problem, especially in data mining problems where a huge amount of samples are provided. Generalization bounds from Statistical Learning Theory tend to become practical in the presence of large samples. At the same time, huge data sets drawn from complex distributions that may be possibly time-varying or partially sampled pose the issue of the stationary nature of the observed data, which is a prerequisite for the reliability of generalization bounds.

The paper has proposed a general and robust criterion for stationarity detection with consequent generalization error validation. The method exploits a clustering-based scheme for efficiently measuring the stationary nature of the observed data, and thereby assessing the consistency of generalization error estimation in data-mining applications.

The crucial aspect of the presented approach has been the empirical nature of the experiments in practical data mining; the cluster-based support of KWMs provided by Vector Quantization was used to build up a sample-based reference model and assess the stationary nature of the observed data accordingly. Indeed, in principle, the underlying model can be applied to any clustering-based classification scheme that prevents an uncontrolled increase in the d_{VC}. The specific analysis presented in this paper was made possible by the tight bounds obtained when applying Vapnik's theory to the KWM model.

Intrusion detection in computer networks, manuscript numeral OCR, Pedestrian detection and Spam filtering have been adopted as case studies. In the first domain, the reference KDD99 data set case showed the validity of the criterion in a truly non-stationary, mission-critical context such as network-security systems. The second domain involving MNIST data made it possible to verify the

criterion effectiveness in huge data sets stemming from incomplete sampling processes of complex distributions. The third dataset confirmed the effectiveness of the approach in another real world environment such as Text Mining and the last, Daimler, provided the stationary counter-example. Moreover a final experiment, SeaConcept, underlined the field on which the proposed method applies.

The future lines of research in this area will aim at associating the divergence values between training and test data with the confidence on the bounds predicted by theory, under the operational assumption that smaller divergence values would most likely indicate more reliable bounds. At the same time, research will also investigate the possible extension of properties (confidence, etc.) of basic theoretical bounds in non-stationary situations.

REFERENCES

[1] B. Mirkin. *Clustering for Data Mining: a Data-recovery Approach*, (2006).

[2] SM .Weiss, N. Indurkhya, T. Zhang, FJ. Damerau. *Text Mining: Predictive Methods for Analyzing Unstructured Information*, Springer Science & Business Media, (2005).

[3] K. Duan, S. Keerthi, A. Poo. Evaluation of simple performance measures for tuning svm hyperparameters, Technical Report CD-01-11, Singapore, (2001).

[4] S. Floyd, M. Warmuth. Sample compression, learnability, and the Vapnik-Chervonenkis dimension, *Machine Learning*, 21 (1995), 1-36.

[5] RC. Williamson, J. Shawe-Taylor, B. Schölkopf, AJ. Smola. Sample based generalization bounds, *NeuroCOLT2 Tech. Rep. Series*, NC-TR-1999-055, (1999).

[6] M . Anthony, N. Biggs. *Computational Learning Theory*, Cambridge Univ. Press, (1992).

[7] V.N. Vapnik. *Estimation of Dependences Based on Empirical Data*, Springer-Verlag, (1982).

[8] S. Ridella, S. Rovetta, R. Zunino. K-Winner Machines for pattern classification, *IEEE Trans. on Neural Networks*, 12(2) (2001) 371-385.

[9] S. Ridella, R. Zunino. Empirical measure of multiclass generalization performance: the K-Winner Machine case, *IEEE Trans. Neural Networks*, 12(6) (2001) 1525–1529.

[10] KDD Cup 1999 Intrusion detection dataset: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.

[11] S. Ridella, S. Rovetta, R. Zunino. Plastic algorithm for adaptive vector quantization, *Neural Computing and Applications*, 7(1) (1998), 37-51.

[12] T . Kohonen. *Self-organization and Associative Memory*, third Ed., Springer Verlag, 1989.

[13] T. Martinetz, K. Schulten. Topology representing networks, *Neural Networks*, 7(3) (1994), 507-522.

[14] D. DeSieno. Adding a conscience to competitive learning, Proc. IEEE Int.Conf. on Neur.Net., SanDiego, 1 (1988), 117-124.

[15] TM. Martinetz, SG. Berkovich, KJ. Schulten, Neural Gas network for vector quantization and its application to time-series prediction, *IEEE Trans. Neural Networks*, 4(4) (1993), 558-569.

[16] MM. Van Hulle. Kernel-based equiprobabilistic topographic map formation, *Neural Computation*, 10 (1998), 1847-1871.

[17] SP. Lloyd. Least squares quantization in PCM, *IEEE Trans. Inf. Theory*, 28(2) (1982), 127-135.

[18] N. Ueda, R. Nakano. A new competitive learning approach based on an equidistortion principle for designing optimal vector quantizers, *Neural Networks*, 7(8) (1994), 1211-1228.

[19] V. Vapnik. Structure of Learning Theory, in: IEEE Int. Workshop on Neural Networks for Signal Processing, Spet. 2nd, 1995, Boston MA, IEEE Press

[20] N. Dalvi, P. Domingos, Mausam, S. Sanghai, D. Verma., Adversarial Classification, Proc. KDD'04, Seattle, Washington, USA, (2004).

[21] V. Vapnik. *Statistical Learning Theory*, John Wiley, New York, 1998, pp. 339-346

[22] I. J. Taneja and P. Kumar, Relative information of type s, Csiszar's f-divergence, and information inequalities, *Information Sciences* 166 (2004) 105–125

[23] W. Lee, S.J. Stolfo. Data mining approaches for intrusion detection, Proc. of the 7th USENIX Security Symp., San Antonio, TX, (1998).

[24] Y. LeCun, L. Bottou, Y. Bengio and P. Haffner: Gradient-Based Learning Applied to Document Recognition, *Proceedings of the IEEE*, 86(11):2278-2324, November 1998. http://yann.lecun.com/exdb/mnist/

[25] S. Ridella, R.Zunino, Using K-Winner Machines for domain analysis, *Neurocomputing* 62 (2004) pp. 367-388

[26] C. Alippi, M. Roveri Just-in-Time Adaptive Classifiers - Part I: Detecting Nonstationary Changes, *IEEE Tansactions On Neural Networks*, Vol. 19, No. 7, July 2008

[27] I. Csiszar, Information-type distance measures and indirect observations Stud. Sci. Math. Hungar, vol. 2, pp 299-318, 1967

[28] S. Ridella, S. Rovetta, R. Zunino, Circular backpropagation networks embed vector quantization, *IEEE Trans. Neural Networks* 10 (4) (1999) 972–975

[29] D. Anguita, S. Ridella, F. Rivieccio, R. Zunino Hyperparameter tuning criteria for Support Vector Classifiers, *Neurocomputing*, October 2003, No.55, pp.109-134

[30] Duan K., Keerthi S., Poo A. Evaluation of simple performance measures for tuning svm hyperparameters , 2001, Technical Report CD-01-11, Dept. of Mechanical Engineering, National University of Singapore, Singapore.

[31] E. Parzen. On estimation of a Probability Density Function and Mode. *Annals of Mathematical Statistics*, 33, 1962

[32] Y. Muto, H. Nagase, and Y. Hamamoto. Evaluation of a modified parzen classifier in high-dimensional spaces, in Proc. 15th Int'l Conf. Pattern Recognition, 2000, vol. 2, pp. 67-70.

[33] A.P. Dempster, N.M. Laird, D.B. Rubin. Maximum-likelihood from incomplete data via the EM algorithm, *J. Royal Statist. Soc*. Ser. B., 39, 1977.

[34] C. Archambeau, J.A. Lee, and M. Verleysen. On the convergence problems of the EM algorithm for finite Gaussian mixtures. In ESANN'03, pages 99–106, Bruges, Belgium.

[35] Sergio Decherchi, Paolo Gastaldo, Judith Redi, Rodolfo Zunino, Non-Stationary Data Mining: the Network Security Issue, ICANN 2008, Prague

[36] I. Katakis, G. Tsoumakas, E. Banos, N. Bassiliades, I. Vlahavas, An Adaptive Personalized News Dissemination System, *Journal of Intelligent Information Systems*, Springer, 32 (2), pp. 191-201, 2009.

[37] SpamAssassin Apache Project, http://spamassassin.apache.org/

[38] H. Shimodaira, Improving predictive inference under covariate shift by weighting the log likelihood function, *Journal of Statistical Planning and Inference, 90(2) pp.227-244, 2000.*

[39] Munder, S., Gavrila, D.M. An Experimental Study on Pedestrian Classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 28, pp. 1863-1868, 2006.*

[40] W. N. Street and Y. Kim, A streaming ensemble algorithm (SEA) for large-scale classification, *Knowledge Discovery & Data Mining, pp. 377-382, 2001.*

[41] S. Decherchi, P. Gastaldo, R. Zunino *"K-Means clustering for Content Based Document Management in Intelligence",* in "Advances In Artificial Intelligence for Privacy Protection and Security", Editors: Augusti Solanas and Antoni Martinez Bellesté, World Scientific Publishing, 2009

Appendix. The K-Winner-Machine training algorithm

The following pseudocode sketches the KWM training algorithm, as proposed in [8]. The conventions are repeated for the reader's convenience:

- $\mathrm{C} = \left\{ c^{(h)}, h = 1,..., N_c \right\}$ is the set of N_c possible pattern classes;
- A real-valued vector ($\mathbf{x}_l$, $c(\mathbf{x}_l)$) denotes a labeled pattern drawn from the space $\Re^D$;
- $\mathrm{W}' = \left\{ (\mathbf{w}_n, c_n), \mathbf{w}_n \in \Re^D, c_n \in C, n = 1,..., N_h \right\}$ is a set of N_h labeled prototypes;
- $\mathbf{w}^*(\mathbf{x}) = \arg\min_{\mathbf{w} \in \mathrm{W}'} \left\{ \|\mathbf{x} - \mathbf{w}\|^2 \right\}$ is the prototype that represents a pattern, $\mathbf{x}$;
- P_n is the data-space partition that is covered by the *n*-th prototype according to a minimum-distance criterion; it coincides with the Voronoi region associated with $\mathbf{w}_n$;
- $\alpha_n^{(h)}, h = 1,.., N_c$ is the share of patterns that lie in P_n and belong to the *h*-th class; as a consequence of this definition, one has that, for each n-th partition: $\left(\sum_{h=1}^{N_c} \alpha_n^{(h)} = 1 \right)$.

K-Winner-Machine training algorithm

1. *Input*: training set of labelled data, X; confidence level: $0 < \eta \leq 1$;
2. (*Unsupervised prototype creation*)
   Apply an unsupervised VQ algorithm to train a set, W, of N_h prototypes.
3. (*Calibration*)
   Calibrate W into a labeled set of prototypes, W′, computed as:
   $\mathrm{W}' = \left\{ (\mathbf{w}_n, c_n), \mathbf{w}_n \in \mathrm{W}, c_n \in \mathrm{C}, n = 1,..., N_h \right\}$, where: $c_n = c_b, b = \max_k \left\{ \alpha_n^{(k)} \right\}$.
4. (*Init*)
   reset optimal bound π^* := 1.
   reset counters $N(k) := 0, \quad \nu(k) := 0, \quad k=1,\ldots,N_h$.
5. (*Concurrence verification and counting*)
   For each training pattern, $\mathbf{x}_l \in \mathrm{X}$:

   Begin loop

   5.1. Sort the set of prototypes, *W″*, arranging them in order of increasing distance from $\mathbf{x}_l$:
   $\mathrm{W}''(\mathbf{x}_l) = \left\{ (\mathbf{w}_{n_r}, c_{n_r}), \ \mathbf{w}_{n_r} \in \mathrm{W}' : r < s \Rightarrow \|\mathbf{x}_l - \mathbf{w}_{n_r}\| \leq \|\mathbf{x}_l - \mathbf{w}_{n_s}\|; \ r, s = 1,..., N_h \right\}$;

5.2. (*Count errors and concurrences for each agreement level*)

For each $k = 1, \ldots, N_h$

Begin loop

5.2.1. *(Determine empirical classification error)*

Let $(\mathbf{w}_1, c_1)$ be the first element of $\mathrm{W}''(\mathbf{x}_l)$ – i.e., the closest prototype to $\mathbf{x}_l$

If $c_1 \neq c(\mathbf{x}_l)$

Then $\nu(k) := \nu(k) + 1$

5.2.2. *(Count concurrences)*

Extract the set $\mathrm{W}_k(\mathbf{x}_l) \subseteq \mathrm{W}''(\mathbf{x}_l)$ including the k closest neighbors to $\mathbf{x}_l$:

$\mathrm{W}_k(\mathbf{x}_l) = \{\mathbf{w}_{n_r} \in \mathrm{W}''(\mathbf{x}_l), r = 1, \ldots, k\}$

5.2.3. Increment k-th counter if a full agreement is found among the elements in $\mathrm{W}_k(\mathbf{x}_l)$:

If $\exists c^* : \forall \mathbf{w}_{n_r} \in \mathrm{W}_k(\mathbf{x}_l)\ c_{n_r} = c^*$

Then $N(k) := N(k) + 1$

end loop

end loop

6. (*Risk estimation*)

For each $k = 1, \ldots, N_h$

Begin loop

If $N(k) > 0$

Begin then

6.1. Compute the empirical error associated with the k-th level:

$\nu(k) := \nu(k) / N(k)$

6.2. Compute the generalization bound, $\pi(k)$, for the k-th agreement level as:

$$\pi(k) = \nu(k) + \frac{\varepsilon(k)}{2}\left(1 + \sqrt{1 + \frac{4\nu(k)}{\varepsilon(k)}}\right), \text{ where } \varepsilon(k) = \frac{4}{N(k)}\left[\lfloor N_h / k \rfloor \cdot \ln N_c - \ln \frac{\eta}{4}\right]$$

6.3. Update level if agreement improves risk bound:

If $\pi(k) < \pi^*$

Then $\pi^* := \pi(k)$

Else $\pi(k) := \pi^*$

end then

Else $\pi(k) := \pi(k-1)$

end loop

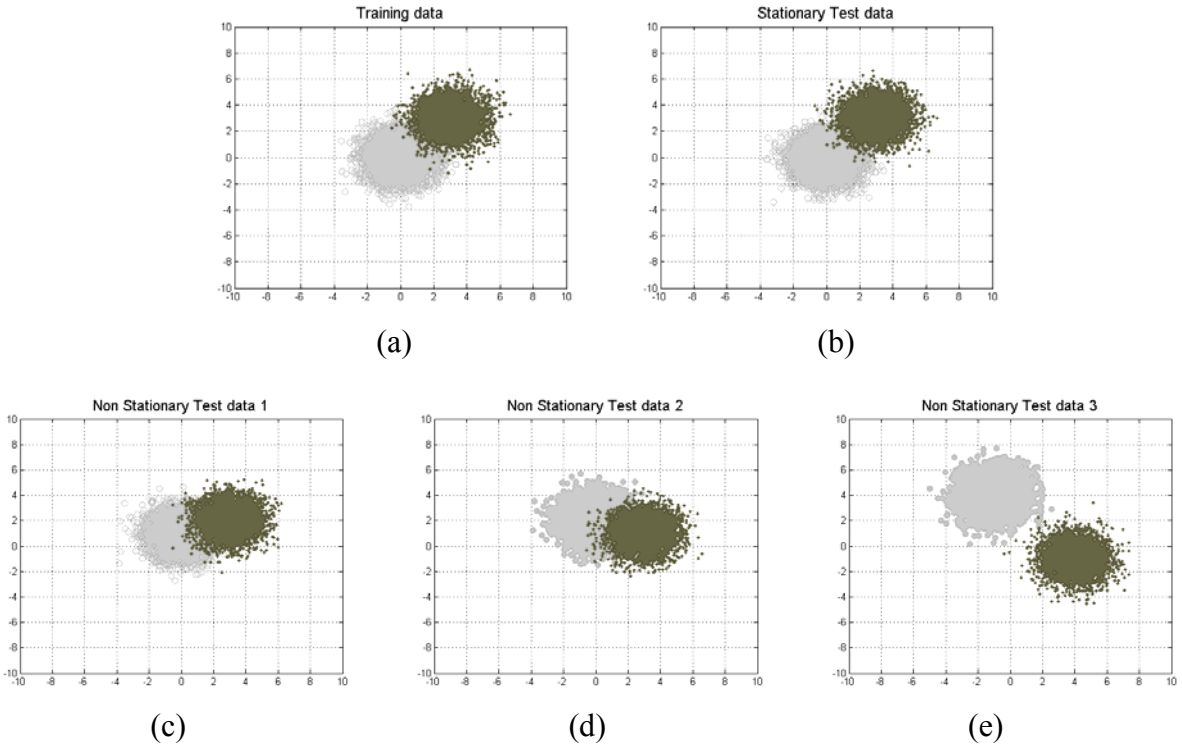

Fig 1. 2-D artificial dataset with a non-stationary data distribution.

a) Training Set, Xtg b) Stationary Test Set, Xts c) Test set, Xts1 d) Test set, Xts2. e) Test set, Xts3

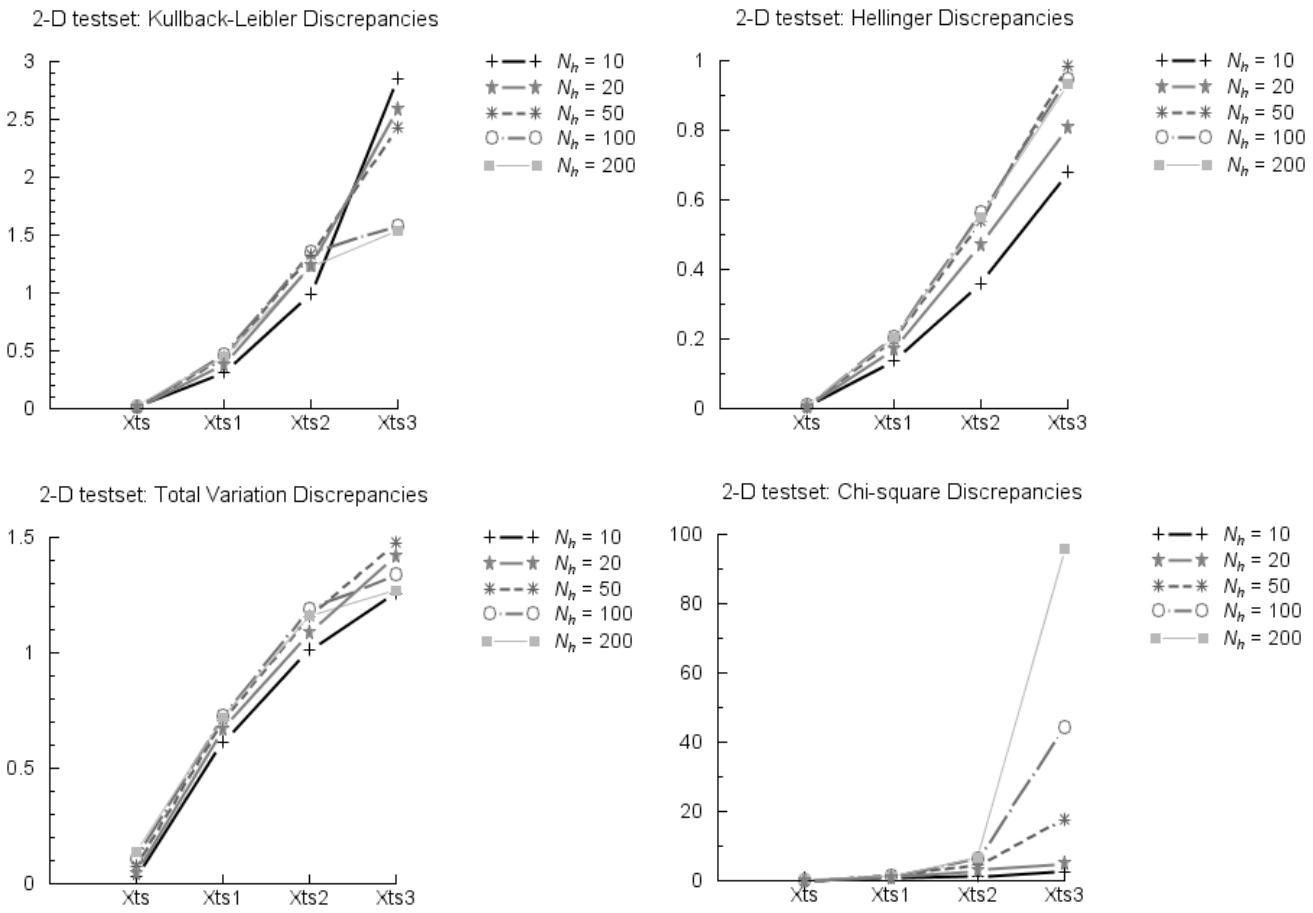

Fig. 2 - Discrepancy measurements for the 2-D artificial experiment.
The curves are parameterized by the number of prototypes, N_h.
Discrepancy values increase in the presence of non-stationary distributions of data.

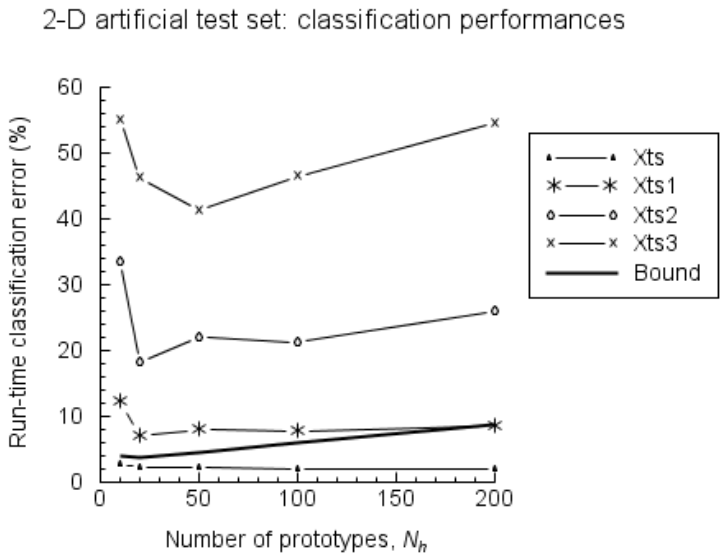

Fig.3 - Generalization bounds and true classification performances.
Theoretical predictions may prove unreliable by the presence of non-stationary distributions;
Xts is the only case involving a stationary distribution.

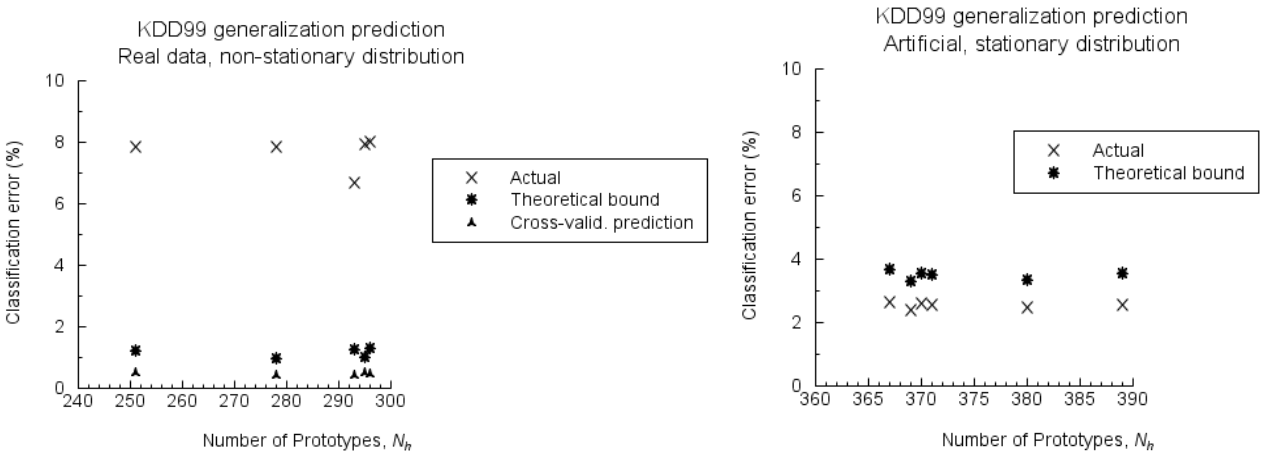

Fig. 4 – KDD dataset: validation of generalization predictions
a) original data b) artificial stationary distribution

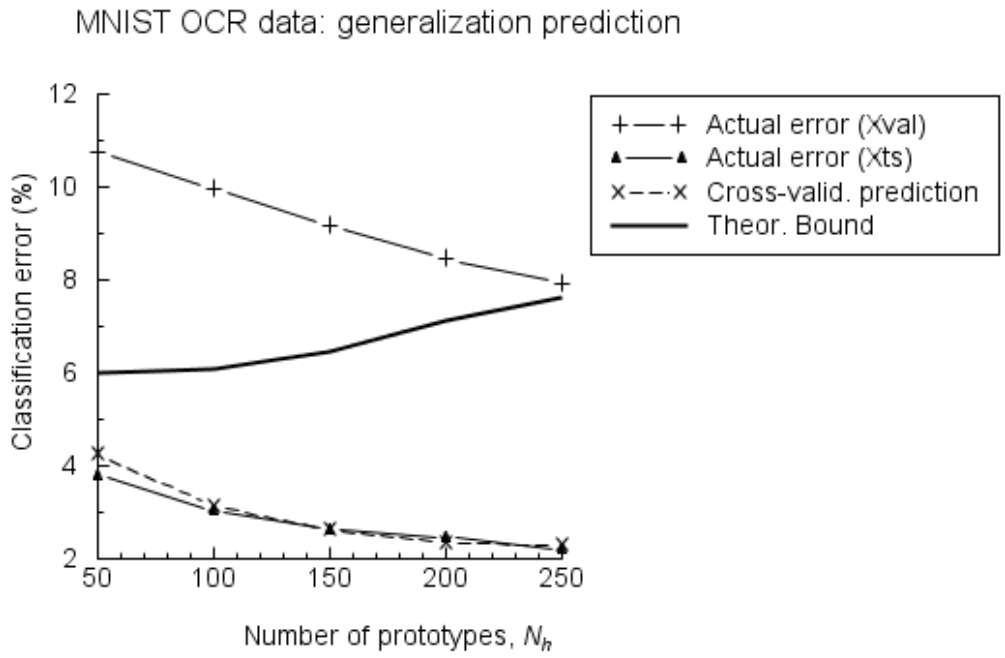

Fig. 5 – MNIST OCR domain: Predicted error performances and actual generalization performances for stationary and non-stationary test sets

Table 1 – Theoretical formulation of divergence measures derived from the general class of *f*-divergences.

Divergence	**Notation**	**Function**
Kullback-Liebler	$D_{KL}\left(P^{(\mathrm{S})}(\mathbf{x}), P^{(\mathrm{T})}(\mathbf{x})\right)$	$f\left(\frac{s_n}{t_n}\right) = \frac{s_n}{t_n}\ln\frac{s_n}{t_n}$
Hellinger	$D_H\left(P^{(\mathrm{S})}(\mathbf{x}), P^{(\mathrm{T})}(\mathbf{x})\right)$	$f\left(\frac{s_n}{t_n}\right) = \left(\sqrt{\frac{s_n}{t_n}} - 1\right)^2$
Total Variation	$D_{TV}\left(P^{(\mathrm{S})}(\mathbf{x}), P^{(\mathrm{T})}(\mathbf{x})\right)$	$f\left(\frac{s_n}{t_n}\right) = \left\lvert\frac{s_n}{t_n} - 1\right\rvert$
Pearson (Chi-square)	$D_P\left(P^{(\mathrm{S})}(\mathbf{x}), P^{(\mathrm{T})}(\mathbf{x})\right)$	$f\left(\frac{s_n}{t_n}\right) = \left(\frac{s_n}{t_n} - 1\right)^2$

Table 2 – Discrepancy values for the artificial dataset pairs.

(a)

Kullback-Leibler divergence, D_{KL}

N_h	Xts	Xts1	Xts2	Xts3
10	0.000852	0.304446	0.98017	2.847543
20	0.001781	0.372826	1.23305	2.582903
50	0.004367	0.43368	1.32496	2.417632
100	0.009302	0.460772	1.34089	1.573968
200	0.017330	0.441207	1.22429	1.528121

(b)

Hellinger divergence, D_H

N_h	Xts	Xts1	Xts2	Xts3
10	0.000428	0.136163	0.3556	0.677328
20	0.000897	0.170551	0.47083	0.806017
50	0.002196	0.194748	0.53416	0.981338
100	0.004583	0.204389	0.5598	0.945832
200	0.008438	0.203124	0.54957	0.933226

(c)

Total-Variation divergence, D_{TV}

N_h	Xts	Xts1	Xts2	Xts3
10	0.0298	0.6068	1.0086	1.2560
20	0.0414	0.6634	1.0820	1.4142
50	0.0718	0.7034	1.1546	1.4696
100	0.1048	0.7216	1.1858	1.3340
200	0.1370	0.7183	1.1605	1.2638

(d)

Pearson (Chi-square) divergence, D_P

N_h	Xts	Xts1	Xts2	Xts3
10	0.00172	0.50750	1.21620	2.22382
20	0.00368	0.77494	2.58869	4.72398
50	0.00901	0.94265	4.53441	17.34433
100	0.01789	0.99850	5.93358	43.91379
200	0.03280	1.04357	6.65349	95.72581

Table 3 – Theoretical bounds and actual classification errors for the artificial dataset pairs.

N_h	Training Error	Theoretical bound	Xts error	Xts1 error	Xts2 error	Xts3 error
10	2.78%	3.93%	2.59%	12.20%	33.40%	54.93%
20	2.05%	3.52%	2.16%	7.00%	18.23%	46.11%
50	1.89%	4.41%	2.18%	7.95%	21.90%	41.20%
100	1.81%	5.89%	1.85%	7.53%	21.10%	46.31%
200	1.67%	8.62%	1.82%	8.44%	25.80%	54.49%

Table 4 - KDD99: measured divergence values
for the original distributions (T,S) and the reference stationary distributions (T_J,S_J).

N_h	Distribution	D_{KL}	D_H	D_{TV}	D_P
250	Real (T,S)	0.988	0.3	0.6910379	88.868469
277	Real (T,S)	0.997	0.301	0.6970229	94.806881
292	Real (T,S)	0.961	0.305	0.6987374	44.565382
294	Real (T,S)	1	0.303	0.6999451	82.754549
295	Real (T,S)	1.03	0.303	0.6969399	62.729348
367	Stationary (T_J,S_J)	0.0026	0.001299	0.0399745	0.0055132
369	Stationary (T_J,S_J)	0.00297	0.00147	0.0421336	0.0061434
370	Stationary (T_J,S_J)	0.0029	0.001414	0.0403167	0.0060123
371	Stationary (T_J,S_J)	0.00296	0.001448	0.0426404	0.0061934
380	Stationary (T_J,S_J)	0.0032	0.00155	0.0442762	0.0066035
388	Stationary (T_J,S_J)	0.003	0.00141	0.0423159	0.0059061

Table 5 - KDD99: training classification errors, test set (actual) error, and theoretical bounds for the original datasets (T,S) and the reference stationary datasets (T_J,S_J)

Nh	Distrib.	Training Error	Actual error	Theoretical bound
251	Real (T,S)	0.58%	7.83%	1.21%
278	Real (T,S)	0.36%	7.81%	0.95%
293	Real (T,S)	0.54%	6.66%	1.23%
295	Real (T,S)	0.37%	7.91%	0.99%
296	Real (T,S)	0.56%	8.00%	1.26%
367	Stationary (T_J,S_J)	2.32%	2.61%	3.65%
369	Stationary (T_J,S_J)	2.01%	2.36%	3.27%
370	Stationary (T_J,S_J)	2.22%	2.58%	3.53%
371	Stationary (T_J,S_J)	2.17%	2.52%	3.47%
380	Stationary (T_J,S_J)	2.03%	2.42%	3.32%
389	Stationary (T_J,S_J)	2.17%	2.51%	3.51%

Table 6 – MNIST OCR domain: divergence values
for the (stationary) test pair (Xtg,Xts) and the validation pair (Xtg,Xval).

N_h	(Xtg, Xts)	(Xtg, Xval)	(Xtg, Xts)	(Xtg, Xval)
	Kullback-Leibler, D_{KL}		Hellinger, D_H	
50	0.000454174	0.093606572	0.0002271	0.0456244
100	0.001051512	0.14979364	0.0005248	0.0713274
150	0.00176112	0.191966419	0.0008796	0.0891992
200	0.002327155	0.223943361	0.001161	0.1048904
250	0.003319328	0.236428103	0.0016586	0.1096719
300	0.003698878	0.252772341	0.0018461	0.1180814
	Total Variation, D_{TV}		Pearson (ChiSq), D_P	
50	0.0244	0.3375183	0.000909213	0.192429682
100	0.0365667	0.4301032	0.002090747	0.296484197
150	0.0459667	0.4692303	0.003517104	0.367802702
200	0.0533	0.5187436	0.004630067	0.460485661
250	0.0654667	0.5222731	0.006656246	0.489166082
300	0.0673	0.5428892	0.007398064	0.551202225

Table 7 – MNIST OCR domain: predicted and empirical classification errors
for the (stationary) test pair (Xtg,Xts) and the validation pair (Xtg, Xval)

N_h	Training Error	Theoretical bound	Cross-valid. prediction	Actual classif. error rate	
				(Xtg,Xts)	(Xtg,Xval)
50	3.84%	6.00%	4.25%	3.79%	10.75%
100	3.01%	6.07%	3.12%	3.02%	9.96%
150	2.59%	6.45%	2.61%	2.60%	9.14%
200	2.44%	7.12%	2.32%	2.43%	8.46%
250	2.20%	7.61%	2.29%	2.19%	7.90%

Table 8 – Spam Assassin: measured divergence values
for the original distributions (T,S) and the reference stationary distributions (T_J,S_J).

N_h	Distribution	D_{KL}	D_H	D_{TV}	D_P
10	Real (T,S)	0.6857924	0.257748859	0.884818132	0.882226675
20	Real (T,S)	0.625897052	0.253369299	0.881237575	0.952071543
50	Real (T,S)	0.6583435	0.263773386	0.837421354	0.95870261
70	Real (T,S)	0.372328109	0.214588784	0.781716201	0.982382522
10	Stationary (T_J,S_J)	0.001462837	0.000733108	0.036357728	0.002974508
20	Stationary (T_J,S_J)	0.003157137	0.001554101	0.050715456	0.006035152
50	Stationary (T_J,S_J)	0.012887348	0.006095046	0.098995085	0.02310027
70	Stationary (T_J,S_J)	0.01575843	0.012957167	0.141682886	0.047113559

Table 9 - Spam: training classification errors, test set (actual) error, and theoretical bounds
for the original datasets (T,S) and the reference stationary datasets (T_J,S_J)

Nh	Distrib.	Training Error	Actual error	Theoretical bound
10	Real (T,S)	12.85%	43.68%	19.49%
20	Real (T,S)	7.95%	32.38%	15.46%
50	Real (T,S)	5.40%	26.80%	16.88%
70	Real (T,S)	7.05%	27.13%	22.47%
10	Stationary (T_J,S_J)	11.3%	11.63%	17.61%
20	Stationary (T_J,S_J)	9.65%	10.3%	17.68%
50	Stationary (T_J,S_J)	7%	7.74%	19.26%
70	Stationary (T_J,S_J)	5.9%	6.64%	20.7%

Table 10 - Daimler: measured divergence values
for the original distributions (T,S) and the reference stationary distributions (T_J,S_J).

N_h	Distribution	D_{KL}	D_H	D_{TV}	D_P
10	Real (T,S)	0.002415877	0.001211102	0.062040816	0.0048926
50	Real (T,S)	0.02385615	0.011747066	0.163965015	0.01275511
100	Real (T,S)	0.063620058	0.03018475	0.251195335	0.07802638
10	Stationary (T_J,S_J)	0.002681746	0.001354378	0.057609329	0.005606701
50	Stationary (T_J,S_J)	0.025458888	0.01252994	0.167696793	0.011140393
100	Stationary (T_J,S_J)	0.056902696	0.026487928	0.244198251	0.054021554

Table 11 - Daimler: training classification errors, test set (actual) error, and theoretical bounds
for the original datasets (T,S) and the reference stationary datasets (T_J,S_J)

Nh	Distrib.	Training Error	Actual error	Theoretical bound
10	Real (T,S)	24.16%	22.67%	35.63%
50	Real (T,S)	15.34%	14.06%	37.08%
100	Real (T,S)	13.00%	14.53%	46.42%
10	Stationary (T_J,S_J)	19.26%	21.37%	29.74%
50	Stationary (T_J,S_J)	14.69%	14.90%	36.16%
100	Stationary (T_J,S_J)	14.11%	14.32%	48.15%

Table 12 – SEAconcepts: measured divergence values

	Concept 1 vs. Concept 2 ($\Delta\vartheta = 1$)				Concept 1 vs. Concept 3 ($\Delta\vartheta = 1$)			
N_h	D_{KL}	D_H	D_{TV}	D_P	D_{KL}	D_H	D_{TV}	D_P
10	0,000436	0,000218	0,024	0,0008707	0,000608	0,000304	0,03072	0,001211
20	0,001429	0,000712	0,0416	0,0028198	0,00162	0,000814	0,04416	0,0033111
40	0,002268	0,001149	0,05232	0,0048014	0,00278	0,001388	0,06096	0,0055544
60	0,004282	0,002156	0,07712	0,0088516	0,004443	0,00224	0,0728	0,0092533

	Concept 1 vs. Concept 4 ($\Delta\vartheta = 1.5$)				Concept 2 vs. Concept 3 ($\Delta\vartheta = 2$)			
N_h	D_{KL}	D_H	D_{TV}	D_P	D_{KL}	D_H	D_{TV}	D_P
10	0,000425	0,000213	0,0232	0,0008607	0,000562	0,000281	0,02912	0,0011305
20	0,001346	0,000672	0,04112	0,0026782	0,00107	0,000534	0,03824	0,0021213
40	0,002728	0,001356	0,05344	0,0053603	0,002219	0,001112	0,05312	0,0044954
60	0,006494	0,003213	0,09168	0,0125705	0,004145	0,002075	0,07248	0,0083941

	Concept 2 vs. Concept 4 ($\Delta\vartheta = 0.5$)				Concept 3 vs. Concept 4 ($\Delta\vartheta = 2.5$)			
N_h	D_{KL}	D_H	D_{TV}	D_P	D_{KL}	D_H	D_{TV}	D_P
10	0,00046	0,00023	0,0224	0,0009184	0,000706	0,000354	0,03152	0,0014328
20	0,001288	0,000648	0,03728	0,00265	0,001532	0,000763	0,04272	0,0030254
40	0,002453	0,001227	0,05664	0,0049329	0,003494	0,001764	0,06	0,0073208
60	0,005226	0,002613	0,07872	0,0105285	0,005818	0,002905	0,08464	0,0116785

Table 13 – SEAconcepts: test errors and theoretical bounds

Concept 1 vs. Concept 2 ($\Delta\mathcal{G} = 1$)			Concept 1 vs. Concept 3 ($\Delta\mathcal{G} = 1$)		Concept 1 vs. Concept 4 ($\Delta\mathcal{G} = 1.5$)	
N_h	Test Error	Theoretical bound	Test Error	Theoretical bound	Test Error	Theoretical bound
10	**23,74%**	**20,03%**	14,51%	20,03%	**23,87%**	**20,03%**
20	**19,42%**	**19,26%**	16,52%	19,26%	19,10%	19,26%
40	19,84%	20,94%	16,84%	20,94%	19,40%	20,94%
60	17,78%	19,99%	15,71%	19,99%	17,42%	19,99%

Concept 2 vs. Concept 3 ($\Delta\mathcal{G} = 2$)			Concept 2 vs. Concept 4 ($\Delta\mathcal{G} = 0.5$)		Concept 3 vs. Concept 4 ($\Delta\mathcal{G} = 2.5$)	
N_h	Test Error	Theoretical bound	Test Error	Theoretical bound	Test Error	Theoretical bound
10	17,86%	25,37%	22,88%	25,37%	**23,97%**	**17,67%**
20	**22,12%**	**21,28%**	17,75%	21,28%	**22,00%**	**19,25%**
40	18,47%	21,45%	16,44%	21,45%	20,42%	20,71%
60	20,58%	20,79%	14,80%	20,79%	**23,52%**	**19,50%**