Anticipative and Dynamic Adaptation to Concept Changes [Extended Abstract]

Ghazal Jaber[1,2,3], Antoine Cornuéjols[1,2], and Philippe Tarroux[3]

[1] AgroParisTech, UMR 518 MIA, F-75005 Paris, France
[2] INRA, UMR 518 MIA, F-75005 Paris, France
[3] Université de Paris-Sud, LIMSI, Bâtiment 508, F-91405 Orsay Cedex, France

1 Introduction

Recent years have witnessed the emergence of a whole new set of applications involving data streams made of pairs $(\mathbf{x}_t, y_t)$, where the "answer" or true label y_t is revealed (sometimes long) after the input $\mathbf{x}_t$. When learning from data streams, it is necessary to rely on on-line learning with the capability to adapt to changing conditions a.k.a. *concept drifts*. Previous works have focused on means to *detect* changes and to *adapt* to them. Ensemble methods relying on committees of base learners have been among the most successful approaches.

Most adaptive strategies operate either by passively tracking the evolving concept or by using an explicit detection mechanism of concept changes before launching an adaptation or relearning process. However, better learning strategies may take advantage of the examination of the history of past concepts in order to anticipate likely future changes or to recognize when a past concept recurs. We have developed ADACC (*Anticipative Dynamic Adaptation to Concept Change*), a system that uses this kind of second order learning to accelerate its adaptation to changing conditions in the environment.

2 Relevant Works

Notable works that have confronted the anticipation of concept change include the PreDet [1] algorithm which uses decision trees as classifiers and anticipates future trees, and RePro [2] which stores the observed concepts in a Markov chain relying on the assumption that concepts repeat over time.

In the following, we first introduce the adaptive architecture of the learning system we take as our basis (Section 3.1). Then, we present the core of this paper which is the anticipating mechanism that builds upon it (Section 3.2).

3 Concept Changes: Adaptive and Anticipative

3.1 Adapting to Concept Changes

The idea is to maintain a pool of base learners $\{h_t^i\}_{1 \le i \le N}$, each of them adapting to the new input data. The main principles of these ensemble methods are the following:

- Each base learner continuously adapts with new incoming data until it is removed from the pool.
- Every τ time steps, the base learners are *evaluated* on a window of size τ_{eval}.
- Based on the results of this evaluation, the deletion procedure *chooses* a base learner to be removed.
- A new based learner is *created* and inserted in the pool. It is protected from possible deletion for a duration τ_{mat}.
- For each new incoming instance $\mathbf{x}_t$, the prediction $H(\mathbf{x}_t)$ results from a *combination* of the prediction of the individual based learners $h_t(\mathbf{x}_t)$.

Our *base learners* can be any supervised predictors. The *evaluation* procedure counts the number of erroneous predictions on the last τ_{eval} time steps. The *deletion strategy* randomly selects one base learner from the worst half of the pool evaluated as above. The *global prediction* uses the prediction from the current best base learner. This simple method leads to fast adaptation when the underlying concept changes.

3.2 Anticipating Concept Changes

In the following, we first describe a mechanism for the *recognition of the relevant states of the world* before presenting the *second-order learning mechanism* that works on these states in order to predict future models.

Recognition of the relevant states of the world

The list of past snapshots (descriptions of past concepts) $\mathcal{M}_{LT} = \{C_1, C_2, \ldots, C_k\}$, ordered according to their time appearance, is the basis of the second order learning mechanism. It serves two purposes. *First*, it provides a sequence of successive states of the environment that can be used by a learning algorithm in order to predict the most likely future state in the series. *Second*, it stores a memory of past successful states of the world, models that should be repeatedly tested against current data in case a recurring concept can be recognized.

The technique we propose to recognize when to take a new snapshot is based on the assumption that, given a stationary environment and a sequence of examples of sufficient length, the base learners converge toward approximately the same, near optimal, concept, as measured by their prediction performance. Therefore, one way to detect that a stationary environment has settled is to check that the diversity of the base learners is low, under some threshold, while their prediction rate peaks.

We use the kappa statistics $\mathcal{K}$ [3] in order to compute the *diversity*. The *stability index* at time t, $I_{stability} = agreement - error$ is computed over the last τ_s received examples where *agreement* and *error* are computed over the best half of the current hypotheses in the pool. Each point in the stability index curve, over some predefined threshold θ_I, is suggestive of a stable environment. In order to avoid storing redundant snapshots, we use again the kappa statistics to measure the agreement between a candidate snapshot h_t^* and elements of $\mathcal{M}_{LT}$. Only when this agreement is low enough (below a threshold θ_d) it is added to $\mathcal{M}_{LT}$.

The long term memory and second order learning

In our experiments, we used Elman's recurrent neural networks as predictors. A network is trained on the pairs of consecutive concepts in $\mathcal{M}_{LT}$: $\{(C_i, C_{i+1})\}_{i=1}^{k-1}$ in order to predict the next likely snapshot $\tilde{C}_{k+1}$. The predicted snapshot is then temporally added to the list of snapshots $\mathcal{M}_{LT}$. It is replaced by the next snapshot C_{k+1} when this one is acquired. Each snapshot in the list is then treated on an equal footing with the base learners in the pool of the adaptive ensemble method. The snapshots are therefore evaluated according to the evaluation strategy used by the adaptive ensemble to evaluate the base learners in the pool. A snapshot is used for prediction if its evaluation record is the best among all candidate hypotheses from both the pool of base learners and $\mathcal{M}_{LT}$.

4 Experiments

We did experiments on an artificial data set to simulate *recurrent* and *predictable concept changes* while controlling the timing of the change and its speed (abrupt, or more or less gradual). The input space $\mathcal{X}$ is d-dimensional and the target concept is a linear decision boundary described by the relation $y(\mathbf{x}) = sign(\sum_{i=1}^{d} w_i x_i + w_0)$. The experiments were carried out on a stream with 7,150 time steps and hence data points. Twelve concept changes were simulated by changing the weights $\{w_i\}_{i=0}^{d}$ of the target hyperplane. The first 7 concepts were obtained through successive additions or substractions of constant values to the weights. The last 6 concepts were recurring concepts. The changes happened either suddenly or gradually between successive target concepts. The transition between consecutive concepts took from 0 to 200 time steps and changes would start happening every 400 to 700 time steps. We report results for the 10-dimensional case. The base learners were perceptrons with 10 input units and one output unit, involving 11 weights (10 + 1 for the bias). The Elman's networks took as input the 11 weights of a snapshot and gave as output the 11 weights of the next predicted snapshot. The parameters for the *anticipative meta-learning* system were set respectively to $\theta_I = 0.9$ and $\theta_d = 0.8$ while $\tau_s = 100$. The remaining parameters concern the adaptive mechanism, and were optimized in order to not unfairly attribute gains to the anticipative process. They were set to $N = 10$ and $\tau_{eval} = \tau_{mat} = 20$.

Figure 1 illustrates the mechanism for the selection of snapshots on the data stream and it shows the evolutions of the prediction performance over 10 repeated experiments.

Even though the concept changes occur at varying dates and with varying speed, the anticipation mechanism is able to predict relevant foreseeable target concepts that, in turn, are quickly recognized as the best for labeling the incoming examples. This brings significant gains in the online performance starting already after the 4th change of concept, and the gain increases thereafter with each new concept change.

5 Conclusions

ADACC is a general framework to endow adaptive online learning systems based on an ensemble approach with second order learning capacity. It provides means (i) to identify the relevant stationary states of the world, (ii) to anticipate likely future states, and (iii) to recognize recurring states if they ever arise. Few parameters are involved in the second-order learning scheme and they do not need to be finely tuned.

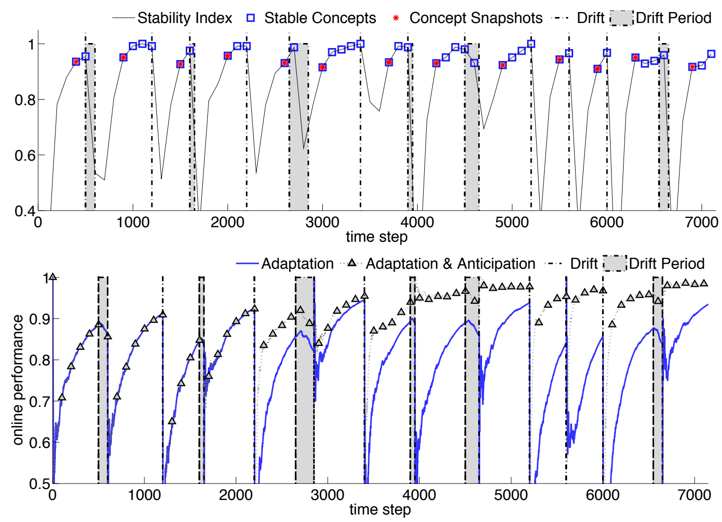

Fig. 1. The *top* plot shows the evolution of the stability index. Small squares indicate the time steps where candidate snapshots are considered, and (red (or black) squares) when they are retained. The *bottom* plot shows the online predictive performance of the adaptive learning strategy (continuous (blue) line) and with the second order learning taking place. They are averaged over 10 repeated experiments. The beginning/end of concept changes are indicated as vertical dotted lines. In case of gradual concept changes, the transition period between consecutive concepts is colored in gray. The online predictive performance is reset every time a transition is complete.

The empirical evaluation shows that second order learning yields substantial gains in prediction performance over a mere adaptation policy. Furthermore, second order learning can only improve and never deteriorate the prediction performance.

Our future plans include testing possible strategies for the management of the memory of snapshots in order to keep its size under control, for instance, by deleting the oldest or the least recurring snapshots from the memory. Another future avenue is to evaluate ADACC on datasets of real-world applications, such as product recommendation systems which evolve with time depending on user preferences or market mood, and can also undergo concept recurrence, for instance, when past fashion comes back.

References

1. Bottcher, M., Spott, M., Kruse, R.: Predicting future decision trees from evolving data. In: Data Mining 2008. ICDM'08. Eighth IEEE International Conference on. IEEE (2008)
2. Yang, Y., Wu, X., Zhu., X.: Mining in anticipation for concept change: Proactive-reactive prediction in data streams. In: Data mining and knowledge discovery, 13(3) (2006)
3. Carletta, J.: Assessing agreement on classification tasks: the kappa statistic. In: Computational linguistics, 22(2), pp. 249–254 (1996)