Theoretical Window Size for Classification in the Presence of Sudden Concept Drift

Indrė Žliobaitė[a], Ludmila I. Kuncheva[b]

[a] *Faculty of Mathematics and Informatics, Vilnius University*
Naugarduko 24, Vilnius LT-03225, Lithuania
zliobaite@gmail.com
[b] *School of Computer Science, Bangor University*
Dean Street, Bangor Gwynedd LL57 1UT, UK
l.i.kuncheva@bangor.ac.uk

Abstract

In classifying sequential data, a new classifier is needed after a sudden concept change. However, the old classifier may be better than a new classifier trained on a small window of new data. We derive a general formula for the size of this window, with a closed-form expression for two equiprobable Gaussian classes. Numerical experiments demonstrate that swapping the classifiers after the window has been acquired is better than using the new classifier right after the change or not modifying the classifier at all.

Keywords: pattern recognition, on-line algorithms, concept drift, variable window size, linear discriminant classifier

1. Introduction

Streamline pattern recognition and machine learning have been criticised for not addressing adequately the challenges of real-life problems [1]. Concept change (also termed concept drift or population drift) is ubiquitous, multifaceted and difficult to handle. Given the complexity of the topic, there is only a handful of idiosyncratic theoretical studies focused on small sub-problems, usually bound by constraints and assumptions [2, 3, 4]. A theoretical break-through is likely to come upon accumulation of a critical mass of such "building blocks". Our study is meant to contribute to this collection.

When a classifier is faced with changes in the underlying problem (concept drift), it needs a mechanism to adapt to these changes. The easiest solution is to keep a window over the incoming data and re-train the classifier on the data in the most recent window. The window size is crucial because it determines the flexibility of the classifier, which needs to match the style and pace of the changes. If the window is too small, the classifier will tend to learn all the noise in the data. Conversely, large windows will make the classifier inert and insensitive to changes.

Window size has been discussed at length in relation to change detection [5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15], including frequent itemset mining [26]. The size reduction of the window after the change detection is typically guided by heuristics. Predefined reduction rate [6] seems to be an obvious starting point. Upon a change detection, the window shrinks to a fixed minimum. Gama et al [12] go a step further by constructing a window that starts at the first sign of the change (entering a "warning zone") and contains all observations up to the point when the concept drift is "confirmed". The observations in this window are supposed to have come after the onset of the change, and are thus used as the new training data. Exhaustive backward search through a host of past of data [7, 10] is the next logical step. The most recent part of the data, where the error is significantly higher than the error on the older part, is retained as the new window. Since exhaustive search may be too demanding computationally, golden section search through possible cut-off points of the past data has been considered [11]. Theoretical results have been derived for splitting a change-detection window [10], in the form of bounds on the false positive and false negative detections. On the other hand, theoretical results that relate the *training* window size with the online classification accuracy in changing environments are still in demand.

Consider the following real-time classification scenario. A sequence of i.i.d. data comes from source S_1. At time t_0 a sudden concept shift occurs, in which source S_1 is replaced by source S_2. Assume that t_0 is known but the probability distributions corresponding to the two sources are unknown. Suppose that we choose a classifier model and train it progressively on the data from S_1 by expanding the training window with each new observation (data point). At t_0 the trained classifier becomes obsolete and should be replaced by a new classifier trained on S_2. Let C_1 be the classifier trained on the data from S_1, and C_2 be the classifier trained on the data from S_2. Since the data comes in a sequence, it would be in deficit straight after the change, and the newly trained C_2 will have erratic performance. On the other hand, if S_1 and S_2 are similar, the old classifier may still be more accurate than the new classifier until a sufficient training window of data coming from S_2 is accumulated.

Here we are interested in finding a relationship between the error jump and the size of the data window used for training the classifier after the concept change. In this way we can estimate the "switch point", i.e., the time point t ($t > t_0$) at which we should stop using C_1 and start using C_2 trained on the past $N^* = t - t_0$ observations. Figure 1 illustrates the problem. It shows

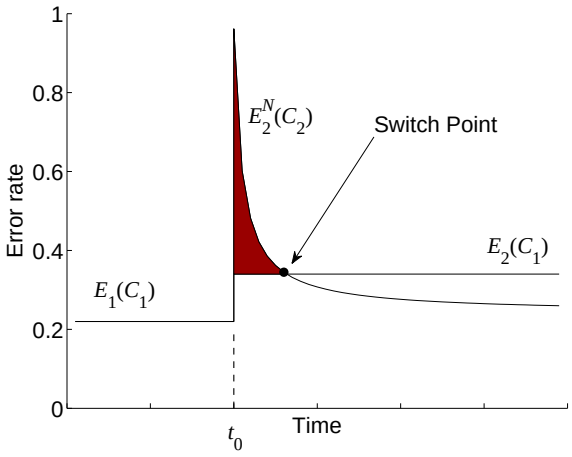


Figure 1: Error rates of C_1 and C_2. The time of the concept shift, t_0, is indicated by a vertical dashed line. The dark-shaded area is the savings in the error if we switch from C_1 to C_2 at the designated switch point.

the error rates of C_1 and C_2, as well as the "saving" in the error if we switch from C_1 to C_2 at the designated switch point. The error reduction depends upon the magnitude of the change, the way the classifier is affected by the training sample size and the asymptotic error achievable by C_2.

The rest of the paper is organised as follows. Section 2 gives the theoretical derivation of the optimal window size N^*. The general case is discussed first, followed by a special case of two equiprobable Gaussian classes in $\Re^n$ with equal covariance matrices. In Section 3 we analyse N^* with respect to the direction and magnitude of the concept drift. Numerical experiments are carried out to evaluate the sensitivity of the window size to inaccurate underlying assumptions. Practical issues are discussed. Section 4 concludes the study and outlines some open problems.

2. Optimal window size after sudden concept drift

2.1. The general case

Let C be the chosen classifier whose parameters are calculated from a sample of size N. Denote by $E^N(C)$ the theoretical error achievable by C on a training data set of size N. Let $E(C)$ be the asymptotic error rate of C obtained as $E(C) = \lim_{N\to\infty} E^N(C)$. Fukunaga and Hayes [16] show that, for any parametric classifier C, regardless of the types of the probability density functions (pdfs) and the priors, the classification error can be expressed approximately as

$$E^N(C) \approx E(C) + \frac{1}{N} f(C), \tag{1}$$

where $f(C)$ is a function that depends on the classifier type, the pdfs, but not on N. Denote by $E_i^N(C_j)$ the generalisation error of classifier C trained on N data points from source S_j with respect to the probability distributions in source S_i, $i, j = 1, 2$. Assuming that the training window for S_1 is sufficiently large, C_1 is trained to reach its asymptotic error $E_1(C_1)$. At the onset of the change at t_0, the error of the classifier jumps to $E_2(C_1)$. It is expected that the change renders C_1 inadequate for the data from S_2, hence $E_2(C_1) > E_1(C_1)$ (Figure 1). If a new classifier is trained starting with the first observation after t_0, and the training set is augmented after each observation, the error of this classifier would be $E_2(C_2) + \frac{1}{N} f(C_2)$. To find the optimal switch point from C_1 to C_2, we solve for N the following equation

$$E_2(C_1) = E_2(C_2) + \frac{1}{N} f(C_2). \tag{2}$$

The switch point is when the size of the training window of data coming from S_2 reaches

$$N^* = \frac{f(C_2)}{E_2(C_1) - E_2(C_2)}. \tag{3}$$

Variants of $f(C)$ are tabulated for various classifiers and pdfs in references [16, 17]. The error values $E_i(C_i)$ can be derived for specific distributions and classifiers [18].

Note that N^* is not merely a window in the standard sense; we can rather view it as the "switch point" from the old to the new classifier.

2.2. Linear Discriminant Classifier (LDC) for two Gaussian classes

Let C be the linear discriminant classifier (LDC) [19] applied to two equiprobable n-dimensional Gaussian classes with identical covariance matrices Σ. Let $\delta^{(j)}$ be the Mahalanobis distance between the class means for source S_j, $j = 1, 2$. The error of LDC for this case is the Bayes error, and is calculated as [18]

$$E_2(C_2) = \Phi\left(-\frac{\delta^{(2)}}{2}\right), \tag{4}$$

where Φ is the cumulative distribution function of the standard normal distribution. The function relating the sample size and the classification error for LDC is [16]

$$f(C) = \frac{1}{2\sqrt{2\pi}\delta}\left[\left(1 + \frac{\delta^2}{4}\right)n - 1\right]\exp\left(-\frac{\delta^2}{8}\right). \tag{5}$$

For $f(C_2)$ we use $\delta = \delta^{(2)}$. The only unknown term in (3) is $E_2(C_1)$ which depends on the type and magnitude of the change. Assuming that only the class means change while the common covariance matrix remains the same from S_1 to S_2, we derive in the Appendix the following expression for $E_2(C_1)$

$$E_2(C_1) = \frac{1}{2}\left\{\Phi\left(-\frac{\mathbf{w}^T\Delta_1}{\delta^{(1)}} - \frac{\delta^{(1)}}{2}\right) + \Phi\left(\frac{\mathbf{w}^T\Delta_2}{\delta^{(1)}} - \frac{\delta^{(1)}}{2}\right)\right\}, \tag{6}$$

where Δ_i is the difference between the means for class ω_i after and before the changes. Let $\mu_i^{(j)}$ be the mean of class ω_i in source S_j, $i, j = 1, 2$, and Σ be the common covariance matrix for the classes in both sources. Then $\Delta_1 = \mu_1^{(2)} - \mu_1^{(1)}$ and $\Delta_2 = \mu_2^{(2)} - \mu_2^{(1)}$. The vector with coefficients $\mathbf{w}$ comes from C_1 trained

on S_1, and is given by $\mathbf{w}^T = (\mu_1^{(1)} - \mu_2^{(1)})^T \Sigma^{-1}$. With all terms in place, the optimal switch point for this special case is

$$N^* = \frac{\frac{1}{2\sqrt{2\pi}\delta^{(2)}} \left[\left(1 + \frac{(\delta^{(2)})^2}{4}\right) n - 1 \right] \exp\left(-\frac{(\delta^{(2)})^2}{8}\right)}{\frac{1}{2}\left\{\Phi\left(-\frac{\mathbf{w}^T\Delta_1}{\delta^{(1)}} - \frac{\delta^{(1)}}{2}\right) + \Phi\left(\frac{\mathbf{w}^T\Delta_2}{\delta^{(1)}} - \frac{\delta^{(1)}}{2}\right)\right\} - \Phi\left(-\frac{\delta^{(2)}}{2}\right)}. \quad (7)$$

Consider as an example two Gaussian classes in $\Re^5$ with $\mu_1^{(1)} = [0.5, 0, 0, 0, 0]^T$, $\mu_2^{(1)} = [-0.5, 0, 0, 0, 0]^T$ for source S_1, and $\mu_1^{(2)} = [1.0, 0.3, 0, 0, 0]^T$, $\mu_2^{(2)} = [0.5, 0, 0, 0, 0]^T$ for source S_2. In both sources the covariance matrix Σ was obtained from an identity matrix of size 5 by setting $\sigma_{1,2} = \sigma_{2,1} = 0.3$. An illustration is shown in Figure 2.

Using (7), we get an optimal training window size $N^* = 42$. This means that after the substitution of S_1 with S_2, the old classifier is expected to be more accurate than the new classifier for the first 42 instances from S_2.

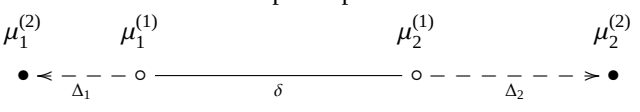
2.3. Applicability of the results

Deriving the theoretical switch point does not automatically offer an algorithm for classification in the presence of concept drift. The obtained result can be used further for constructing plug-and-play algorithms. Such an algorithm requires a multitude of choices to be made, e.g., classifier model, change detection method, pdf approximations, error approximations, $f(C)$ approximation, etc. Then we are faced with the 'credit apportionment' problem; the success or failure of such an algorithm can be attributed to any of the choices. The collection of choices might haphazardly smother or highlight the benefit from the optimal window size. Our experiments are designed to showcase the theoretical window size in comparison with other window sizes. Since we are not proposing an adaptive classifier algorithm we do not run comparisons with other adaptive classifier models. The experiments in this study are only meant as an illustration and not proof of concept.

3. Analysis and simulations

3.1. Optimal N^* in relation to the magnitude and the direction of the drift

For the 1-dimensional case, $x \in \Re$, we can investigate the relationship between the direction and the magnitude of the drift in the class means and the optimal switching point N^* (7). Without loss of generality, assume that $\mu_1^{(1)} < \mu_2^{(1)}$. Let $\delta = \mu_2^{(1)} - \mu_1^{(1)}$ be the distance between the two means in the distributions of source S_1. In the new distributions coming from source S_2, $\mu_1^{(2)} = \mu_1^{(1)} + \Delta_1$ and $\mu_2^{(2)} = \mu_2^{(1)} + \Delta_2$. The common variance in both cases was 1. The set-up is depicted below



We varied Δ_1 and Δ_2 independently in the interval $[-\delta, \delta]$. Figure 3 gives a colour plot and a surface plot of $\log(N^*)$ as a function of Δ_1 and Δ_2 for $\delta = 4$.

The ridge along the diagonal section from $(\delta/2, -\delta/2)$ to $(-\delta, \delta)$ reflects the case where the two centres migrate symmetrically about the midpoint, i.e., when $\Delta_1 = -\Delta_2$. The denominator in (7) collapses to 0 because the old classifier is optimal for the new distributions ($E_2(C_1) = E_1(C_1)$), and N^* approaches infinity. For visualisation purposes we cut off the peak by resetting all denominator values smaller than 10^{-5} to 10^{-5}. This is the cause of the flat top at the upper left corner $(-4, 4)$. The part of the diagonal starting from $(0, 0)$ and ending at $(-4, 4)$ corresponds to the case where the centres move apart, while the part from $(0, 0)$ down to $(-2, 2)$ corresponds to the two centres moving symmetrically towards one another. In both cases, the old classifier is optimal (regardless of the error), and $N^* \to \infty$. At $(-2, 2)$, the centres fall on top of one another, and no classifier can be better than random chance. Hence the ridge across from $(-\delta, 0)$ to $(0, \delta)$. For all pairs (Δ_1, Δ_2) on this line, $\Delta_2 - \Delta_1 = \delta$, which means that $\mu_1^{(2)} = \mu_2^{(2)}$ (the concept change merges the two classes into one). For this case the old classifier will be as useless as any classifier, $E_2(C_1) = E_1(C_1) = 0.5$, therefore $N^* \to \infty$.

The points (Δ_1, Δ_2) on the right diagonal correspond to the case $\Delta_1 = \Delta_2$, i.e. the classes are shifted together to the left or to the right. The old classifier in this case becomes progressively more inadequate with the size of the offset.

The darker subregions of A and B correspond to very small N^* (negative $\log(N^*)$) reflecting the case where even a very coarse and undertrained classifier for source S_2 is better than the old classifier C_1 trained on source S_1. Small values of N^* are not necessarily related with large error. Consider the pair $(\Delta_1 = -\delta, \Delta_2 = 0)$. Class 1 moves to the left by δ while class 2 stays put. The separability increases substantially, which leads to a smaller $f(C_2)$ term. On the other hand, due to the increased separability $E_2(C_2)$ may be much smaller than $E_2(C_1)$. Therefore N^* will be small. This situation is most prominently expressed in region C where the two means "swap places" so that, while $\mu_1^{(1)} < \mu_2^{(1)}$, after the concept change $\mu_2^{(2)} < \mu_1^{(2)}$. In this case C_1 will give the opposite labels in S_2 and will be worse than chance. The old classifier should be immediately replaced with C_2, even though C_2 might act as the largest probability classifier before proper training.

3.2. Numerical experiments

In order to examine equation (7) we generated four data sets commonly used in concept drift research [20, 24]. We also used two real data sets from the UCI repository [25] adding to them artificial drift.

Gaussian data. Two hundred observations were generated as the sequential data, 100 before the concept change, and 100 after. The number of features was chosen to be $n = 4$. The means in source S_1 were $\mu_1^{(1)} = [-1, 0, 0, 0]^T$ and $\mu_2^{(1)} = [1, 0, 0, 0]^T$. The concept drift was set at $\Delta_1 = [0.5, 0.2, 0, 0, 0]^T$ and $\Delta_2 = [0.7, 0, 0, 0, 0]^T$. The prior probabilities in both sources were 0.5/0.5. The common covariance matrix, Σ, was constructed from an identity matrix of size 4 by setting $\sigma_{1,2} = \sigma_{2,1} = 0.3$.

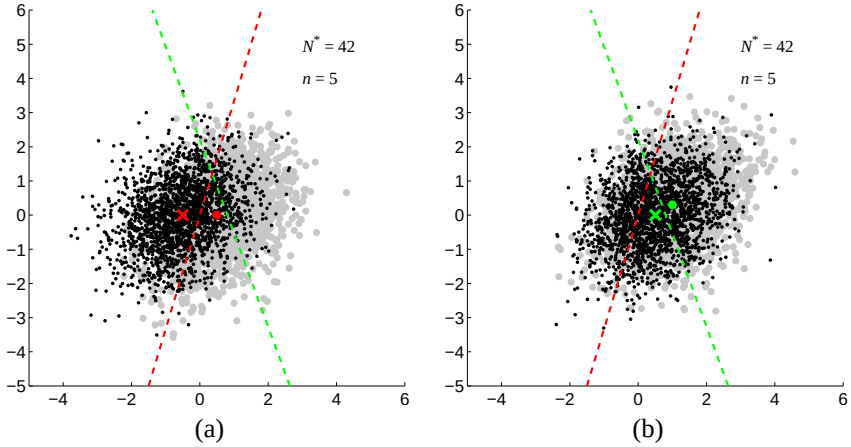


Figure 2: Data from sources S_1 (a) and S_2 (b), plotted in the first two dimensions. The centres of the two classes are marked. The optimal discriminant lines for the two sources are shown in both plots.

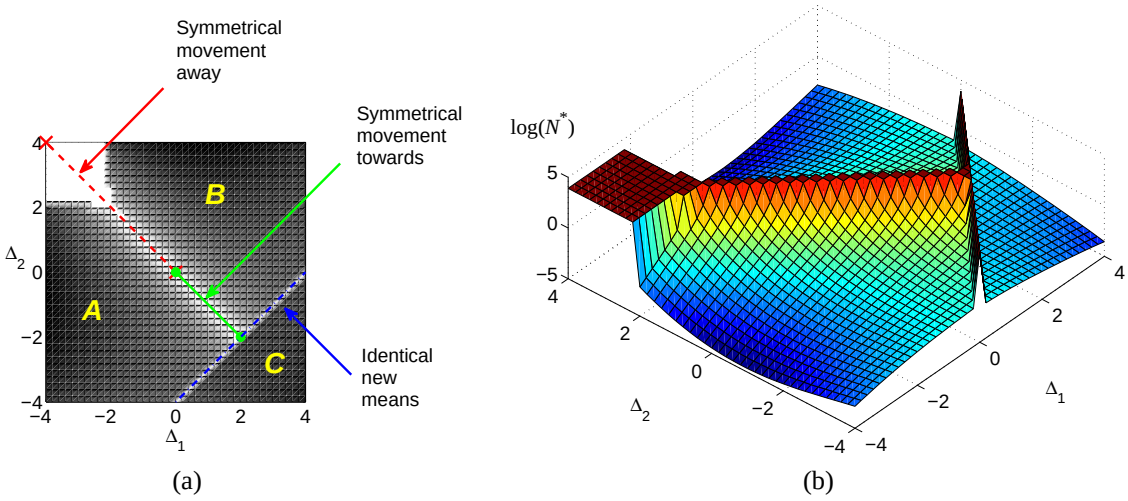


Figure 3: Plots of $\log(N^*)$ as a function of the offsets of the means, Δ_1 and Δ_2. (a) Colour plot. Dark colour corresponds to smaller N^*; (b) Surface plot (with restricted height) for $\log(N^*)$.

STAGGER data [5]. Each data point was described by three features, each with three possible categories: size $\in$ {small, medium, large}, colour $\in$ {red, green, blue} and shape $\in$ {square, circular, triangular}. The numerical representation of a data point consisted of 9 bits, 3 for each feature. For example, a large, red, square object was encoded as the vector $[0, 0, 1, 1, 0, 0, 1, 0, 0]^T$ and treated as a point in $\Re^9$. Three classification tasks were to be learned in a course of 120 points. From point 1 to point 40, the classes to be distinguished were [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.

Moving-hyperplane data [21, 22, 23]. The initial data and the 4 subsequent concept changes are shown in Figure4. The separation line started at 0° and was rotated to 45°, 90°, 135°, 180°. To form sequential data, 50 i.i.d points were drawn from each source before the next rotation.

SEA data [24]. Each data point was described by three features, $\mathbf{x} = [x_1, x_2, x_3]^T$, where $\mathbf{x}$ were uniformly randomly generated from $[0, 10]^3$. Only the first two features were relevant. An instance belonged to class 1 if $x_1 + x_2 \leq \Theta$ and to class 2 otherwise, where Θ was a threshold value, different for each concept. There were four concepts $\Theta = 8; 9; 7; 8.5$. We generated 200 instances for each concept. No label noise was added so the two classes were perfectly separable by a hyperplane in the feature space $[0, 10]^3$, parallel to the x_3 axis.

Vote data [25]. This data set represents the 1984 United States Congressional Voting Records (435 data points of which 267 democrats and 168 republicans with 16 binary features).

Ionosphere data [25]. This data set represents a two-class problem where a radar returns 'good' and 'bad' signals. The

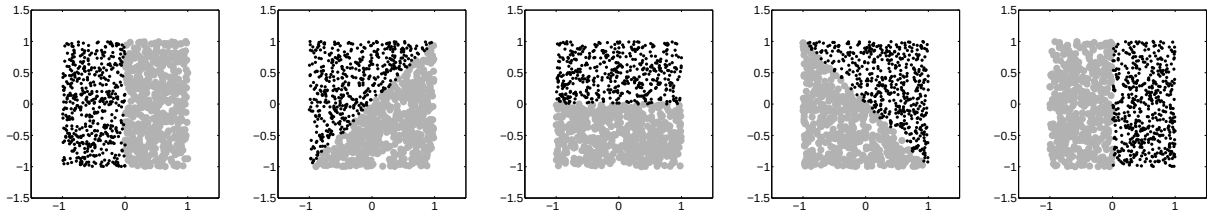


Figure 4: The five stages of the moving hyperplane data (0°,45°,90°,135°,180°)

data consists of 351 instances (136 + 215), each having 34 features.

To form different i.i.d data sequences with concept drift, the real data was first randomly permuted. Then feature pairs were swapped at every 50th instance (two features for the Vote data and four features for the Ionosphere data).

We compared the following scenarios

- A. *No update.* Running C_1 all the way through the sequential data.
- B. *Update without forgetting.* Online updating of C_1 without forgetting past data.
- C. *Complete forgetting.* Switching classifiers at t_0 by dismissing C_1 and starting the training of C_2.
- D. *Partial forgetting.* Switching classifiers at N^*, where the means in both distributions are estimated from the sequential data.

In order to bypass the issue of change detection, we assumed that the change points are known for scenarios A, C and D.

We did not compare with the plug-and-play algorithms because this would mix the effect of change detection with the classifier switch point.

For each data point submitted as a part of the sequence, an independent testing set of 100 objects was generated and labelled according to the current class description. The classifier was retrained and tested after each observation. The four classifiers were trained and tested on identical data in order to enable pairwise comparison. The nearest mean classifier (NMC) was used. The evaluation of (7) required only the class means to be estimated from the old and the new distributions. We should mention the following three issues

- *Multiple changes.* We note that in scenario A, the classifier was trained on the first bout of sequential data, up to the first change. The same classifier was applied in any subsequent changes. In the calculation of $N*$, however, we used a different C_1 after each change. For example, with the moving hyperplane data, classifier C_2 trained after rotation to 45° becomes the "old" classifier with respect to the next rotation to 90°. Following that, the new classifier was taken as C_1 with respect to the next rotation to 135°, and so on.
- *Single class.* If only points from one of the classes were available, then NMC would label all the data to that class. As the points came randomly, the probability of error of C_2 at the first few observations was likely to reach level $(1 - \max_i P(\omega_i))$, i.e., 0.5. In that case $N^* = \infty$, because there was no difference as to which classifier to use. That means no switch was recommended. In this case $\delta = 0$ and due to that $f(C) = \infty$, while $E_2(C_2) = E_2(C_1) = (1 - \max_i P(\omega_i))$.
- *Premature switch.* Suppose that we start counting the observations from t_0 onwards. At observation t_i we re-evaluated the window, denoted N_i^*. If $i < N_i^*$, then the old classifier was still useful, otherwise we switched to C_2. Since we did not switch back and since there was noise in the estimate of N^*, it was likely that the switch came earlier than necessary.

One hundred runs were carried out for the artificial data. Figure 5 plots the testing error rate for the three data sets. Table 1 shows the overall error and its 95% confidence interval. The overall error is the testing error averaged across the whole online run. Since all 4 classifiers were trained and tested on identical data, paired *t*-tests were carried out between scenario D on the one hand, and A, B, and C, on the other hand. The results are indicated in the table.[1] Symbol '$\circ$' means that the scenario was found to be significantly worse than D, and '$\bullet$' means that the scenario was found to be significantly better than D.

For the real data, three hundred runs were carried out, using different random permutations of the data. The running classification error was estimated, i.e., the classifier was trained on data instances $1, 2, \ldots, t$ and tested on instance $t + 1$. Table 1 shows the overall error and its 95% confidence interval.

The window approach D was significantly better than the other three approaches except for the STAGGER data where immediate switching to C_2 was significantly better than the window approach. The reasons for this exception are several: (1) there was no peak of the error $E_2^N(C_2)$ above $E_1(C_2)$ for small N; (2) the concepts were very different and also had different priors; (3) there was no noise in the data. Therefore the new classifier was more useful right away. The statistical differences were estimated only as an illustration. They depend on the chosen time length of the sequential data. Our approach is

[1] Note that the *t*-test results cannot be recovered from the individual confidence intervals.

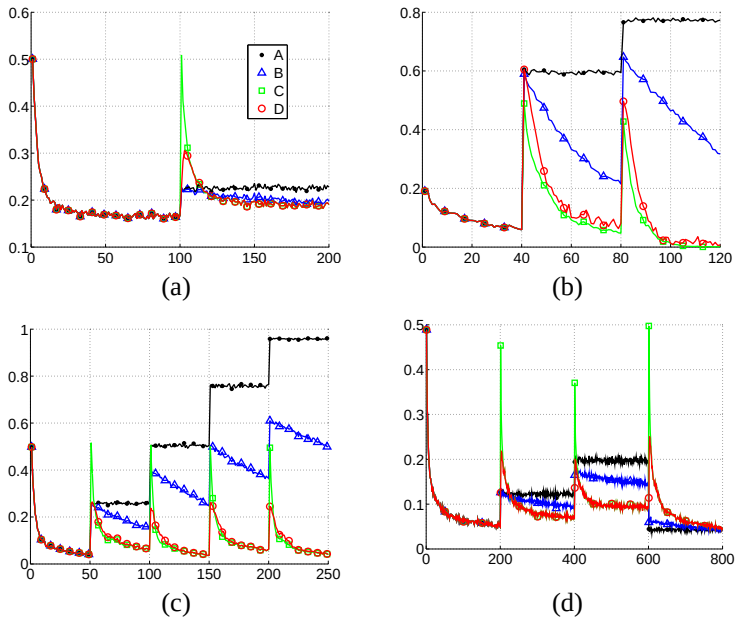


Figure 5: The 4 scenarios with the Gaussian data (a); STAGGER data (b); the moving hyperplane data (c) and SEA data (d) where A = No update; B = Update without forgetting; C = Complete forgetting; D = Partial forgetting with optimal window size.

Table 1: Total error for the 4 scenarios (in %) with 95% confidence intervals

Data	A = No update	B = No forgetting	C = Complete forgetting	D = Partial forgetting
Gaussian	20.53 (±0.17) ∘	19.52 (±0.14)	19.74 (±0.15) ∘	19.47 (±0.15)
Stagger	48.92 (±0.49) ∘	31.00 (±0.76) ∘	9.99 (±0.41) •	12.57 (±0.54)
Moving hyperplane	51.36 (±0.54) ∘	31.78 (±0.48) ∘	9.78 (±0.28) ∘	9.62 (±0.28)
SEA	11.12 (±0.26) ∘	9.85 (±0.14) ∘	9.14 (±0.19) ∘	8.90 (±0.19)
Votes	15.01 (±0.13) ∘	12.19 (±0.07) ∘	12.73 (±0.08) ∘	11.19 (±0.08)
Ionosphere	28.60 (±0.52) ∘	26.89 (±0.24) ∘	27.92 (±0.21) ∘	26.54 (±0.22)

Note: Symbol '∘' means that the scenario was found to be significantly worse than D, and '•' means that the scenario was found to be significantly better than D

meant to act as an 'oracle' identifying the most accurate of the two original classifiers at each time and switch as soon as N^* is reached. Thus there were intervals where the running error of D coincides with that for either C_1 or C_2. In the part before the change, the NMC classifiers for the 4 scenarios used identical training and testing data and had the same running error. The magnitude of the difference depends on the number of drifts w.r.t. the length of the series observed. If the sequential data was let to run long enough, the significance of the error peak being shaved off would be smoothed over, and scenarios C and D would have been indistinguishable.

Note that only the Gaussian data satisfied the assumptions of the model; the other five cases illustrated that the theory might have been useful even when the assumptions did not hold.

Assuming that sufficient data is available from the old distribution, the estimates of $\mu_1^{(1)}$ and $\mu_2^{(1)}$ would be stable. The inaccuracy of estimating $\mu_1^{(2)}$ and $\mu_2^{(2)}$ from a small sample of sequential data coming after the change will induce instability of the estimate of N^* causing premature switch. With the Gaussian data, where the distributions are guessed correctly, this will lead to curve D (partial forgetting) being closer to curve C (complete forgetting) rather than following curve A (no update), or even better curve B (update but no forgetting). Bias in N^* comes from wrongly guessed distributions as for the STAGGER data and moving hyperplane data.

3.3. *Practical issues*

To calculate the optimal switch point N^* from data, we need to know the time of the change, t_0, the errors $E_2(C_1)$ and $E_2(C_2)$, and the term $f(C_2)$ that accounts for the error component coming from inaccurate estimates of the parameters of the classifier.

Change detection methods can be employed to determine t_0. If the change time t_0 is detected at a later time $t_d < t_0 + N^*$, nothing is lost because the optimal classifier has been running all the way to the detection. Large changes can be detected quicker than small changes. For small changes, however, N^* is larger, thus allowing for a larger detection time.

Application of the switch point calculation is not straightforward if we drop the assumption of Gaussian densities. In that case the estimation of the three terms in (3) requires suitable data sets to train and test the chosen classifier. With estimation techniques in place, an algorithm for resizing of a moving training window can be developed.

4. Conclusion

Here we derive an optimal window size for online classification of sequential data after a sudden concept change. The window size N^* can be computed from the classification errors after the change, and a term that gauges the contribution of the inaccuracy of parameter estimates to the classification error. A special case of equiprobable Gaussian classes is detailed for the linear discriminant classifier (LDC). The general formula for calculating N^* is expanded so that the window size can be calculated using only the estimates of the means in S_1 and S_2, and the common covariance matrix. The common covariance matrix is the same before and after the change, so it can be estimated from S_1, where we assume practically unlimited data. The analysis reveals that N^* depends strongly on the change, and for large changes the old classifier should be dismissed quite early on (the valleys in Figure 3 (b)). On the other hand, having a window where C_1 is better than C_2 gives a lee way for the change detection. If the change is detected within N^* observations after t_0, the two classifiers can be swapped at the optimal moment. The numerical simulations demonstrated that the window approach does cut the error peak when it exists.

There could be various extensions of this work. Derivation of an optimal window size for other classifier models can be approached in a similar way as we showed for LDC. Different types of concept drift can be explored within the proposed framework with a view to derive theoretically an optimal variable window size. The practical impact of such theories depends upon how robust the proposed methods are with respect to violation of the underlying assumptions, hence empirical verification will be needed.

Appendix A. Theoretical generalisation error of LDC after the concept change

Following Raudys' derivation of the classification error [18], here we derive the expression for the error of the linear discriminant classifier C trained on source S_1 and applied to the probability distributions in source S_2 (notation $E_2(C_1)$). In source S_1, the two classes are distributed as $\mathbf{x} \sim \mathcal{N}(\mu_i^{(1)}, \Sigma)$, $\mathbf{x} \in \Re^n$, $i = 1, 2$. Superscript in parentheses will be used to indicate the source. We assume that the prior probabilities are $P(\omega_1) = P(\omega_2) = 1/2$. The discriminant function of LDC is

$$\begin{aligned} g(\mathbf{x}) &= \mathbf{w}^T\mathbf{x} + w_0 && \text{(A.1)}\\ &= \left(\mu_1^{(1)} - \mu_2^{(1)}\right)^T \Sigma^{-1}\mathbf{x} && \text{(A.2)}\\ &+ \frac{1}{2}\left[\left(\mu_2^{(1)}\right)^T \Sigma^{-1}\mu_2^{(1)} - \left(\mu_1^{(1)}\right)^T \Sigma^{-1}\mu_1^{(1)}\right]. && \text{(A.3)} \end{aligned}$$

If $g(\mathbf{x}) \geq 0$ then $\mathbf{x}$ is assigned to class ω_1, otherwise, to class ω_2. The probability that C will make an error is

$$E(C) = \frac{1}{2}\Big(Pr(g(\mathbf{x}) < 0|\omega_1) + Pr(g(\mathbf{x}) \geq 0|\omega_2)\Big). \qquad \text{(A.4)}$$

Conditioned by either ω_1 or ω_2, the discriminant function $g(\mathbf{x})$ becomes a linear combination of normally distributed variables, and is normally distributed itself. The error can be calculated using the cumulative distribution function Φ of the standard normal distribution. For this, we need the expectation and the standard deviation of $g(\mathbf{x})$, conditioned by the respective class label. For source S_2, class ω_1 has mean $\mu_1^{(2)} = \mu_1^{(1)} + \Delta_1$. Therefore the expectation of $g(\mathbf{x})$ for the "new" ω_1 is

$$\mathcal{E}_1^{(2)}[g(\mathbf{x})] = \mathbf{w}^T\mathcal{E}_1^{(2)}[\mathbf{x}] + w_0 = \mathbf{w}^T(\mu_1^{(1)} + \Delta_1) + w_0. \qquad \text{(A.5)}$$

Expanding $\mathbf{w}$ and w_0 leads to

$$\mathcal{E}_1^{(2)}[g(\mathbf{x})] = \frac{1}{2}\left(\delta^{(1)}\right)^2 + \mathbf{w}^T\Delta_1, \qquad \text{(A.6)}$$

where $\delta^{(1)} = \left(\mu_1^{(1)} - \mu_2^{(1)}\right)^T \Sigma^{-1}\left(\mu_1^{(1)} - \mu_2^{(1)}\right)$ is the Mahalanobis distance between the class means in source S_1. In the same way we arrive at the expectation of $g(\mathbf{x})$ conditioned by the new class ω_2

$$\mathcal{E}_2^{(2)}[g(\mathbf{x})] = -\frac{1}{2}\left(\delta^{(1)}\right)^2 + \mathbf{w}^T\Delta_2. \qquad \text{(A.7)}$$

The variance of $g(\mathbf{x})$ is the same for both classes, and is given by

$$\begin{aligned} \mathcal{V}[g(\mathbf{x})] &= \mathcal{V}[\mathbf{w}^T\mathbf{x} + w_0] = \mathbf{w}^T\Sigma\mathbf{w} && \text{(A.8)}\\ &= \left(\mu_1^{(1)} - \mu_2^{(1)}\right)^T \Sigma^{-1}\,\Sigma\,\Sigma^{-1}\left(\mu_1^{(1)} - \mu_2^{(1)}\right) && \text{(A.9)}\\ &= \left(\delta^{(1)}\right)^2. && \text{(A.10)} \end{aligned}$$

Then

$$Pr(g(\mathbf{x}) < 0|\omega_1) = \Phi\left(-\frac{\mathbf{w}^T\Delta_1}{\delta^{(1)}} - \frac{\delta^{(1)}}{2}\right) \qquad \text{(A.11)}$$

and

$$\begin{aligned} Pr(g(\mathbf{x}) \geq 0|\omega_2) &= 1 - \Phi\left(-\frac{\mathbf{w}^T\Delta_2}{\delta^{(1)}} + \frac{\delta^{(1)}}{2}\right) && \text{(A.12)}\\ &= \Phi\left(\frac{\mathbf{w}^T\Delta_2}{\delta^{(1)}} - \frac{\delta^{(1)}}{2}\right) && \text{(A.13)} \end{aligned}$$

By substituting the conditional probabilities (A.11) and (A.13) back in (A.4), we arrive at the error on S_2 of LDC trained on S_1

$$\begin{aligned} &E_2(C_1)\\ &\quad = \tfrac{1}{2}\left\{\Phi\left(-\tfrac{\mathbf{w}^T\Delta_1}{\delta^{(1)}} - \tfrac{\delta^{(1)}}{2}\right) + \Phi\left(\tfrac{\mathbf{w}^T\Delta_2}{\delta^{(1)}} - \tfrac{\delta^{(1)}}{2}\right)\right\}. && \text{(A.14)} \end{aligned}$$

References

[1] D.J. Hand, "Classifier technology and the illusion of progress (with discussion)", in *Statistical Science*, vol. 21, pp. 1–34, 2006.

[2] P. L. Bartlett, S. Ben-David, S.R. Kulkarni, "Learning changing concepts by exploiting the structure of change", in *Machine Learning*, vol. 41, pp. 153–174, 2000.

[3] P. Domingos, G. Hulten, "Mining High-Speed Data Streams", in *Proceedings of the Sixth International Conference on Knowledge Discovery and Data Mining*, pp. 71–80, 2000.
[4] J. Case, S. Jain, S. Kaufmann, A. Sharma, F. Stephan, "Predictive learning models for concept drift", in *Theoretical Computer Science*, vol. 261, no. 2, pp. 323–349, Elsevier Science Publishers Ltd., 2001.
[5] G. Widmer, M. Kubat, "Learning in the presence of concept drift and hidden contexts," in *Machine Learning*, vol. 23, pp. 69–101, 1996.
[6] R. Klinkenberg, I. Renz, "Adaptive information filtering: Learning in the presence of concept drifts", in *AAAI-98/ICML-98 workshop Learning for Text Categorization*, pp. 33–40, AAAI Press, 1998.
[7] R. Klinkenberg, T. Joachims, "Detecting concept drift with support vector machines", in *Proceedings of the Seventeenth International Conference on Machine Learning (ICML)*, pp. 487–494, Morgan Kaufmann, 2000.
[8] M. Lazarescu, S. Venkatesh, "Using selective memory to track concept drift effectively", in *Intelligent Systems and Control*, vol. 388, ACTA Press, 2003.
[9] M. Scholz, R. Klinkenberg, "An ensemble classifier for drifting concepts", in *Proceedings of the 2nd Workshop on Knowledge Discovery from Data Streams*, pp. 53–64, Porto, Portugal, 2005.
[10] A. Bifet, R. Gavaldà, "Learning from time-changing data with adaptive windowing", in *Proceedings of the Seventh SIAM International Conference on Data Mining*, pp. 443–448, Minneapolis, Minnesota, USA, 2007.
[11] I. Koychev, R. Lothian, "Tracking drifting concepts by time window optimisation", in *Proceedings the 25th SGAI International Conference on Innovative Techniques and Applications of Artificial Intelligence, AI-2005*, pp. 46–59, Springer Verlag, 2005.
[12] J. Gama, P. Medas, G. Castillo, P. Rodriguez, "Learning with drift detection", in *Advances in Artificial Intelligence (SBIA), the 17th Brazilian Symposium on Artificial Intelligence*, ser. LNCS, vol. 3171, pp. 286–295, Springer Verlag, 2004.
[13] M. Baena-García, J. Campo-Ávila, R. Fidalgo, A. Bifet, R. Gavaldà, R. Morales-Bueno, "Early drift detection method", in *4th International Workshop on Knowledge Discovery from Data Streams*, pp. 77–86, 2006.
[14] C. G. Atkeson, A. W. Moore, S. Schaal, "Locally weighted learning", in *Artificial Intell. Review*, vol. 11, no. 1-5, pp. 11–73, 1997.
[15] M. Last, "Online classification of nonstationary data streams", in *Intelligent Data Analysis*, vol. 6, no. 2, pp. 129–147, 2002.
[16] K. Fukunaga, R. R. Hayes, "Effects of sample size in classifier design", in *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 11, no. 8, pp. 873–885, 1989.
[17] S. Raudys, A. K. Jain, "Small sample size effects in statistical pattern recognition: Recommendations for practitioners", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 3, pp. 252–264, 1991.
[18] S. Raudys, *Statistical and neural classifiers: an integrated approach to design*, Springer-Verlag, 2001.
[19] R. O. Duda, P. E. Hart, D. G. Stork, *Pattern Classification*, 2nd ed, John Wiley & Sons, 2001.
[20] A. Narasimhamurthy, L. I. Kuncheva, "A framework for generating data to simulate changing environments", in *Proc. IASTED, Artificial Intelligence and Applications*, pp. 384–389, Innsbruck, Austria, 2007.
[21] J. Z. Kolter, M. A. Maloof, "Dynamic weighted majority: A new ensemble method for tracking concept drift", in *Proc 3rd International IEEE Conference on Data Mining*, pp. 123–130, IEEE Press, 2003.
[22] G. Hulten, L. Spencer, P. Domingos, "Mining time-changing data streams", in *In Proc. 7th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 97–106, ACM Press, 2001.
[23] F. Ferrer-Troyano, J. S. Aguilar-Ruiz, J. C. Riquelme, "Incremental rule learning based on example nearness from numerical data streams", in *2005 ACM Symposium on Applied Computing, SAC 05*, pp. 568–572, ACM Press, 2005.
[24] W. N. Street, Y. Kim, "A streaming ensemble algorithm (SEA) for large-scale classification", in *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 377–382, ACM Press, 2005.
[25] A. Asuncion, D. J. Newman, "UCI Machine Learning Repository", [online] http://www.ics.uci.edu/ mlearn/MLRepository.html, *Irvine, CA: University of California, School of Information and Computer Science*, 2008.
[26] J. Bailey, E. Loekitoa, "Efficient incremental mining of contrast patterns in changing data", in *Information Processing Letters*, vol. 110, no. 3, pp. 88–92, 2010.