Expert Systems With Applications 71 (2017) 216–229

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

Applying a kernel function on time-dependent data to provide supervised-learning guarantees

Lucas de Carvalho Pagliosa*, Rodrigo Fernandes de Mello

Institute of Mathematics and Computer Science, University of São Paulo, Av. Trabalhador São-carlense, 400, São Carlos, SP, Brazil

ARTICLE INFO

Article history:
Received 28 July 2016
Revised 28 October 2016
Accepted 19 November 2016
Available online 25 November 2016

Keywords:
Statistical Learning Theory
Time dependency
Kernel function
Takens' immersion theorem
Supervised-learning algorithms

ABSTRACT

The Statistical Learning Theory (SLT) defines five assumptions to ensure learning for supervised algorithms. Data independency is one of those assumptions, once the SLT relies on the Law of Large Numbers to ensure learning bounds. As a consequence, this assumption imposes a strong limitation to guarantee learning on time-dependent scenarios. In order to tackle this issue, some researchers relax this assumption with the detriment of invalidating all theoretical results provided by the SLT. In this paper we apply a kernel function, more precisely the Takens' immersion theorem, to reconstruct time-dependent open-ended sequences of observations, also referred to as data streams in the context of Machine Learning, into multidimensional spaces (a.k.a. phase spaces) in attempt to hold the data independency assumption. At first, we study the best immersion parameterization for our kernel function using the Distance-Weighted Nearest Neighbors (DWNN). Next, we use this best immersion to recursively forecast next observations based on the prediction horizon, estimated using the Lyapunov exponent. Afterwards, predicted observations are compared against the expected ones using the Mean Distance from the Diagonal Line (MDDL). Theoretical and experimental results based on a cross-validation strategy provide stronger evidences of generalization, what allows us to conclude that one can learn from time-dependent data after using our approach. This opens up a very important possibility for ensuring supervised learning when it comes to time-dependent data, being useful to tackle applications such as in the climate, animal tracking, biology and other domains.

1. Introduction

Technology advances have allowed and motivated the study of evolving phenomena such as population growth, climate change, stock market, spread of diseases, etc. (Robledo & Moyano, 2007; Tucker, 1999). Those phenomena are characterized by the continuous production of data along time, which need to be modeled in order to classify, predict and understand behavior changes (de Mello, 2011). Despite this type of data is typically referred to as time series by the areas of Physics and Statistics, it has also been named as data streams by Machine Leaning (ML). In the context of ML, two main types of approaches have been used to tackle data stream modeling: the supervised and the unsupervised one.

Supervised machine learning approaches attempt to find the best function $f : \mathcal{X} \rightarrow \mathcal{Y}$ to map examples in the input space (whose dimensions correspond to attributes) $x \in \mathcal{X}$ into the correct label or class $y \in \mathcal{Y}$. Function f is also referred to as classifier, when $\mathcal{Y}$ has discrete labels, or to as regression function, when labels are continuous. Although supervised algorithms require class information, they rely on the Statistical Learning Theory (SLT) (Luxburg & SchÈµlkopf, 2011) to provide the formal framework to ensure learning conditions. On the other hand, unsupervised learning attempts to model the structural organization present in data space, according to similarity measures (Kanungo et al., 2002; Murtagh & Contreras, 2011). Besides this second approach of learning does not require class information, it does not count either on the same theoretical foundation to support results.

To take full advantage of SLT, a set of assumptions must be satisfied: (i) examples must be independent from each other and sampled in an identical manner; (ii) no assumption is made about the joint probability distribution, otherwise one could simply estimate its parameters; (iii) labels can assume nondeterministic values due to noise and class overlapping; (iv) the joint probability distribution is fixed, i.e., it cannot change along time; and (v) data distribution is still unknown at the time of training, thus it must be estimated using training examples. However, some of those assumptions are very difficult to hold when dealing with

* Corresponding author. Lucas de Carvalho Pagliosa
E-mail addresses: lucas.pagliosa@usp.br (L. de Carvalho Pagliosa), mello@icmc.usp.br (R.F. de Mello).

http://dx.doi.org/10.1016/j.eswa.2016.11.028
0957-4174/

data streams, specially (i) and (iv). Assumption (i), for instance, makes inappropriate the modeling of data observations or examples when they present some dependencies to others. For example, consider data produced by a time series in form:

$$x(t+1) = rx(t)(1-x(t)) \tag{1}$$

in which a next data observation $x(t+1)$ depends on the previous one $x(t)$ and a constant r. This dependency makes the Empirical Risk Minimization Principle (Vapnik, 1992), defined by the SLT (more details in Section 3), inconsistent and, therefore, it no longer guarantees learning for any classifier inferred upon the respective time series. In this scenario, the sampling of a data observation changes the probability of others to be sampled next.

Assumption (iv), on the other hand, forces the joint probability distribution $P(\mathcal{X}, \mathcal{Y})$ to be static, i.e., it cannot change along time. This distribution defines how input space $\mathcal{X}$ is related to output space $\mathcal{Y}$, i.e., how data attributes are associated to classes. This assumption is necessary once the SLT considers the Law of Large Numbers (Révész, 1967) to formalize and guarantee the empirical risk (or training error) as a fair estimator to the real risk (error for unseen examples), what is also known as the Bias-Variance Dilemma (Geman, Bienenstock, & Doursat, 1992). Assumption (iv) is in fact an issue for the current data stream applications, being even more relevant when approaching concept drift (Gama, Žliobaitė, Bifet, Pechenizkiy, & Bouchachia, 2014; Zliobaite, 2010) as it assumes changes in the joint probability distribution along time.

Besides the issues associated to assumptions (i) and (iv), several studies still employ supervised learning in the context of data streams and other temporal data modeling. Faria, Gama, and Carvalho (2013) survey and discuss a list of studies devoted to data stream one-class (Denis, Gilleron, & Letouzey, 2005; Spinosa, de Leon F. de Carvalho, & Gama, 2009; Tan, Ting, & Liu, 2011; Tax & Duin, 2008) and multi-class classification (Al-Khateeb et al., 2012; Al-Khateeb, Masud, Khan, & Thuraisingham, 2012; Farid & Rahman, 2012; Masud, Gao, Khan, Han, & Thuraisingham, 2011; Masud et al., 2010). Besides relevant contributions, those studies relax both assumptions (i) and (iv) and, therefore, provide results that are not fully supported by SLT when dealing with time-dependent data.

After studying both assumptions for data stream scenarios, we decided to tackle assumption (i) in this paper, which we believe to be the basic step to provide learning guarantees to the time-dependent domain. We know assumption (iv) must also be addressed, however we leave it for future work. Thus, instead of relaxing assumption (i), we propose to hold it by using a kernel function, based on Dynamical System concepts, to reconstruct data streams into multidimensional spaces (a.k.a. phase spaces). By ensuring assumption (i), we intend to provide learning guarantees according to the SLT, so one can have higher confidence about modeling and predicting data stream observations along time. This kernel function applies Takens' immersion theorem (Takens, 1981) to map time dependencies of data stream observations into the axes of a phase space (Alligood, Sauer, & Yorke, 1996; Kantz & Schreiber, 1997).

In order to illustrate our proposal, consider a data stream produced by the system defined in Eq. (1), which was reconstructed by the kernel function into the phase space shown in Fig. 1b. The abscissa is associated to $x(t)$ and the ordinate to $x(t+1)$, therefore time is no longer represented by an explicit axis such as in Fig. 1a. After this reconstruction, we organize data examples (which are here seen as points in the phase space) as shown in Table 1 in order to proceed with classification or regression.

As reported in the ML literature, the Bayes theorem is useful to formalize learning. According to this theorem, every data example $x \in \mathcal{X}$ must have meaningful attribute values to estimate the output class $y \in \mathcal{Y}$, therefore input attributes must have some dependency on the expected outcome (Bishop, 2006). So, by applying the kernel function, we intend to obtain examples that represent the dependencies among data stream observations over time (in this situation, every output $x(t+1)$ depends on $x(t)$). As consequence, one can uniformly sample data examples from this dataset (as illustrated in Table 1) to infer some classification or regression model, what can be seen as a selection of points in the phase space under the same probability distribution (Fig. 2)–similar approaches have been considered in literature (Carlsson & Mémoli, 2013).

Table 1
Data stream observations organized in a tabular form after applying the kernel function (Takens' immersion theorem using embedding parameters $m = 2$ and $d = 1$, see more details in Section 2).

Points	$\mathcal{X}$	$\mathcal{Y}$
$(x(t), x(t+1))$	$x(t)$	$x(t+1)$
$(x(t+1), x(t+2))$	$x(t+1)$	$x(t+2)$
$(x(t+2), x(t+3))$	$x(t+2)$	$x(t+3)$
$(x(t+3), x(t+4))$	$x(t+3)$	$x(t+4)$
$(x(t+4), x(t+5))$	$x(t+4)$	$x(t+5)$
⋮	⋮	⋮

In this context, if the inferred model produces a good generalization (i.e., it learns) for data examples, so assumption (i) was held. To summarize, if: (i) dependencies are defined in terms of every data example, i.e., every row such as in Table 1; (ii) data examples can be uniformly sampled to compose a training set in order to build a classification or regression model; and (iii) we obtain good enough results using some cross-validation criterion for training and test sets; thus we conclude the kernel function produces some immersion in another multidimensional space which is capable of translating data stream observations into independent and identically distributed (i.i.d.) examples.

Therefore, the main contribution of this paper is to hold assumption (i) so the uniform convergence is respected as required by the SLT to ensure learning (Scholkopf & Smola, 2001). As insightful implication, the uniform convergence makes possible to find the best classifier or regression function to represent a given problem (within the algorithm bias), what improves forecasting results for any application domain. In practical terms, instead of relaxing assumption (i), one can rely on the theoretical framework provided by SLT to ensure learning for both classification and regression tasks on time-dependent data.

We have performed two different sets of experiments using the Distance-Weighted Nearest Neighbors (DWNN) algorithm, a variation of the k-Nearest Neighbors (kNN), which has been proved to learn according to SLT (Luxburg & SchÈµlkopf, 2011): (i) the first set assessed different phase-space reconstructions (kernel function parameterizations) for synthetic and real-world data streams, and then we proceeded by selecting the space producing the best classifier, i.e., with the greatest generalization capability (the one in which the empirical risk is the best estimator for the real risk Luxburg & SchÈµlkopf, 2011); (ii) the second set considered a recurrent forecasting on top of the best phase-space reconstructions to quantify the accumulated error and evaluate the generalization capacity of the inferred classifiers. Then, we conclude those classifiers provide good enough results and confirm the kernel function (under the correct parameterization) is capable of translating time-dependent data stream observations into i.i.d. examples. From this, we corroborate the phase-space reconstruction provides a relevant tool to compose training sets when one needs to model and predict time series or any other time-dependent data such as commonly found in the Data Stream scenario.

This paper is organized as follows: Section 2 briefly introduces the importance of the phase-space reconstruction to analyze time-