Adaptive Online Prediction Using Weighted Windows

Yoshida, Shin-ichi
NTT West

Hatano, Kohei
Department of Informatics, Kyushu University

Takimoto, Eiji
Department of Informatics, Kyushu University

Takeda, Masayuki
Department of Informatics, Kyushu University

出版情報：IEICE Transactions on Information and Systems. E94.D (10), pp.1917-1923, 2011-10.
The Institute of Electronics, Information and Communication Engineers
バージョン：

PAPER Special Section on Information-Based Induction Sciences and Machine Learning

Adaptive Online Prediction Using Weighted Windows

Shin-ichi YOSHIDA[†], *Nonmember*, **Kohei HATANO**[††a)], **Eiji TAKIMOTO**[††], *Members*, *and* **Masayuki TAKEDA**[††], *Nonmember*

SUMMARY We propose online prediction algorithms for data streams whose characteristics might change over time. Our algorithms are applications of online learning with experts. In particular, our algorithms combine base predictors over sliding windows with different length as experts. As a result, our algorithms are guaranteed to be competitive with the base predictor with the best fixed-length sliding window in hindsight.

key words: *machine learning, data stream, online learning, sliding window*

1. Introduction

Data stream arises in many applications. For example, developments of distributed censor devices enable us to collect data which are generated constantly over time. Also, there are more and more huge data available and such huge data can be viewed as data stream as well if we want to deal with them by "one-pass" scan. Researches on data stream have become popular in various areas in computer science such as databases, algorithms [1], data mining and machine learning [2].

There are two notable properties of data stream. The first property is that nature of the data stream might change over time. The underlying distribution which generates the data might change gradually over time or change suddenly at some trial (concept drifts). So, prediction algorithms for data stream need to adapt concept drifts. The second property is that whole the data stream is too huge to keep since the new data comes endlessly. Therefore, prediction algorithms also need to choose some partial data.

A natural approach to deal with data stream is to use a sliding window. The sliding window keeps only recent data. As a new instance comes in, then the oldest instance in the window is discarded from the window and the new one is added to the window. Then prediction algorithms use only the data of the sliding window to make predictions on future data.

For time-changing data streams, it is reasonable to assume that recent data is more informative than older data. So, sliding window approaches seem to work well for prediction tasks on data streams. However, it is not trivial to determine the size of sliding window in advance. If the size is too large, accuracy of prediction might become worse when the nature of the data stream changes, since older data affects the prediction. On the other hand, if the size is too small, accuracy of prediction might become worse as well when the data is rather stationary.

There are some researches to make the size of the sliding window adaptive [3]–[6]. These proposed methods heavily depend on the choice of parameters, e.g., a threshold to determine when to discard data in the window. For example, given a single temporal outlier, ADWIN [3] discards the all data in the window even if the data stream is rather stationary, which might get the accuracy worse.

In this paper, we take an alternative approach. Instead of choosing a fixed-sized window or changing the size of the window adaptively, we combine the predictions using multiple windows with different sizes. More precisely, we employ the approach of online learning with experts [7]–[18]. We consider M "sub-windows" which contains the k newest elements of the window ($k = 1, \ldots, M$).

We assume a fixed predictor, called the base predictor, which works with a sliding window so that it makes predictions using only the data in the window. Since we have M sliding windows of size 1 through M, we get M predictors: the k-th predictor is the base predictor running with the sliding window of size k. Using these M predictors as experts and applying Weighted Average Algorithm by Kivinen and Warmuth [19], we obtain an online prediction algorithm, called the Weighted Window (WW, for short). The WW is guaranteed to perform almost as well as the best expert, i.e., the base predictor with the best fixed size window. More precisely, we show that the WW has $O(\ln M)$ regret, where the regret of a prediction algorithm is defined as the cumulative loss of the algorithm minus that of the predictor with the best fixed size window.

Furthermore, we apply the method of Hazan and Seshadhri [20] to make the prediction algorithm more adaptive. In particular, by combining multiple copies of WWs over different intervals, we obtain the second algorithm called the Weighted Windows with follow the leading Hisotry (WWH, for short). The WWH is guaranteed to perform almost as well as the best experts for all intervals in the data stream. More precisely, we show that for any interval I in the data stream, the regret of WWH measured for I is bounded above by $O(\ln M \ln T + \ln^2 T)$, where T is the length of the data stream.

Note that our contribution is not to develop new tech-

Manuscript received January 7, 2011.
Manuscript revised May 4, 2011.

[†]The author is with NTT West, Osaka-shi, 540–8511 Japan.
[††]The authors are with Department of Informatics, Kyushu University, Fukuoka-shi, 819–0395 Japan.
a) E-mail: hatano@inf.kyushu-u.ac.jp

DOI: 10.1587/transinf.E94.D.1917

niques for online learning with experts, but to apply online learning with experts framework to sequence prediction using sliding windows in order to make it robust.

In our experiments over artificial and read time-series data, WW and WWH outperform other previous methods and compete fairly with predictions with the best fixed window.

2. Preliminaries

For a fixed integer N, let $\mathcal{X}$ be the domain of interest. A member $\boldsymbol{x}$ in $\mathcal{X}$ is called an instance. For integers r and $s (r \leq s)$, we denote by $[r, s]$ the set consisting of sequential integers $r, \ldots, s$. In particular, we write $[s]$ for short if $r = 1$.

2.1 Online Prediction with Experts

We consider the following protocol of online prediction. At each trial $t = 1, \ldots, T$,

1. the adversary gives an instance $\boldsymbol{x}_t$ to the learner,
2. the learner guesses a prediction $\hat{y}_t \in [0, 1]$ for $\boldsymbol{x}_t$,
3. the adversary gives the true value $y_t \in [0, 1]$ to the learner, and
4. the learner incurs loss $\ell(y_t, \hat{y}_t)$.

Here the function $\ell : [0, 1] \times [0, 1] \to \mathbb{R}$ is called the *loss function*. In particular, in the setting of online learning with experts ([7], [8]), the learner can use predictions of experts. More precisely, the learner is given M experts in advance. At each trial t, each expert is given an instance $\boldsymbol{x}_t$ and returns its prediction $\hat{y}_{t,i} \in [0, 1]$.

The goal of the learner is to predict as well as the best expert in hindsight. More detailed goals are the following:

- Regret:
$$\sum_{t=1}^{T} \ell(y_t, \hat{y}_t) - \min_{i=1,\ldots,M} \sum_{t=1}^{T} \ell(y_t, \hat{y}_{t,i}).$$
- Adaptive regret [20]
$$\sup_{I=[r,s]\subset[T]} \left\{ \sum_{t=r}^{s} \ell(y_t, \hat{y}_t) - \min_{i=1,\ldots,M} \sum_{t=r}^{s} \ell(y_t, \hat{y}_{t,i}) \right\}.$$

A loss function ℓ is called α-exp *concave* if the function $e^{-\alpha\ell(\hat{y},y)}$ is concave w.r.t. $\hat{y}$. It is known that some natural loss functions such as square loss, log loss, relative entropy and Hellinger loss are α-exp concave for some α (see, e.g., [9]).

Let us discuss the difference between regret and adaptive regret. Regret measures the difference between cumulative losses of the algorithm and the best expert for all T trials. However, low regret does not necessarily imply a good performance over time-changing data. This is because, for data changing its tendency over time, the best expert for all T trials might not adapt changes in data and predicts badly for some intervals. On the other hand, If the adaptive regret is bounded, then the regret w.r.t. the best expert for any interval is bounded as well. So, minimizing the adaptive regret is more challenging goal, especially for time-changing data streams.

2.2 Sliding Window

A *sliding window* is popular in the task of prediction of data streams which might change over time. A sliding window of size k keeps k newest instances. More formally, the sliding window W of size k at trial t is a sequence of instances of size k,

$$W = \begin{cases} \cup_{j=1}^{t-1} \{(\boldsymbol{x}_j, y_j)\} & \text{if } t - 1 \leq k \\ \cup_{j=t-k}^{t-1} \{(\boldsymbol{x}_j, y_j)\} & \text{if } t - 1 > k \end{cases}.$$

We assume a base prediction algorithm associated with a sliding window. The algorithm uses the examples in the sliding window to make predictions.

In general, behaviors of the prediction algorithm using a sliding window depend on the size of the window. When the size of the sliding window is large, predictions using the window tends to be insensitive to outliers. So, the predictions are robust with respect to temporal noises. However, if the tendency of the data stream changes, the predictions tend to become worse, since older data in the sliding window affects so that the prediction algorithm to adapt the change more slowly. On the other hand, when the size of the sliding window is small, the predictions are more sensitive to changes in the data stream. Thus the prediction algorithm can adapt the change quickly. But, its disadvantage is that the predictions become sensitive to temporal noises as well. Therefore, in order to predict adaptively w.r.t. data streams, we need to determine the size of the sliding window appropriately.

2.3 Our Goal

Given a base predictor, our goal is to predict as well as the base predictor using the best fixed-sized sliding window. Specifically, we aim to construct online prediction algorithms whose regret or adaptive regret w.r.t. the base predictor with the best fixed sliding window.

3. Algorithms

In this section, we propose two algorithms, which are modifications of existing algorithms having regret and adaptive regret bounds, respectively.

3.1 Weighted Window

The first algorithm, which we call Weighted Window (WW), the special case of Weighted Average Algorithm [19] with base predictors with sliding windows as experts. More precisely, WW has a sliding window of size M, and the sliding window induces M sub-windows. Each sub-window, which is denoted as $W[i]$ $(i = 1, \ldots, M)$, has the at most i newest examples. We regard the base predictor with each sub-window $W[i]$ as an expert. That is, each expert predicts using the base predictor and the data in the sub-window $W[i]$.

Algorithm 1 WW(Weighted Window)

1. $\boldsymbol{w}_1 = (\frac{1}{M}, \ldots, \frac{1}{M})$.
2. For $t = 1, \ldots, T$
   a. The sliding window W contains at most M newest examples before trial t.
   $$W = \begin{cases} \cup_{j=1}^{t-1}\{(\boldsymbol{x}_j, y_j)\} & \text{if } t-1 \le M, \\ \cup_{j=t-M}^{t-1}\{(\boldsymbol{x}_j, y_j)\} & \text{if } t-1 > M. \end{cases}$$
   Each sub-window $W[i]$ contains at most i newest examples ($i = 1, \ldots, M$).
   b. Receive an instance $\boldsymbol{x}_t$.
   c. Each expert E_i predicts $\hat{y}_{t,i}$ using the sub-window $W[i]$ ($1 \le i \le M$).
   d. Predict $\hat{y}_t = \sum_{i=1}^{M} w_{t,i}\hat{y}_{t,i}$.
   e. Receive the true outcome y_t.
   f. Update the weight vector.
   $$w_{t+1,i} = \frac{e^{-\alpha\ell(y_t,\hat{y}_{t,i})}}{\sum_{i=1}^{M} e^{-\alpha\ell(y_t,\hat{y}_{t,i})}}.$$

Finally, Weighted Average Algorithm combines the experts' predictions by computing the weighted average. The details of WW is given in Algorithm 1.

An advantage of WW is that it predicts adaptively w.r.t. changes of tendency in the data stream. For example, when the tendency of the data changes drastically, experts corresponding to small sub-windows would have larger weights and older data no longer affects the predictions of WW. Similarly, if the tendency of the data does not change, experts corresponding to large sub-windows would have larger weights and predictions of WW would become resistant to temporal outliers in the data stream.

The regret bound of WW is directly follows from that of Weighted Average Algorithm [19].

Theorem 1 (Kivinen & Warmuth [19])**.** *Suppose that the loss function ℓ is α-exp concave. Then the regret of WW is at most* $(1/\alpha)\ln M$.

By Theorem 1, WW is guaranteed to perform almost as well as predictions with best fixed window of size less than M.

3.2 Weighted Window with Follow the Leading History

The second algorithm, Weighted Window with follow the leading History (WWH), is a modification of Follow the Leading History (FLH) [20] with many copies of WWs as experts. Specifically, WWH differs from FLH in that experts of FLH use all the past instances given to them while those of WWH use instances in their sliding windows only. This change makes, as we will show later, practical improvement in changing environments.

At each trial i, WWH generates a copy of WW denoted as WW^i as an expert. Each WW^i has a lifetime $lifetime_i$ and it is only active in $lifetime_i$ trials. Each WW^i runs WW through the data given in the lifetime. Each WW has a sliding window of size M and M sub-windows as sub-experts. At each trial, WWH combines the prediction of experts which are active at the trial.

More precisely, an expert WW^i, which is generated at trial i, is active at trial t if $i + lifetime_i \ge t$. The $lifetime_i$ of expert WW^i is given as follows: If i is represented as $i = r2^k$, where r is some odd number and k is an integer, we fix $lifetime_i = 2^{k+2} + 1$. Note that r and k are unique for each i. For example, if i is odd, then $k = 0$ and $r = i$. Similarly, if i is even, there also exist unique k such that $k \ge 1$ and odd number r satisfying $i = r2^k$. Let A_t be the set of indices of active experts at trial t. Then the following lemma holds [20]. Note that the lemma holds not only for FLH but also for WWH.

Lemma 1 (Hazan & Seshadhri [20])**.**

1. *For any $s \le t$, $[s, (s+t)/2] \cap A_t \ne \emptyset$.*
2. *For any t, $|A_t| = O(\log T)$.*
3. *For any t, $A_{t+1} \setminus A_t = \{t+1\}$.*

The description of WWH is given in Algorithm 2. At each trial WWH combines the predictions of active experts by computing the weighted average. Then WWH generates a new expert. Finally, WWH removes the experts whose lifetimes are zero, and normalize the weights of active experts.

3.3 Analysis

We show a regret bound of WWH. First, we use the lemma for FLH [20].

Lemma 2 ([20])**.** *Suppose that for an interval $I = [r, s]$ WW^r is active. Then, regret of WWH w.r.t. WW^r for the interval I is $\frac{2}{\alpha}(\ln r + \ln |I|)$.*

By Lemma 2 and Theorem 1, we have the following lemma.

Lemma 3. *Suppose that, during the interval $I = [r, s]$, WW^r is active. Then, regret of WWH w.r.t. any sub-window for the interval I is at most $\frac{2}{\alpha}(\ln r + \ln |I| + \ln M)$.*

Then we analyze the regret of WWH w.r.t. any interval $I = [r, s]$ and any sub-window.

Lemma 4. *For any interval $I = [r, s]$, the regret of WWH w.r.t. I is $O(\frac{1}{\alpha}(\ln M + \ln s)\ln |I|)$.*

Proof. By Lemma 1, for the trial s and the interval $I = [r, s]$, there exists $i \in A_s$ such that (i) $i \in [r, \frac{r+s}{2}]$, and (ii) the expert WW^i is generated at trial i and is active at trial s. Therefore, by Lemma 3, the regret of WWH for any sub-window and the interval I is at most $\frac{2}{\alpha}(\ln i + \ln |I| + \ln M)$.

Similarly, for the interval $[r, i]$, there exists i' such that $i' \in [r, \frac{r+i}{2}]$, we can evaluate the regret of WWH for any sub-window and the interval $[i', i]$. Note that, by this argument, the interval for which the regret is evaluated becomes at most a half of the original interval. So, there are at most $\log_2 |I|$ intervals to consider. Thus the regret of WWH for any sub-window and the interval $I = [r, s]$ is at most

Algorithm 2 WWH (Weighted Window with follow the leading History)

1. Let A_1={1} and $w_{1,1} = 1$. Generate the expert WW^1 having WW as its prediction algorithm.
2. For $t = 1, \ldots, T$
   a. Receive an instance $\boldsymbol{x}_t$.
   b. For each $i \in A_t$, the expert WW^i predicts $\hat{y}_{t,i}$
   c. Predict $\hat{y}_t = \sum_{i \in A_t} w_{t,i} \hat{y}_{t,i}$.
   d. Receive the true outcome y_t.
   e. Update:
   $$\hat{w}_{t+1,i} = \frac{w_{t,i} e^{-\alpha \ell(y_t, \hat{y}_{t,i})}}{\sum_{j \in A_t} w_{t,j} e^{-\alpha \ell(y_t, \hat{y}_{t,j})}}$$
   f. Add the new expert WW^t:
   $$\bar{w}_{t+1,i} = \begin{cases} \frac{1}{t+1} & \text{if } i = t+1, \\ (1 - \frac{1}{t+1}) \hat{w}_{t+1,i} & \text{if } i \neq t+1. \end{cases}$$
   g. Let A_{t+1} be the set of indices of active experts at trial $t+1$. For each $i \in A_{t+1}$, let
   $$w_{t+1,i} = \frac{\bar{w}_{t+1,i}}{\sum_{j \in A_{t+1}} \bar{w}_{t+1,j}}.$$

$$\frac{2}{\alpha}(\ln s + \ln |I| + \ln M) \cdot \log_2 |I| = O\left(\frac{1}{\alpha}(\ln M + \ln s) \ln |I|\right).$$

□

Finally, we prove the adaptive regret bound of WWH.

Theorem 2. *The adaptive regret of WWH w.r.t. the best fixed-sized window is* $O(\ln M \ln T + \ln^2 T)$.

Proof. By Lemma 4, the regret of WWH for any interval $I = [r, s]$ and the best sub-window is $O\left(\frac{1}{\alpha}(\ln M + \ln s) \ln |I|\right)$. Since, $s, |I| \leq T$, we complete the proof. □

4. Experiments

We evaluate our proposed algorithms and other previous methods over synthetic and real time series data.

The data we deal with has the following form: $S = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \ldots, (\boldsymbol{x}_t, y_t)\} (1 \leq t \leq T)$, where each $\boldsymbol{x}_t = t$, and $y_t \in [0, 1]$ $(1 \leq t \leq T)$. At each trial t, each prediction algorithm is supposed to predict $\hat{y}_t \in [0, 1]$, given $\boldsymbol{x}_t = t$. The loss function we consider here is square loss, i.e., $\ell(y, \hat{y}) = (y - \hat{y})^2$. It can be shown that square loss $\ell(y, \hat{y})$ is α-exp concave for $\alpha \leq 1/2$ when $y, \hat{y} \in [0, 1]$. Since larger α implies smaller regret (as stated in Theorem 1), we fix $\alpha = 1/2$ when we use WW in our experiments.

We assume that the base prediction algorithm associated with each sub-window performs least square regression. Then, given M examples $(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_M, y_M) \subset (\mathbb{R} \times [0, 1])^M$, the prediction is $\hat{y} = ax + b$, where

$$a = \frac{\sum_{i=1}^{M} (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^{M} (x_i - \bar{x})^2}.$$

$$b = \bar{y} - a\bar{x}.$$

Here, $\bar{x}, \bar{y}$ are denoted as the averages of $\boldsymbol{x}_i$ and y_i $(i = 1, \ldots, M)$, respectively. For WW, a naive implementation takes $O(M^2)$ time to make a prediction. But, it is possible to reduce the computation down to $O(M)$.

The algorithms we evaluate are WW, WWH, FLH [20], ADWIN (ADaptive WINdowing) [3], KAARch (Kernel Aggregating Algorithm for Regression with Changing dependencies) [17], and the best fixed-sized sub-window. Note that all algorithms use least square regression as the base prediction algorithm.

For ADWIN, we set $\delta = 0.9$. For KAARCh, we set $a = \sqrt{Y^2 c^2 T(T-1)/2s(T)}$ as suggested in [18].

4.1 Experiments for Artificial Data

We use the following artificial data sets: Each data set consists of a sequence of 1000 examples.

Radical : a sequence where the values radically changes at $t = 500$ (in Fig. 1).

Gradual : a sequence where the values gradually change at $t = 300, \ldots, 700$ (in Fig. 2).

Temporal : a sequence where an outlier appears after each 200 steps (in Fig. 3).

Random : a sequence where a trend changes at random trials and the degree of the gradient changes as much as randomly determined (in Fig. 4).

For each artificial data, we further add random noises at each trial, where each random noise is generated i.i.d. from $N(0, 0.05)$.

The cumulative loss of each algorithm for each data is shown in Figs. 1, 2, 3, 4, respectively. We set the size of the window $M = 10$ and $\alpha = 1/2$.

For all artificial data sets except from Gradual data, WW and WWH perform better than other algorithms. Further, cumulative losses of WW, WWH and the best window are close. For Gradual data, ADWIN performs best among all algorithms other than the best window. Curiously, even though WWH has a stronger theoretical guarantee (i.e., the adaptive regret), its performance is slightly worse than that of WW. Other algorithms sometimes perform well but sometimes not. In particular, FLH seems not to adapt changes of the data well at later trials. ADWIN shows a good performance for data with gradual changes such as Gradual data, but behaves badly when temporal noises appear in the data such as Temporal data. We omit the plots of KAARCh since its performance is much worse than others.

4.2 Experiments on Real Data

As a real data, we use Nikkei 225, a stock market index for the Tokyo Stock Exchange. The data consists of 6311 daily average stocks (the closing price) of 225 Japanese representative companies, ranging from 1984/1/4 to 2009/8/27 (in

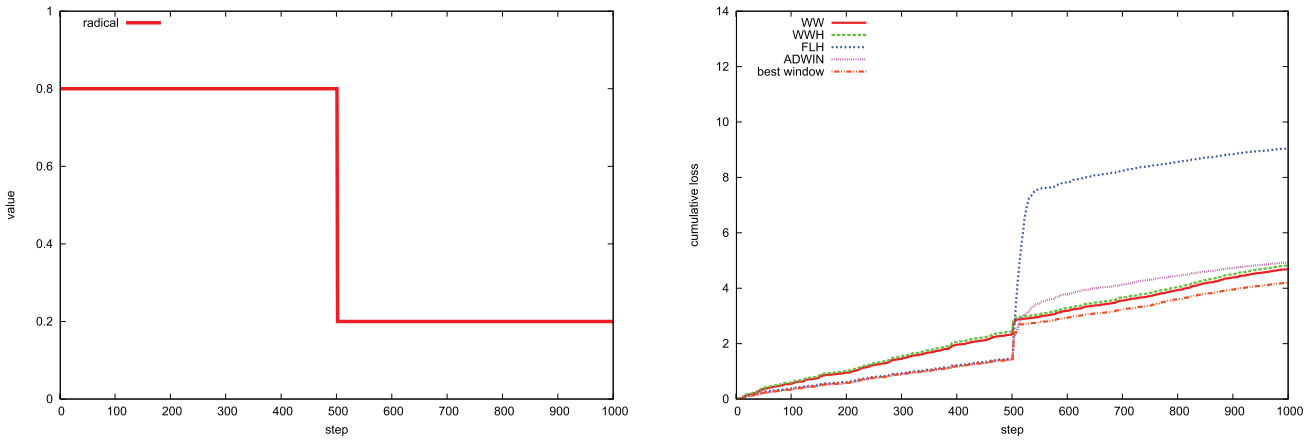

Fig. 1 Radical data (left) and the cumulative losses of the algorithms (right).

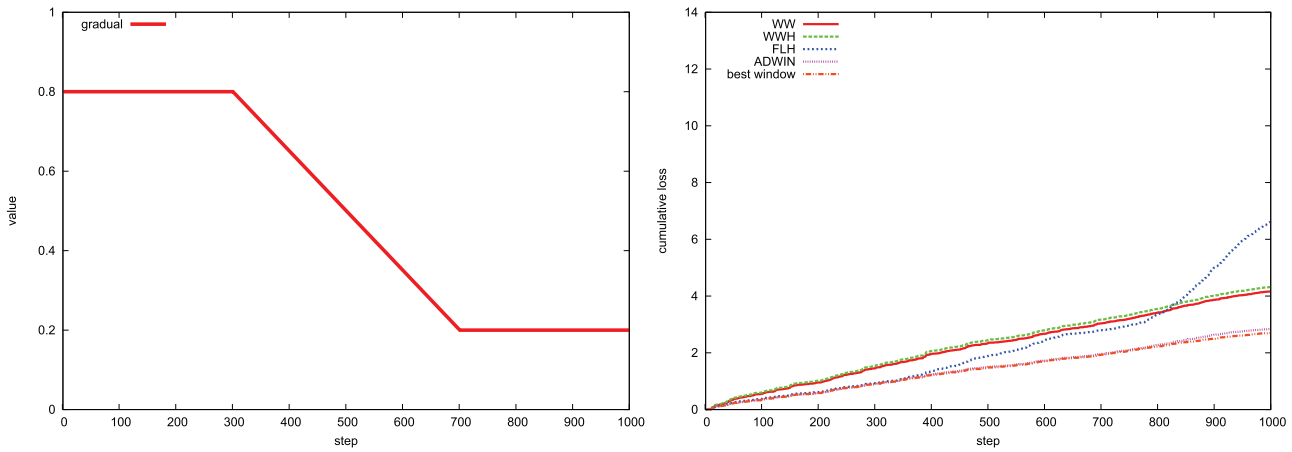

Fig. 2 Gradual data (left) and the cumulative losses of the algorithms (right).

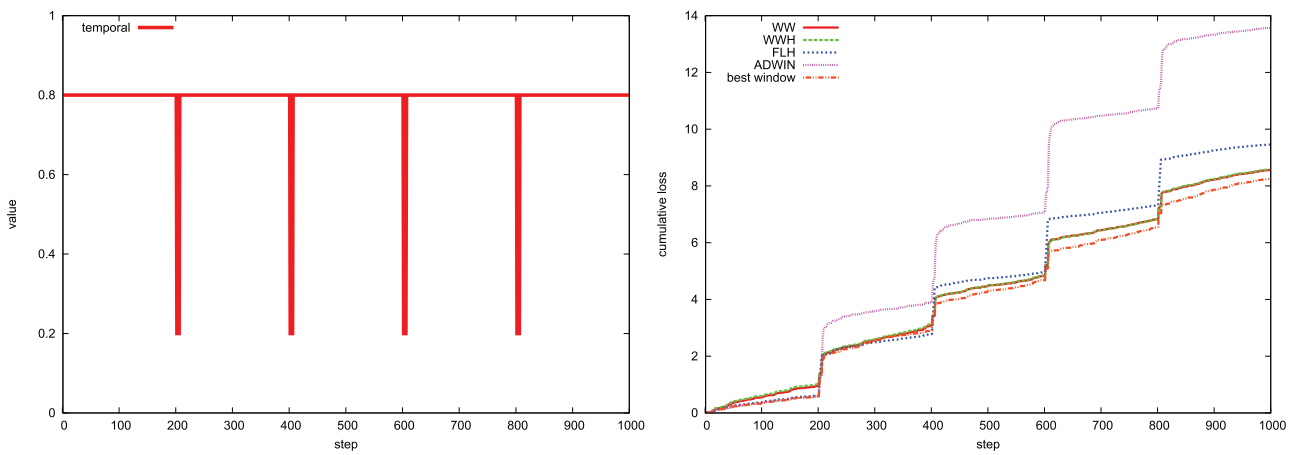

Fig. 3 Temporal data (left) and the cumulative losses of the algorithms (right).

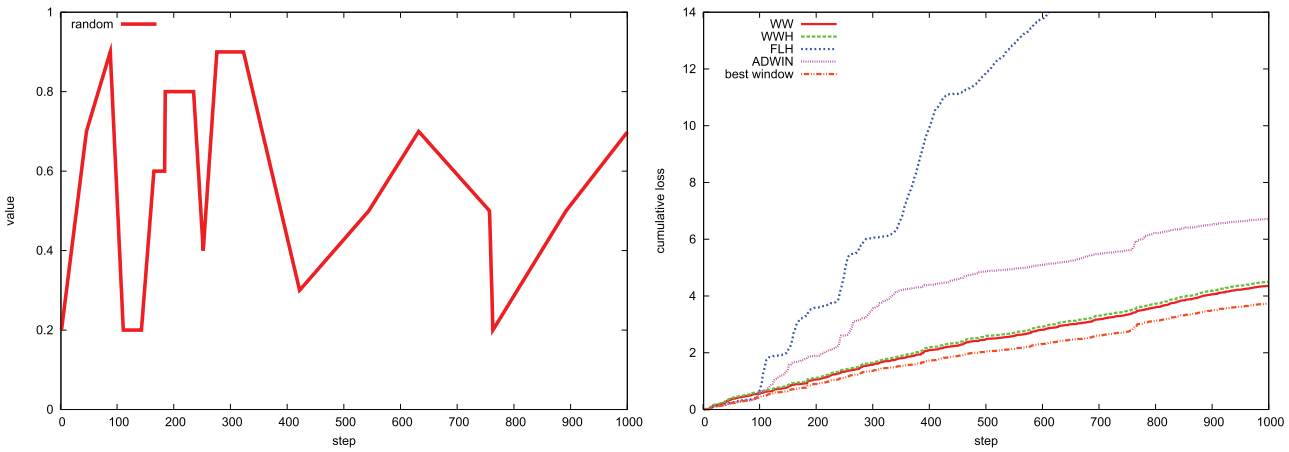

Fig. 4 Random data (left) and the cumulative losses of the algorithms (right).

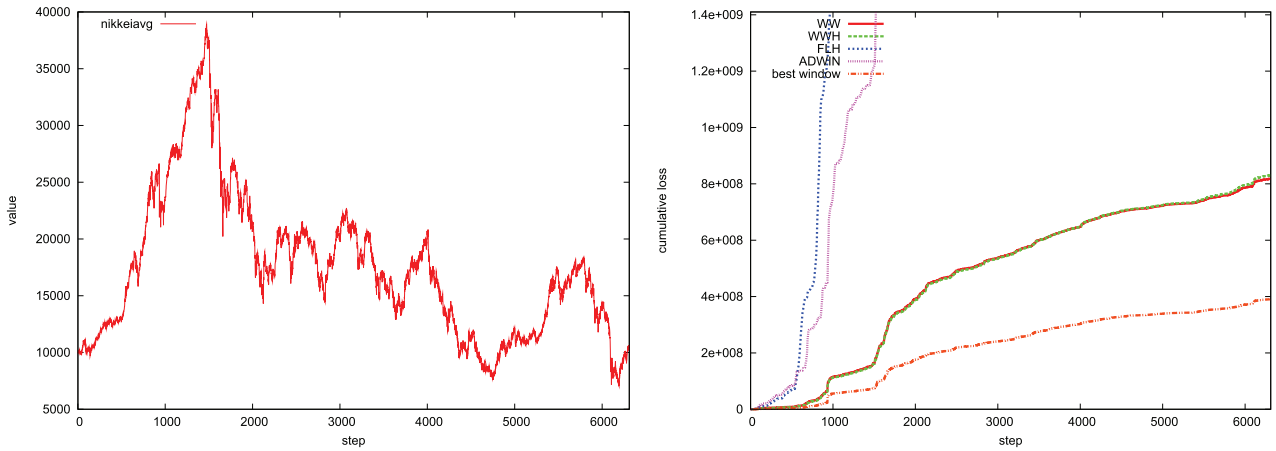

Fig. 5 Nikkei 225 data (left) and the cumulative losses of the algorithms (right).

Fig. 5). We set $M = 50$ for WW and WWH.

For Nikkei 225 data, again, WW and WWH perform the best among other algorithms except from the best window (We omit the plot of KAARCh since the performance is much worse). Similar to the results for artificial data, WWH performs slightly worse than WW.

5. Conclusion

In this paper, we propose algorithms for predicting of data streams, based on techniques of online learning with experts. The first algorithm, WW, combines slide windows of different sizes, and we show it predicts almost as well as the best sub-window. The second algorithm, WWH, further combines WWs over different intervals, having the adaptive regret small.

Acknowledgements

We thank anonymous reviewers for providing many useful comments and suggestions to improve initial manuscript. This research was partially supported by MEXT Grand-in-Aid for Young Scientists (B) 21700171.

References

[1] S. Muthukrishnan, "Data streams: Algorithms and applications," Foundations and Trends in Theoretical Computer Science, vol.1, no.2, 2005.
[2] C.C. Aggarwal, ed., Data Streams: Models and Algorithms, Springer, 2007.
[3] A. Bifet and R. Gavaldà, "Learning from time-changing data with adaptive windowing," Proc. 7th SIAM International Conference on Data Mining (SDM'07), pp.443–449, 2007.
[4] J. Gama, P. Medas, G. Castillo, and P. Rodrigues, "Learning with drift detection," SBIA Brazilian Symposium on Artificial Intelligence, pp.286–295, 2004.
[5] R. Klinkenberg and T. Joachims, "Detecting concept drift with support vector machines," Proc. International Conference on Machine Learning (ICML), 2000.
[6] G. Widmer and M. Kubat, "Learning in the presence of concept drift and hiden contexts," Mach. Learn., vol.23, no.1, pp.69–101, 1996.
[7] V. Vovk, "Aggregating strategies," Proc. 3rd Annual Workshop on Computational Learning Theory, pp.371–386, 1990.
[8] N. Littlestone and M.K. Warmuth, "The weighted majority algorithm," Inf. Comput., vol.108, no.2, pp.212–261, 1994.
[9] N. Cesa-Bianchi and G. Lugosi, Prediction, Learning, and Games, Cambridge University Press, 2006.
[10] M. Herbster and M. Warmuth, "Tracking the best linear predictor,"

J. Machine Learning Research, vol.1, pp.281–309, 2001.
[11] O. Bousquet and M.K. Warmuth, "Tracking a small set of experts by mixing past posteriors," J. Machine Learning Research, vol.3, pp.363–396, 2002.
[12] J.Z. Kolter and M.A. Maloof, "Dynamic weighted majority: An ensemble method for drifting concepts," J. Machine Learning Research, vol.8, pp.2755–2790, 2007.
[13] P. Auer, N. Cesa-Bianchi, and C. Gentile, "Adaptive and self-confident on-line learning algorithms," J. Comput. Syst. Sci., vol.64, pp.48–75, 2002.
[14] M. Herbster and M.K. Warmuth, "Tracking the best expert," Mach. Learn., vol.32, no.2, pp.151–178, 1998.
[15] C. Monteleoni and T.S. Jaakkola, "Online learning of non-stationary sequences," Advances in Neural Information Processing Systems 16 (NIPS'03), 2004.
[16] V. Vovk, "Competitive on-line statistics," International Statistical Review, vol.69, no.2, pp.213–248, 2001.
[17] S. Busuttil and Y. Kalnishkan, "Online regression competitive with changing predictors," Proc. 18th Conference on Algorithmic Learning Theory (ALT'07), pp.181–195, 2007.
[18] S. Busuttil and Y. Kalnishkan, "Weighted kernel regression for predicting changing dependencies," Proc. 18th European Conference on Machine Learning, pp.535–542, 2007.
[19] J. Kivinen and M.K. Warmuth, "Averaging expert predictions," Proc. 4th European Conference on Computational Leanring Theory (EuroCOLT'99), pp.153–167, 1999.
[20] E. Hazan and C. Seshadhri, "Efficient learning algorithms for changing environments," Proc. 26th Annual International Conference on Machine Learning (ICML'09), 2009.

Shin-ichi Yoshida received B.E and M.E. degrees from Kyushu University in 2008 and 2010, respectively. He now works for NTT West.

Kohei Hatano received Ph.D. from Tokyo Institute of Technology in 2005. Currently, he is an assistant professor at Department of Informatics in Kyushu University. His research interests include boosting, online learning and their applications.

Eiji Takimoto received Dr. Eng. degree from Tohoku University in 1991. Currently, he is a professor at Department of Informatics in Kyushu University. His research interests include computational complexity, computational learning theory, and online learning.

Masayuki Takeda received Dr. Eng. degree from Kyushu University in 1996. Currently, he is a professor at Department of Informatics in Kyushu University. His research interests include string algorithms, data compression, and discovery science.