When to rebuild or when to adjust scorecards

Authors and Affiliation and Postal address:

K. M. Jung

Department of Informational Statistics, Kyungsung University, Busan 608-736, South Korea

Lyn C. Thomas

Mee Chi So[1]

Southampton Management School, University of Southampton, Highfield, Southampton, United Kingdom, SO17 1BJ

Email: L.Thomas@soton.ac.uk, M.So@soton.ac.uk

Tel: +44 (0)23 8059 8964

Fax: +44 (0)23 8059 7677

Abstract:

Data based scorecards, such as those used in credit scoring, age with time and need to be rebuilt or readjusted. Unlike the huge literature on modelling the replacement and maintenance of equipment there have been hardly any models which deal with this problem for scorecards. This paper identifies an effective way of describing the predictive ability of the scorecard and from this describes a simple model for how its predictive ability will develop. Using a dynamic programming approach one is then able to find when it is optimal to rebuild and when to readjust a scorecard. Failing to readjust or rebuild a scorecard when they aged was one of the defects in credit scoring identified in the investigations into the sub-prime mortgage crisis.

Key words: consumer credit risk; maintenance; readjust; credit cards; behavioural scores.

[1] Corresponding author

1. Introduction

Credit scoring has proved one of the most successful applications of Operations Research in the finance industry (Anderson, 2007; McNab and Wynn, 2000; Thomas, 2009; Finlay, 2008). It uses data on past borrowers to predict which consumer applicants and borrowers of consumer credit are likely to default within a given time period (be Bad) or not default (be Good). Almost all lenders of consumer credit now use scoring as part of their risk assessment. Its success has led to data-mining based scoring being used in a wide range of applications. These include direct marketing (Malthouse, 2001), customer lifetime value (Glady et al., 2009), customer churn (Burez and van den Poel, 2009), intensive care assessment in hospitals (Arts et al., 2005), and parole decisions (Petersilia, 2009).

Just as equipment deteriorates over time so does scorecard performance. This happens for several reasons – changes in the population who are being scored, changes in the economic and commercial environment, and changes in the operating policy of the lender. Yet though there has been a significant literature on modelling the optimal maintenance and replacement policies of deteriorating equipment, there has been hardly any work on when to readjust (essentially repair) and when to rebuild data based scoring systems. In credit scoring, this failure to model when a scorecard needs rebuilding was recognised as one of the causes of the sub-prime mortgage crisis of 2007-8. The US Securities and Exchange report (2008) identified that credit bureau scorecards had aged significantly over time. This is clear in the analysis by Demyanyk and van Hemert (2008), Ashcraft and Schuermann (2008) and Rona-Tas and Hiss (2008). The power of the score to predict delinquency or foreclosure dropped considerably between 2001 and 2006. Yet the same scorecard was in operation throughout and a new one was not introduced until 2008.

The aim of this paper is to build a model which predicts when a scorecard will need to be rebuilt and when it needs readjusting. Such a model ensures improved scorecard performance but also indicates the likely workload on the analytics team involved in building and maintaining the scorecard.

Traditionally it would take a team of analysts three to six months to build a new scorecard. Much of this effort is involved in acquiring and cleaning an appropriate data set (Anderson, 2007; Thomas, 2009). Then they need to identify, analyse and develop the characteristics that relate most strongly to the Good/Bad status of the borrower at the end of the performance period. Finally a classification technique such as logistic regression is used to build the final scorecard. (Details can be found in Anderson (2007) and Thomas (2009)). Even with modern data mining tools this process still takes time.

Even the simpler readjustment action is not without a cost. In application scorecards this adjustment is equivalent to moving the accept/reject decision for new borrowers. Such decisions involve analysis and discussion between the marketing and risk departments to get some understanding of what is happening and why it is occurring.

Behavioural scoring also assesses borrowers' default risk for the Basel Accord regulations. These regulations require validation of the score to probability of default (being a Bad) relationship including recording and justifying any changes made in this relationship. The analysis involved in such justification means readjustment of a behavioural score is also not costless.

The dearth of replacement and maintenance models for data based scorecards is due to the lack of a compact description of the state of the scorecard and so to a model of the dynamics of such state movements. With scorecards the state of the system should reflect the accuracy of the scorecard's prediction. Either the state description is too large like a ROC curve which is a function over a compact interval or it only partially describes the discrimination of the scorecard like the error functions at a specific cut-off. Scorecards are monitored every month using a number of performance measures (Anderson, 2007; McNab and Wynn, 2000) but these only describe the past performance of the scorecard and there is no forecast of what will happen in future periods. Thus the decisions about when to rebuild or repair a scorecard are static ones depending on how badly the scorecard performed in the previous periods. The objective of this paper is to show how one of the measures of performance – the log odds to score graph – gives an efficient description of the state of the scorecard. Its dynamics can then be represented by a simple diffusion model. This allows one to determine optimal rebuilding and readjusting strategies, which include forecasts of the future ageing of the scorecard.

The literature on the replacement and maintenance of equipment and of complex systems is long and distinguished. The survey articles by McCall (1965), Pierskallar and Voelker (1976), Sherif and Smith (1981), Thomas (1986), Valdes-Flores and Feldman (1989), Sarkar et al. (2011) give an indication of the developments. Initially the systems were modelled as states of a Markov chain (Derman, 1963) with the dynamics given by the transition probabilities. Deciding when to repair and when to replace leads to a Markov decision process formulation. Subsequent models made the state the age of the system, or the mileage of the vehicle. Regular inspection led to replacement strategies based on condition monitoring (for example Liao et al. (2006)). Here the condition of the system is described by physical measures like temperature or vibration values. Other models combined the replacement and repair decision such as Lui et al. (1995).

The main approach to modelling replacement and maintenance problems is dynamic programming, particularly Markov Decision Processes, White (1969), Puterman (1994), Buerle and Rieder (2011). This techniques optimizes a sequence of decisions when the state the system has stochastic dynamics.

There are many real applications of this approach (White, 1988; 1993). However nowhere in the literature are there models of when to rebuild or repair scorecards, even in books on dynamic programming and replacement modelling (Denardo, 1982; Bertsekas, 2012; Dohi and Nakagawa, 2013).

The literature on software maintenance looks at when existing software programmes need to be updated to deal with the new needs of the organisation. Xiong et al. (2011) model the problem as a continuous time Markov chain; Kulkarni et al. (2009) look at a queuing theory approach; while Feng et al. (2006) use dynamic programming. Krishnan et al (2004) derive a Markov decision process model where the decisions are: keep the system operating, introduce a minor patch release or a major upgrade. These models are based on the idea of software ageing because of fragmentation, resource leakage, numerical error accrual and data corruption (Grottke et al., 2008). However scorecards age for different reasons and the objective in maintaining credit scorecards is to improve the existing usage of the scorecards rather than to develop new applications.

Crook et al. (1992) were one of the first to investigate how credit scorecards age. They looked at the difference in scorecards built on data which was just before and during an economic downturn. Varga (2009) modelled when one should build a new scorecard. In it the state of the scorecard was the percentage of defaults (Bads) and non-defaults (Goods) who were misclassified at a specific cut-off score. This approach does not allow one to adjust the cut-off score. So there is no opportunity for readjusting the scorecards, which is equivalent to changing this cut-off. The model in this paper has a more sophisticated description of the state of the scorecard which allows reclassification as well as rebuilding actions.

In section two we define log odds scores, explain why are they important and how the scorecard performance can be described in terms of the log odds to score graph. In the following section we formulated a general discrete time Markov decision process model, based on the dynamics of this graph. This is used to determine the optimal rebuilding and readjusting strategies. Section four looks at aspects of this strategy including the optimal cut-off scores after rebuilding and readjusting the cut-off scores and the control limit form of the rebuilding decision. In section five we model a specific form of the dynamics of the scorecard's ageing and in section six we apply this to a case study describing the performance of a scorecard used by a portfolio of over 2,000,000 credit card accounts over two years 2002-2003. In the final section we draw some conclusions on the effectiveness of the model and its possible applications.

2. Credit scorecards

2.1 Building credit scorecards

Almost all credit scores are log odds scores (Thomas, 2009) or linear transformations of log odds scores. Scorecards built using logistic and linear regression, which are the most common ways of building scorecards, are of this form. Let $P(B \mid \mathbf{x})$ be the probability a borrower with characteristics $\mathbf{x}$ is "Bad" and defaults within a given time horizon, and let $P(G \mid \mathbf{x}) = 1 - P(B \mid \mathbf{x})$ be the probability the borrower will be "Good" and not default. Then a log odds, $s(\mathbf{x})$ satisfies

$$s(\mathbf{x}) = \log\left(\frac{P(G \mid \mathbf{x})}{P(B \mid \mathbf{x})}\right) \quad (1)$$

For a linearly transformed log odds score one has

$$a + bs(\mathbf{x}) = \log\left(\frac{P(G \mid \mathbf{x})}{P(B \mid \mathbf{x})}\right) \Leftrightarrow P(G \mid \mathbf{x}) = \frac{1}{1 + e^{-a-bs(\mathbf{x})}} \quad (2)$$

Logistic regression produces a log odds score $\tilde{s}$ but this score is often linear transformed by $(\tilde{s} - a)/b$ to have an agreed marginal good:bad odds at some specific score and a doubling of the log odds for some specified increase in the score.

To use such a scorecard in decision making one needs a cut-off score *c*, which will categorise the borrowers into two classes and hence can be used for a binary decision. For application scores, those above the cut-off are Accepted (because they are predicted as Good), while those below the cut-off are categorised as Rejected, (because they are predicted as Bad). For behavioural scores the cut-off might separate those for whom a credit limit increase is appropriate from those for whom it is not. Thus a log odds scoring decision system consists of three parameters *(a,b,c)*. In practice *a* and *b* are fixed when the scorecard is built but vary because of the scorecard ageing while *c* is a decision by the lender when rebuilding or readjusting the scorecard. Currently users plot the score to log odds graph each month as a visual representation of the way the scorecard is performing. What is new in this paper is the realization that the three parameters (*a,b,c*) give a concise and accurate description of the state of the scorecard and the current decision using it. Moreover by forecasting how *a* and *b* will change over time one can estimate when the scorecard should be rebuilt and what readjustment should be done.

2.2 Readjusting scorecards

Scorecards "age" over time for a number of reasons, including when changes in economic conditions affect the default rates so that the score to default rate relationship changes even though the relationship of the individual characteristics to the outcome is unchanged. As Thomas (2009) showed, Bayes theorem means that scores can be split into two components. One is the current log odds of non-defaults to defaults and the other is the weights of evidence of the characteristics of the individual borrower. This means the score one would like to use is *s*(x,t)* the score of a borrower with characteristics ***x*** at time *t*. It is given by

$$s^*(\mathbf{x},t)=\log\left(\frac{P(G|\mathbf{x},t)}{P(B|\mathbf{x},t)}\right)=\log\left(\frac{(1-d(t))P(\mathbf{x}|G,t)}{d(t)P(\mathbf{x}|B,t)}\right)=\log\left(\frac{1-d(t)}{d(t)}\right)+woe(\mathbf{x},t) \quad (3)$$

where *d(t)* is the default rate of the whole population at time *t* while *woe(x,t)* is the weights of evidence of a borrower with characteristics ***x*** at time *t* (i.e. $woe(\mathbf{x},t)=\log\left(\frac{P(\mathbf{x}|G,t)}{P(\mathbf{x}|B,t)}\right)$.

However the actual score that the scorecard produces is *s(x,t)* where if the scorecard was last rebuilt or readjust at time t_0 and there has been no change in the weight of evidence term, this is

$$s(\mathbf{x},t)=\log\left(\frac{1-d(t_0)}{d(t_0)}\right)+woe(\mathbf{x},t_0)=\log\left(\frac{1-d(t_0)}{d(t_0)}\right)+woe(\mathbf{x},t)=s^*(\mathbf{x},t)+\log\left(\frac{(1-d(t_0))d(t)}{d(t_0)(1-d(t))}\right) \quad (4)$$

Thus one would like to adjust the score by the subtracting the second term in equation (4) from the score in order to deal with the ageing. This is akin to *a* changing.

A second reason scorecards age is that the relationship between the characteristics and the Good/Bad status changes over time. So the weights of evidence term, $woe(\mathbf{x},t)$ may also change for certain characteristics. This can occur if there is a population drift, so that the new borrowers entering the portfolio are different from those who are leaving the portfolio. Thus it is reasonable to assume that the parameters *(a,b)* which describe the relationship between the score and the default probability will change over time, i.e. the scoring system at time *t* is described by *(a(t),b(t),c(t))* where *c(t)* is the cut-off score in operation at time t.

3. Model Formulation

Scorecards age and need rebuilding if for example the score to log odds relationship in (2) flattens so the gradient *b* tended to zero. Then the score is not discriminating well between the Goods and the Bads. Rebuilding will cost *B* and results in a new scorecard, which is scaled so that its score to log odds relationship is given by the parameters a_B and b_B and the cut off score is set at c_B.

A less severe change to the existing scorecard is to adjust the cut-off score, which is equivalent to readjusting the scorecard by adding or subtracting the same constant to each borrower's score. As (4) shows this would be an effective way of dealing with changes in the underlying population default rate. With application scores most lenders change the cut-offs and keep the score as is. With behavioural scores they do the same for credit limit changes. Changing *a* and keeping *c* fixed which is what happens if behavioural scores are used for Basel Accord purposes, is equivalent to keeping *a* fixed and changing *c*. For consistency we take "recalibrating or readjustment" to be readjusting the cut-off score both for application and behaviour scorecards. Readjusting involves some analysis and so its cost, *R*, is not negligible, even if lower than the rebuilding cost.

If there is no rebuilding or readjustment we assume the "ageing" of the scorecard is given by a discrete time Markov process, *t=1,2,…* with transitions given by a conditional density function $p(a(t+1),b(t+1)\,|\,a(t),b(t))$. The process is discrete time because the borrower's score and default status is only recalculated at monthly intervals.

The state of the scorecard at any time *t* is *(a(t),b(t),c(t))* where A*(t)* and B*(t)* are the random variables whose values for the parameters in the score to log odds linear relationship (2). *C(t)* is the variable whose value c*(t)* is the current cut-off score. In application scoring, the profit is the difference of two terms. Good borrowers who are accepted and then repay produce a profit of *L*. Bad borrowers who are accepted and then default cost the lender *D*. If the state of the system is *(a,b,c)* the profit from *N* applicants in a period is given by *Q(a,b,c)* and its expectation by *E*(*Q(a,b,c))*

$$\begin{aligned} E(Q(a,b,c)) &= N\left(-D\int_c^\infty P(B\,|\,s)f(s)ds + L\int_c^\infty P(G\,|\,s)f(s)ds\right) \\ &= N\left(-D\int_c^\infty \frac{e^{-a-bs}f(s)}{1+e^{-a-bs}}ds + L\int_c^\infty \frac{f(s)}{1+e^{-a-bs}}ds\right) \end{aligned} \tag{5}$$

Implicit in (5) is that the score distribution *f(s)* is stationary over time. This is the case for the application score of a given vintage segment of the portfolio. If there is churn in the portfolio then the simplest assumption is that the new borrowers have the same distribution as those who are leaving. The non-stationary assumption can be relaxed but one must then estimate how the score distribution varies over time. Also the resultant optimal policy will be non-stationary and hence difficult to implement. In this paper we therefore concentrate on the case when the score distribution is stationary.

We are now in a position to build the Markov Decision Process model. Let *V(a,b,c)* be the future optimal expected discounted profit of operating, rebuilding and readjusting such a scorecard system over an infinite time horizon given the current scorecard is in state *(a,b,c)*. The profits and costs are discounted by a monthly discount factor β. *V(a,b,c)* satisfies the optimality equation

$$V(a,b,c)=\max\begin{cases} E(a,b,c)+\int \beta V(x,y,c)p(x,y\mid a,b)dxdy \\ \max_{c_R}(-R+V(a,b,c_R(a,b)) \\ \max_{c_B}(-B+V(a_B,b_B,c_B) \end{cases} \tag{6}$$

The first expression on the right hand side corresponds to continuing with the existing scorecard; the second expression to readjusting the scorecard by changing the cut-off score *c;* the third corresponds to building a new scorecard where the slope of the score – log odds curve is given by b_B , its intercept is a_B and the cut-off chosen is c_B .

4. Properties of the Optimal Policy

This section comments on the optimal cut-offs if the scorecard is readjusted or rebuilt. It then proves the optimal rebuilding policy is a control limit one, under reasonable conditions on the dynamics of the log odds to score graph,

The profitability of the portfolio should be the dominant objective of a lender. If the scorecard is in state *(a,b,.)* profitability is maximisd by a cut-off $c_M(a,b)$ which accepts all those who are currently profitable and rejects those who are not. This means choosing a cut-off score where the Good:Bad odds at the cut-off score equals the ratio of *D* to *L*. This follows since the profit is $LP(G|s)-DP(B|s)$ and so is zero when

$$\frac{p(G\mid s(x)=c_M)}{p(B\mid s(x)=c_M)}=\frac{D}{L}\Leftrightarrow a+bc_M(a,b)=\log\left(\frac{D}{L}\right) \tag{7}$$

The cut-off $c_M(a,b)$ with this property maximises the expected profit per period $E(a,b,c)$.

$c_M(a,b)$ is often chosen as the cut-off score (or c_R value) after readjustment. However this is only optimal if the subsequent dynamics of the scorecard means the value of the log odds curve does not change at the cut-off score c_R. Otherwise it may be better to anticipate the likely changes when setting the cut-off. The following corollary and counter example display these results.

Lemma 1. Suppose the dynamics of the scorecards is such that the parameters, *a(t)* and *b(t),* change only under the condition the profit maximizing cut-off score ($c_M(a(t),b(t))$) remains unchanged in the subsequent time periods. Under such an assumption, the optimal cut-off score when rebuilding is $c_R(a,b)= c_M(a,b)$.`

Assume a scorecard is readjusted when the score to log odds parameters are *a* and *b*. If the subsequent dynamics of the scorecard is such that the transition probabilities satisfy $p(a(t+1),b(t+1)\,|\,a(t),b(t))=0$ unless $a(t+1)+b(t+1)c_M(a,b)=a+bc_M(a,b)$ then the optimal readjustment of the cut-off is $c_R(a,b)=c_M(a,b)$.

Proof.

Given the conditions in the corollary, the log odds to score graph continues to pass through the point $(c_M(a,b), \log(D/L))$, so at time t the scorecard must be in a state $(a(t),b(t))$ where

$$a(t)+b(t)c_M(a,b)=a+bc_M(a,b)=\log\left(\frac{D}{L}\right) \qquad (8)$$

Hence from (7) $c_M(a,b)$ maximises $E(a(t),b(t),c)$ for all such states to which the scorecard can move. The total expected profit between this readjustment and the next readjustment is the expectation of the weighted sum of terms of the form $E(a(t),b(t),c)$ as t varies over time. Since maximises every one of the possible terms in this expectation, it must maximise the expectation. Hence $c_R(a,b)=c_M(a,b)$.

It is easy though to construct cases where because the scorecard is changing over time, the cut-off that maximises the expected profit in the first period is not optimal over a future horizon. It is better to have a cut-off score that anticipates these changes. A simple case of this is when $D=L=1$, $N=200$, $R=10$, $\beta=0.999999$ (so it is essentially 1 and we can ignore it); let 50% of the population have scores of -1 and 50% have scores of +1, and currently $a=0$ and $b=1$. Then $c_M(0,1)=0$. We assume that the slope of the log odds curve remains 1 and that the intercept goes up by 1 each period so that $p(a+1,1\,|\,a,1)=1$ and $p(x,y\,|\,a,1)=0$ for any other x, y. It is easy to show that a cut-off score of -2 which accepted all the portfolio is more profitable than $c_M(0,1)=0$.

We now show that given reasonable conditions on the dynamics of the scorecard, the rebuilding decision is a limit policy, that is one rebuilds the scorecard if *a* (or *b*) go below some limits. To do this we develop properties of the value function using the standard approach in Markov decision processes of induction on the iterates of value iteration. In this model the iterates of value iteration satisfy

$$V_{n+1}(a,b,c) = \max \begin{cases} E(Q(a,b,c)) + \int \beta V_n(x,y,c)p(x,y \mid a,b)dxdy \\ \max_{c_R}(-R + V_n(a,b,c_R(a,b)) \\ \max_{c_B}(-B + V_n(a_B,b_B,c_B) \end{cases}$$

$$V_0(a,b,c) \equiv 0$$

(9)

Lemma 2. $V_n\,(a,b,c)$ converges to the optimal value function $V(a,b,c)$ which satisfies the optimality equation (6).

Proof. The proof is the standard one for convergent Markov decision processes (Puterman, 1994).

The reasonable assumption about the dynamics of the score to log odds graph is as follows.

Assumption A. There is a stochastic ordering property of the graph in both the a and b dimensions. |Concisely if $\bar{P}(x,y|a,b) = \int_y^\infty dw \int_x^\infty dv p(v,w|a,b)$ and $a \geq \acute{a}$ and $b \geq \acute{b}$, then $\bar{P}(x,y|a,b) \geq \bar{P}(x,y|\acute{a},b)$ and $\bar{P}(x,y|a,b) \geq \bar{P}(x,y|a,\acute{b})$. (10)

So if the graph has a higher a value (or b value) this period, it is likely to have a higher a value (or b value) next period. This assumption is reasonable, since for parameter a it says if one portfolio has a lower risk now than another then next period it will continue to have a lower risk. For parameter b it says if a scorecard has a larger difference in good:bad odds per unit score now than another scorecard that larger difference will still be the case next period. With these properties we can show the monotonicity of the value function in a and b.

Lemma 3. If Assumption A holds then

i) $V(a,b,c)$ is non decreasing in a;

ii) $V(a,b,c)$ is non decreasing in b.

Proof.

i) The proof is by induction on the iterates of value iteration. Trivially it is true for $n=0$ since $V_0(a,b,c)=0$ for all a,b,c. Assume the hypothesis holds for V_n. Then in equation (10), the second expression is non-decreasing from the induction hypothesis while the third expression is constant. For the first expression E(Q(a,b,c)) increases as a increases since increasing a increases $P(G|s)$ and decreases P(B|s). Considering $V_n(x,y,c)\bar{P}(x,y|a,b)$ as the integral of its derivative between (0,0)and (∞,∞) gives

$$\int V_n(x,y,c)p(x,y|a,b)dxdy = \int \frac{\delta}{\delta x \delta y} V_n(x,y,c)\bar{P}(x,y|a,b)dxdy + V_n(0,0,c) \qquad (11)$$

The right hand side of (11) is non-deceasing in a since the derivative term is positive because of the induction hypothesis and $\overline{\mathrm{P}}(x,y|a,b)$ is non-decreasing in a because of Assumption A. Hence all three expressions in the maximisation in (10) are increasing in a and so the induction hypothesis is true for n+1. Hence it is true in the limit for $\mathrm{V}(a,b,c)$.

ii) The proof follows exactly that of i).

This Lemma leads to the optimality of a control policy for when to rebuild the scorecard.

Theorem 2. If assumption A holds then there exist function $a^*(b)$ and $b^*(a)$ so that in (a,b,c) the scorecard is rebuilt provided

i) $a \leqslant a^*(b)$ or

ii) $b \leqslant b^*(a)$.

Proof.

Let $a^*(b) = max\{a|V(a,b,c) = max_{c_B}(-B + V(a_B,b_B,c_B))\}$ then if $a \le a^*(b)$, $V(a,b,c) \le V(a^*(b),b,c) = max_{c_B}(-B + V(a_B,b_B,c_B))$ where $V(a,b,c) \le V(a^*(b),b,c)$ follows from lemma 3. Hence one rebuilds the scorecard in state (a,b,c).

The proof for (ii) is similar.

In the next section we derive a model of the dynamics of the score to log odds relationship which satisfies Assumption A and allows us to solve the case study in section 6.

5. Dynamics of the Score to Log odds relationship

We assume that the score log odds relationship will remain approximately linear (i.e. the score is a linear transformation of the log odds) but that the coefficients change over time. At a time *t* after the last scorecard rebuild the relationship is

$$a(t) + b(t)s(\mathbf{x},t) = \log\left(\frac{P_t(G\,|\,\mathbf{x})}{P_t(B\,|\,\mathbf{x})}\right) \qquad (12)$$

Lenders monitor the performance of scorecards by plotting such score to log odds graphs but there has been no published analysis of how the relationship changes over time. Our empirical investigations suggests that the simplest reasonable model would be to assume that *a(t)* and *b(t)* both follow Brownian motion around a linear trend. Figure 1 shows the plots of *a(t)* and *b(t)* over a 24 month period for the portfolio we will use in section 6. These plots support the idea that the relationship is Brownian motion around a linear trend

[Figure 1 about here]

A linear dynamic model is also suggested in Spader and Quercia (2009) and Sousa et al (2013). The negative correlation in the Brownian motion is also supported by the data and suggests that the score to log odds relationship is more stable close to the mean portfolio scores than at the extremes where there are usually fewer cases.

One could argue that a piece-wise linear approximation would give a closer fit to the score-log-odds graph. However this would lead to a much more complicated state space and non-stationary optimal policies which are difficult to implement. The linear approximation describes the critical features of the relationship between score and log odds.

In continuous time the dynamics of the parameters is given by

$$\begin{aligned} da(t) &= \mu_a dt + \sigma_a dw_1 \\ db(t) &= \mu_b dt + \sigma_b dw_2 \end{aligned} \tag{13}$$

However since the score to log odds relationship is only measured every month, the process is a discrete time one, namely

$$\begin{aligned} a(t) &= a(t-1) + \mu_a + \sigma_a W_1 \\ b(t) &= b(t-1) + \mu_b + \sigma_b W_2 \end{aligned} \tag{14}$$

where the W_i are discrete time equivalents of Brownian motion which only take values +1 and -1 with $E\{W_i\}=0,\ Var\{W_i\}=1,$ and their correlation coefficient is $\rho(W_1,W_2)=E\{W_1W_2\}=\rho$ .These conditions are satisfied by the joint probability distributions

$$P\{W_1=+1,W_2=+1\}=P\{W_1=-1,W_2=-1\}=\frac{1+\rho}{4}$$

$$P\{W_1=+1,W_2=-1\}=P\{W_1=-1,W_2=+1\}=\frac{1-\rho}{4}$$

This satisfies Assumption A. The state space (*a,b,c*) can be simplified to a countable state space. Let *g(a)* be the greatest common denominator of μ_a and σ_a so $\mu_a = k_a g(a)$ and $\sigma_a = l_a g(a)$ then $a(t)=a(t-1)+(k_a \pm l_a)g(a)$. Thus we can identify *a(t)* with the integer *n(t)* where $a(t)=a_B+n(t)g(a)$. Similarly if $g(b)=\gcd(\mu_b,\sigma_b)$, $\mu_b=k_b g(b)$ and $\sigma_b=l_b g(b)$, then $b(t)=b(t-1)+(k_b \pm l_b)g(b)$ and hence we can identify $b(t)$ with $m(t)$ where $b(t)=b_B+m(t)g(b)$.

There are only a finite number of cut-off score values *c* to consider since a scorecard only produces a finite number of scores. One can translate the scorecard states *(a,b,c)* into states *(n,m,c)* where

$a = a_B + (n - n_B)g(a)$ and $b = b_B + (m - m_B)g(b)$. If one defines the corresponding optimal expected profit of operating, rebuilding and readjusting the scorecard as *v(n,m,c)* rather than *V(a,b,c)*, then the optimality equation (6) becomes

$$v(n,m,c) = \max \begin{cases} E(Q(n,m,c)) + \beta \begin{pmatrix} ((1+\rho)/4)(v(n+k_a+l_a, m+k_b+l_b, c) + v(n+k_a-l_a, m+k_b-l_b, c) + \\ ((1-\rho)/4)(v(n+k_a-l_a, m+k_b+l_b, c) + v(n+k_a+l_a, m+k_b-l_b, c) \end{pmatrix} \\ \max_{c_R}(-R + v(n,m,c_R(n,m)) \\ \max_{a_B}(-B + v(n_B, m_B, c_B) \end{cases} \qquad (15)$$

This dynamic programming still has an infinite set of states. However a standard approximation technique (Puterman, 1994) to obtain a finite state space is to impose absorbing barriers on the state space. If one wishes to restrict the state space to $(n,m,c), N_L \le n \le N_U, M_L \le m \le M_U$ then the first term on the RHS of (15) becomes

$$E(Q(n,m,c)) + \beta \begin{pmatrix} ((1+\rho)/4)v(\min(n+k_a+l_a, N_U), \min(m+k_b+l_b, M_U), c) + \\ ((1+\rho)/4)v(\max(n+k_a-l_a, N_L), \max(m+k_b-l_b, M_L), c) + \\ ((1-\rho)/4)(v(\max(n+k_a-l_a, N_L), \min(m+k_b+l_b, M_U), c) + \\ ((1-\rho)/4)v(\min(n+k_a+l_a, N_U), \max(m+k_b-l_b, M_L), c) \end{pmatrix} \qquad (16)$$

This gives a finite state space so we can now solve a real example numerically.

6. Case Study

The behavioural scores and the good:bad status twelve months later was obtained for a sample of over two million credit card accounts for a two year period from January 2002 until December 2003. Although this is only one data set it is quite a substantial one. The model in our paper can be applied to other data sets. The results on the form of the optimal policy would continue to apply even if the details of the optimal policy will be different.

In this sample the scorecard took values in the range 34 to 307. The score range was split into 40 bands, with boundary values considered to be the cut offs *c=1,2,..,40*. The distribution of scores over the 24 months of the sample period in these 40 bands is given by Table 1 A system stability index test showed the score distribution at the end of the period was not significantly different from the one at the start of the period.

[Table 1 about here]

The log odds to score graph was derived each month *t=1,…,24* by plotting for each scoreband the log(number of Goods in band/Number of Bads in band*)* against the mean score of the band. The linear regression line was then fitted to the 40 points on the score to log odds graph . This gave the a(t) and b(t) values for that month t.

[Table 2 about here]

Table 2 shows the estimates of *a(t)* (intercept of the log odds score) and *b(t)* (slope of the log odds score) at each time period. Recall Figure 1 shows the trends in *a(t)* and *b(t)*. Since *b(t)* is decreasing over time, this means the scorecard is degrading and becoming less efficient. .

The data in Table 2 can be used to estimate the parameters $\mu_a, \sigma_a, \mu_b, \sigma_b, \rho$ needed for the dynamics of the state of the system. $\mu_a = 0.175, \mu_b = -0.000625$ are obtained from a linear regression of the values of the slope and the intercept against the time period.rs $\sigma_a = 0.001875, \mu_b = 0.525, \rho = 0.972$ are the standard deviation and variances and covariance of the slope and intercept values. In order to solve a finite state space problem we approximate these values in the discrete space dynamic programme by

$$g(a) = 0.35, l_a = 0.5, k_a = 1.5, g(b) = -0.00125, l_b = 0.5, k_b = -1.5$$

This means the movement in the a and b values between different periods are multiples of 0.35 and 0.00125 respectively and so we define the finite state space in terms of parameters n and m where

$$a = 2.3 + 0.35(n-11), 0 \le n \le 21; b = 0.024 - 0.00125(m-17), 0 \le m \le 25$$

If the scorecard is rebuilt we choose the *a* and *b* values that were close to the actual values when it was rebuilt, namely a_R=2.3, b_R=0.024. These translate to *m*=17 and *n*=11.

[Table 3 about here]

The optimal policy is calculated using this discretization of the state and action spaces. Table 3 displays the results for 21 different problems. They vary in their values of *D*, *L* and *B* values keeping R=1. For example, in scenarios 1a, we assume the debt incurred is *D*=250, the lost profit is *L*=5, the cost of rebuilding a scorecard is *B*=30 and the cost of readjusting a scorecard is *R*=1. Column 2 to column 4 in Table 3 listed the details of these scenarios. We use Matlab to compute the results and the numbers of iterations of value iteration needed to solve the scenario are listed in column 4. For all problems, it takes less than 10 minutes for the dynamic programming computations of (15) and (16). Details of the optimal policy are shown in Table 3. Column 6 to 8 list the percentage of states where the optimal policy for that scenario requires scorecard readjustment, scorecard rebuilding or doing nothing.

Comparing the results of problems 1a and 2a, if the cost of rebuilding the scorecard is not very much higher than the cost of readjusting the scorecard (i.e. problem 1a, *B:R=30:1*), 28% states need to rebuild the scorecard. On the contrary, if the cost of rebuilding the scorecard is high (i.e. scenario 1c), only 13.9% requires rebuilding the scorecard.

[Table 4 about here]

Table 4 shows the optimal policies of problems 1a, 2a and 3a where the cut-off is $c=3$. The code "B" means the optimal policy is to rebuild the scorecard; the code "K" means keeping the current cut-off unchanged. If the model suggests changing the cut-off, the optimal value is listed in the table. For example, the optimal cut-off for n=5 and m=3 is level 1, which corresponds to accepting everyone.

The results show that when *m* increases (i.e. the slope *b* decrease), the model recommends rebuilding the scorecard. As the log odds graph flattens, there is no way to improve the discriminating power of the scorecard but to rebuild it. This applies when it is costly to rebuild the scorecard (i.e. scenario 1c) and when it is relatively cheap to rebuild the scorecard (i.e. scenario 1a). Since the expected loss is relatively low in these three scenarios, in many states, the optimal policy is to reduce the cut-off to 1, which corresponds to accepting everyone.

[Tables 5 and 6 about here]

Table 5 shows three extreme cases. For scenarios 7a, 7b and 7c, the expected loss is far higher than the expected profit (where *D/L*=5000). So in the majority of the states in these three scenarios, the optimal policy is to increase the cut-off, i.e. decrease the population being accepted, or to rebuild the scorecard. If the default rate of the overall population is low (i.e. *n* is high), the optimal policies are either to keep the cut-off unchanged or to reduce the cut-off to 2. Overall this leads to a much more conservative risk profile.

If we compare Tables 5 and 6, where the only difference is in what the current cut-off level is (c=3 in Table5, c=8 in Table 6), the optimal policy only differs when one is readjusting the cut-off in one case but doing nothing in the other. So in Scenario 7a of Table 5, in the state $m=9$ and $n=5$, one keeps the cut-off at level 8 if it is there already (as in Table 6). It moves the cut-off to level 9 if it is already at level 3 as in Table 5.

The examples confirm that rebuilding follows a control limit form in that for any point with rebuilding as the decision all the decisions to the right and above it are also rebuild. Note that taking no action is only optimal in about 2% of the states in any scenario, while readjusting is much more common.

7. Conclusion

Some lenders monitor their scorecards by drawing the score to log odds graph. This paper shows how the parameters of the linear approximation (*a,b*), of this graph together with the cut-off score *c*, is a concise way to describe the discrimination of the scorecard. It allows dynamic programming models to be developed which optimise when scorecards should be rebuilt or readjusted. The dynamic programming model shows that the current way of readjusting after a rebuild is suboptimal in that it does not allow for the likely future dynamics of the scorecard. It also proves that the optimal rebuilding policy is a control limit both in the intercept and the slope of the log-odds score graph.

We apply this model to a data set of over two million credit cards to determine when the scorecard should be rebuilt and when it needs readjusting. Even when the cost of readjusting is made high-probably higher than the cost in a real situation - the optimal policy suggests lenders should readjust much more often than they do in practice. Moreover one can simulate the optimal policy to find out how frequently the scorecard needs rebuilding. This will give management an indication of the size of the analytics team they will need for rebuilding all their scorecards. Thus the model is useful for determining the logistic support needed to keep a scorecard discriminating well.

Although we have described the model in terms of a credit scorecard, the same approach can be applied problems and other scorecards. For example some lenders have three possible actions they can undertale on a new applicant for credit. Accept, Review and Reject. There is a cost for reviewing but a review changes the probability of a customer with a particular score being Good or Bad. The decision then involves two cut-off scores c_1 and c_2. and the expected cost $E(Q(a,b,c_1,c_2)$ is the sum of the costs under the three decisions. The need to rebuild and readjust scorecards occurs in any area where scorecards are in use, O this approach can be useful in propensity to purchase scorecards, churn scorecards, as well as scorecards in health care and the legal system.

References

Anderson R (2007). *The Credit Scoring Toolkit Theory and Practice for Retail Credit Risk Management and Decision Automation*. Oxford University Press, Oxford.

Arts D G T, de Keizer N F, Vroom M B and de Jonge E (2005). Reliability and accuracy of Sequential Organ Failure Assessment (SOFA) scoring. *Critical Care Medicine* **33**: 1988-1993.

Ashcraft A B and Schuermann T (2008). Understanding the securitization of subprime mortgage credit. Federal Reserve Bank of New York Staff Report no 318.

Bertsekas D P (2012). *Dynamic Programming and Optimal Control*. Athena Scientific, Nashua.

Bauerle N., Rieder U, (2011), Markov decision Processes with Applications in Finance, Springer,Heidelberg.

Burez J and van den Poel D (2009). Handling class imbalance in customer churn predictions. *Expert Systems with Applications* **36**: 4626-4636.

Crook J N, Thomas L C and Hamilton R (1992). The degradation of the scorecard over the business cycle. *IMA Journal of Mathematics in Business and Industry* **4**: 111-123.

Demyanyk Y and van Hemert O (2008). Understanding the subprime mortgage crisis. Proceedings of Federal Reserve Bank of Chicago, May 2008, 171-192.

Denardo E V (1982). *Dynamic Programming, Models and Applications*. Prentice-Hall, Englewood Cliffs.

Derman C (1963). On optimal replacement rules when changes of state are Markovian. In Mathematical Optimization Techniques, ed. R. Bellmann, University of California Press, Berkeley, 201-212.

Dohi T and Nakagawa N (2013). *Stochastic Reliability and Maintenance Modelling*. Springer, Heidleberg.

Elwany A H, Gebraeel N Z and Maillart L M (2011). Structured replacement policies for components with complex degredation processes and dedicated sensors. *Operations Research* **59**: 684-695.

Feng Q, Mookerjee V S and Sethi S P (2006). Optimal policies for the sizing and timing of software maintenance projects. *European Journal of Operational Research* **173**: 1047-1066.

Finlay S (2008). *The Management of Consumer Credit, Theory and Practice*. Palgrave Macmillan, Basingstoke.

Glady N, Baesens B and Croux C (2009). Modeling churn using customer lifetime value. *European Journal of Operational research* **197:** 402-411.

Grottke M, Matias R and Trivedi K S (2008). The fundamentals of software ageing. Proceedings 1st International Workshop on Software Aging and Rejuvenation, 1-6.

Krishana M S, Mukhopadhyay T and Krebel C H (2004). A decision model for software maintenance. *Information Systems Research.* **15**: 396-412.

Kulkarni V G, Kumar S, Mookerjee V S and Sethi S P (2009). Optimal allocation of effort to software maintenance: a queuing theory approach. *Production and Operations Management* **18**: 506-515.

Liao H, Elsayed E A and Chan L-Y (2006). Maintenance of continuously monitored degrading systems. *European Journal of Operational Research* **175**: 821-835.

Lui K-G, Makis V and Jardine A K S (1995). A replacement model with overhauls and repair. *Naval Research Logistics* **42**: 1063-1079.

Malthouse E C (2001). Assessing the performance of direct marketing scoring models. *Journal of Interactive Marketing* **15:** 49-62.

McCall J J (1965). Maintenance policies for stochastically failing equipment: A survey. *Management Science* **11**(5): 493-524.

McNab H and Wynn A (2000). *Principles and Practice of Consumer Credit Risk Management*. CIB Publishing, Canterbury.

Petersilia J (2009). *When Prisoners Come Home, Parole and Prisoner Reentry*. Oxford University Press, Oxford.

Pierskalla W P and Voelker J A (1976). A survey of maintenance models: the control and surveillance of deteriorating systems. *Naval Research Logistics Quarterly* **23**: 353-388.

Puterman M (1994). *Markov Decision Processes: Discrete Stochastic Dynamic Programming*. John Wiley, New York.

Rona–Tas A and Hiss S (2008). Consumer and Corporate Credit Ratings and the Subprime mortgage crisis in the U.S. with some lessons for Germany, SCHUFA, Weisbaden.

Sarkar A, Panja S C and Sarka B (2011). Survey of maintenance policies for the last 50 years. *International Journal of Software Engineering and Applications* **2**: 130-148.

Sherif Y S and Smith M L (1981). Optimal maintenance models for systems subject to failure-A review. *Naval Research Logistics Quarterly* **28**(1): 47-74.

Spader J., Quercia R.G.,(2009) Thriftiness is not enough. Dynamics of consumer credit history scores among low and moderate income home owners, Center for Community Capital, University of North Carolina, Chapel Hill

Sousa M.R., Gama J., Brandao E., (2013), Intriducing time-changing economics into credit scoring, FEB Working Paper 513, PEP-UP, University of Porto

Thomas L C (1986). A survey of maintenance and replacement models for maintainability and reliability of multi-item systems. *Reliability Engineering and System Safety* **16**(4): 297-309.

Thomas L C (2009). *Consumer credit models: pricing, profit and portfolios*. Oxford University Press, Oxford, UK.

US Securities and Exchange Commission (2008). Summary report on issues identified in the Commission Staff's Examination of Select Credit Rating Agencies by the Staff of the Securities and Exchange Commission, July 2008.

Valdez-Flores C and Feldman R M (1986). A survey of preventative maintenance models for stochastically deteriorating single-unit systems. *Naval Research Logistics* **36**: 419-446.

Varga G (2009). Investment decision in a new credit score system. FCE Consulting, Brazil.

Wang H (2002). A survey of maintenance policies of deteriorating systems. *European Journal of Operational Research* **139**: 469-489.

White D J (1969), Dynamic Programming, Holden-Day, San Francisco.

White D J (1988). Further real applications of Markov Decision Process. *Interfaces* **18**: 55-61.

White D J (1993). A survey of applications of Markov Decision Processes. *Journal of the Operational Research* **44**: 1073-1096.

Xiong C-J, Xie M and Ng S-H (2011). Optimal software maintenance policy considering unavailable time. *Journal of Software Maintenance and Evolution Research and Practice* **23**: 21-33.

Table 1. Distribution of scores

Band	Scoreband Lower limit	Scoreband Upper limit	% in Band	Band	Scoreband Lower limit	Scoreband Upper limit	% in Band
1	34	144	1.71	21	250	252	2.47
2	145	158	1.85	22	253	255	2.40
3	159	168	2.04	23	256	257	1.77
4	169	176	2.12	24	258	260	3.23
5	177	182	1.94	25	261	261	1.23
6	183	188	2.18	26	262	263	3.07
7	189	193	1.98	27	264	266	3.18
8	194	197	1.59	28	267	268	2.68
9	198	203	2.60	29	269	271	3.67
10	204	206	1.35	30	272	272	2.13
11	207	212	2.97	31	273	273	2.47
12	213	217	2.35	32	274	274	0.99
13	218	222	2.92	33	275	277	3.35
14	223	227	2.88	34	278	279	0.86
15	228	230	1.58	35	280	280	1.73
16	231	235	2.78	36	281	282	3.90
17	236	239	2.19	37	283	285	3.73
18	240	243	2.64	38	286	289	4.55
19	244	247	2.75	39	290	294	3.79
20	248	249	1.35	40	295	307	5.03

Table 2. Estimates of a(t) and b(t) at each time period

Time Period	a(t):Intercept	b(t):Slope	Time Period	a(t):Intercept	b(t):Slope
1 (JAN 2002)	-1.4715	0.0269	13 (JAN 2003)	2.0228	0.0136
2 (FEB 2002)	-0.8866	0.0253	14 (FEB 2003)	2.4301	0.0132
3 (MAR 2002)	-0.8567	0.0235	15 (MAR 2003)	2.1875	0.0129
4 (APR 2002)	0.8989	0.0185	16 (APR 2003)	2.4029	0.0137
5 (MAY 2002)	0.7725	0.0176	17 (MAY 2003)	2.2516	0.0129
6 (JUN 2002)	1.2673	0.0168	18 (JUN 2003)	2.9714	0.011
7 (JUL 2002)	0.9466	0.0167	19 (JUL 2003)	2.8581	0.01
8 (AUG 2002)	1.7658	0.0164	20 (AUG 2003)	3.1495	0.0145
9 (SEP 2002)	1.3969	0.0167	21 (SEP 2003)	2.8273	0.00951
10 (OCT 2002)	2.0485	0.0153	22 (OCT 2003)	3.3631	0.00948
11 (NOV 2002)	1.7942	0.0148	23 (NOV 2003)	3.0513	0.00957
12 (DEC 2002)	2.3302	0.0138	24 (DEC 2003)	3.1424	0.0103

Figure 1. Dynamics of the slope and intercept for the log odds to score graph

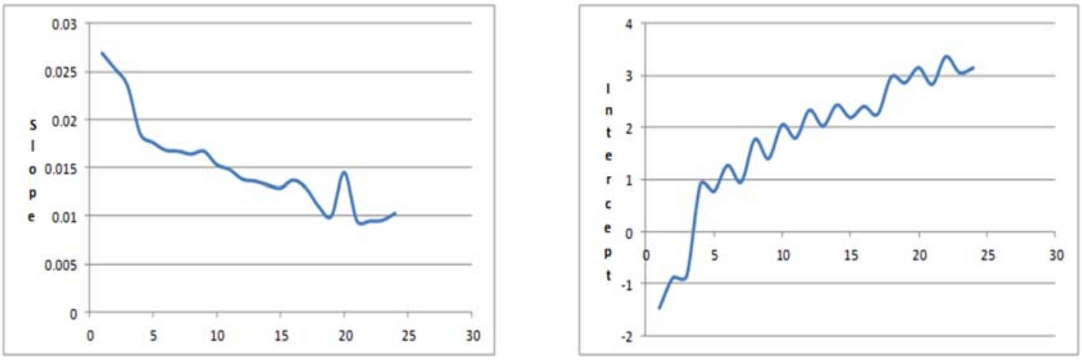

Table 3. Scenarios setting and the corresponding results

Scenarios	D/R	L/R	B/R	No. of Iterations	Change the cut-off (%)	Rebuild the scorecard (%)	No action (%)
1a	250	5	30	690	69.8	28.4	1.8
2a	400	5	30	692	69.2	28.9	1.8
3a	500	5	30	694	68.9	29.3	1.8
4a	750	5	30	696	68.1	30.1	1.8
5a	1000	5	30	698	67.0	31.2	1.7
6a	2500	5	30	720	67.7	30.5	1.8
7a	5000	5	30	731	66.3	31.8	1.9
1b	250	5	90	693	78.9	19.0	2.0
2b	400	5	90	693	77.4	20.6	2.0
3b	500	5	90	695	76.3	21.7	2.0
4b	750	5	90	696	74.8	23.2	2.0
5b	1000	5	90	698	73.9	24.2	1.9
6b	2500	5	90	720	72.1	25.9	2.0
7b	5000	5	90	732	69.9	28.0	2.1
1c	250	5	180	693	83.9	13.9	2.1
2c	400	5	180	693	81.4	16.4	2.2
3c	500	5	180	695	80.6	17.3	2.1
4c	750	5	180	696	78.3	19.6	2.1
5c	1000	5	180	699	77.6	20.4	2.1
6c	2500	5	180	721	74.1	23.8	2.1
7c	5000	5	180	733	72.6	25.1	2.3

Table 4. The optimal policy for Scenario 1a, 1b and 1c where c=3

n													m												
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
													Scenario 1a												
1	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B
2	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B
3	2	2	2	2	2	2	2	2	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B
4	2	2	2	2	2	2	2	2	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B
5	1	1	1	1	1	1	1	1	2	2	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B
6	1	1	1	1	1	1	1	1	2	2	2	2	2	B	B	B	B	B	B	B	B	B	B	B	B
7	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	B	B	B	B	B	B	B	B	B
8	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	B	B	B	B	B	B	B	B	B
9	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	B	B	B	B	B	B	B	B	B
10	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	B	B	B	B	B	B	B
11	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	B
12	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	B
13	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	B
14	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	B
15	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
16	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
17	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
18	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
19	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
20	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
21	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
													Scenario 1b												
1	2	2	2	2	2	2	2	2	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B
2	2	2	2	2	2	2	2	2	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B
3	2	2	2	2	2	2	2	2	2	2	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B
4	2	2	2	2	2	2	2	2	2	2	2	2	2	2	B	B	B	B	B	B	B	B	B	B	B
5	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	B	B	B	B	B	B	B	B	B
6	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	B	B	B	B	B	B	B	B	B
7	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	B	B	B	B	B	B	B	B	B
8	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	2	B	B	B	B	B	B	B
9	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	B	B	B
10	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	B
11	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	B
12	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	B
13	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
14	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
15	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
16	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
17	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
18	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
19	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
20	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
21	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
													Scenario 1c												
1	2	2	2	2	2	2	2	2	2	K	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B
2	2	2	2	2	2	2	2	2	2	2	2	2	B	B	B	B	B	B	B	B	B	B	B	B	B
3	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	B	B	B	B	B	B	B	B	B
4	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	B	B	B	B	B	B	B	B	B
5	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	B	B	B	B	B	B	B	B	B
6	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2	B	B	B	B	B	B	B	B
7	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	2	2	B	B	B	B	B	B
8	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	B
9	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	B
10	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	B
11	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	B
12	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
13	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
14	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
15	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
16	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
17	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
18	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
19	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
20	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
21	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Table 5. The optimal policy for Scenario 7a, 7b and 7c where c=3.

n	m																								
	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25
	Scenario 7a																								
1	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B
2	5	5	5	5	5	5	5	5	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B
3	4	4	4	4	4	4	4	4	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B
4	K	K	K	K	K	K	K	K	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B
5	K	K	K	K	K	K	K	K	9	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B
6	2	2	2	2	2	2	2	2	7	7	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B
7	2	2	2	2	2	2	2	2	5	5	5	5	5	5	5	B	B	B	B	B	B	B	B	B	B
8	2	2	2	2	2	2	2	2	4	4	4	4	4	4	4	4	B	B	B	B	B	B	B	B	B
9	2	2	2	2	2	2	2	2	K	K	K	K	K	K	K	K	B	B	B	B	B	B	B	B	B
10	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	B	B	B	B	B	B	B	B	B
11	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	8	B	B	B	B	B	B	B	B
12	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	5	5	B	B	B	B	B	B	B
13	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	K	K	K	K	K	B	B	B	B
14	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	K	K	K	K	2	2	2	2	B
15	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	B
16	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	B
17	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	B
18	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	1	K
19	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2
20	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2
21	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	Scenario 7b																								
1	6	6	6	6	6	6	6	6	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B
2	5	5	5	5	5	5	5	5	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B
3	4	4	4	4	4	4	4	4	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B
4	K	K	K	K	K	K	K	K	11	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B
5	K	K	K	K	K	K	K	K	9	9	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B
6	2	2	2	2	2	2	2	2	7	7	7	7	B	B	B	B	B	B	B	B	B	B	B	B	B
7	2	2	2	2	2	2	2	2	5	5	5	5	5	5	5	5	B	B	B	B	B	B	B	B	B
8	2	2	2	2	2	2	2	2	4	4	4	4	4	4	4	4	B	B	B	B	B	B	B	B	B
9	2	2	2	2	2	2	2	2	K	K	K	K	K	K	K	K	B	B	B	B	B	B	B	B	B
10	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	11	B	B	B	B	B	B	B	B
11	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	8	8	B	B	B	B	B	B	B
12	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	5	5	5	5	B	B	B	B	B
13	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	K	K	K	K	K	K	K	K	B
14	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	K	K	K	K	K	K	K	2	B
15	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	B
16	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	B
17	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	B
18	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	1	K
19	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2
20	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2
21	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	Scenario 7c																								
1	6	6	6	6	6	6	6	6	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B
2	5	5	5	5	5	5	5	5	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B
3	4	4	4	4	4	4	4	4	13	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B
4	K	K	K	K	K	K	K	K	11	11	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B
5	K	K	K	K	K	K	K	K	9	9	9	9	B	B	B	B	B	B	B	B	B	B	B	B	B
6	2	2	2	2	2	2	2	2	7	7	7	7	7	7	7	7	B	B	B	B	B	B	B	B	B
7	2	2	2	2	2	2	2	2	5	5	5	5	5	5	5	5	B	B	B	B	B	B	B	B	B
8	2	2	2	2	2	2	2	2	4	4	4	4	4	4	4	4	B	B	B	B	B	B	B	B	B
9	2	2	2	2	2	2	2	2	K	K	K	K	K	K	K	K	14	B	B	B	B	B	B	B	B
10	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	11	11	B	B	B	B	B	B	B
11	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	8	8	8	B	B	B	B	B	B
12	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	5	5	5	5	5	5	5	B	B
13	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	K	K	K	K	K	K	K	K	B
14	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	K	K	K	K	K	K	K	2	B
15	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	B
16	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	B
17	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	5
18	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	1	K
19	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2
20	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2
21	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Table 6. The optimal policy for Scenario 7a, 7b and 7c where c=8.

n	m																								
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**
	Scenario 7a																								
1	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B
2	5	5	5	5	5	5	5	5	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B
3	4	4	4	4	4	4	4	4	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B
4	3	3	3	3	3	3	3	3	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B
5	3	3	3	3	3	3	3	2	K	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B
6	2	2	2	2	2	2	2	2	7	7	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B
7	2	2	2	2	2	2	2	2	5	5	5	5	5	5	5	B	B	B	B	B	B	B	B	B	B
8	2	2	2	2	2	2	2	2	4	4	4	4	4	4	4	4	B	B	B	B	B	B	B	B	B
9	2	2	2	2	2	2	2	2	3	3	3	3	3	3	3	3	B	B	B	B	B	B	B	B	B
10	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	B	B	B	B	B	B	B	B	B
11	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	K	B	B	B	B	B	B	B	B
12	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	5	5	B	B	B	B	B	B	B
13	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	3	3	3	3	3	B	B	B	B
14	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	B
15	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	B
16	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	B
17	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	B
18	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	1	3
19	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2
20	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2
21	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	Scenario 7b																								
1	6	6	6	6	6	6	6	6	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B
2	5	5	5	5	5	5	5	5	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B
3	4	4	4	4	4	4	4	4	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B
4	3	3	3	3	3	3	3	3	11	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B
5	3	3	3	3	3	3	3	2	K	K	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B
6	2	2	2	2	2	2	2	2	7	7	7	7	B	B	B	B	B	B	B	B	B	B	B	B	B
7	2	2	2	2	2	2	2	2	5	5	5	5	5	5	5	5	B	B	B	B	B	B	B	B	B
8	2	2	2	2	2	2	2	2	4	4	4	4	4	4	4	4	B	B	B	B	B	B	B	B	B
9	2	2	2	2	2	2	2	2	3	3	3	3	3	3	3	3	B	B	B	B	B	B	B	B	B
10	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	11	B	B	B	B	B	B	B	B
11	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	K	K	B	B	B	B	B	B	B
12	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	5	5	5	5	B	B	B	B	B
13	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	3	3	3	3	3	3	3	3	B
14	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	B
15	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	B
16	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	B
17	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	B
18	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	1	3
19	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2
20	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2
21	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
	Scenario 7c																								
1	6	6	6	6	6	6	6	6	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B
2	5	5	5	5	5	5	5	5	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B
3	4	4	4	4	4	4	4	4	13	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B
4	3	3	3	3	3	3	3	3	11	11	B	B	B	B	B	B	B	B	B	B	B	B	B	B	B
5	3	3	3	3	3	3	3	2	K	K	K	K	B	B	B	B	B	B	B	B	B	B	B	B	B
6	2	2	2	2	2	2	2	2	7	7	7	7	7	7	7	7	B	B	B	B	B	B	B	B	B
7	2	2	2	2	2	2	2	2	5	5	5	5	5	5	5	5	B	B	B	B	B	B	B	B	B
8	2	2	2	2	2	2	2	2	4	4	4	4	4	4	4	4	B	B	B	B	B	B	B	B	B
9	2	2	2	2	2	2	2	2	3	3	3	3	3	3	3	3	14	B	B	B	B	B	B	B	B
10	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	11	11	B	B	B	B	B	B	B
11	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	K	K	K	B	B	B	B	B	B
12	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	5	5	5	5	5	5	5	B	B
13	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	3	3	3	3	3	3	3	3	B
14	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	B
15	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2	B
16	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	B
17	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	2	5
18	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	2	2	2	2	2	2	1	3
19	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2
20	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2
21	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1