Handling the risk of obsolete information is there a one-size-fits-all strategy?

Dr. C. Anagnostopoulos
Statistics Section, Department of Mathematics, Imperial College London
joint work with Dr N. Adams, Prof. D.J. Hand, Dr. D. Leslie

August 29, 2013

UNOFFICIAL SUMMARY

- scorecard performance deteriorates over time – obsolete information?
- ‘population drift’: gradual changes in population characteristics lead to worse accuracy of scorecard over time
- common remedy: regularly update the scorecard (cost-benefit analysis)
- common ‘monitoring’ tool: observed rate of performance deterioration
- following the decision to update, one must decide whether/by how much to take historical data into account

Observations:

- for certain types of data, and certain types of scorecards, counterintuitive answers crop up (e.g., performance deterioration might be aggravated by rebuilding the scorecard)
- alternative criterion: gradient information

Introduction

What is drift, and why is it a problem

Characteristics of credit risk (as opposed to, say, planetary orbits):

- Credit risk is an **evolving process**, affected by a multitude of interconnected factors (e.g., macroeconomic and demographic variables, regulatory initiatives, human behaviour, etc.) These can be very volatile, or hard to measure.

Two ways to handle this:

A. Condition on time-varying covariates – hope that residuals are stationary

B. Model the data shift by an explicit model – hope that your assumptions are correct

Either might prove insufficient, leading to **drift**, and model deterioration. In which case:

C. Occasionally rebuild the scorecard.

Introduction

What is drift, and why is it a problem

Characteristics of credit risk (as opposed to, say, planetary orbits):

- Credit risk is an **evolving process**, affected by a multitude of interconnected factors (e.g., macroeconomic and demographic variables, regulatory initiatives, human behaviour, etc.) These can be very volatile, or hard to measure.

Two ways to handle this:

A. Condition on time-varying covariates – hope that residuals are stationary

B. Model the data shift by an explicit model – hope that your assumptions are correct

Either might prove insufficient, leading to **drift**, and model deterioration. In which case:

C. Occasionally rebuild the scorecard. *A type of local/weighted likelihood method.*

> *If the model is properly specified, a good fit in some region will not induce a poor fit in another region. However, when the model is not properly specified, then a good fit in some region may well detract from quality of fit in another.*
> D.J. HAND, V. VINCIOTTI, 2003

Motivation for this talk

Typical scenario

1. build a scorecard using historical data up to a certain date
2. deploy scorecard to infer credit risk of new applicants thereafter
3. monitor scorecard performance continuously
4. when performance deterioration exceeds a given threshold, go to step 1.

Such **one-size-fits-all** approaches to handling drift are highly desirable for practical purposes – but are they sensible? In this talk, we address this question. In particular:

- we list various types of drift; various types of scorecards; and various types of handling drift, some scorecard-specific, others scorecard-agnostic (like the above)
- we rely on a real-data example throughout, taking a practical approach.
- we reveal a flaw in the typical approach above, and discuss more generally the challenges involved in constructing one-size-fits-all strategies for handling drift.

. . . all within the framework of *online supervised learning*, which we introduce next.

Framework

The data

4635 credit card accounts, 755 of which defaulted within 12 months. **Features:**

- age of applicant at time of application
- employment status (employed / homemaker / retired / self-employed / student)
- home ownership status at time of application (council tenant / homeowner / living with parents / private tenant)
- months at current address at time of application
- application channel (cold call / doordrop / inserts / internet / mail / other)
- month of account opening (in months from January 2008)

What is a scorecard

A way of *scoring* applicants in terms of their propensity to default. Mathematically:

$$X \to f(X), \text{ where } f(X) = \hat{p}(\text{default} \mid X)$$

- $S()$ is typically constructed using historical data $(c_i, X_i)_{i=1:T}$, where c_i indicates whether the ith application (described by X_i) defaulted within 12 months.
- $S()$ can be turned into a *classifier* by thresholding. Conversely, most classifiers rely on a *scoring function* that can be used as a scorecard.

The problem

Types of ‘drift’ (purposedly informal notion)

- concept drift, population drift, . . . ; abrupt/smooth, regime-changes/short bursts

Types of scorecards/classifiers

- sampling (QDA) vs diagnostic (logistic)
- parametric (LDA) vs non-parametric (k-NN), linear (LDA) vs non-linear (QDA)
- offline/online, recursive vs incremental implementations, etc.

Adaptivity mechanisms

- sliding window
- changepoint detection and reset
- forgetting factors

What interplays exist? Can there be a **one-size-fits-all strategy?**

Alternatives / contrasts to adaptivity

Dynamic Modelling

Explicitly **model** temporal variation:

- time-series seasonality analysis
- state-space models – both smooth (Kalman filters) and abrupt (regime-switching)
- random-effect over time (similar to above)
- changepoint identification (retrospective)
- can offer more reliable forecasting accuracy estimates (tend to grow over time)

Time-Varying Covariates (TVC)

Condition on temporal variation:

- macroeconomic / behavioural variable selection
- should render population drift irrelevant

Difficulties:

- explicit modelling requires makes model much harder to fit
- model selection for dynamic modelling - a whole new world
- variable selection and lags for TVCs

... and you may **still** get it wrong. Some degree of temporal adaptivity is practically expedient in most real-time decision-making systems, no matter how good the model.

Different types of drift

Useful factorisation of the problem via Bayes' theorem:

$$P(C|X) = \frac{P(X|C)P(C)}{P(X)} \qquad \text{posterior} = \frac{\text{class-conditional} \times \text{prior}}{\text{population}}$$

Any of these quantities may 'drift':

- the overall proportion of defaults may increase ('prior' drift)
- the population characteristics of bad (resp. good) credit customers may change ('class-conditional population' drift) – e.g., defaults may now include more homeowners than before
- the overall population characteristics may change, irrespective of their propensity to default ('population' drift) – e.g., due to an ageing population

Depending on the problem, drift may be easier to understand in terms of one of these factors – but the phenomenon of interest is 'concept drift[1]', i.e., change in $P(C \mid X)$.

'Change' can mean several things:

- slow, gradual change
- an abrupt change into a new regime
- short-lived (or even instantaneous) bursts of irregular activity

We ignore recurring patterns throughout – they trivialise the discussion.

[1] Schlimmer & Grange, 1986; Widmer & Kubat, 1996

Different types of drift

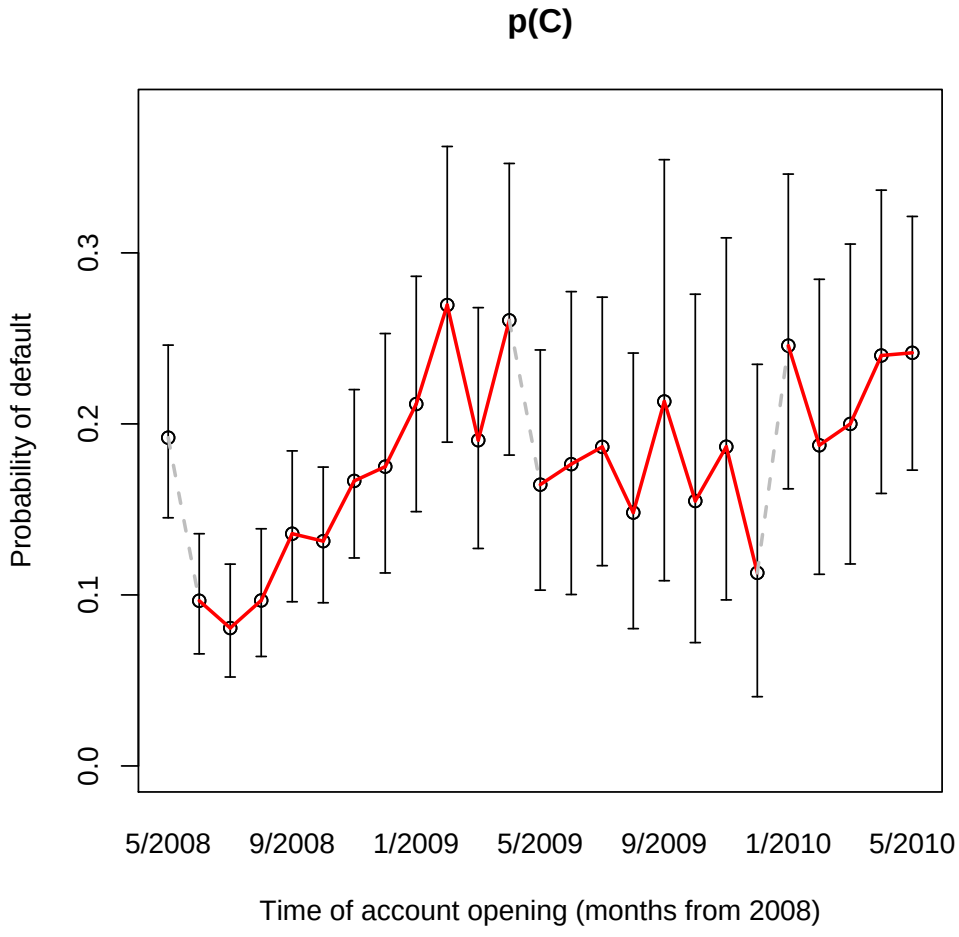
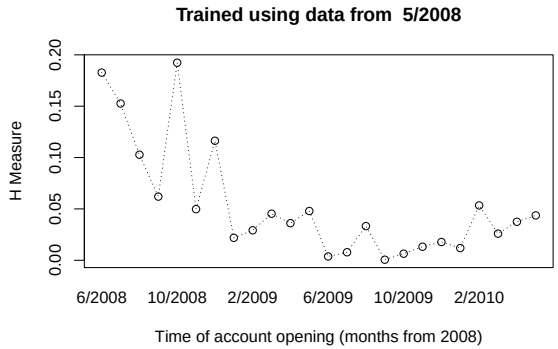


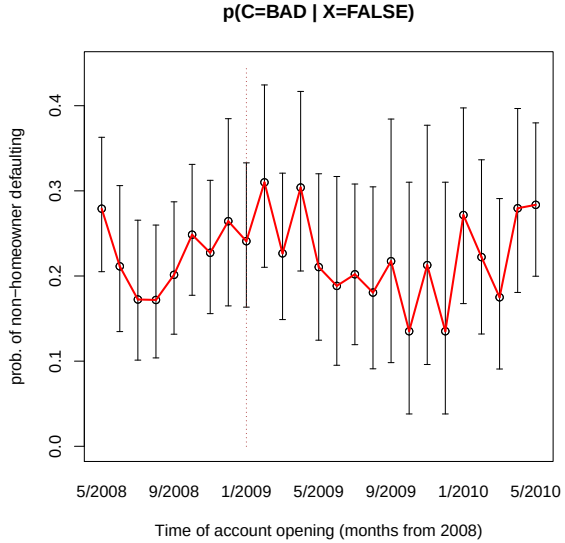
Figure: A *crude* 'drift detector' via sequential hypothesis testing. Contiguous red regions are static, or slowly drifting; regions in gray indicate abrupt changes to a new regime, or bursts of anomalies.

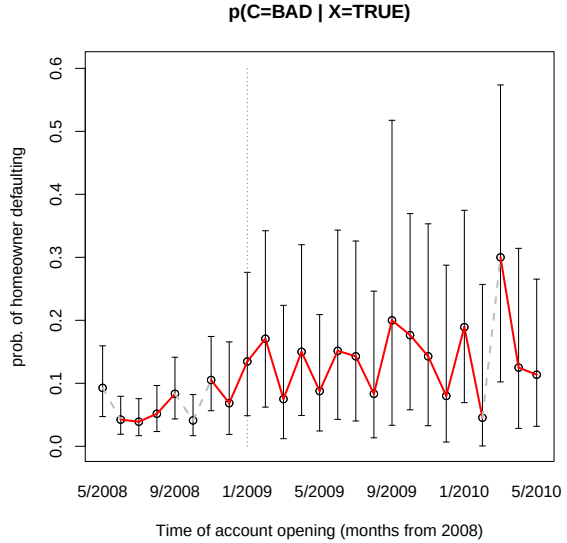
Toy example: default $\sim$ homeowner status

How does this affect performance? Clear signs of deterioration:



But reset offers no advantage at all in this case! Why? Because despite clear presence of population/prior drift, there is little evidence of concept drift:





Something **has** changed, but this (trivial) scorecard is unable to capture it.

Two less trivial classifiers

Logistic regression

Diagnostic approach, conditioning on x, log-odds are a linear combination thereof:

$$\text{logit}(p) = X^T\beta = \sum_{j=1}^{p} X_j\beta_j, \quad c \sim Bin(p)$$

Given an estimate $\hat{\beta}$, the score is $S(x) = p_{\hat{\beta}}(c = \text{TRUE} \mid x) = \text{logit}^{-1}(x^T\hat{\beta})$. The estimate is produced by maximising the *likelihood*, $\hat{\beta}_{1:T} = \underset{\beta}{\text{argmin}}\, \mathcal{L}\left(\beta; (c_i, x_i)_{i=1}^T\right)$.

Linear Discriminant Analysis

Sampling approach, modelling X given C as Gaussian, with a binomial prior on C:

$$p(X \mid C = T) = N(\mu^{(T)}, \Sigma),\ p(X \mid C = F) = N(\mu^{(F)}, \Sigma),\ p(C) = Bin(p)$$

MLE yields prior probabilities, the means per class, and the intra-class covariance $\hat{\Sigma} = \frac{1}{n}\sum_{i=1}^{T}\left(x_i - \mu^{(c_i)}\right)\left(x_i - \mu^{(c_i)}\right)^T$. Score is posterior probability (related to LRT):

$$\hat{p}(C = T \mid X) = \frac{\hat{p}(X \mid C = T)\hat{p}(C = T)}{\hat{p}(X \mid C = T)\hat{p}(C = T) + \hat{p}(X \mid C = F)\hat{p}(C = F)}$$

Remarks from the literature

"Performance deterioration does not necessarily imply concept drift", as the problem may have simply become inherently harder. *Why not reset anyway, 'to be safe'?*

- cost in estimation variance
- over-reaction in the case of smooth drift, as well as for abrupt jumps that are detected with delay (last few datapoints are relevant)
- in flexible modelling (e.g., kernel regression), deterioration may occur due to population drift entering a previously unseen part of the data space

Simulated examples

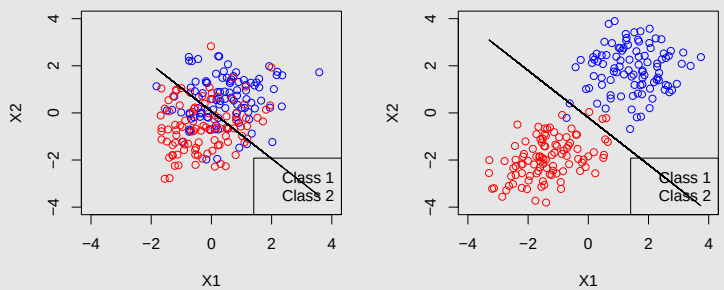


Figure: Two Gaussian populations drifting apart, with the resp. LDA fit

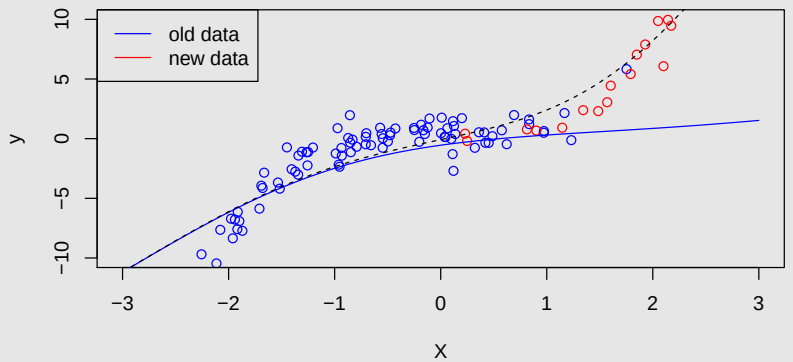


Figure: Counterintuitive effects of drift in non-parametric inference.

Remarks from the literature

"Logistic regression *conditions* on X, so, unlike LDA, it should be unaffected by drift in $p(X)$. *But under mis-specification, this is not necessarily the case.*

Simulated examples.

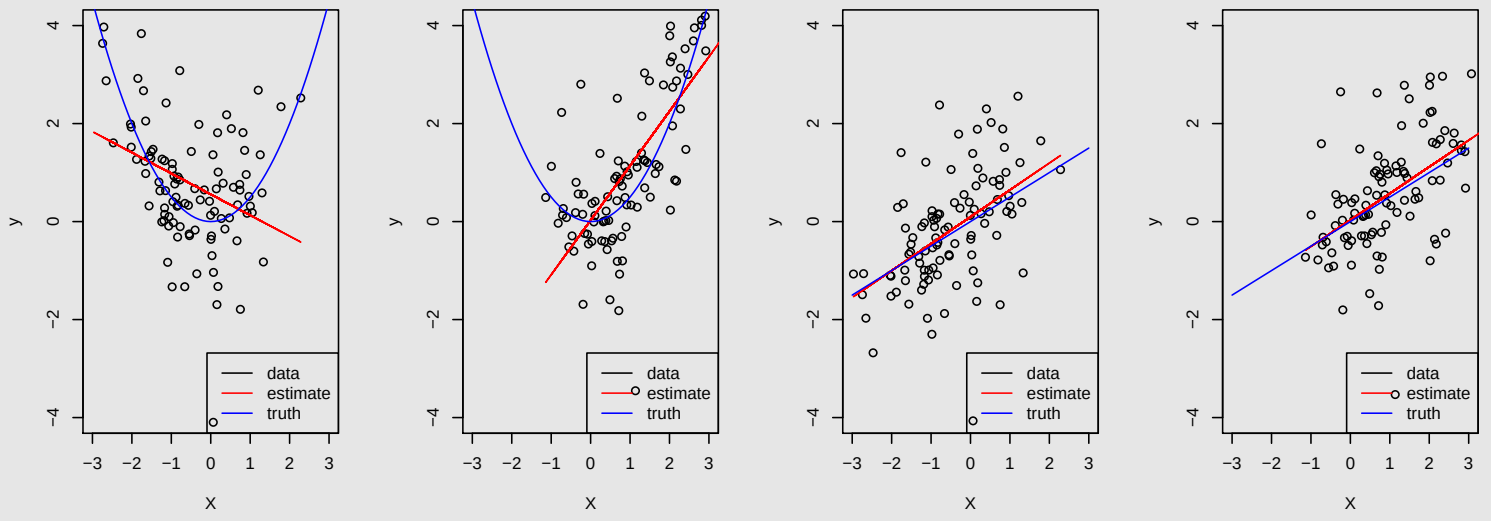


Figure: A linear regression model is fit to data generated from $y = \beta X^2 + \epsilon$ (two leftmost plots); and from $y = \beta X + \epsilon$ (two rightmost plots). For each setup, the X population drifts.

Remarks from the literature

"Drift will manifest as model mis-specification, as historical data come from a different model". But can we detect this in practise using model diagnostics?

Credit applications data.

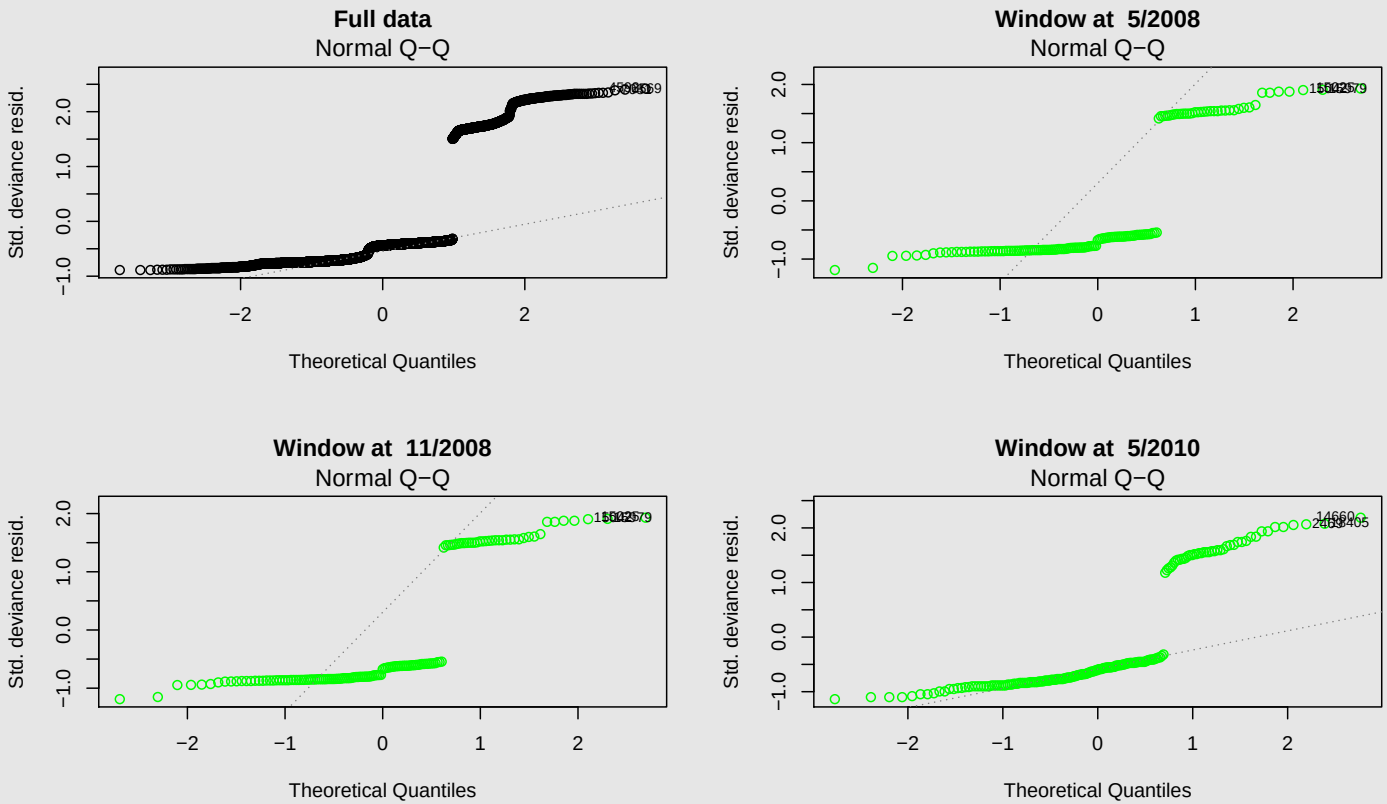


Figure: QQ plots for the logistic model against the full data (left), and window estimators.

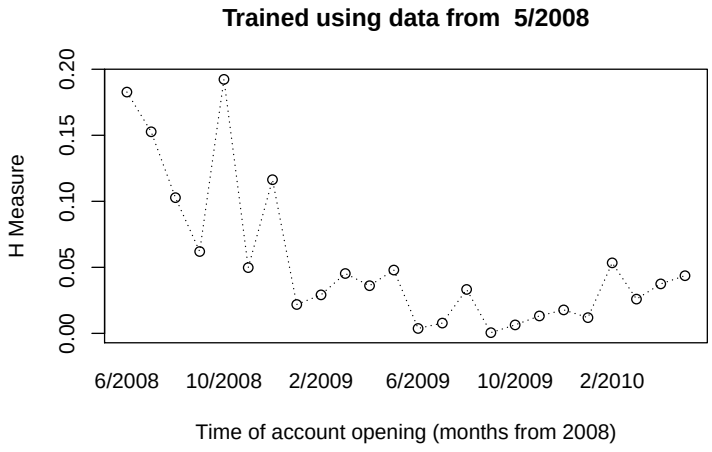
Further remarks from the literature

- the decision to reset the scorecard because of a drop in performance was already present in the CUSUM and control literature (changepoint detection)
- in Hashemi (2009) and Anagnostopoulos (2010) it was observed that monitoring **performance alone is not necessarily a good indicator** of concept drift.
- in Yang et al. (2005), it was challenged **whether 'recent data' are necessarily more valuable**, and a history of concepts was maintained instead. Recurring patterns had also been discussed in Widmer and Kubat (1996).
- Alaiz-Rodriguez et al (2008) assessed **whether simpler classifiers handle drift better**, motivated by remarks in Hand (2003). Their results were mixed.
- in the adaptive filtering literature (Haykin, 1996), forgetting factors were employed to handle non-stationarity of signals in a smooth fashion.

Recap

The problem:

- When dynamic modelling and TVCs fail you, drift may lead to model deterioration
- But discarding the data history may not always help



A proposed solution:

- A better way to discard data
- A better way to decide when to discard data

A formal framework

Denote the data by x_t, the (parametric) scorecard by θ and consider two loss functions:

- $L(x;\theta)$ is the *fit* loss that is minimised to build the scorecard: $\hat{\theta}_t = \underset{\theta}{\text{argmin}} L(x_{1:t};\theta)$
- $M(x;\theta)$ is the *performance* loss that is being monitored thereafter: $M(x_T;\hat{\theta}_t)$.

Although in practice data availability may mean that $T >> t$, we assume $T = t+1$.
Assume $L() = M()$; which must actually hold, approximately, for sensible results.

Model-free assumptions

- at time t, target $\theta_{t+1} = \underset{\theta}{\text{argmin}}\, \mathbb{E}_{t+1}[L(x_{t+1};\theta)]$ (if $L()$ is a log-lik., standard inference.)
- in the absence of drift, use $\hat{\theta}_t = \underset{\theta}{\text{argmin}}\, L(x_{1:t},\theta) = \underset{\theta}{\text{argmin}} \sum_{i=1}^{t} L(x_i,\theta)$
- if drift is suspected, could implement a window estimator:

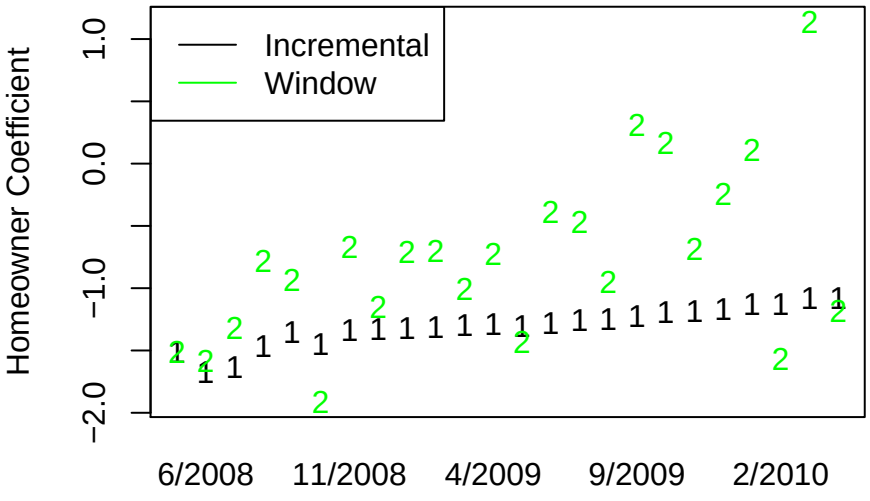
$$\hat{\theta}_t^{(\text{window})} = \underset{\theta}{\text{argmin}} \sum_{i=(t-w+1):t} L(x_i,\theta) \tag{1}$$

Unless $\hat{\theta}_t^{(\text{window})}$ is changing, no point worrying about drift!

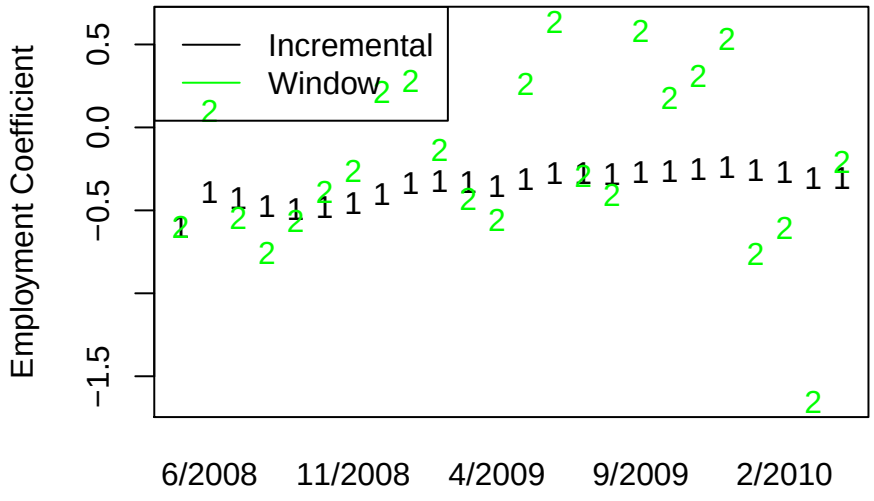
- A different take to the problem than Population Stability Indices: look at parameter estimate changes, not scoring distribution (decile) changes.

Parameter tracking

Homeowner coefficient for logistic classifiers



Employment coefficient for logistic classifiers

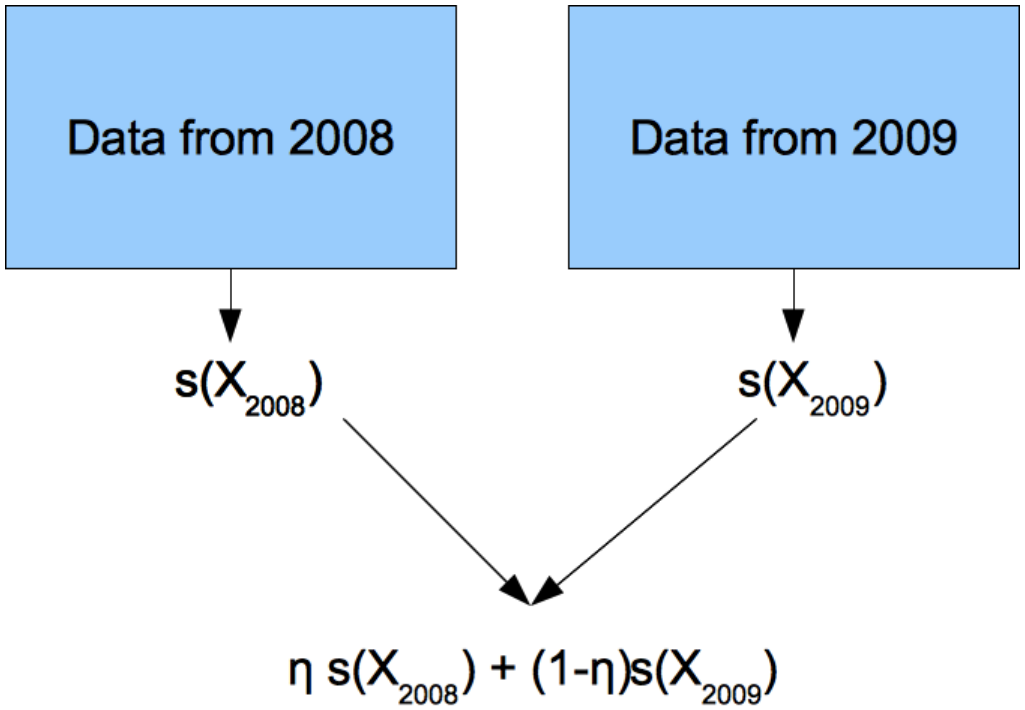


- Estimated coefficients seem to be drifting – but less data means more variability, and less reliable individual estimates. Classic bias-variance – can handle it.
- Note: windows here are non-overlapping, so no spurious auto-correlation.

Sliding windows are a ‘nasty’ smoother, due to sharp cutoffs. Smooth alternatives.

Data (is) history

To keep it or not to keep it? Not necessarily a binary choice:



$$\eta \begin{cases} = 0, & \text{reset at January 2009} \\ = 1, & \text{jointly consider 2008 and 2009} \\ \in (0,1), & \text{interpolates between the two.} \end{cases}$$

Choose η. Only as simple as this (convex sum) in simple models. See later.

Parameter tracking

Several smooth alternatives to sliding windows:

$$\hat{\theta}_t^{(\lambda)} = \operatorname*{argmin}_{\theta} \sum_{i=1}^{t} \lambda^{t-i} L(x_i, \theta) \qquad \text{(forgetting)}$$

$$\hat{\theta}_t^{(\alpha)} = \hat{\theta}_{t-1}^{(\eta)} + \alpha \nabla_\theta L(x_t; \theta)|_{\theta=\hat{\theta}_{t-1}^{(\eta)}} \qquad \text{(SGD)}$$

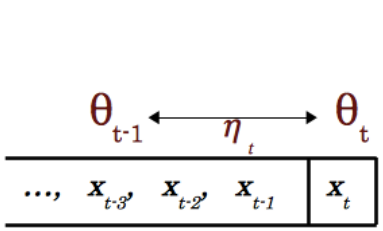
$$\hat{\theta}_t^{(\eta)} = (1-\eta)\hat{\theta}_{t-1}^{(\eta)} + \eta \operatorname*{argmin}_{\theta} L(x_t; \theta) \qquad \text{(convex updates – ensembles)}$$

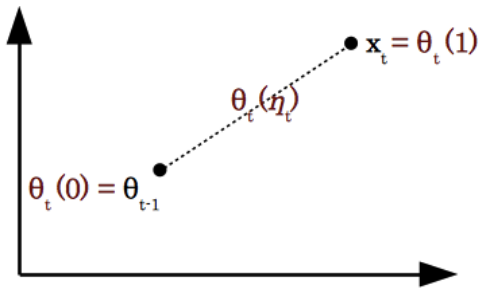
(2)

where η, λ, α may also be time-varying. In the case of a simple average, with squared loss, all three match, with $\eta_t = \frac{1}{t}$ yielding the static estimator:

$$\hat{\theta}_t = (1-\eta_t)\hat{\theta}_{t-1} + \eta_t x_t$$

Here, $\eta_t = 0$ resets, whereas $\eta_t = 1$ does not update at all, ignoring x_t.





Relationships to dynamic modelling

Relax the i.i.d. assumption:

$$x_t \sim f(X; \theta_t)$$

where θ_t typically follows a Markov process, e.g.,

$$\theta_t \sim g(\Theta; \theta_{t-1}, \phi)$$

The estimate $\hat{\theta}_t$ is then produced via inferential machinery.

- Requires exact assumptions about the 'drift'
- Optimal solution is typically not closed-form
- Optimal/approximate solutions not necessarily online

Relationships to dynamic modelling

The simplest possible state-space model is the following:

$$x_t = \mu_t + \epsilon_t,\ \epsilon_t \sim N(0, \sigma_X^2),$$
$$\mu_t = \mu_{t-1} + \eta_t,\ \eta_t \sim N(0, \sigma_\mu^2),$$

The posterior estimates are also known as the *filtering* density:

$$P(\mu_t; x_1, \ldots, x_t) = N(\hat{\mu}_t, \hat{\Sigma}_t^t)$$

and are given by the Kalman Filter (where $\rho = \frac{\sigma_\mu^2}{\sigma_X^2}$ is the SNR):

$$\hat{\mu}_t = (1 - k_t)\hat{\mu}_{t-1} + k_t x_t,\ \hat{\Sigma}_t^t = k_t \Sigma_V,\ k_t = \frac{\rho + k_{t-1}}{1 + \rho + k_{t-1}}$$

The Kalman gain k_t is a *learning rate*. It is also a **deterministic function** of the SNR.

- estimating ρ is tricky, even in this toy example (EM algorithm)
- *in practise, σ_μ^2 is often 'inflated' when errors grow*
- KF yields a distribution, not a point estimate – can fix this, see next slide
- a dynamic model is often just an operational device. Estimation of SNR-type parameters is not too different from data-dependent tuning of η.

Power priors

Bayesian update for current data D, historical data D_0:

$$P(\theta \mid D, D_0) \propto P(D \mid \theta) P(D_0 \mid \theta) \pi_0(\theta) = L(\theta; D) L(\theta; D_0) \pi_0(\theta)$$

where $\pi_0(\theta)$ is data-independent *initial* prior. Now generalise:

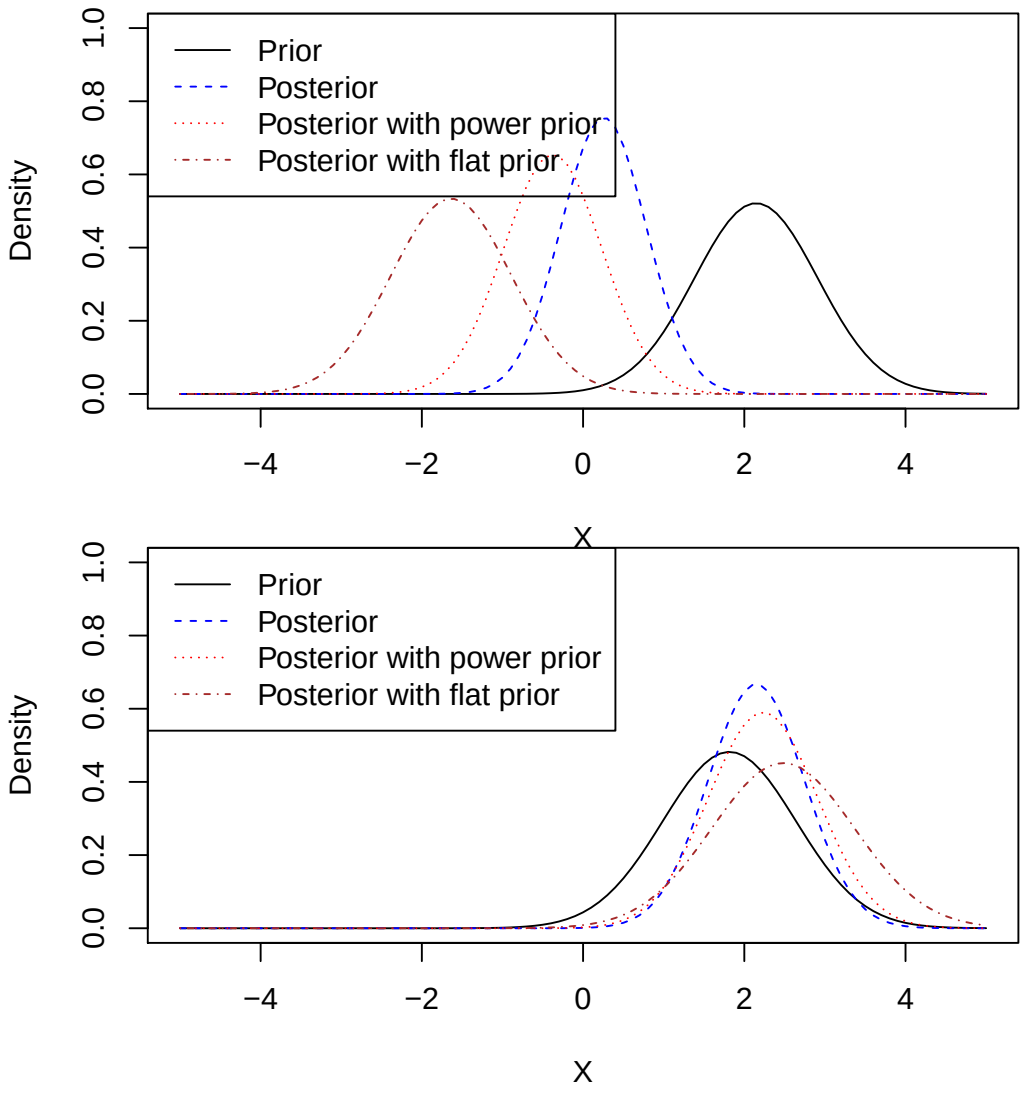
$$P^{(\lambda)}(\theta \mid D, D_0) \propto= L(\theta; D) L^{\lambda}(\theta; D_0) \pi_0(\theta)$$

where $\lambda \in [0, 1]$ is precisely the same as a *forgetting factor*:

- For $\lambda = 0$, ignore D_0.
- For $\lambda = 1$, equivalent to considering $D \cup D_0$.
- For $\lambda \in (0, 1)$, 'interpolate' between the two:

$$P^{(\lambda)}(\theta \mid D, D_0) = \underset{g}{\operatorname{argmin}} \{\lambda \mathrm{KL}(g, P(\theta \mid D)) + (1 - \lambda) \mathrm{KL}(g, P(\theta \mid D, D_0))\}$$

Power priors



Choosing η: the hopeful smile.

Recent work incorporates such ‘forgetful’ informative priors in dynamic trees:

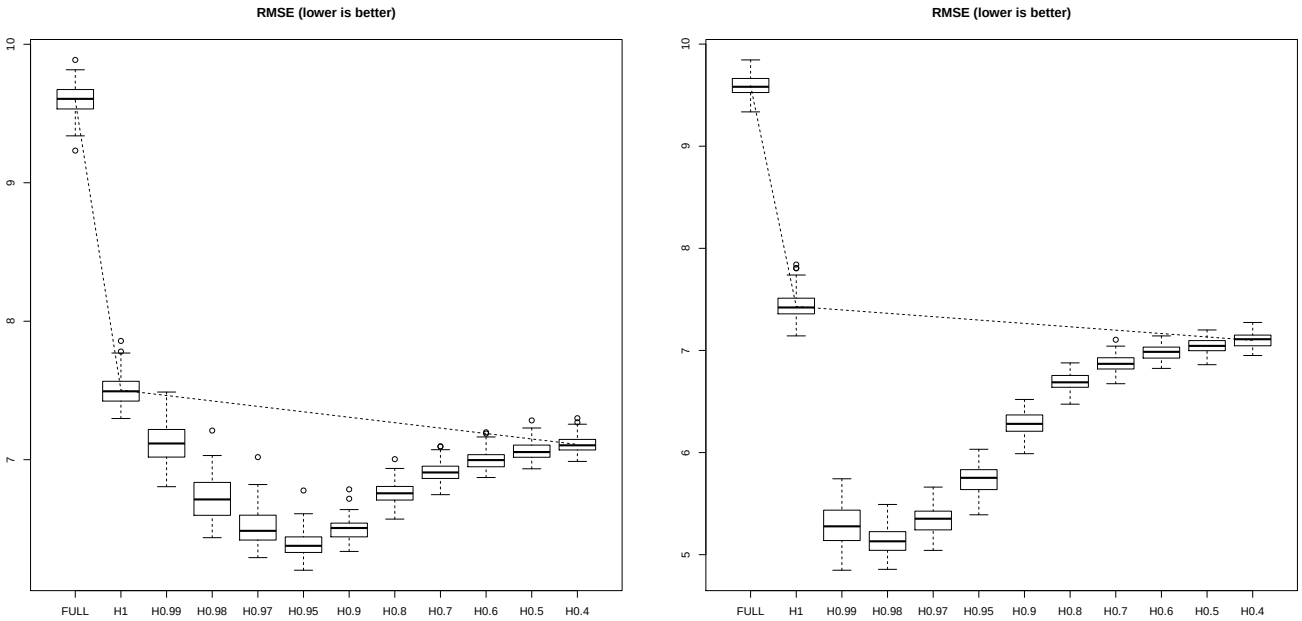


Figure: Performance of dynamic regression trees against slow (left) and fast (right) drift, measured by one-step-ahead squared error for the full data model, as well as various degrees of forgetting, ranging from $\lambda = 1$ (no forgetting) to $\lambda = 0.4$. The *U*-shaped curve is typical.

Choosing η: a three-way comparison

As explained above, a model update/reset is simply a choice of a value for η_t: how far along the path starting from the old estimate and ending at the new one should we go?

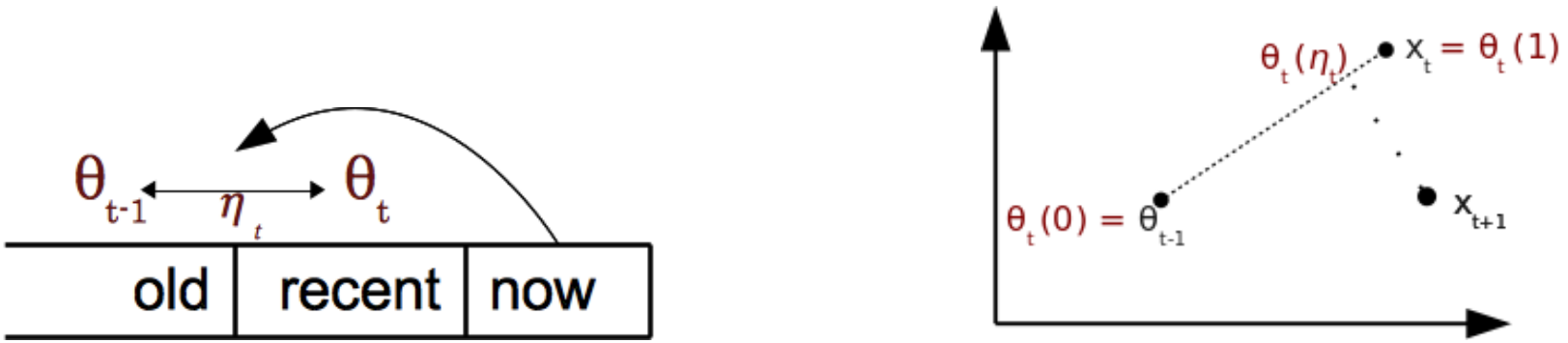


Figure: $\eta_t = \underset{\eta}{\text{argmin}}\, L\left(x_{t+1}; \hat{\theta}_t^{(\eta)}\right)$

Rely on a three-way comparison to avoid overfitting (akin to development / hold-out / out-of-time) and take an empirical approach:

- as far as it takes to minimise the one-step-ahead loss

! η/λ is a *smooth* parameter – easier to optimise than w.

One-size-fits-all? Perhaps, subject to constraints.

+ can handle outliers / abrupt jumps / smooth drift
- but this minimisation can be very noisy
- requires continuous incremental updating

Implies an appealing formal notion of drift: the optimal θ_t is varying but memoryless.

An illustration study

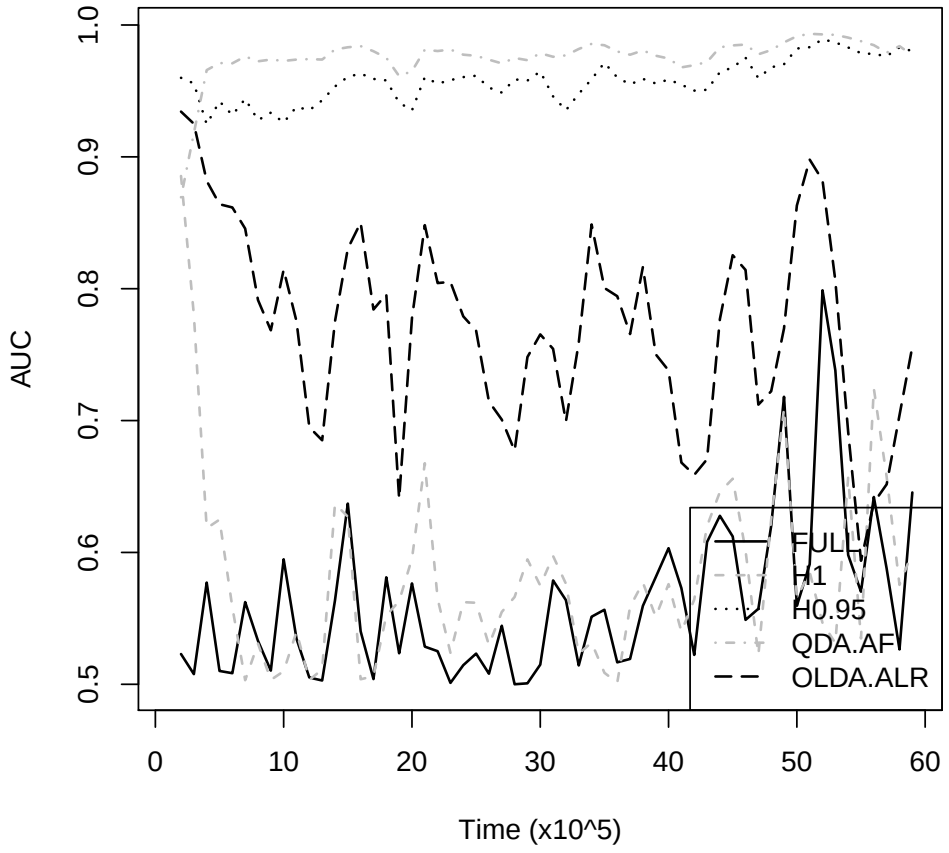


Figure: Methods investigated: trees – FULL uses the full data offline; H1 uses the data incrementally; H0.95 uses the data incrementally with fixed forgetting. Discriminant analysis: QDA-AF uses adaptive forgetting, OLDA-ALR uses a variant of performance monitoring.

Conclusions

Performance deterioration may not necessarily imply a need to forget:

- the problem may have just become harder
- resetting involves a cost in terms of estimation variance

Each model will be affected differently by drift:

- confounding between data shift and model mis-specification
- discrepancies between the way models are fit, and assessed

Can reason empirically about obsolete information:

- forget performance – are your parameter estimates changing?
- smooth, not sharp, forgetting – express updates as convex combinations
- use threeway comparison to decide whether to include data history

In practise ...

Downsides:

- requires iterative model refitting (not rebuilding)

Upsides:

- relatively easy to explain, and easier to implement than dynamic modelling
- if used cautiously, unlikely to cause drop in performance