CRC Working Papers

Model Shift and Model Risk Management

By Dr. Alan Forrest

January 2024

Contact
Credit Research Centre
University of Edinburgh Business School
29 Buccleuch Place
Edinburgh, EH8 9JS
credit.research.centre@ed.ac.uk

Model Shift and Model Risk Management

Alan Forrest, 12th January 2024.

Abstract

This paper describes a programme, based on the concept of model shift, leading to automated validation and monitoring. This means not only the automatic performance of the validation tests and calculation of the monitoring metrics, but their automated specification and implementation.

It describes an efficient method to explore data shifts and to calculate quickly, approximately and formulaically many important quantities used in model validation: the resulting model shifts and the top model impact sensitivities; the greatest weaknesses in the model specification; and the tests and monitoring to perform to quantify those weaknesses. From here the way opens to continual model updating, real-time validation and bespoke dynamic monitoring.

Of parallel interest, this programme motivates and makes accessible to applied modellers, modern statistical geometric ideas that continue to be a topic of advanced mathematical research.

1. Introduction: Why Model Shift is useful and important

This note is motivated by model risk and model validation in banking, and by a strong need to simplify, quantify and systematise model risk measurement. It unfolds as follows:

1. Model validation challenges and tests model specification and robustness. Many such challenges are variations on the "What if…?" question: "If the development data, assumptions and context were different, how differently would I build the model?"
2. Almost all such "What if…?" questions can be expressed as a data shift to model shift question. In a suitable mathematical environment (see appendix): "If the development data were changed (data shift), in what way or by how much does the model change (model shift)?"
3. As the number of such investigations can be large and ramified, each investigation should be as efficient as possible, ideally by going directly and quickly from data shift to model shift, without having to redevelop each shifted model from first principles.
4. This paper exploits such an approach to this model shift problem for categorical exponential family models, a class of models that includes logistic regression scorecards and many other kinds of model used in banking risk management. Mathematically we view the data and model both as single points in a high dimensional geometric space; the data point can appear anywhere in the space, but the model point is constrained to a pre-defined model subspace. Fitting a model to data is the same as finding the model point in the model subspace that is closest to the data point. The model shift problem then becomes a study of the related perturbations of the data and model points and of the differential geometry of the model subspace.
5. The mathematical development behind this is the topic of a second paper, and the appendix here illustrates one of its results: a computationally quick matrix calculation of model shift from the data shift, to first order approximation.

From this practical implementation of model shift flow all the program's benefits: improved approaches to model refresh, testing, validation and monitoring. This leads up to the automated specification of validation tests and of monitoring, and opens the way to real-time continuous model control and validation.

2. Practical Applications

This section shows how this program can be implemented, presenting six applications of the model shift idea that build up progressively and naturally.

1. Dynamic model reweighting (Application 1);
2. Systematic quantification of model specification error and improved model testing and validation (Applications 2, 3 and 4);
3. Systematic specification of model monitoring, tuned to model risks (Application 5);
4. Automated validation of model specification and automated specification of model monitoring. (Application 6).

Application 1: update model parameters dynamically

After a model is built, data continues to stream in and model implementation includes a feedback loop that monitors the performance of the model on the new data; detects when the model is under-preforming or is otherwise challenged; and then in that case changes the model in some way, with a quick overlay correction of outputs, a more granular reweight, or a complete redevelopment.

In most implementations, this feedback is slow and model changes are momentous: monitoring is usually on a monthly or quarterly cycle, often requiring committee decision to act; and changing a model can take months. In modern digital environments however, speed and agility are critical to a model's success, and such slow feedback is unacceptable. Thus we seek models that update dynamically in a stable and automatable way.

The approach described in the introduction of moving from data shift to model shift by an approximating linear map gives the quick feedback mechanism for such dynamic modelling. New data is added to the development dataset shifting that data point, by a small amount at first but potentially drifting far from the original development in time. At every time point the model shift and model parameter shift can be computed by matrix multiplication.

This matrix calculation is an approximation, so eventually too great a data shift will give an estimated parameter shift that is unacceptably approximate. As we approach that point (whose emergence can be tracked and detected in monitoring) a model refit can be planned in advance and the model parameters rebased, as in the usual process. But with this dynamic process the model change is anticipated, and in the meantime we have kept an old model in good condition through dynamic adjustment.

The transparency of this feedback process is assured by the steadiness of the drifting data and the linearity of the calculation, a matrix multiplication. In such a simple case, control boundaries or damping feedbacks are straightforward to arrange by classical approaches eg Kalman Filtering which are not discussed further.

Better approximations of model shift, to second or higher orders, are available and computable in principle, but are beyond the scope of this programme.

Application 2: use model shifts to quantify and prioritise validation investigations

Often a model validator's concerns can be expressed as "What if…?" scenarios. Here investigatory hypothetical data shifts test the validation concern, resulting in model shifts that reveal the model's sensitivities to these concerns and thereby detect and quantify model weaknesses or instabilities.

This list of validator's concerns and "what if" scenarios could be large, and each scenario may have several data shift directions and lengths to consider. Further if there is strong interaction between concerns, we may need to compound the "what if" scenarios combinatorially. Therefore the list of data shifts to explore might easily be many hundreds or thousands long. It is clearly not feasible to rebuild hypothetical models for all these cases, and a quick direct formula, such as the matrix multiplication in the appendix, is the only hope of producing all these model shifts (approximately).

Therefore to make progress in this case, we find model shifts by matrix multiplication of each data shift under investigation. This quickly quantifies the impact of each data shift and allows rapid triage:

- it lets us pick out the material model shifts for further investigation or action
- it allows us to drop immediately from consideration those data shifts that have little impact on the model specification;
- it gives an ordering of model shift impacts that allows us to prioritise the most sensitive or impactful data shifts.

My experience so far is that a first order approximation is good enough for this triage approach, giving roughly correct direction and proportion to each model shift.

Often the step from validator's concern to data shift is short and direct and little analysis beyond this triage is needed. If the model shift is large enough (or is in some other way concerning), then the validator raises a finding with an action for further investigation, model limitation, conservatism/adjustment or monitoring; if the model shift is not large then the model is likely to be robust against this particular concern, which is also worth noting as a positive comfort but does not raise a finding.

Application 3: find the business impacts of specific data shifts
Application 2 presumed a way of sizing or examining a model shift to tell when it is acceptable or unacceptable. This can be agreed qualitatively among experts in any particular case, and Application 2's approach will help to structure and inform a productive conversation between validator and developer about model sensitivities. Application 3 adds further quantification to this discussion, and so allows a more automated, less judgemental approach.

A model has many quantitative metrics or impacts that are important to the model manager, validator and business user, and which a model shift might change:

- Model specification itself: algorithm structure and parameters
- Statistical performance measures – accuracy, gini, etc.
- Direct impact – the impact of immediate use of the model's output – eg PD in RWA
- Indirect impact – the onward use of the model's output or judgemental observations based on the model – eg PD in IFRS9 SICR staging or in stress testing

Changes or instabilities in these values challenge the robustness of the original model; or suggest corrections, adjustments or conservatism; or give advice or overrides to the model user. The first three kinds of impact metric are formulaic quantitative functions of the model, and the fourth can be quantified to some extent.

To a first approximation, the model shift perturbs these metrics' values linearly, using a matrix (a derivative) that can be readily computed from the impact metric's formula[1]. Thus the matrix that takes data shift to model shift (Application 2) combines with this matrix from model shift to metric shift, to give a single matrix from data shift to metric shift. Application 3 then reduces to Application 2 with a suitably adapted matrix.

Because the impact metrics described in this Application are easily interpreted and familiar to the business users, it is straightforward to agree the acceptable boundaries of metric values, beyond which data shifts will not be allowed to shift the impact. This augments the triage process of Application 2 and enhances the validation conversation, now tied better to business outcomes.

Application 4: improve the validation narrative by finding the data shifts that affect the model impacts most sensitively
This application reverses the flow of analysis of Applications 2 and 3, asking "what are the directions in which the data can be shifted to have most effect on the model or model impacts?".

The "what if" scenarios investigated in Application 2 come from our imagination, and the list of data shifts is limited by computer capacity, and these may not explore well the space of possible data shifts which can be extremely high dimensional[2]. Therefore, we don't rely only on

[1] Where the impact has no derivative, eg where it meets a step change or floor, then this approach breaks down, though other combinatorial techniques may allow a solution. This is not discussed in this note.

[2] In high dimensions, almost all data shift directions have impacts that are close to "average sensitivity" and seldom do we find an extreme impact by chance. This loosely described and counter-intuitive statement can be made mathematically precise as Concentration of Measure.

scenarios, but search mathematically for those data shifts that cause the greatest impacts for smallest input.

To support this, note that the mathematical spaces involved (see appendix) come with a natural definition of length which we use to scale the degree of sensitivity.

Recall that data shift to model shift or business impact is computed by matrix multiplication, so that to find the most sensitive data shifts is to find the data shift of unit length that maximises its length after matrix multiplication. This is a well-understood optimisation problem whose solution can be found from the maximum eigenvalue and eigenvector of (a function of) the matrix in question; which in turn is a standard computation. The eigenvector that has maximum eigenvalue gives the direction of maximum sensitivity.

This eigen structure allows other sensitive eigenvectors to be picked out, with corresponding lower eigenvalues to show their strength of sensitivity. Thus we rank top downwards the sensitive eigenvector directions.

Each data shift direction picked out in this way can be described in words by looking at its coordinates and noting contrasts and patterns. This helps to understand and bring to life the scenario it describes. For example, we could find strong sensitivity to changes in the distribution of a particular factor, or of an interacting group of factors. This might have a good business interpretation and could become an insightful finding or query about the stability of those factors.

Such a way of writing enriches the validation narrative greatly, describing the most sensitive data shifts in business-relevant terms. Where these scenarios are realistic and intuitive, the business will be motivated to understand and manage the risk and mitigate it in model design or user guidance.

Application 5: specify bespoke model monitoring by the most sensitive data shifts
As part of model risk management suggested in Application 4, it is natural to seek model monitoring metrics that detect the sensitive data shifts, with thresholds tuned to the business impacts or model change caused by the shift.

Knowing the sensitive data shift direction, it is straightforward to set up scalar product calculations that detect most accurately the movements in that sensitive direction. This data shift measurement becomes therefore a bespoke monitoring metric for this particular sensitivity. A group of such metrics (say the top 3 or 5 sensitivities) could be added simply to regular model monitoring.

The tuning of thresholds involves tracking back from acceptable model impact constraints; this may be a complex negotiation with experts at its first set up, but is standardizable and automatically renewable (eg for observed type 1 and type 2 errors) thereafter.

It turns out that the bespoke metric described here is a variation of the familiar and much used Population Stability Index; it uses the PSI summands but re-weights them before summing. It is therefore an advantage to implement this metric within an existing PSI environment.

Note this is the way to set up monitoring of data shift sensitivities in general, including those scenario-generated sensitivities detected in Applications 2 and 3 above; we have found some data shifts that cause us concern, and so can design PSI-like metrics that detect those shifts best.

Note that each choice of model impact (Application 3) will determine different eigen structures and hence different monitoring proposals. It is important therefore to choose explicitly the model impacts to be measured, and note that they may need to be different for different models, audiences and contexts for monitoring.

Additional refinements – population shifts

There may be reason to restrict our attention to particular kinds of data-shift, which adds another matrix adjustment to the calculations and modifies the eigen-structure above, but in no essential way is the analysis or approach changed.

One important kind of data shift to consider is population shift – a data shift that moves only the input population distribution but not the output law (eg for PD we artificially double the number of customers in the LTV>80% group, but we don't change the observed default rates in that group). We may wish to restrict attention to population shifts, as a particular validation challenge and as a kind of advance warning monitoring.

A population shift that affects a model sensitively is a strong validation concern and a sign of a poorly specified model, because in that case the model can be broken by a change of input population alone, irrespective of output behaviour.

On the other hand, such population changes are quicker to detect: they can be measured now without having to wait for the model outcomes to play out (e.g. for PD the default event could be 12 months ahead). Therefore population shift monitoring has a special place in the monitoring suite and is the foundation of a dynamic early warning model monitoring suite.

Application 6: automate validation and monitoring specification

All these previous applications are automatable in almost all their steps.

Starting with the development dataset and the MLE optimised model, the calculation of the matrices above (of all kinds) and the determination of eigenvectors are automatable.

Then, choosing the appropriate model impacts to measure and their acceptable boundaries, the specification of bespoke monitoring and thresholds is also automatable.

Then the top sensitivities for model shift and impact can be printed out to a human validator who then checks and selects them for attention, describing them in words in a query for developers or as a finding. This helps speed up the writing and deepen validation findings, and gives confidence in the completeness of the challenge to specification risk.

Thus we set up a dynamic updating model (Application 1), with a validation and monitoring feedback loop also automated and shortened (Application 5). In this case dynamic monitoring and validation of specification would shadow, possibly in real time, the dynamic updating of model design and parameters.

Appendix – the impact of data shift on model parameters

The first order model shift can be described explicitly as a matrix multiplication of the data shift. The following gives a convenient set up and formula for such a matrix. The derivation of this and similar formulae, and their theoretical basis are the topics of a second paper.

For simplicity we look at the case of a logistic regression, with categorical input factors, and binary output $\{1,-1\}$. The number of input factor combinations is N, supposed to be large, and the degrees of freedom of the regression is M, usually much smaller.

The development data on which the model is fitted are represented as frequencies in a contingency table – this table has $2N$ cells (assuming no structural zeros). We normalise these frequencies to sum to 1, obtaining a "data point" x.

Likewise the logistic model predicts a binomial distribution output for each combination of inputs. With the distribution of input factors given by the development data, the model predicts or describes a frequency distribution on the same contingency table. Thus another data point is created from the model – the "model point", y.

For data and model points we use a functional notation so that where a is a combination of input characteristics and b is a choice of output, then $x(a,b)$ is the proportion of the development data that has those characteristics and output. Likewise $y(a,b)$ is the model's prediction of this proportion.

We assume that all values of y are strictly positive and note practical challenges to this assumption at the end.

Note the following properties:

1. Data point and model point have the same input population distribution: $x(a,1)+x(a,-1) = y(a,1)+y(a,-1)$ for all input combinations a.
2. For each input combination a, $log(y(a,1)/y(a,-1))$ is the log odds ratio predicted by the model, which in logistic regression is a linear form of combinations of input characteristics. By contrasting suitable choices of a, the model parameters can be retrieved, assuming no singularity. Thus the model point determines the parameters of the model.

Let D be the design matrix of the regression. This is a matrix with M columns (indexed by model parameters) and N rows (indexed by the combinations of input factors). The design matrix is a standard construction of regression showing how the model factors are coded as combinations of the cells of the contingency table.

Let Y^+ be the NxN diagonal matrix indexed by input factor combinations, whose a entry is $y(a,1)$.
Let Y^- be the diagonal matrix derived from $y(a,-1)$ likewise.

Let $Y = Y^+ \, Y^- \, (Y^+ + Y^-)^{-1}$.

Let $Z = Y^+ \, (Y^-)^{-1}$ the diagonal matrix of modelled odds ratios, so that $Y = (I + Z)^{-1} \, (I + Z^{-1})^{-1} \, (Y^+ + Y^-)$.

Let $C = D^T \, Y \, D$, an MxM matrix, assumed non-singular.

Suppose that the data are shifted, so that the data point *x* moves to *x+dx* . Write dx^+ for the vector indexed by input factor combinations, *a*, $dx^+(a) = dx(a,1)$. Likewise, let $dx^-(a) = dx(a,-1)$. Then it can be shown that the parameters of the model are shifted by

$$dp = C^{-1}D^T\,[\,(I + Z)^{-1}\,dx^+ - (I + Z^{-1})^{-1}\,dx^-\,]$$

Where some $y(a,b)=0$, then there are no data with the input factor combination *a*. This is quite likely where there are large numbers of factors or many classes, even for large development data sets.

Sparse contingency tables are a long-recognised problem in certain analyses, and practical solutions are a study in themselves. We do not explore this here, but note two possible approaches.

1. The matrices *Z* and *Y* above may be defined directly from the modelled odds, even if cells are zero; so the calculation can proceed. The inaccuracies of a linear approximation in this case are another matter, not pursued in this note.
2. Alternatively, we can add a small non-zero offset to each cell count in the contingency table. Statistics students are familiar with the trick of adding 0.5 to every cell count to help with sparse contingency table analysis, and this and variations are acceptable in many statistical contexts. The biasing effect of this adjustment is not assessed here.