Predicting Concept Changes using a Committee of Experts

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

[1] Université de Paris-Sud, LIMSI, Bâtiment 508, F-91405 Orsay Cedex, France
[2] AgroParisTech, Dept. MMIP, 16 rue Claude Bernard, 75005 Paris Cedex, France

Abstract. In on-line machine learning, predicting changes is not a trivial task. In this paper, a novel prediction approach is presented, that relies on a committee of experts. Each expert is trained on a specific history of changes and tries to predict future changes. The experts are constantly modified based on their performance and the committee as a whole is thus dynamic and can adapt to a large variety of changes. Experimental results based on synthetic data show three advantages: (a) it can adapt to different types of changes, (b) it can use different types of prediction models and (c) the committee outperforms predictors trained on a priori fixed size history of changes.

Key words: Committee of experts, on-line learning, concept changes, prediction.

1 Introduction

Different types of concept changes exist in the litterature. *Concept drifts* [9] refer to the change of the statistical properties of the concept's target value. For example, the behavior of customers in shopping might evolve with time and thus the concept capturing this behavior evolves as well. The speed of the change can be gradual or sudden. A sudden drift is sometimes referred to as a *concept shift* [7, 8]. Another type of change, known as *virtual concept drift* [6], *pseudo-concept drift* [4], *covariate shift* [1] or *sample selection bias* [3] occurs when the distribution of the training data changes with time. In this context, several problems should be adressed:

- What is the optimal size of the memory in order to predict well the future trends? A long history may allow for more precise predictions, but it can also be misleading in case of sudden change of regime. This is the basis of the well-known *stability-plasticity* dilemma.
- What is the nature of the change ruling the evolving environment? Do we consider change as a stable function or can the change itself vary with time?

Most current approaches assume that the change is a temporal function that can be learnt using standard temporal series methods, for instance using linear regression [2] or a hidden markov model [8]. Therefore, they consider the change as a stable function that can be predicted using time information only.

In this paper, we suggest a general approach to anticipate how a concept changes with time. We solve the stability-plasticity dilemma by using a committee of experts where each expert is a predictor trained on its own history size. Our method is inspired by [5] where a dynamic committee of experts is used to learn under *concept drift*. By analogy, we use a committee of experts to learn under *concept change drift*. Therefore, we look at the bigger picture by considering *the change* itself as a *dynamic function* that can vary with time.

The rest of this paper is organized as follows. Section 2 discusses related works. We present our approach in Section 3. In Section 4, we test our approach on two scenarios: the first is designed to show how our committee allows a fast adaptation to sudden changes while preserving a good prediction accuracy on gradual changes, the second mixes different types of predictors: neural networks and polynomial regression models. Finally, Section 5 summarizes our results.

2 Related Work

In learning under concept drift, some approaches aim at removing the effect of change [4] while others suggest techniques to detect change and adapt their model accordingly [5]. The PreDet [2] algorithm is one of the few works directly related to ours. It anticipates future decision trees by predicting for each node the evaluation measure of each attribute, this value being used to determine which attribute will split the node. The system uses linear regression models trained on a fixed size history to predict future changes. The size of the history requires a priori knowledge on the speed of change. Another prediction system, RePro [8], stores the observed concepts as a markov chain. Once a change is detected, it uses the markov chain to predict the future concept. It assumes that concepts repeat over time.

3 Prediction Algorithm

The prediction scenario works as follows. The training examples are received in data sets or *batches* of fixed size. A *training example* is represented by a pair $(\mathbf{x}, y)$, where $\mathbf{x} \in R^p$ is a vector in a p-dimensional feature space and y is the desired output or target.

For each batch S_i a concept C_i is learnt. The concept C_i can be a classification rule, a decision tree or any other model that learns the model of the training data in S_i. By analyzing the sequence of concepts $\big(C_1, ..., C_i\big)$, the future concept C_{i+1} is predicted. The main parts of the prediction algorithm are presented next. The pseudo-code is shown in Algorithm 1.

To simplify the discussion, we represent a concept C as a vector of parameters of dimension n: $C = [c_1, c_2, ..., c_n]$. If the concept is a neural network for example, it can be represented as a vector of the network weight values.

After each batch S_i is received, a concept C_i is learnt. A change sample δ_t corresponds to a change between two consecutive concepts: $\delta_t = (C_t, C_{t+1})$. At timestep $t + 1$, the total history of changes is the sequence $\big(\delta_1, ..., \delta_t\big)$.

3.1 Building the Committee

In our approach, a committee of predictors is learned, each of which trained on a different history size.

The algorithm starts by defining a set of predictors with possibly different structures. For instance, the predictors can be neural networks with different numbers of hidden neurons and activation functions, or they can be decision trees, linear regression models, Bayes rules, SVMs etc. The set of predictors will be referred to as the "base of predictors" and has size n_base.

The committee P is initially empty. When the first change sample δ_1 is observed, a base of predictors b_1 trained on δ_1 is added to the committee. When the second change sample δ_2 is observed, each predictor in the base b_1 adds δ_2 to its history and is retrained. In addition, a new base of predictors b_2, trained on δ_2 only, is added to the committee. This process continues until a maximum number of base of predictors max_base in the committee is reached. At this point, the base of predictors $\{b1, b2, ..., b_{max\_base}\}$ have history sizes: $\{max\_base, max\_base-1, ..., 1\}$ respectively. In subsequent steps, the committee P is updated. At each timestep t, a small but significant number of predictors (e.g. 25%) is selected and their history size is increased; the history size of the remaining predictors remains unchanged. The worst n_base predictors are removed from the committee and replaced with a new base of predictors trained on the last seen change δ_{t-1} only.

3.2 Prediction

At timestep t, each committee member $p \in P$ give its prediction of the next concept $\tilde{C}^p_{t+1}$.

$$\tilde{C}^p_{t+1} = [\tilde{c}^p_{(t+1,1)}, ..., \tilde{c}^p_{(t+1,n)}] \tag{1}$$

where $\tilde{c}^p_{(t+1,i)}$ is the *i-th* parameter of the concept C_{t+1}, predicted by the *p-th* predictor for timestep $t+1$.

Each predictor in the committee predicts all the parameters of the next concept. However, the final prediction of the committee is formed by selecting the best prediction for each parameter $c_{(t+1,i)}$ of the concept independently. This is motivated by the fact that selecting the whole predictions $\tilde{C}^p_{t+1} = [\tilde{c}^p_{(t+1,1)}, ..., \tilde{c}^p_{(t+1,n)}]$ of a predictor p assumes that all the parameters $c_{(t+1,i)}$ evolve at the same speed, which may not be the case. The final prediction of the committee is: $\tilde{C}_{t+1} = [\tilde{c}_{(t+1,1)}, ..., \tilde{c}_{(t+1,n)}]$, where $\forall i \in \{1, .., n\}$

$$\tilde{c}_{(t+1,i)} = \arg\min_{p \in P} \parallel c_{(t+1,i)} - \tilde{c}^p_{(t+1,i)} \parallel \tag{2}$$

We choose the best predictions by defining an evaluation function $eval(p, i)$ that measures the performance of a predictor p in predicting the parameter i of the concept C. Equation 2 then becomes: $\tilde{c}_{(t+1,i)} = \tilde{c}^{p^*}_{(t+1,i)}$, where $p^* = \arg\max_{p \in P} \big(eval(p, i)\big)$.

Algorithm 1 The Concept Change Prediction Algorithm

$P \leftarrow \phi$, $t \leftarrow 2$, $maxsize\_P = n\_base * max\_base$
$C_1 \leftarrow$ train(C_1,S_1)

while batches are received **do**
  $C_t \leftarrow$ train(C_t,S_t)
  $\delta_{t-1} = (C_{t-1}, C_t)$ {the last sample change}

  {Remove the lowest performing predictors}
  if size(P) >= $maxsize\_P$ **then**
    for $k = 1 \rightarrow n\_base$ **do**
      {$p \in P$ has the lowest prediction performance}
      $P \leftarrow P \setminus p$
    end for
  end if

  {Update remaining predictors}
  for $k = 1 \rightarrow$ size(P) **do**
    $p_k \in P$ is the k_{th} predictor
    $hist_k$ is the history size of p_k
    $r \leftarrow rand[0, 1]$
    if $r >= 0.75$ **then**
      $hist_k \leftarrow hist_k + 1$
    end if
    Retrain p_k on the last $hist_k$ sample changes
  end for

  {Add new predictors}
  b_t is a base of predictors trained on δ_t
  $P \leftarrow P \cup b_t$

  {Predict next concept change}
  $H \leftarrow \phi$ is the set of predictions
  for $k = 1 \rightarrow$ size(P) **do**
    $p_k \in P$ is the k_{th} predictor
    $H \leftarrow H \cup \tilde{C}_{t+1}^{p_k}$
  end for

  $\tilde{C}_t \leftarrow \phi$ is the final prediction
  if $H \neq \phi$ **then**
    for $i = 1 \rightarrow$ size(n) **do**
      $p^* = \arg\max_{p \in P} \left(eval(p, i)\right)$
      $\tilde{c}_{(t,i)} = \tilde{c}_{(t,i)}^{p^*}$
    end for
  end if

end while

4 Experiments

The first experiment in Section 4.1 is designed to show how the prediction committee adapts to different types of change. In the second experiment, in Section 4.2, we show that by mixing different types of predictors in our committee (neural networks, polynomial regression models), we take advantage of each type of predictor and get better prediction results. Finally, we show that our predictors, whose history size change dynamically with time, outperform predictors trained on a fixed size window.

4.1 Experiment 1

We simulate a *concept drift* by continuously moving the hyperplane corresponding to a decision function (the target concept) in a d-dimensional space. A hyperplane is described by the equation $\sum_{i=1}^{d-1} w_i x_i = w_0$.

Sequence of concepts. For each slowly modified hyperplane, we generate a batch of 1000 training examples $(\mathbf{x}, y)$, where $\mathbf{x} \in [0,1]^d$ is a randomly generated vector of dimention d and $y = sign(\sum_{i=1}^{d} w_i x_i - w_0)$. The value of w_0 is set to $1/2 \sum_{i=1}^{d} w_i$ so that nearly half of the y's are positive and the other are negative. In this experiment, $d = 6$ and the hyperplane weights w_i are initially set to random values, that are gradually incremented, not necessarily with the same increment for each weight, until time step 101, then decreased until they reach their initial values. We learn a sequence of perceptrons each of which trained on the corresponding batch of training examples.

Prediction. In order to predict the perceptron changes, we use feed-forward neural networks that anticipate the new values of the perceptron weights given the current ones. The maximum size of the committee is set to 10 predictors.

At timestep t, each predictor in the committee gives its prediction of the next hyperplane weights. For each weight w_i, we define the best predictor as the committee member that predicted w_i with the least mean error on previous timesteps $t-1$, $t-2$ and $t-3$[3]. At timestep t, we also compute for each predictor its mean square error on all the weights to predict. The worst predictor on previous timesteps $t-3$, $t-2$ and $t-1$ is removed.

Results and Discussion. The prediction results for the perceptron weight w_1 is presented in Figure 4.1. The same behavior is observed for the other weights. The committee predicts well when the weight value increases. When the values start suddenly to decrease at time step 101, it corrects its prediction error rapidly

[3] In all our experiments, we evaluate the predictors' performance on a window size of 3. The window size is set to a small value to adapt to the recent predictors' performance

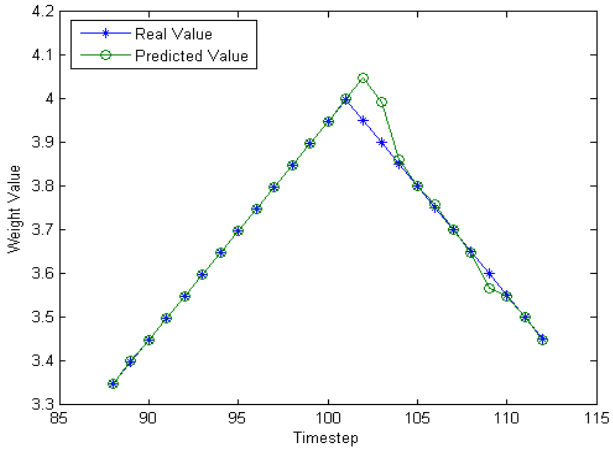

Fig. 1. The prediction results of the perceptron weight w_1 in the time interval $[85, 115]$, during which the weight suddenly starts decreasing. The line with asterisks represents the real value of w_1 whereas the line with circles represents the predicted value.

and regains its former prediction performance afterwards. This ability to adapt to rapid changes is due to the dynamic committee of predictors whose members are trained on different history sizes. We show in Figure 2 the evolution in the history size of three of the committee members.

The predictor A is added to the committee at timestep 2. Its history keeps growing until timestep 105 where it is replaced by a new predictor, trained on a history of size 1. Indeed, the sudden change in the weight value deteriorates the predictor's performance, causing its elimination from the committee. Predictor B is added at timestep 8, and is also removed soon after the sudden change. Predictor C, added to the committee at timestep 5, is replaced before the sudden change because it is the lowest performing committee member. It is also common for a newly added predictor to be replaced soon after it is added to the committee, as we see for predictor C during time interval $[90, 130]$. This occurs when the change is gradual and thus the performance of a newly added predictor will be bad compared to the other committee members.

4.2 Experiment 2

In this section, the hyperplane in a six-dimensional space undergoes more complex changes in the weight values than in experiment 4.1 (see Figure 3). A sequence of 250 different hyperplanes $(\boldsymbol{H}_1, \boldsymbol{H}_2, ..., \boldsymbol{H}_{250})$ is generated, where each hyperplane $\boldsymbol{H_i} = [w_{i,1}, w_{i,2}, ..., w_{i,6}]$ is represented by its weight vector.

Neural Networks vs Polynomial Regression In this *first set of experiments*, we tested our prediction approach using different types of predictors. We were

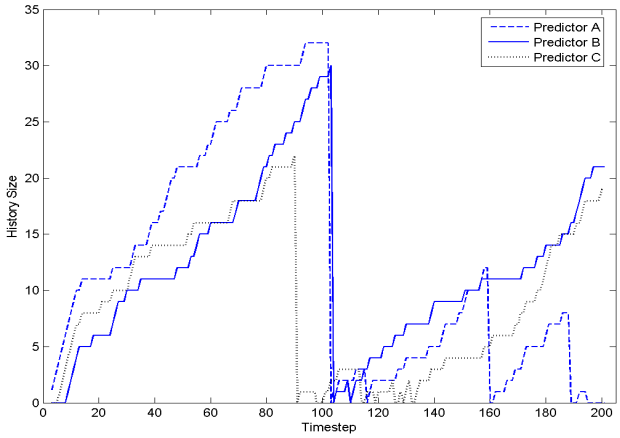

Fig. 2. The evolution in the history size of three of the committee predictors.

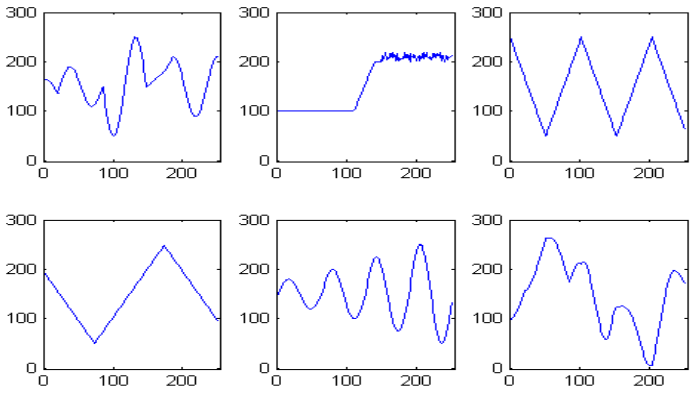

Fig. 3. The evolution of the six weights of the hyperplane described in experiment 4.2.

specially interested in comparing neural networks and polynomial regression models as predictors.

In each experiment, we compared feed-forward neural predictors to simple predictors which consider the next hyperplane equals to the current one. The prediction results are reported in exp. 1,2 and 3 of Table 1. Our prediction approach beats the simple prediction approach in nearly 85.6% of the time.

In the *second set of experiments*, we tested our prediction approach using polynomial regression models instead of neural networks as predictors. We repeated the previous tests using a base of predictors that consists of 3 polynomial predictors with degree 1,2 and 3 respectively. The results are reported in exp. 4,

Exp.	Base of Pred.	Max. Base of Pred.	S_b_O Per. (%)	S_b_O MSE ratio	S_O MSE ratio
1	1 FF	8	85.94	3.61	2.05
2	1 FF	13	85.94	4.03	0.26
3	1 FF	20	85.14	5.02	0.77
4	3 PR	8	89.95	2.11	1.68
5	3 PR	13	91.16	2.19	1.76
6	3 PR	20	89.55	2.27	1.77
7	1 FF, 3 PR	8	87.95	3.18	1.30
8	1 FF, 3 PR	13	89.95	3.38	1.77
9	1 FF, 3 PR	20	85.94	3.64	1.88

Table 1. The prediction results with different predictor types and committee sizes, using our prediction approach. *Exp* is the index of the experience. *Base of Pred.* is the base of predictors; 1 FF stands for one feed forward neural network and 3 PR stands for three polynomial regression models with degree 1,2 and 3 respectively. *Max. Base of Pred.* is the maximum number of base of predictors in the committee. During the experiments, we predict the weights of 250 hyperplanes. For each prediction, we compute the prediction MSE: the mean square error between the predicted values and the real values. The *S_b_O MSE* is the percentage of time our prediction MSE is smaller than the simple prediction MSE. The *S_b_O MSE ratio* is the ratio between the simple prediction MSE and our prediction MSE, when our prediction MSE is smaller than the simple prediction MSE. The *S_O MSE ratio* is the ratio between the simple prediction MSE and our prediction MSE.

5 and 6 of Table 1. Globally, neural networks beat polynomial regression models by having a smaller prediction error when they are better than the simple prediction scenario. On the other hand, polynomial regression models beat the neural networks by having a smaller prediction error on average.

In the *third set of experiments*, we mixed both type of predictors: the base of predictors contains a feed forward neural network and 3 polynomial regression models with degree 1, 2 and 3 respectively. The prediction results are reported in exp. 7, 8 and 9 of Table 1. By mixing neural networks with polynomial regression models, we take advantage of both types of predictors: the *S_O MSE ratio* and the *S_b_O MSE Per.* increase compared to when we only used neural networks while the *S_b_O MSE ratio* increases compared to when we only used polynomial regression models.

Dynamic History Size vs Fixed History Size. Prediction performances are compared with our committee and with predictors using a fixed history size. Five experiments were conducted. In the first four experiments, the history size was set to 2, 4, 8 and 15 respectively. In the fifth experiment, the history size of the predictor grows with time. The results are reported in Table 2. It appears that using fixed window size predictors requires a priori knowledge of the suitable window size for the prediction task. Choosing the wrong window size might give catastrophic results.

Exp	Predictor	History Size	S_b_O Per. (%)	S_b_O MSE ratio	S_O MSE ratio
1	1 FF	3	82.32%	4.4	0.21
2	1 FF	5	87.95%	3.9	1.7
3	1 FF	9	70.28%	2.44	1.07
4	1 FF	15	17.67%	1.4	0.29
5	1 FF	*growing*	4.47%	1.2	0.0108

Table 2. The prediction results using predictors with a fixed size history. *Exp* is the index of the experience. *Predictor* is the type of predictor used in the experience; 1 FF stands for one feed forward neural network. *History Size* is the fixed history size of the predictor. The last three columns are explained in Table 1.

5 Conclusion

We have presented an approach to predict future concept changes using a dynamic and diverse committee of experts. Each expert in the committee is a predictor that anticipates the future changes of an evolving concept, taking into account the observed history of changes. The committee can be comprised of different types of experts (neural networks, polynomial regression models, SVMs etc...) with different history sizes. It is also *dynamic* by constantly updating its members. The experiments show that the diversity in the history size allows us to adapt to different types of changes while using multiple types of experts improves the prediction results.

References

1. S. Bickel, M. Bruckner, and T. Scheffer. Discriminative learning under covariate shift. *The Journal of Machine Learning Research*, 10:2137–2155, 2009.
2. M. Bottcher, M. Spott, and R. Kruse. Predicting future decision trees from evolving data. In *Data Mining, 2008. ICDM'08. Eighth IEEE International Conference on*, pages 33–42.
3. W. Fan, I. Davidson, B. Zadrozny, and P. S. Yu. An improved categorization of classifierś sensitivity on sample selection bias. 2005.
4. S. Ruping. Incremental learning with support vector machines. In *Proceedings of the 2001 IEEE International Conference on Data Mining*, 2001.
5. K. O. Stanley. Learning concept drift with a committee of decision trees. *Computer Science Department, University of Texas-Austin*, 2001.
6. N. A. Syed, H. Liu, and K. K. Sung. Handling concept drifts in incremental learning with support vector machines. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, page 321, 1999.
7. G. Widmer and M. Kubat. Learning in the presence of concept drift and hidden contexts. *Machine learning*, 23(1):69–101, 1996.
8. Y. Yang, X. Wu, and X. Zhu. Mining in anticipation for concept change: Proactive-reactive prediction in data streams. *Data mining and knowledge discovery*, 13(3):261–289, 2006.
9. I. Zliobaite. Learning under concept drift: an overview. Technical report, Tech. rep., Vilnius University, Faculty of Mathematics and Informatics, 2009.