Temporally-adaptive linear classification for handling population drift in credit scoring

Niall M. Adams[1], Dimitris K. Tasoulis[1], Christoforos Anagnostopoulos[3],David J. Hand[1,2]

[1]Department of Mathematics
[2]Institute for Mathematical Sciences
Imperial College London

[3]Statistical Laboratory
University of Cambridge

August 2010

Contents

- Credit scoring
- Streaming data and classification
- Our approach: incorporate self-tuning forgetting factors
- Adaptation for credit scoring
- Experimental results

Research supported by

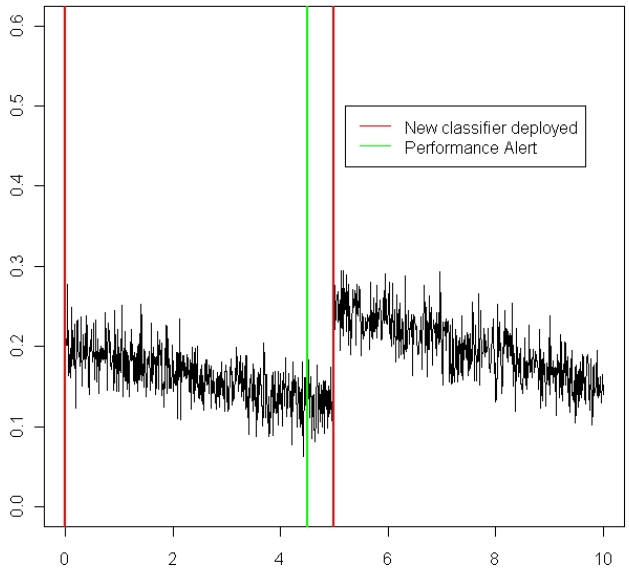
- the EPSRC/BAe funded ALADDIN project:
  www.aladdinproject.org

- Anonymous UK banks

Credit Application Scoring

- Credit application classification (CAC) is one important application of credit scoring
- There is a legislative requirement for certain products, like UPLs, to provide an explanation for rejecting applications
- this manifest as a preference for simple models: primarily logistic regression
- LDA often competitive in this context
- CAC usually subject to population drift: distribution of prediction data different to training data. Common problem in many applications.
- Objective here is to see how streaming technology might be adapted to handle drift without an explicit drift model.

- Many approaches proposed to handle population drift. Most not suitable for CAC.
- approach in consumer credit is to monitor for CAC performance degradation, and then rebuild: define new window of recent training data.



- This is a method related to a classification performance metric.
- We will deploy streaming methods, which respond to changes in model parameters, to reduce degradation between rebuilds (which are inevitable).

- CAC is often posed as a two class problem
- classes are good or bad risk, according to some definition, often similar to "bad if 3 or more months in arrears"
- data extracted from application form - personal details, background, finances - and other sources (e.g. CCJs).
- Variety of transformations explored at classifier building stage
- Some more complex timing data issues in CAC which we ignore

Streaming Data I

A data stream consists of a sequence of data items arriving at high frequency, generated by a process that is subject to unknown changes (generically called drift).

Many examples, often financial, include:

- credit card transaction data (6000/s for Barclaycard Europe)
- stock market tick data
- computer network traffic

The character of streaming data calls for algorithms that are

- efficient, one-pass - to handle frequency
- adaptive - to handle unknown change

Streaming Data II

A simple formulation of streaming data is a sequence of p-dimensional vectors, arriving at regular intervals

$$\ldots, x_{t-2}, x_{t-1}, x_t$$

where $x_i \in \mathbb{R}^p$.

Since we are concerned with K-class classification, need to accommodate a class label. Thus, at time t we can conceptualise the label-augmented streaming vector $y_t = (C_t, x_t)'$, where $C_t \in \{c_1, c_2, \ldots, c_k\}$.

However, in real applications C_t arrives at some time $s > t$, and the streaming classification problem is concerned with predicting C_t on the basis of x_t in an efficient and adaptive manner.

Streaming Data and Classification

Implicit assumption: single vector arrives at any time.

Assumption common in literature, which we use, is that the data stream is structured as

$$\ldots, (C_{t_3}, x_{t_2}), (C_{t_2}, x_{t_1}), (C_{t_1}, x_t),$$

That is, the class-label arrives at the next tick.

We will treat the streaming classification problem as: predict the class of x_t, and adaptively (and efficiently) update the model at time x_{t+1}, when C_t arrives.

This is naive, but the problem is challenging even formulated thus. Will return to label timing later.

Streaming Data and Classification

Can use the usual formulation for classification

$$P(C_t|x_t) = \frac{p(x_t|C_t)P(C_t)}{p(x_t)} \tag{1}$$

and construct either

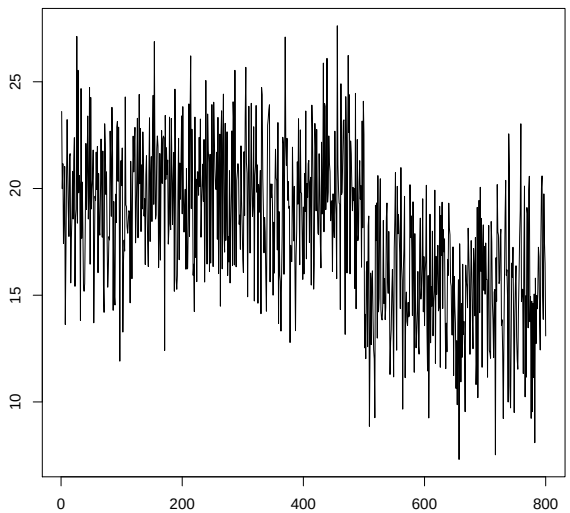
- Sampling paradigm classifiers, focusing on class conditional densities
- Diagnostic paradigm classifiers, directly seeking the posterior probabilities of class membership

Note the we will usually restrict attention to the $K = 2$ class problem.

Eq.1 where population drift can happen: the prior, $P(C_t)$, the class conditionals, $p(x_t|C_t)$, or both.

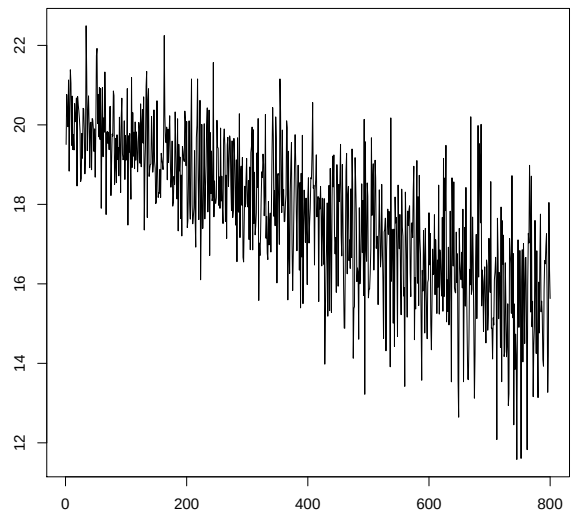
Notional drift types

1. Jump



(in mean)

2. Gradual change



(in mean and variance) Trend, seasonality etc.

Drift: CAC Examples

Consumer credit classification (conditionals)

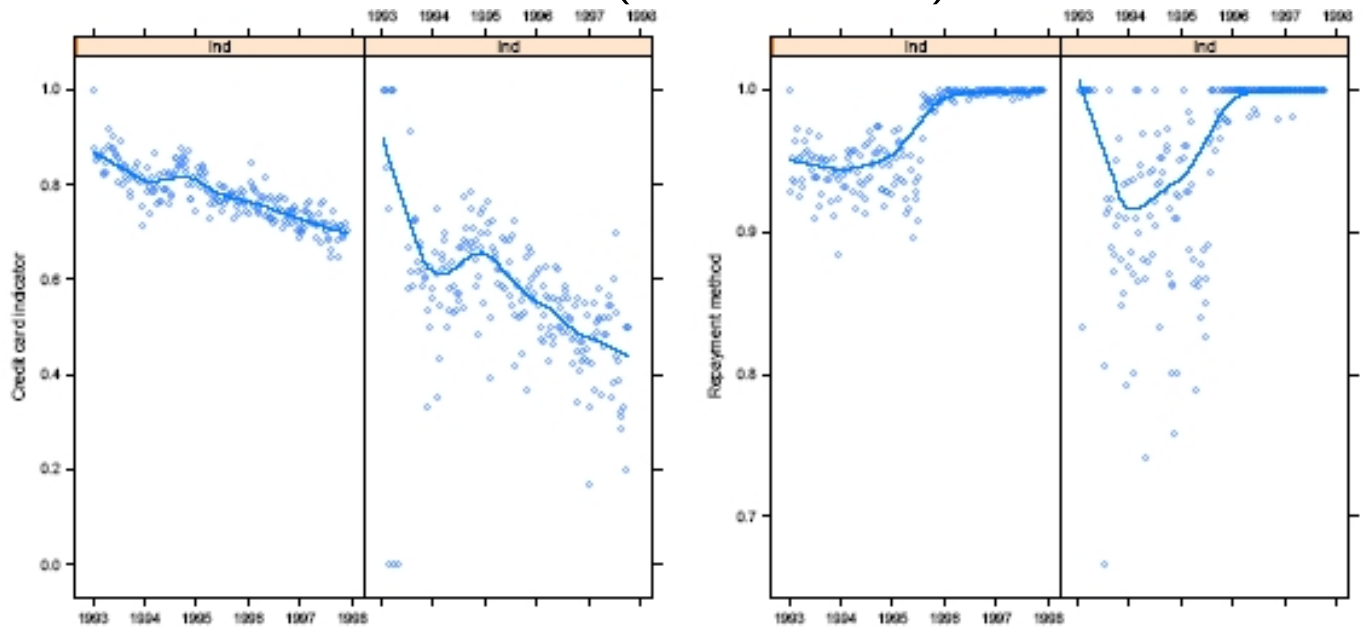


Figure 1: (a): Weekly averages for credit card indicator. (b): Weekly averages for repayment method indicator. Each plot includes good risk (left) and bad risk (right).

Consumer credit classification (prior)

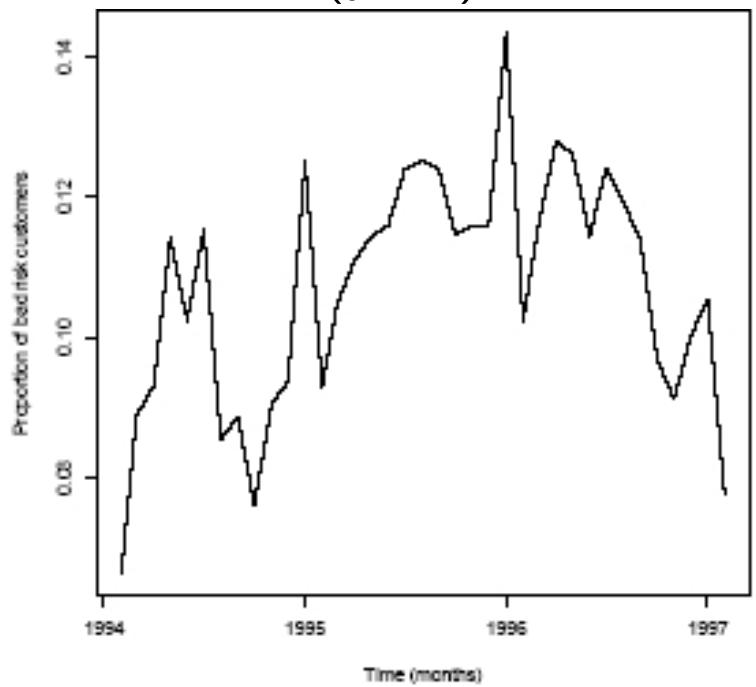


Figure 2: Proportion of bad risk accounts, by month, over the entire observation period.

Methods

A variety of approaches for streaming classification have been proposed, including

- Data selection approaches with standard classifiers. Most commonly, use of a fixed or variable size window of most recent data. But how to determine size in either case?
- Ensemble methods. One example is the adaptive weighting of ensemble members changing over time. This category also includes learning with expert feedback.

As noted above, CAC usually uses a static classifier with responsive rebuilds.

Forgetting-factor methods

We are interested in modifying standard classifiers to incorporate forgetting factors - parameters that control the contribution of old data to parameter estimation.

We adapt ideas from adaptive filter theory, to tune the forgetting factor automatically.

Simplest to illustrate with an example: consider computing the mean vector and covariance matrix of a sequence of n multivariate vectors. Standard recursion

$$m_t = m_{t-1} + x_t,\ \hat{\mu}_t = m_t/t,\ m_0 = 0$$
$$S_t = S_{t-1} + (x_t - \hat{\mu}_t)(x_t - \hat{\mu}_t)^T,\ \hat{\Sigma}_t = S_t/t,\ S_0 = \mathbf{0}$$

For vectors coming from a non-stationary system, simple averaging of this type is biased.

Knowing precise dynamics of the system gives chance to construct optimal filter. However, not possible with streaming data (though interesting links between adaptive and optimal filtering).

Incorporating a forgetting factor, $\lambda \in (0, 1]$, in the previous recursion

$$
\begin{aligned}
n_t &= \lambda n_{t-1} + 1, \; n_0 = 0 \\
m_t &= \lambda m_{t-1} + x_t, \; \hat{\mu}_t = m_t / n_t \\
S_t &= \lambda S_{t-1} + (x_t - \hat{\mu}_t)(x_t - \hat{\mu}_t)^T, \; \hat{\Sigma}_t = S_t / n_t
\end{aligned}
$$

λ down-weights old information more smoothly than a window. Denote these forgetting estimates as $\hat{\mu}_t^\lambda$, $\hat{\Sigma}_t^\lambda$, etc.

n_t is the effective sample size or memory. $\lambda = 1$ gives offline solutions, and $n_t = t$. For fixed $\lambda < 1$ memory size tends to $1/(1-\lambda)$ from below.

Setting λ

Two choices for λ, fixed value, or variable forgetting, λ_t. Fixed forgetting: set by trial and error, change detection, etc (cf. window).

Variable forgetting: ideas from adaptive filter theory suggest tuning λ_t according to a local stochastic gradient descent rule

$$\lambda_t = \lambda_{t-1} - \alpha \frac{\partial \xi_t^2}{\partial \lambda}, \quad \xi_t \text{: residual error at time } t, \ \alpha \text{ small} \tag{2}$$

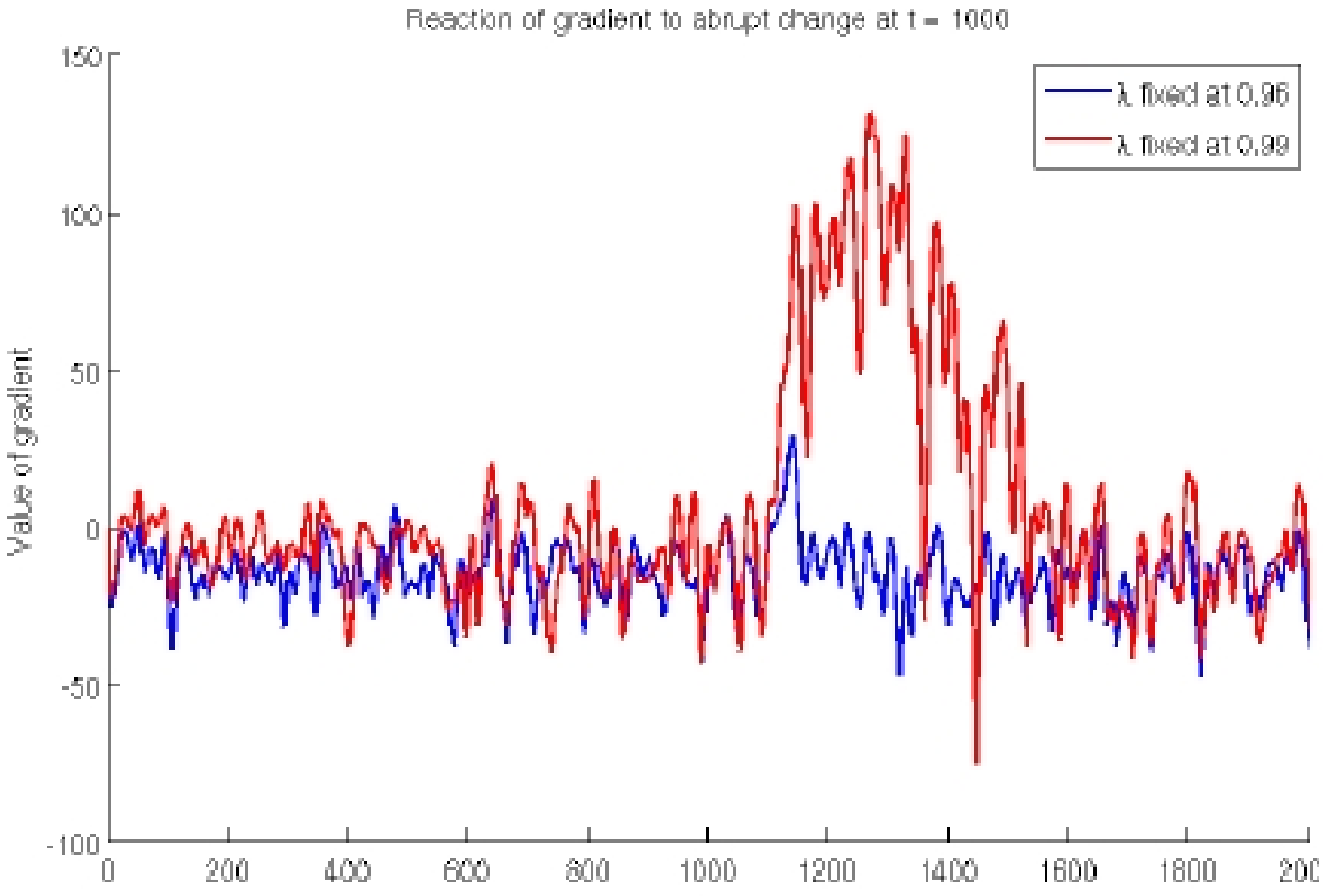
Efficient updating rules can implemented via results from numerical linear algebra ($O(p^2)$).

Performance very sensitive to α. Very careful implementation required, including bracket on λ_t and selection of learning rate α.

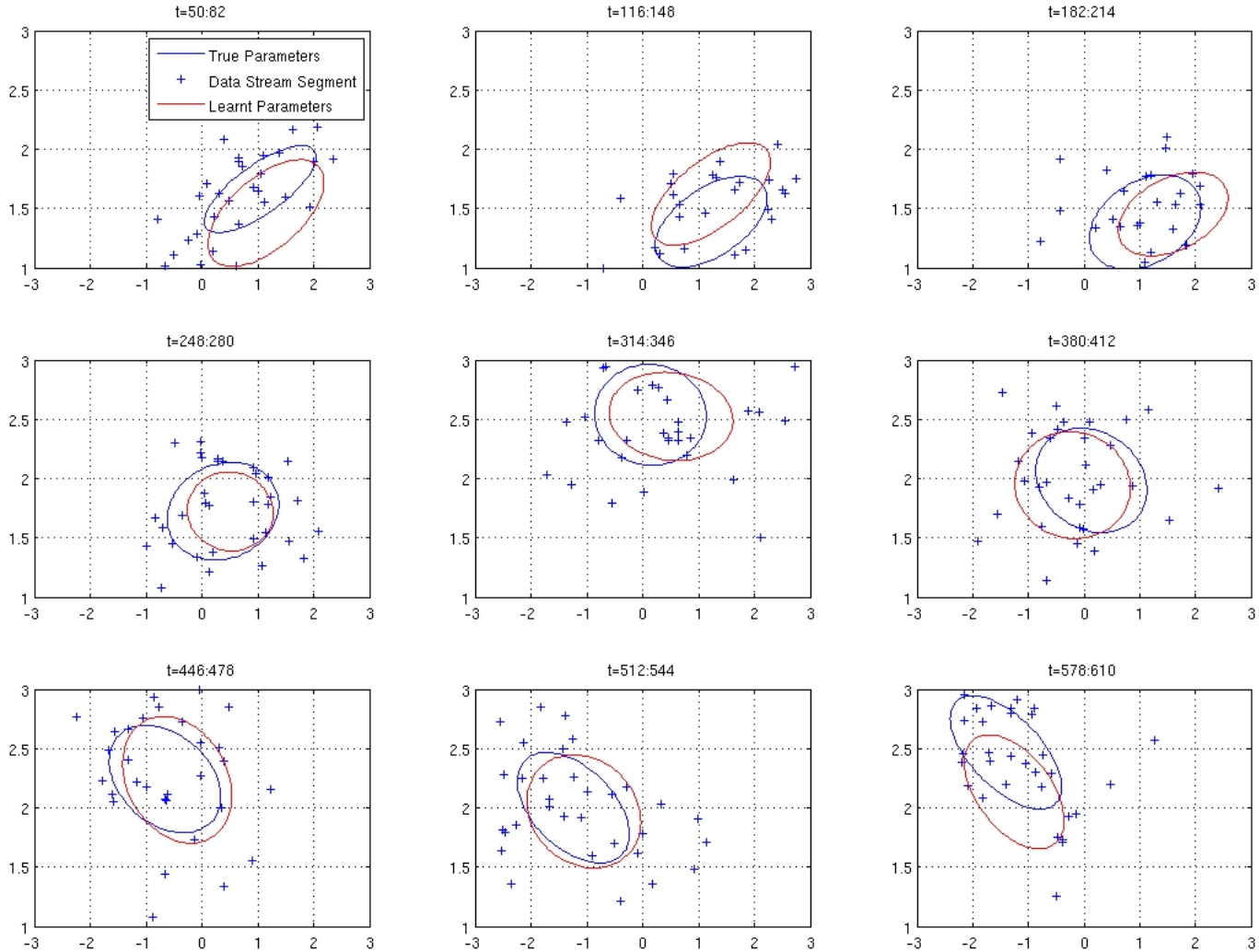
Framework provides an adaptive means for balancing old and new data. Note slight hack in terms of interpretation of λ_t.

Tracking illustrations

Does fixed forgetting respond to an abrupt change?
5D Gaussian, two choices of λ, change in σ_{23}: gradient



Tracking mean vector and covariance matrix in 2D.



Adaptive-Forgetting Classifiers

Our recent work involves incorporating these self-tuning forgetting factors in

- Parametric
  - Covariance-matrix based
  - Logistic regression
- non-parametric
  - Multi-layer perceptron

(sampling paradigm) (diagnostic paradigm)

We call these AF (adaptive-forgetting) classifiers.

Streaming Quadratic Discriminant Analysis

QDA can be motivated by reasoning about relationship of between and within group covariances, or assuming class conditional densities are Gaussian.

For static data, latter assumption yields discriminant function for jth class

$$g_j(x) = \log(P(C_j)) - \frac{1}{2}\log(|\Sigma_j|) - \frac{1}{2}(x-\mu_j)^T\Sigma_j^{-1}(x-\mu_i) \quad (3)$$

where μ_j and Σ_j are mean vector and covariance matrix, respectively, for class j.

Frequently, plug-in ML estimates for unknown parameters: μ_j, Σ_j, $P(C_j)$.

Idea here is to plug-in the AF estimates, $\hat{\mu}_t^{\lambda}$ etc.

Results in CA's thesis show that the AF framework above can be generalised, using likelihood arguments, to the whole exponential family. Thus, the priors, $P(C_t)$ can also be handled in a streaming manner.

The approach is then:

- Forgetting factor for prior (binomial/multinomial)
- Forgetting factor for each class

The class of x_t is predicted when it arrives. Immediately thereafter, the class-label arrives, and the true class parameters are updated.

This will be problematic for large K or very imbalanced classes: few updates complicates the interpretation of the update equation for λ_t (Eq. 2).

Streaming LDA

The discriminant function in Eq.3 reduces to a linear classifier under various constraints on the covariance matrices (or mean vectors).

We consider the case of a common covariance matrix: $\Sigma_1 = \Sigma_2 = \ldots = \Sigma_K = \Sigma$. Again, we will substitute streaming estimates μ_j^λ, Σ^λ.

Have a couple of implementations options. One approach is

- Forgetting factor for prior
- Forgetting factor for each class
- Compute pooled covariance matrix, using streaming prior

Performance assessment

Performance assessment and summary is difficult for data streams, particularly with real data, because of the unknown character of the drift. We use time-averaged point wise performance measures. CAC practitioners often favour either

- the bad rate among accepts (BRA) - the proportion of bad risk among the accepted population, for a fixed population acceptance level.
- The area under the ROC curve (despite recently discovered interpretation issues (Hand, 2009)).

We consider BRAA computed monthly, for a fixed proportion of accepts. Then, consider the relative difference between the BRAA for a target classifier with the base classifier.

Timing issues

We treat the time increment as a day. Within this, there are the following possibilities per day

1. no data - we ignore
2. one labeled data - proceed as above
3. more than one labeled data

Two choices:

- immediate updating - update with every new application, arbitrary order
- daily updating - update using the mean vector of a day's applications

Data and Results

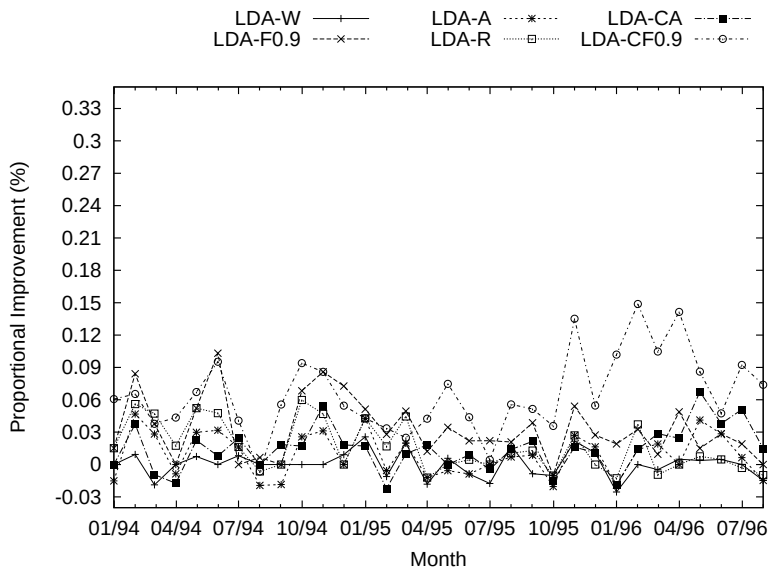
92258 UPL applications from 1993 -1997. Twenty predictor variables, typical of the application.

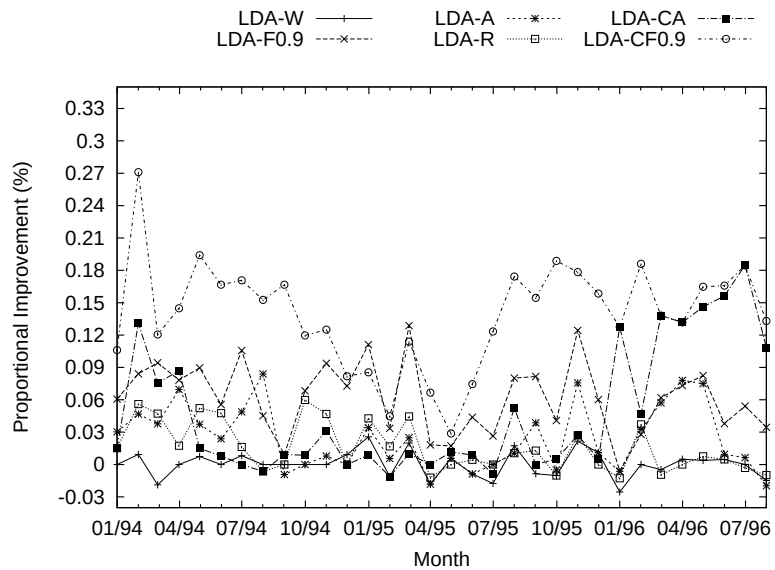
Report performance improvement in BRAA compared to LDA on first year of data.

Comparison includes

- contiguous windows
- Moving window
- fixed λ LDA
- variable λ LDA

LEFT: Daily, RIGHT: Immediate





- AF LDA methods consistently outperform the benchmark
- Best performance for fixed λ - but how to set in advance?
- No real difference between daily and immediate updating

Conclusion

AF methods have some merit for reducing performance degradation between classifier rebuilds. We have also developed AF versions of logistic regression which exhibits similar behaviour (Anagnostopoulos et al, 2009; Pavlidis et al, 2010).

Need to give proper attention to

- timing issues. Labels arrive in a much more complicated manner, and the methodology needs extension to handle this.
- optimisation parameters. Setting/changing.

References

- Adams, N.M., Tasoulis, D.K., Anagnostopoulos, C. and Hand, D.J. 'Temporally-adaptive linear classification for handling population drift in credit scoring', In Lechevallier, Y. And Saporta. (eds), COMPSTAT2010, Proceedings of the 19th International Conference on Computational Statistics, 2010, Springer, 167-176.
- Anagnostopoulos, C. 'A statistical framework for streaming data analysis', PhD Thesis, Department of Mathematics, Imperial College London, 2010.
- Anagnostopoulos, C., Tasoulis, D.K., Adams, N.M. and Hand, D.J., 'Streaming Gaussian classification using recursive maximum likelihood with adaptive forgetting', Machine Learning, (2010), under review.
- Anagnostopoulos, C., Tasoulis, D.K., Adams, N.M. and Hand, D.J. 'Temporally adaptive estimation of logistic classifiers on data streams'. Adv. Data An. Classif.,3(3) (2009),243-261.
- Hand, D.J. 'Measuring classifier performance: a coherent alternative to the area under the ROC curve', Mach. Learning, 77(1) (2009), 103-123.
- Haykin, S. 'Adaptive filter theory', 4th edition, Prentice Hall (1996).
- Kelly, M.G., Hand, D.J. and Adams, N.M., 'The impact of changing populations on classifier performance' in 'KDD 99, Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining', Chaudhuri, S. and Madigan, D. ed(AAAI), 1999, 367371.
- Pavlidis, N.G., Adams, N.M and Hand, D.J., 'λ-Perceptron: an adaptive classifier for data streams', Pattern Recogn., (2010) doi:10.1016/j.patcog.2010.07.026 .
- Pavlidis, N.G., Tasoulis, D.K., Adams, N.M. and Hand, D.J. 'Adaptive consumer credit classification', J. Oper. Res. Soc., (2010), under review.
- Weston, D.J., Anagnostopoulos, C., Tasoulis, D.K., Adams, N.M. and Hand, D.J. 'Handling missing feature values for a streaming quadratic discriminant classifier', Data Mining and Knowl. Disc., (2010), under review.