A stable and online approach to detect concept drift in data streams

Brazilian Conference on Intelligent Systems, 3th, 2014, São Carlos.
http://www.producao.usp.br/handle/BDPI/48597

Downloaded from: Biblioteca Digital da Produção Intelectual - BDPI, Universidade de São Paulo
A Stable and Online Approach to Detect Concept Drift in Data Streams

Fausto Guzzo da Costa  
Instituto de Ciências Matemáticas e de Computação  
Universidade de São Paulo  
São Carlos, Brazil  
Email: fausto at icmc dot usp dot br

Rodrigo Fernandes de Mello  
Instituto de Ciências Matemáticas e de Computação  
Universidade de São Paulo  
São Carlos, Brazil  
Email: mello at icmc dot usp dot br

Abstract—The detection of concept drift allows to point out when a data stream changes its behavior over time, what supports further analysis to understand why the phenomenon represented by such data has changed. Nowadays, researchers have been approaching concept drift using unsupervised learning strategies, due to data streams are open-ended sequences of data which are extremely hard to label. Those approaches usually compute divergences of consecutive models obtained over time. However, those strategies tend to be imprecise as models are obtained by clustering algorithms that do not hold any stability property. By holding a stability property, clustering algorithms would guarantee that a change in clustering models correspond to actual changes in input data. This drawback motivated this work which proposes a new approach to model data streams by using a stable hierarchical clustering algorithm. Our approach also considers a data stream composed of a mixture of time-dependent and independent observations. Experiments were conducted using synthetic data streams under different behaviors. Results confirm this new approach is capable of detecting concept drift in data streams.

Keywords—data stream; concept drift; clustering; stability;

I. INTRODUCTION

Data streams are open-ended sequences of data produced by different types of applications, such as monitoring the industrial production, the weather and the Internet. They are characterized by large volumes of data produced at high frequencies. In the last decades, the attention of several researchers have been focused on the design of approaches to detect concept drift in data streams [1], [2]. Such detection allows to know when the phenomenon changes its behavior over time, what supports further analysis to understand why it has changed. There are two main approaches to detect concept drift [2], [3]. The first is based on supervised learning which requires labeled data to produce an initial model to be compared to future data. After detecting a concept drift, the supervisor needs to label more data and another model has to be produced for further comparisons. However, when we suppose a data stream application, besides data is produced at high volumes we also consider it is produced at high frequencies, what makes such a supervision unfeasible. On the other hand, this approach has a formal framework which provides learning guarantees according to the Empirical Risk Minimization (ERM) principle [4]. Consequently, it is possible to obtain theoretical boundaries to guarantee the correct concept drift detection.

The second approach is based on unsupervised learning which is responsible for detecting data structural changes. As the main advantage, this approach does not require data labeling, although it does not rely on a strong formalization yet. However, several studies have already introduced important theoretical contributions to provide learning guarantees for the unsupervised scenario [5]. Among such studies, Carlsson and Mémoli [6] provide a framework which yields learning guarantees. In their work, it is defined a stability property for hierarchical clustering algorithms, which ensures that a data set \( X \) or any permutation of \( X \) will produce the same hierarchical clustering model, which is also referred to as a dendrogram. Moreover, Carlsson and Mémoli [6] prove that if \( X \) is only composed of independent and identically distributed (i.i.d.) observations, the produced dendrogram is equivalent to an ultrametric space. Therefore, it is possible to compare two dendrograms using the Gromov-Hausdorff distance and detect changes.

Within the unsupervised learning approach, several strategies were developed in order to detect concept drift. But most of them do not respect a stability property, like the permutation invariant property proposed by Carlsson and Mémoli [6]. Consequently those strategies tend to induce different clustering models given small perturbations in the input data. Therefore, the divergence computed between the clustering models may not indicate differences in input data, consequently the detected concept drift is imprecise.

This drawback motivated this work, whose objective is the online detection of concept drift in data streams, respecting the stability concept proposed by Carlsson and Mémoli [6]. By online we mean as data is collected over time. Based on the framework proposed by Carlsson and Mémoli [6], we developed a new approach which creates consecutive data windows, model them and compare those models to indicate when a concept drift has occurred in the input data.

Our approach considers data streams composed of a mixture of dependent and independent observations, this is due to real-world data usually contains both behavior embedded together [7]. Since the framework proposed by Carlsson and Mémoli [6] does not accept dependent observations, we propose a previous stage to decompose the data stream into deterministic (composed of dependent observations) and stochastic (composed of independent observations) components. Such stage is based on the work by Rios and Mello [7], which uses the Empirical Mode Decomposition method [8] to decompose a time series into several monocomponents and a residue. Next,
these monocomponents are analyzed using the Mutual Information [9] method in order to separate them into deterministic and stochastic ones. Finally, the monocomponents of the same type are summed, forming the deterministic and the stochastic components.

After this decomposition stage, we still have the deterministic component with dependent observations. In order to remove the temporal dependencies, we propose a transformation stage using Takens’ immersion theorem [10]. Primarily the embedded and the separation dimensions of this data are computed with the support of the False Nearest Neighbors [11] and the Auto-Mutual Information [12] methods. Then, the theorem is applied to unfold, i.e., reconstruct data, transforming observations into i.i.d. Consequently, it is possible to employ the concepts proposed by Carlsson and Mémoli [6]. Afterwards, the deterministic component already immersed and the stochastic component are modeled using the stable hierarchical clustering algorithm, producing a dendrogram each. Then, these dendrograms are compared with the dendrograms of the previous data window by using the Gromov-Hausdorff distance [13]. As result, we have two divergences computed, one between the stochastic models and another between the deterministic ones. These divergences can be used to detect concept drift, however, in this paper we mainly consider mixtures of deterministic components and their models. However, we also evaluate one scenario composed of a mixture between deterministic and a stochastic data. As future work, we plan to study more scenarios under the influence of stochasticity.

Experiments were conducted using two scenarios: i) synthetic data streams generated by using a mixture of deterministic behaviors; ii) another synthetic scenario in which the data stream is composed of a mixture between deterministic and stochastic components. In this stage, we believe the use of synthetic data makes more sense, since we know what to expect from experiments. Results confirm that the developed algorithm is capable of detecting concept drift in data streams.

This paper is organized as follows: Section II presents the related work; Section III describes the Online Concept Drift Approach; Section IV presents the experiments and discusses results, supporting our hypothesis; Section V shows the concluding remarks.

II. RELATED WORK

Klinkenberg and Renz [1] organize approaches for detecting concept drift in data streams into three categories: i) performance measures; ii) properties of the classification model; and iii) properties of the input data. Algorithms of the first category use performance measures on labeled data to quantify input data modifications, i.e., they are related to supervised learning. Algorithms of the second category compute the differences of input data based on classification models, i.e., they also consider supervised or semi-supervised learning. In the third category, algorithms use data properties for detecting concept drift, which are usually implemented using statistical or unsupervised learning algorithms.

The algorithms from the first and second categories are related to supervised or semi-supervised learning, i.e., they require (at least parts of) input data to be labeled to produce models. An initial model, representing the normal behavior, is compared against further models. As a concept drift is detected, new data needs to be labeled to produce the base model which will be compared to future models. In this circumstance, a specialist is required to label data. However, in the present scenario of data streams, in which data is produced at large volumes and at high frequencies, this labeling is not a feasible task. Besides this problem related to labeling more data, this supervised learning approach has the advantage of relying in guarantees provided by the Statistical Learning Theory [14], what makes possible to state that a concept drift has been correctly detected.

Considering the problem of labeling new data, we believe the third category of algorithms is more appropriate for the data streams scenario. In this unsupervised learning category, the algorithms are generally used to detect concept drift by comparing models created by clustering algorithms. Algorithms such as BIRCH [2] and DenStream [15] divide their functionality in online and offline stages. As new data is received by those algorithms, the online stage organizes them into a limited number of structures called Clustering Features (CFs). The offline stage executes to group CFs by using traditional clustering algorithms (K-means and DBScan, respectively), generating models. Besides these algorithms, there are the incremental algorithms, such as Growing Neural Gas [16] and SONDE [17], which modify the clustering model as new data is received. All those algorithms compare consecutive models in attempt to detect concept drift. However the divergence resultant of such comparison does not necessarily correspond to the actual modification in the input data. This lack of guarantees happens due to those algorithms do not hold any stability property such as the one proposed by Carlsson and Mémoli [6].

A. Stability property

Hierarchical Clustering (HC) algorithms are part of the unsupervised learning area. They create nested families of partitions representing a hierarchical decomposition of the input data, called dendrograms. These algorithms are organized into two types: agglomerative and divisive. The first creates a cluster for every data point. At each iteration, a set of clusters is selected according to their relative distance, normally the minimum distance between clusters, and are merged in order to create a new cluster. This process finishes when all data points are in the same cluster. Divisive algorithms work in the reverse way, i.e., they begin with all data points into a single cluster and, at every iteration, clusters are splitted up. This process finishes when all points are in single clusters.

Carlsson and Mémoli [6] formalize a stability property for agglomerative HC algorithms in which, during the merging process, if two or more points are close enough according to a threshold, all of them are merged. For example, consider function \( D(X, Y) \) to compute the Euclidean distance between clusters \( X \) and \( Y \). Now, let the distance between clusters \( A \) and \( B \) be equal to \( \epsilon \), i.e., \( D(A, B) = \epsilon \), and let \( D(D, E) = \epsilon \), \( D(E, F) = \epsilon \). In this case, clusters \( A \) and \( B \) will be merged forming set \( \{A, B\} \) as well as clusters \( D, E \) and \( F \) will form another cluster represented by set \( \{D, E, F\} \). Following this principle, the adaptation for the agglomerative HC algorithm proposed by Carlsson and Mémoli [6] will produce identical dendrograms for any permutation of the input data.
Furthermore, Carlsson and Mémoli [6] proved that dendrograms constructed by i.i.d. observations are equivalent to an ultrametric space. Thus, using the Gromov-Hausdorff distance it is possible to compute divergences of two dendrograms, i.e., compare two dendrograms. This results in a framework that allows to compute divergences between the outputs (or models) of HC algorithms, since they hold this stability property. As a consequence, the divergence between models represents actual changes in input data.

III. ONLINE CONCEPT DRIFT APPROACH

The next sections detail the online approach to detect concept drift in data streams. At the beginning, we introduce the decomposition stage used to obtain the deterministic and stochastic components that constitute the data stream. Afterwards, the transformation stage is responsible for converting the deterministic component into independent and identically distributed (i.i.d.) data. Then, both the deterministic and the stochastic components are in an i.i.d. form what makes possible to model them by using the stable clustering algorithm proposed by Carlsson and Mémoli [6]. Consecutive models obtained along time are compared and the resultant divergences are used to detect concept drift. Besides presenting the complete process involving deterministic and stochastic components, we only conducted experiments considering deterministic sources of data. The conclusions obtained on these data is enough to confirm the importance in holding the stability property as a means to make possible the comparison of consecutive clustering models. As next step, we will consider mixtures of deterministic and stochastic components forming the data stream.

A. Decomposition

The decomposition stage is responsible for separating the contribution that the deterministic and the stochastic influences have on every observation of the data stream. The deterministic component corresponds to dependent behavior, i.e., it provides information about how an observation is obtained through the combination of past ones. The stochastic component corresponds to independent and identically distributed behavior, in which observations have almost no dependency among themselves, making possible to assume them as i.i.d.

In this paper, we consider the decomposition approach proposed by Rios and Mello [7]. This approach takes an observation \( x(t) \) of time series \( X_t \), regardless its linearity, stationarity and stochasticity, and decomposes it by using the Empirical Mode Decomposition (EMD) [8] method. This results in a set of monocomponents \( h_j(t) \) and a residue \( r(t) \), such that \( x(t) = \sum_{j=1}^m h_j(t) + r(t) \).

By observing that every monocomponent \( h_j(t) \) extracted by EMD is more stochastic than the next, i.e., \( h_{j+1}(t) \), the authors proposed a next step which computes the Mutual Information [9] between consecutive monocomponents to verify if they follow the same behavior. This strategy permits to observe from which monocomponent the behavior starts to be predominantly deterministic, then it uses this information to separate the stochastic monocomponents from the deterministic ones. As last step, the strategy sums all stochastic monocomponents to form the stochastic component \( S(t) \) of time series \( X_t \) and sums the other monocomponents plus the residue to form the deterministic one, \( D(t) \), such that \( x(t) = S(t) + D(t) \).

The deterministic component corresponds to the part of the original time series observations which present dependencies. This means that every observation of the deterministic component can be modeled or represented by combining past occurrences. On the other hand, the stochastic component contains the part of the original time series which does not present any dependency or barely present it. Consequently, one can assume the stochastic component as i.i.d. data.

In this paper we consider a data stream as a time series, i.e., observations are collected over time and there is the possibility of having a mixture of deterministic and stochastic behaviors. From that, we employ the decomposition approach proposed by Rios and Mello [7] as a preprocessing step which is followed by the clustering stage.

B. Data transformation

As the framework provided by Carlsson and Mémoli [6] has the limitation of only using i.i.d. data, what is necessary due to the stability property guarantees that the same model is obtained given any permutation in the input data, it is essential to remove the dependencies among observations of the deterministic component \( D(t) \) obtained after the decomposition stage. In order to remove such dependencies, we apply Takens’ immersion theorem [10] to reconstruct a time series \( D(t) = d_0, d_1, \ldots, d_{n-1} \) into a multidimensional space, also referred to as phase space, \( d_m(m, \tau) = (d_m, d_{m+\tau}, \ldots, d_{m+(m-1)\tau}) \), in which \( m \) is the embedded dimension and \( \tau \) is the time delay (also referred to as separation dimension). This theorem translates series observations as points in the phase space which have temporal dependencies embedded in axes.

To illustrate Takens’ immersion theorem consider Figure 1 which shows two 250-observation time series produced with the Logistic map. At first, we observe series \( A_i \) and series \( B_i \) have very different plots. However, after applying Takens’ immersion theorem to reconstruct them into the phase space, we observe both have very similar attractors [18]. Those attractors represent the dependency relations among observations. In this case, the \( x \)-axis represents either an observation \( a(t) \) or \( b(t) \) while the \( y \)-axis corresponds to either \( a(t+1) \) or \( b(t+1) \), thus every observation at time instant \( t+1 \) actually depends on the observation at \( t \).

In this scenario, the temporal dependencies are embedded into axes, this means there is no axis or dimension to represent time. In fact, this makes possible to see axis values as states and the relation among states as transitions. In this case, we can see \( a(t) \) as current state and \( a(t+1) \) as the next one. By this conclusion, one can observe that it is possible to map (in a discrete manner) this phase space to states such as in a Markov Chain.

C. Modeling

As the stochastic component \( S(t) \) and the transformed deterministic component \( D'(t) \) are now composed only by i.i.d. observations, it is possible to model them by the permutation-invariant HC algorithm proposed by Carlsson and Mémoli [6]. Such algorithm generates a proximity dendrogram based on
and $A_t$ over time. The top-right figure corresponds to observations of time series $B_t$ over time. The bottom-left figure illustrates the phase space of time series $A_t$ obtained after Takens’ immersion theorem with embedded dimension $m = 2$ and separation dimension $\tau = 1$. The bottom-right figure illustrates the phase space of time series $B_t$ obtained after Takens’ immersion theorem with embedded dimension $m = 2$ and separation dimension $\tau = 1$.

Observe that after step (iv) all sets of clusters within the minimum distance $r_i$ among clusters; iv) afterwards, it selects all sets of clusters which can be connected through this minimal distance $r_i$ plus an additional neighborhood $\epsilon$, i.e., $r_i + \epsilon$, and merges them (\(\epsilon\) is a threshold defined by the user); finally, it goes back to step (ii) until all observations are in the same cluster.

D. Model Comparison and Concept Drift Detection

As formalized by Carlsson and Mémoli [6], the dendrograms obtained by HC algorithms respecting their stability property are equivalent to an ultrametrics space. So it is possible to compute the divergence between two dendrograms, $\theta_X$ and $\theta_Y$, by using the Gromov-Hausdorff distance, $d_{GH}(\theta_X, \theta_Y)$. In our work, such divergence is computed between dendrograms, therefore, we can compare successive dendrograms obtained for the stochastic and the deterministic components, having two divergences computed: $\alpha_s = d_{GH}(\theta_{S(t)}, \theta_{S(t-1)})$ and $\alpha_d = d_{GH}(\theta_{D(t)}, \theta_{D(t-1)})$, in which the $\theta_{S(t-1)}$ and $\theta_{D(t-1)}$ are the dendrograms obtained in the previous time window $t - 1$ and $\theta_{S(t)}$ and $\theta_{D(t)}$ are the dendrograms obtained in the current window. Besides we plan to compare only consecutive dendrograms in practical scenarios, i.e., dendrograms obtained in consecutive time windows, we also conducted experiments comparing all dendrograms obtained at different time windows against each other, this is useful to point out the divergences among them and improve the understanding about results.

The comparison between two dendrograms is done as follows. Let $R_D^{(t)}$ and $R_D^{(t-1)}$ be vectors of distance from the proximity dendrograms $\theta_{D(t)}$ and $\theta_{D(t-1)}$, respectively, such that, $r_i \in R_D^{(t)}$ and $r'_i \in R_D^{(t-1)}$. The Gromov-Hausdorff distance is given by the maximum divergence between the values into these vectors, i.e., $d_{GH}(\theta_{D(t)}, \theta_{D(t-1)}) = \max(\{abs(r_i - r'_i)\}, 0 < i \leq N)$. Consider the two proximity dendrograms from Figure 2, $\theta_{D(t)}$ and $\theta_{D(t-1)}$, in which $R_D^{(t)} = (0.6, 0.4, 0.3)$ and $R_D^{(t-1)} = (0.7, 0.3, 0.1)$. After computing the absolute difference between the vectors we obtain $(0.1, 0.1, 0.2)$, which maximum corresponds to the Gromov-Hausdorff distance, i.e., $d_{GH}(\theta_{D(t)}, \theta_{D(t-1)}) = 0.2$.

IV. Experiments

In this paper we compute the divergence between models to detect concept drift in data streams. Our approach decomposes data streams in deterministic and stochastic components and models both. Consecutive models are then compared against each other and divergences $\alpha_d$ (in between deterministic components) and $\alpha_s$ (in between stochastic components) are used to issue a new concept. We organize experiments in two types: i) considering a mixture of solely deterministic components to form data streams; ii) considering another scenario in which the data stream is composed of a mixture of deterministic and stochastic influences. Both situations confirm that our approach is effective.

A. Setup

The experiments were executed on synthetic data streams with 10,000 observations each, representing data streams with deterministic behavior. They were created by combining pairs of the following deterministic series: sine, Logistic map and Lorenz system [18]. Let $F_t$ and $G_t$ be time series generated by a deterministic behavior, and $w_1$ and $w_2$ be complementary weights that vary from zero to one according to the sigmoid function $w_2(t, s, t_e) = \frac{1}{1 + e^{-(s - t_e)}}$ and $w_2(t) = 1 - w_2(t)$, in
which \( t \) is the current time instant, \( s \) is a smoothing factor and \( t_n \) is the instant in which both weights are equal to 0.5. The data stream \( S_t \) is formulated as \( s(t) = w_1(t) \cdot f(t) + w_2 \cdot g(t) \), in a way that it starts having only one behavior that changes to the other using the sigmoid function.

In all experiments we considered parameters to make the data stream start modifying its behavior at time instant 4,500 and have the next behavior at instant 5,500. Meanwhile, a different mixture of both behaviors is observed. For example, at time instant 5,000, the data stream has 50% of each behavior. Figure 3 illustrates the central 2,000 observations of a data stream starting with a sinusoidal behavior and changing to a Logistic-map-based one. In total, nine data streams were generated and tested, having the following behavior changes: i) sinusoidal behavior is kept all the way; ii) sinusoidal behavior which changes to a Logistic-map-based one; iii) sinusoidal behavior followed by a Lorenz system; iv) Logistic-map behavior followed by the sinusoidal one; v) Logistic-map behavior is kept all the way; vi) Logistic-map behavior followed by the Lorenz system; vii) Lorenz-system behavior followed by the sinusoidal one; viii) Lorenz-system behavior followed by the Logistic map; ix) Lorenz-system behavior is kept all the way.

The parameters used to generate the data streams were: i) for the sinusoidal series, amplitude equals to 1, period equals to 40 and phase equals to 0; ii) for the Logistic map, the initial condition is 0.5 and rate is 3.8, which develops a chaotic behavior, difficulting modeling and therefore prediction using Statistical tools [19]; and iii) the sigmoid function was parameterized with smoothing factor \( s = 0.01 \) and transition point at \( t = 5000 \). After producing all data streams, every one was normalized in range \([0,1]\). During the transformation stage, the embedded dimension and the time delay were set to \( m = 3 \) and \( \tau = 5 \), respectively.

The dendrograms produced by our approach require a data-stream window, i.e., a subsequence of observations. In experiments, we considered 1,000-length windows, using an overlap of 500 observations. As soon as a window is filled with data, it is processed by our approach which produces two dendrograms, one for the deterministic and another for the stochastic component. Deterministic dendrograms are then compared against each other using the Gromov-Hausdorff distance to produce divergence value \( \alpha_s \). Stochastic dendrograms are also compared to produce \( \alpha_s \). Next section details all results.

### B. Results

In experiments, we compared all dendrograms obtained for every data-stream window against each other, producing divergence matrices. Every matrix resultant of a data stream with changing behavior should present high divergence at middle points. Results of the first experiment (considering data streams formed as a mixture between deterministic components) are shown in as heat maps in Figure 4 in which each cell represents the Gromov-Hausdorff distance computed between deterministic dendrograms of two data windows. The colors varies from dark gray, which represents no divergence, to light gray, which represents high divergence. As one can see in Figure 3, the most critical time regarding behavior changes is between time instants 4,500 and 5,500, as expected.

By analyzing Figure 4, we observe a pattern on every heat map composed of two different behaviors, which contain four quadrants, being two at lower divergences (the ones comparing the data-stream windows under the same behavior) and two at higher divergences (the ones comparing the data-stream windows under different behaviors). This pattern confirms that our approach is capable of pointing out differences in the data-stream behavior and, consequently, it detects concept drift. Moreover, data streams composed of two identical behaviors had detected low divergences in the middle of the heat maps, as expected. This low divergences still happen due to the concatenation between behaviors (even when the same) is not smooth.
The second type of experiment considered a mixture of deterministic and stochastic components to form the data stream. In this manner, we could test all steps involved in our approach, specially the decomposition one. In this last experiment, the data stream was generated by mixing two composed behaviors: i) the Logistic-map behavior was summed up with noise produced by a Uniform distribution (parameterized with minimum equals to −0.5 and maximum to +0.5); and ii) the sinusoidal behavior was summed up with noise produced using a Normal distribution (parameterized with mean equals to 0 and standard deviation to 0.1). In that manner, the Logistic behavior (deterministic) is already mixed with the stochastic one (uniformly generated noise) as well as the sinusoidal (deterministic) is mixed with the normally distributed noise (stochastic). The data stream starts with behavior (i) and changes to (ii) according to the sigmoid function having $s = 0.01$ and $t = 5,000$. For this experiment, the divergences for the deterministic and also for the stochastic components were computed. The heat maps representing the results are shown in the Figure 5, in which the divergences had the same pattern found into the previous experiment, making possible to point out differences in between deterministic components along time as well as stochastic components along time. This makes our approach adequate to separate deterministic and stochastic behaviors and compare them to detect concept drift in data streams.

V. CONCLUDING REMARKS

This paper introduces a new approach to detect concept drift in data streams composed of different mixtures of deterministic and stochastic components. This approach decomposes the data stream making possible to hold the stability property formalized by Carlsson and Mémoli [6]. By holding this property, it guarantees that a change in clustering models corresponds to actual changes in the input data, what is not ensured by other unsupervised approaches [2], [17]. Experiments confirmed this approach is capable of detecting concept drift in data streams. As next step, we will conduct experiments analyzing data streams formed by the combination of different stochastic and deterministic models.

![Heat maps obtained from the comparison of dendrograms using a data stream that starts with a Logistic-map behavior plus uniformly distributed noise and changes to a Sinusoidal behavior plus normally distributed noise.](image)

**Fig. 5:** Heat maps obtained from the comparison of dendrograms using a data stream that starts with a Logistic-map behavior plus uniformly distributed noise and changes to a Sinusoidal behavior plus normally distributed noise. The heat map on the left represents the divergences obtained for the deterministic component, while the heat map on the right corresponds to divergences for the stochastic component. Both components were decomposed and compared using our approach.

ACKNOWLEDGMENTS

This paper is based upon works supported by CAPES (Coordination for the Improvement of Higher Level Personnel), Brazil, under grant no. DS-5230736/D, FAPESP (São Paulo Research Foundation), Brazil, under grant no. 2011/51305-0, and by CNPq (The National Council for Scientific and Technological Development), Brazil, under grant no. 303280/2011-5. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of CAPES, FAPESP and CNPq.

REFERENCES