Adaptive Learning Rate for Online Linear Discriminant Classifiers

Ludmila I. Kuncheva and Catrin O. Plumpton

School of Computer Science,
Bangor University, Dean Street, Bangor Gwynedd LL57 1UT, UK
{l.i.kuncheva,c.o.plumpton}@bangor.ac.uk

Abstract. We propose a strategy for updating the learning rate parameter of online linear classifiers for streaming data with concept drift. The change in the learning rate is guided by the change in a running estimate of the classification error. In addition, we propose an online version of the standard linear discriminant classifier (O-LDC) in which the inverse of the common covariance matrix is updated using the Sherman-Morrison-Woodbury formula. The adaptive learning rate was applied to four online linear classifier models on generated and real streaming data with concept drift. O-LDC was found to be better than balanced Winnow, the perceptron and a recently proposed online linear discriminant analysis.

1 Introduction

In online classification, the data points[1] arrive one at a time. The classifier predicts a label, and immediately after that receives the correct label. The new data point is used to update the classifier. Although such instant feedback may not be available in real-life scenarios, this is a standard assumption that underpins incremental learning. The major advantage of this feedback is that the classification error can be monitored and changes in the classification environment (concept drift or concept shift) can be detected almost at their onset. A good online classifier should be able to self-tune to respond to changes. The classical online models, such as the perceptron [1] and the Winnow family [2,3,4,5] use a constant learning rate parameter to guide the classifier updates in case of a misclassification. Typically the value of the learning rate is fixed in advance through cross-validation experiments on a training data set. Using a fixed learning rate acts as a "forgetting mechanism" for the classifier because the newcoming points are given more weight in the online training. Here we propose to modify the learning rate as a function of a running estimate of the classification error. An increasing error rate may herald the onset of a change in the classification environment. In this case, an increased learning rate is desirable, so that the classifier "forgets more quickly" old observations. The larger the jump in the error is, the larger the increase of the learning rate should be.

[1] The term is used here as a synonym of *observations, examples, instances.*

N. da Vitora Lobo et al. (Eds.): SSPR&SPR 2008, LNCS 5342, pp. 510–519, 2008.

A notable omission from the toolbox of online linear classifiers is the classical Linear Disctiminant Classifier (LDC) [6] where the classes are assumed to be Gaussian with a common covariance matrix. The problem with the online implementation of LDC is that the update requires the inverse of the new covariance matrix. While the covariance matrix itself can be updated easily with a new observation, its inverse will have to be computed anew. Even though this can be done in constant time, running LDC online may become prohibitively expensive, especially for large number of features. Hence we propose here an Online Linear Discriminant Classifier (O-LDC) which avoids the matrix inverse at each step. A learning rate, λ is introduced to weight the contribution of the new data point.

The rest of the paper is organized as follows. Related work is discussed in Section 2. Section 3 introduces O-LDC. Experimental results with generated and real data for changing environments are presented in Section 6. Section 7 concludes the study.

2 Online Linear Classifiers

Consider data points in the n-dimensional real space, $\mathbf{x} \in \Re^n$. Let $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ be a data set, $\mathbf{x}_j \in \Re^n$, labelled in c classes.

2.1 Perceptron for Streaming Data (2 Classes)

The value for the learning rate η is chosen first. The coefficients of the linear discriminant function between the two classes, $\mathbf{w} = [w_0, \ldots, w_n]^T$, are initialised as small random numbers. Let $\mathbf{x}$ be the new observation, and $y(\mathbf{x}) \in \{-1, +1\}$ be its class label. Denote by $\mathbf{z} = [1 \quad \mathbf{x}^T]^T$ the augmented data vector. The first element, 1, multiplies the bias coefficient w_0. The predicted label ($+1$ or -1) for data point $\mathbf{x} \in \Re^n$, is calculated as

$$y_{\text{predicted}} = \text{sign}\left(\mathbf{z}^T \mathbf{w}\right),$$

where 'sign' is the signum function[2]. If $y_{\text{predicted}} \neq y(\mathbf{x})$, the weights are updated by $\mathbf{w} \leftarrow \mathbf{w} - \eta\, \mathbf{z}\, y_{\text{predicted}}$. Otherwise, the weights do not change.

2.2 Balanced Winnow for Streaming Data (2 Classes)

Balanced Winnow shares the same online update concept with the perceptron in that the weights are updated only upon a wrong prediction. First, a value for the learning rate β is chosen, $0 < \beta < 1$. There are two sets of weights, a positive set $\mathbf{w}^+$ and a negative set $\mathbf{w}^-$, all of which are initialized as positive random numbers. Using the augmented vector $\mathbf{z}$, the prediction formula is

$$y_{\text{predicted}} = \text{sign}\left(\mathbf{z}^T(\mathbf{w}^+ - \mathbf{w}^-)\right).$$

[2] Sign$(a) = 1$, if $a \geq 0$ and sign$(a) = -1$, if $a < 0$ The value of the function for $a = 0$ is irrelevant here and can be taken to be either $+1$ or -1.

When $y_{\text{predicted}} \neq y(\mathbf{x})$, the weights are updated by the following clause

if $y(\mathbf{x}) = +1$ then

$$w_i^+ \leftarrow \beta^{-z_i} w_i^+, \quad w_i^- \leftarrow \beta^{z_i} w_i^-, \quad i = 0, 1, \ldots, n. \tag{1}$$

else

$$w_i^+ \leftarrow \beta^{z_i} w_i^+, \quad w_i^- \leftarrow \beta^{-z_i} w_i^-, \quad i = 0, 1, \ldots, n. \tag{2}$$

2.3 Online Linear Discriminant Analysis (LDA) (c Classes)

The original data space is transformed as $\mathbf{y} = A^T\mathbf{x}$ where A is an $n \times m$ transformation matrix such that separability between the classes is maximised. The nearest mean classifier is applied in the new space. The procedure is equivalent to calculating linear discriminant functions in the original space. Fisher's linear discriminant analysis uses the Rayleigh quotient to measure separability. This criterion has the between-class scatter matrix, B, in the numerator and the within-class scatter matrix, W in the denominator. To make LDA suitable for online updates, Hiraoka et al. [7] propose an alternative criterion. The transformation matrix A is found by an iterative maximisation of the following potential function

$$\phi(A) = tr\left(A^T B A \left(I - \frac{1}{2} A^T W A)\right)\right),$$

assuming that the number of features in the new space, m, is chosen and fixed. It is stated in Ref [7] that the number of features m in the new space should be less than the number of classes c. In our case, two-class comparisons are carried out, which leaves only one dimension for the new space. In a set of pilot experiments we found that this seems to hamper Hiraoka's online LDC, hence we will use here $m = n$ (assuming non-singular covariance matrix). The learning rate parameter, ζ, for this classifier is a part of the iterative algorithm for computing A. Large learning rate places more weight on the new objects, as in the perceptron algorithm.

3 Online Linear Discriminant Classifier (O-LDC) (c Classes)

3.1 Online LDC for Static Environments

Let c be the total number of classes, $P^{(i)}$ be the prior probability for class i, $\mu^{(i)} \in \Re^n$ be the class mean, and Σ be the $n \times n$ covariance matrix, common for all classes. The linear discriminant classifier calculates c discriminant functions

$$g_i(\mathbf{x}) = \ln\ P^{(i)} - \frac{1}{2}{\mu^{(i)}}^T \Sigma^{-1} \mu^{(i)} + {\mu^{(i)}}^T \Sigma^{-1}\mathbf{x}, \quad i = 1, \ldots, c. \tag{3}$$

The class with the largest $g_i(\mathbf{x})$ is assigned to $\mathbf{x}$. This classifier guarantees minimum classification error (Bayes error) when the classes have normal distributions

and equal covariance matrices. In an online setting the classifier is initialised by calculating estimates for the priors, the class means and the common covariance matrix from a training data set. A stream of data points is then submitted for classification. In order to update the discriminant functions so as to correspond to the new training set after each new observation, we need to update the class means and the *inverse* of the covariance matrix.

Let $\{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ be a data set, $\mathbf{x}_j \in \Re^n$, and $\mathbf{m}_{N_i}^{(i)}$ be the maximum likelihood estimate of the mean for class i, where N_i is the number of points from class i ($N_1 + N_2 + \ldots + N_c = N$). Upon receiving data point $\mathbf{x}_{N+1}$ with label k, we update of the mean for class k using

$$\mathbf{m}_{N_k+1}^{(k)} = \frac{1}{N_k+1}(N_k \mathbf{m}_{N_k}^{(k)} + \mathbf{x}_{N+1}). \tag{4}$$

Denote by S_N the maximum likelihood estimate of the covariance matrix Σ.

$$S_N = \frac{1}{N} \sum_{i=1}^{c} \sum_{\mathbf{x} \in \text{ class } i} \left(\mathbf{x} - \mathbf{m}_{N_i}^{(i)}\right) \left(\mathbf{x} - \mathbf{m}_{N_i}^{(i)}\right)^T.$$

Knowing that $\mathbf{x}_{N+1}$ comes from class k, denote $\mathbf{z} = \mathbf{x} - \mathbf{m}_{N_k+1}^{(k)}$. Then

$$S_{N+1} = \frac{N}{N+1}\left(S_N + \frac{1}{N}\mathbf{z}\mathbf{z}^T\right). \tag{5}$$

To apply the online updates, we reorganize (5) as

$$S_{N+1} = \frac{N}{N+1}\left(S_N + \sqrt{\frac{1}{N}}\,\mathbf{z}\ \ \sqrt{\frac{1}{N}}\,\mathbf{z}^T\right) \tag{6}$$

If A is an invertible matrix and $\mathbf{v}$ is a vector, the Sherman-Morrison-Woodbury formula states that

$$\left(A + \mathbf{v}\mathbf{v}^T\right)^{-1} = A^{-1} - \frac{A^{-1}\,\mathbf{v}\,\mathbf{v}^T\,A^{-1}}{1 + \mathbf{v}^T A^{-1}\mathbf{v}}. \tag{7}$$

Applying (7) to (6),

$$S_{N+1}^{-1} = \frac{N+1}{N}\left(S_N^{-1} - \frac{S_N^{-1}\,\mathbf{z}\,\mathbf{z}^T\,S_N^{-1}}{N + \mathbf{z}^T\,S_N^{-1}\,\mathbf{z}}\right), \tag{8}$$

which is the update of the covariance matrix for $\mathbf{x}_{N+1}$.

Finally, the priors are estimated as $P_N^{(i)} = N_i/N$ and updated as

$$P_{N+1}^{(i)} = \begin{cases} \frac{N_i}{N+1}, & i \neq k \\ \frac{N_i+1}{N+1}, & i = k \end{cases}. \tag{9}$$

With $P_{N+1}^{(i)}$, $\mu_{N+1}^{(i)}$ and S_{N+1}^{-1} in place, the new discriminant functions $g_i(\mathbf{x})$ can be calculated.[3]

[3] In training O-LDC, when the covariance matrix of the training data appeared to be singular, we replaced it by the identity matrix.

3.2 Online LDC for Changing Environments

To enable updates that will accommodate changing environments, we introduce a learning rate λ to account for the weight of the new data point, $0 < \lambda < 1$. For $\lambda \to 0$, there is no update, while for $\lambda \to 1$ the classifier forgets everything except the new data point. When $\lambda = 1/2$, the update is exactly the one given by equations (4), (8) and (9). For values from 0 to 0.5 the classifier is "reluctant" to train while with $\lambda > 1/2$ it is "eager" to train. To accomplish this, the update formulas for the means are

$$\mathbf{m}_{N_k+1}^{(k)} = \frac{1}{(1-\lambda)N_k + \lambda} \times \left((1-\lambda) N_k \mathbf{m}_{N_k}^{(k)} + \lambda \, \mathbf{x}_{N+1} \right). \tag{10}$$

The prior probabilities are updated as

$$P_{N+1}^{(i)} = \begin{cases} \frac{(1-\lambda)N_i}{(1-\lambda)N+\lambda}, & i \neq k \\ \frac{(1-\lambda)N_i+\lambda}{(1-\lambda)N+\lambda}, & i = k \end{cases}. \tag{11}$$

Finally, by weighting the $\mathbf{z}$-term in (5) by λ, and S_N by $(1-\lambda)$, while keeping the overall "soft" count at $(1-\lambda)N + \lambda$, we derive the update for the covariance matrix

$$S_{N+1}^{-1} = \frac{(1-\lambda)N + \lambda}{(1-\lambda)N} \times \left(S_N^{-1} - \frac{S_N^{-1} \, \mathbf{z} \, \mathbf{z}^T \, S_N^{-1}}{\frac{(1-\lambda)N}{\lambda} + \mathbf{z}^T \, S_N^{-1} \, \mathbf{z}} \right), \quad \lambda \neq 0, \lambda \neq 1. \tag{12}$$

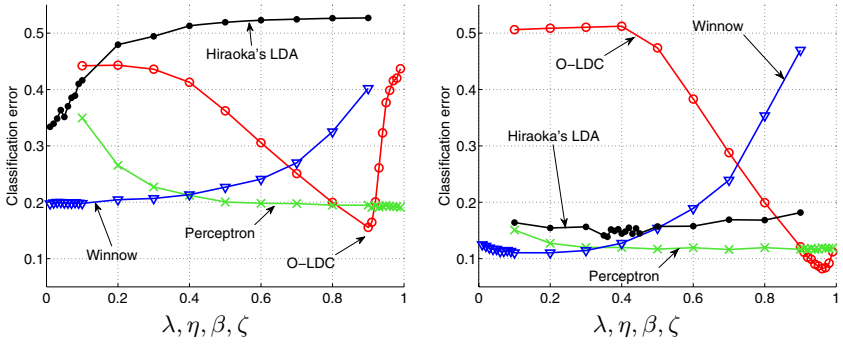
4 Adaptive Learning Rate

4.1 Fixed Learning Rate - An Illustration

The effect of the learning rate on the classification accuracy is demonstrated below on two synthetic data sets simulating abrupt changes (STAGGER data) and gradual changes (moving plane data).

STAGGER data. We tested O-LDC, Perceptron, Winnow and Hiraoka's LDA on the popular STAGGER data used in [8]. Each data point is described by 3 features, each with three possible categories: size $\in$ {small, medium, large}, colour $\in$ {red, green, blue} and shape $\in$ {square, circular, triangular}. Three classification tasks were to be learned in a course of 120 points. From point 1 to point 40, the classes to be distinguished are [size = small AND colour = red] vs all other values; from 41 to 80, [colour = green OR shape = circular] vs all other values; and from 81 to 120, [size = small OR size = large] vs all other values. For each data point submitted as a part of the streaming data, an independent testing set of 100 objects was generated and labelled according to the current class description. The classifier was tested after each submission.

Here we are interested in the error rates of the four online linear classifiers with a *pre-fixed* parameter value that does not change with time. Therefore we varied the parameters that control the learning speed and then fine-tuned them



(a) STAGGER data (b) moving plane data

Fig. 1. Classification error versus learning rate

around the best value, all on the *testing* set. With each parameter value and each method we carried out 100 runs. The classification error was calculated as the mean error across the whole online training. Figure 1 (a) plots the error versus the parameter values.

Although O-LDC gives the best error rate, the range of values for the optimal λ is small. On the other hand, the perceptron and Winnow are fairly robust with respect to their learning rate. As expected, larger values of η and smaller values of β lead to faster learning, hence better overall results. The dip in the O-LDC error curve shows a balance between memorising past data and learning new data.

Moving plane data. A gradual changing environment was simulated by rotating the linear class boundary about the origin in a 2d space [9]. All data points came from a uniform distribution in the unit square. The class definition (concept) was changed with each new data point by rotating the boundary at a further angle of 1°. Starting with $\theta = 0$ and finishing with $\theta = 359°$, we had a sequence of 360 data points. All classifiers were initialized at $\theta = 0$ using a random set of 10 points. An additional testing data set of 100 points was generated for each angle θ and used to evaluate the classifier performance. Figure 2 gives four snapshots of the moving-plane class configurations for $0 \leq \theta \leq 90°$.

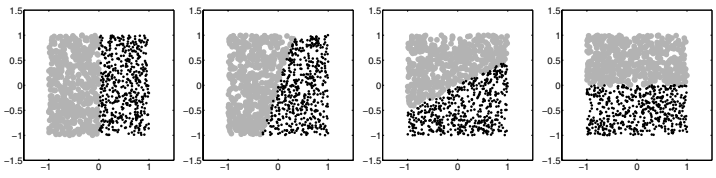


Fig. 2. Snapshots of the moving-plane data for angle $0 \leq \theta \leq 90°$

Table 1. Average error for the *best* choice of parameter values, the 95% confidence intervals and the optimal parameter values

Dataset	O-LDC	Perceptron	Winnow	Hiraoka's LDA
STAGGER	**0.156**±0.024	0.191±0.023	0.197±0.025	0.334±0.029 •
	($\lambda = 0.90$)	($\eta = 0.91$)	($\beta = 0.01$)	($\zeta = 0.01$)
Moving plane	**0.082**±0.005	0.116±0.003 •	0.110±0.009 •	0.139±0.005 •
	($\lambda = 0.96$)	($\eta = 0.70$)	($\beta = 0.20$)	($\zeta = 0.36$)
ELEC2.2	**0.162**±0.0342	0.169±0.035 •	0.171±0.035 •	0.190±0.036 •
	($\lambda = 0.50$)	($\eta = 0.90$)	($\beta = 0.10$)	($\zeta = 0.50$)

We varied the parameters of the algorithms, λ, η, β and ζ in the same way as with the STAGGER data. The results are presented in Figure 1 (b). A pattern similar to that with the STARGGER data is observed. O-LDC reaches the best error rate but only for a small range of λ. This time Hiraoka's LDA shows a good and stable performance across all tested values of ζ but is dominated by the perceptron. To evaluate the statistical significance of the differences, Table 1 shows the errors averaged across the length of the runs and then across the 100 runs. Only the *best* case is shown. The table also contains the 95% confidence intervals. The best results are indicated in boldface and the statistically significant differences in comparison with O-LDC are marked with '•'.

5 Variable Learning Rate

Let $W(t)$ be a window containing the past M objects in the streaming data, $W(t) = \{\mathbf{x}_{t-M+1}, ..., \mathbf{x}_t\}$. Denote by E_t the error rate of the objects in W, calculated in the course of the online classification. The error difference

$$\Delta_e = E_{t-M} - E_t$$

can be construed as a measure of the change in the environment in the past M steps. Large negative Δ_e will signify an abrupt deterioration in the classification accuracy. We propose to modify the learning rates of the four classifiers by taking the magnitude of Δ_e into account. The larger the change in the error, the larger the change in the learning rate in the direction of forgetting old data.

O-LDC	$\lambda \leftarrow \lambda^{(1+\Delta_e)}$
Perceptron	$\eta \leftarrow \eta^{(1+\Delta_e)}$
Winnow	$\beta \leftarrow \beta(1 + \Delta_e)$
LDA	$\zeta \leftarrow \zeta^{(1+\Delta_e)}$

For λ, η and ζ, the larger the learning rate, the more responsive the classifier. As all learning rates are within the interval [0,1], a power of $(1 + \Delta_e)$ will lead

to an increase when $\Delta_e < 0$ and a decrease when $\Delta_e > 0$. The Winnow learning rate, β, on the other hand, already is used as the base for an exponent in the weights update ((1) and (2)). Thus it receives a more "gentle" update by only multiplying the old value by $(1 + \Delta_e)$. Negative Δ_e will derease β as smaller β makes the balanced Winnow more responsive. In all four models, we only update the learning rate if the current data point is misclassified.

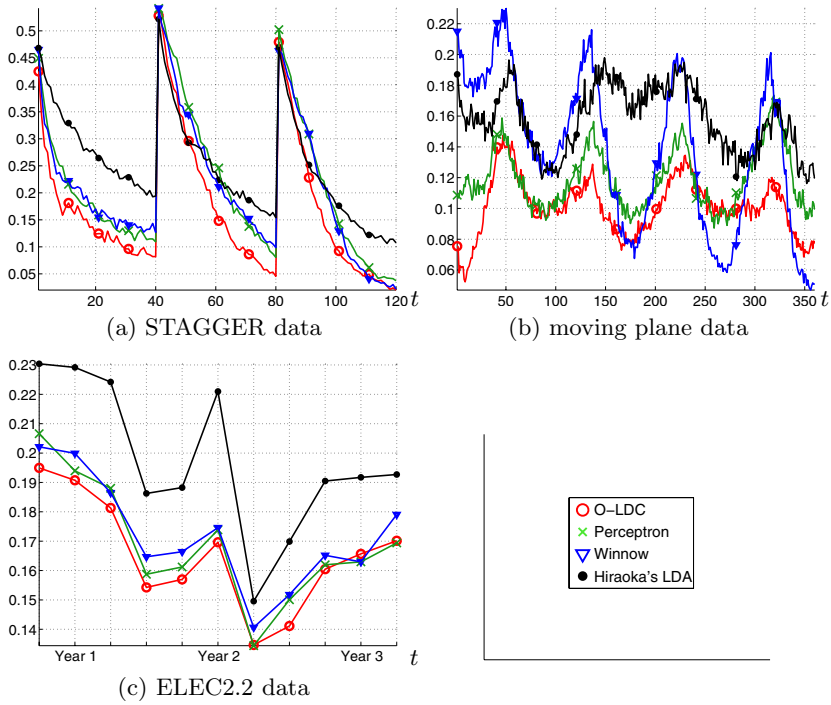
At first glance it seems that we merely replace one parameter choice (the learning rate) with another (the moving window size). However, additional experiments with different window sizes showed that all four algorithms are much less sensitive to the window size than to the choice of their respective learning rates.

6 Experimental Results

The purpose of the experiment was to find out whether the automatic adaptation of the learning rate can lead to results similar to these with the optimal parameter values. The behaviour of the four online *linear* classifier models: O-LDC, perceptron, Winnow and Hiraoka's LDA was examined on the two synthetic data sets STAGGER and moving plane, and also on a real data set (ELEC2) used in other studies of classification in changing environments [10,11,12].

Figures 3 (a) and (b) show the patterns of the error progression along the online training for the two synthetic data sets. The two error peaks for the STAGGER data correspond to the concept shifts at observation 40 and 80. The "wavy" pattern for the moving plane data is caused by the inertia in the reaction of the classifier to the changes. Table 2 shows the mean classification errors for the four classifiers *with the adaptive learning rate*. In all cases the classifiers achieve error rates similar to these with the optimal choice of their respective learning rates (note that the optimal learning rate values were found on the *testing* data). O-LDC gives best overall results, where the difference is statistically signifiant ($p < 0.05$).

Electricity Market data set. This data set is one of the few publically available benchmark data sets for changing environments [11]. The version named Elec 2.2. was used here. It consists of 45,312 data points, each represented by three fatures: day of the week, time of the day and electricity demand of New South Wales, Australia at the time. The data set is a collection of successive measurements every 30 minutes, spanning the period from May 1996 to December 1998. The class label of each point is either UP or DOWN, referring to whether the electricity price at the specified time is higher or lower than the average price of the preceding 24 hours. In our experiments we used the error of the new data point in the sequence as the testing error before retrieving the correct label and adding the data point to the training set. Thus the overall error from an experiment is the average of correct/wrong predictions on the whole data set. Table 1 shows the error rates with the optimal parameter choice and Table 2, with the adaptive leanring rate. The error progression is displayed in Figure 3 (c)



(a) STAGGER data (b) moving plane data

(c) ELEC2.2 data

Fig. 3. Classification error vs number of online observations t

Table 2. Average error with *adaptive learning rate* and 95% confidence intervals

Dataset	O-LDC	Perceptron	Winnow	Hiraoka's LDA
STAGGER	**0.171**±0.022	0.216±0.023	0.211±0.022 •	0.251±0.017 •
Moving plane	**0.101**±0.002	0.119±0.002 •	0.138±0.005 •	0.158±0.002 •
ELEC2.2	**0.165**±0.035	0.169±0.035	0.172±0.035	0.198±0.036 •

for the four online linear classifiers. Again, O-LDC shows superior performance compared to the perceptron, balanced Winnow and the LDA.

7 Conclusions

Linear online classifiers were found to be sensitive to the choice of their learning rate parameter. We propose a strategy for automatic updating of the learning rate based upon the magnitude of the error change during the online training. We also propose an online linear discriminant classifier (O-LDC) that is able to work with streaming data and changing environments. The inverse of the covariance matrix is updated through the Sherman-Morrison-Woodbury formula. The experiments demonstrated that the adaptive learning rate leads to error rates very close to those with the optimal learning rates. The results favoured

O-LDC in comparison with the perceptron, the balanced Winnow and a recently proposed online linear discriminant classifier, which we refer to as Hiraoka's LDA.

References

1. Rosenblatt, F.: Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms. Spartan Books, Washington (1962)
2. Littlestone, N.: Learning quickly when irrelevant attributes abound: A new linear threshold algorithm. Machine Learning 2, 285–318 (1988)
3. Grove, A., Littlestone, N., Schuurmans, D.: General convergence results for linear discriminant updates. Machine Learning 43(3), 179–210 (2001)
4. Wang, B., Jones, G.J.F., Pan, W.: Using online linear classifiers to filter spam emails. Pattern Analysis and Applications 9(4), 339–351 (2006)
5. Carvalho, V.R., Cohen, W.W.: Notes on single-pass online learning. Technical Report CMU-LTI-06-002, Carnegie Mellon University (2006)
6. Duda, R.O., Hart, P.E., Stork, D.G.: Pattern Classification, 2nd edn. John Wiley & Sons, Chichester (2001)
7. Hiraoka, K., Yoshizawa, S., Hidai, K., Hamahira, M., Mizoguchi, H., Mishima, T.: Convergence analysis of online linear discriminant analysis. In: Proc. Int. Joint Conf. on Neural Networks, vol. 3, pp. 387–391 (2000)
8. Widmer, G., Kubat, M.: Learning in the presence of concept drift and hidden contexts. Machine Learning 23, 69–101 (1996)
9. Narasimhamurthy, A., Kuncheva, L.I.: A framework for generating data to simulate changing environments. In: Proc. IASTED, Artificial Intelligence and Applications, Innsbruck, Austria, pp. 384–389 (2007)
10. Gama, J., Medas, P., Castillo, G., Rodrigues, P.: Learning with drift detection. In: Bazzan, A.L.C., Labidi, S. (eds.) SBIA 2004. LNCS, vol. 3171, pp. 286–295. Springer, Heidelberg (2004)
11. Harries, M.: Splice-2 comparative evaluation: Electricity pricing (1999)
12. Bifet, A., Gavaldà, R.: Learning from time-changing data with adaptive windowing. In: Proceedings of the Seventh SIAM International Conference on Data Mining, Minneapolis, Minnesota, USA, pp. 443–448 (2007)