ESTIMATION OF TRANSITION PROBABILITY MATRICES IN CREDIT RISK ANALYSIS

by

Laurent El Ghaoui, Maksim Oks and Ankur Varma

Memorandum No. UCB/ERL M00/60

27 July 2000

ESTIMATION OF TRANSITION PROBABILITY MATRICES IN CREDIT RISK ANALYSIS

by

Laurent El Ghaoui, Maksim Oks and Ankur Varma

Memorandum No. UCB/ERL M00/60

27 July 2000

ELECTRONICS RESEARCH LABORATORY

College of Engineering
University of California, Berkeley
94720

Estimation of Transition Probability Matrices in Credit Risk Analysis*

Laurent El Ghaoui[†], Maksim Oks[‡], Ankur Varma[§]

July 27, 2000

Abstract

In this report we discuss the problem of estimation of the single-period (month, year, etc) transition probability matrix of a credit rating from historical rating data. In a first approach, we assume that the data is not subject to errors, and that the process is stationary. We derive an efficient numerical procedure based on a convex, constrained least-squares problem, and show how to account for sparse historical data (rare events) in this setting. In the second part, we address the issues of non-stationarity of the process (including business cycles) and uncertainty in the data.

*Research supported by a grant from BNP-Paribas Group, DREI, RECORD.

[†]EECS Department, 267 M Cory Hall, University of California at Berkeley, CA 94720-1770, USA. Internet: elghaoui@eecs.berkeley.edu

[‡]IEOR Department, UC Berkeley.

[§]EECS Department, UC Berkeley.

Contents

1 Introduction

1.1 The estimation problem

Credit rating companies such as Moody's and Standard & Poor's have been providing ratings to the investors on the companies that hold debt. The credit rating is basically the "opinion of the future ability, legal obligation, and willingness of a bond issuer or other obligor to make full and timely payments on principal and interest due to investors" [1]. The companies are rated using rating symbols (Aaa-through-C for Moody's and AAA-through-C for Standard & Poor's). The "higher" the rating, the better is the credit quality in the opinion of the rating company, with the "triple A" representing the highest credit quality. The lowest rating, C, indicates the lowest credit quality, meaning that there's high risk of a default, i.e a non-payment or a missed payment. (For a complete set of definitions of each rating category, see for example "Moody's Rating Definitions" in [1].)

Since the ratings basically reflect the likelihood or probability of a default (in the opinion of the rating company, which is based on extensive economical and statistical studies), an investor might be interested not only in the current rating of a given company but also in the probabilities of it changing the rating to a different state, in particular, to the default state. By viewing the process as a Markov Chain, we are led to the problem of estimating the *transition probability matrix* of ratings.

1.2 Previous approach and resulting challenges

A previous set of results for this problem have been obtained at the DREI of BNP-Paribas group in 1998-9. In the report [13], the authors formulated the estimation problem as an optimization problem, more precisely a constrained least-squares problem. The variable is the transition probability matrix sought; the constraints reflect some prior knowledge or assumptions about the matrix. Based on the historical (estimated) transition to default data vector, and the assumption of stationarity of the underlying process, one can form consistency relations that impose (linear) constraints on the variables. Due to various errors in the data, these overdetermined equations cannot be satisfied exactly. The squared norm objective is used in the optimization problem in an attempt to minimize the discrepancy arising from those errors. The report [13] describes an iterative procedure to solve the resulting constrained least-squares problem. Difficulties with this procedure (lack of convergence) are reported.

The existing approach described above suffers from various shortcomings that were identified early on. These lead to four main challenges that our research will address.

- *Non-stationarity:* the main assumption in the above setting is that the process should be stationary, which translates into a constant transition probability matrix. In practice, this assumption does not hold, for various reasons. For example, business cycles might affect the transition probability matrix. We need to formulate the estimation problem in order to account for non-stationarity, and, especially, for the presence of business cycles.

- *Sparse data:* some transitions are very rarely observed. This means that some of the (estimated) probabilities of transitions to default are subject to large errors. We need to smooth out this effect in order to come up with more reliable estimates.

- *Errors in the data.* The data in the optimization problem are probabilities of transitions to default. As noted above, these are estimated, from historical frequencies of transitions to default. The historical data may result in the estimates containing errors. Even with a "smoothing" process in place (as in the case of rare transitions), we need to take these errors into account when estimating the transition probability matrix. This raises several issues: can we estimate the size, or variance, of the errors? Can we analyze the impact of these errors on the resulting esimate of the transition probability matrix, as well as on some probabilities of future transitions to default?

- *Efficient numerical solutions:* All the problems above need to be solved with efficient and reliable numerical methods, that have the potential to be implemented in a decision-making tool. In particular, some of the caveats of the iterative procedure developed in the earlier report [13] need to be addressed. Thus, our solutions need to rely on algorithms that converge *globally* in an *unsupervised* manner.

1.3 Report overview

This (first) report concentrates on the following aspects.

First, we revisit in section 2 the "basic approach" described in the report [13], and come up with an efficient numerical procedure to address the problem. We also show how to account for sparse data (rare transitions) and smooth out the resulting estimation errors. A numerical example based on the rating history of 2268 companies accross 60 time periods is presented. Related Matlab routines are presented in the Appendix A, while a numerical value of the transition matrix is given in Appendix B.

The rest of the report is devoted to laying the theoretical ground for addressing the more difficult problems of non-stationarity and data errors. In section 3, the non-stationarity problem is formulated as one of computing *optimal bounds* on a time-varying transition probability matrix, that are consistent with the constraints and the consistency relations. We solve this problem using a linear programming technique. In section 4, we examine how errors in the data can be taken into account within the framework of the previous section. The final section outlines the direction of further work, and some challenges that lie ahead.

2 Basic Problem

In this section we describe the mathematical model that we used as the basis for estimating the transition probability matrix. First we introduce some notation and definitions.

2.1 Mathematical model

2.1.1 Notation and definitions

Let s be the number of possible rating cathegories or states $1, \ldots, s$, where the last state s is absorbing. We call this state the **default**. Once a company defaults (enters state s), it is assumed it stays there forever. Let n be the number of periods (months, years, etc.) over which the data of ratings is available.

Let $C_k \in \mathbf{R}^s$ be the vector of probabilities of default over the period of k time units, i.e.

$$C_k = \begin{pmatrix} C_{1k} \\ \vdots \\ C_{sk} \end{pmatrix}$$

where C_{ik} is the probability of a company starting in state i and defaulting over a period of k time units. By our convention that default is an absorbing state it follows that $C_{sk} = 1$ for all $k \in 1 \ldots n$. We also define $C_0 \in \mathbf{R}^s$ to be the vector $(0, \ldots 0, 1)^T$.

We make the assumption that the transition probability matrix $P \in \mathbf{R}^{s \times s}$ is stationary, i.e independent of time and other factors (such as economic cycle, etc).

2.1.2 Consistency relations

Suppose we knew the true vectors $C_k,\ k = 1, \ldots, n$. If the process by which the companies change rating states from one period to the next were truly a stationary markovian process with the transition probability matrix P, the following *consistency* relations would hold:

$$C_k = P^k C_0, \quad k = 1, \ldots, n$$

or, equivalently,

$$C_k = P C_{k-1} \quad k = 1, \ldots, n. \tag{1}$$

We take the second of these relations to be the basis of our mathematical model. It is more convenient for our purposes to write this relation in a different form. Let $x \in \mathbf{R}^{s^2}$ be the vectorized form of P (row by row), i.e.

$$x = (p_{11}, p_{12}, \ldots, p_{1s}, p_{21}, \ldots, p_{ss})^T \in \mathbf{R}^{s^2}$$

Then the relation (1) is equivalent to:

$$Fx = f \tag{2}$$

in which the "unknown" transition probability matrix P is represented by the vector x, and the matrix $F \in \mathbf{R}^{sn \times s^2}$ is given by

$$F = \begin{bmatrix} C_0^T & 0 & \cdots & 0 \\ 0 & C_0^T & \cdots & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & \cdots & C_0^T \\ C_1^T & 0 & \cdots & 0 \\ 0 & C_1^T & \cdots & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & \cdots & C_1^T \\ \vdots & \vdots & \vdots & \vdots \\ C_{n-1}^T & 0 & \cdots & 0 \\ 0 & C_{n-1}^T & \cdots & 0 \\ 0 & 0 & \ddots & 0 \\ 0 & 0 & \cdots & C_{n-1}^T \end{bmatrix} \tag{3}$$

while the vector $f \in \mathbf{R}^{sn}$ is:

$$f = \begin{pmatrix} C_1 \\ C_2 \\ \vdots \\ C_n \end{pmatrix} \tag{4}$$

2.1.3 Hard constraints on the transition matrix

So far we have only described our desire that the matrix P fits the consistency relation (1) or (2). We should also include "hard" constraints requiring P to be a transition probability matrix as well as requiring it to satisfy some of the additional desired properties explained below.

1. $P_{sj} = 0, \; j \neq s$
   $P_{ss} = 1$ — default is an absorbing state.

2. $0 \leq P_{ij} \leq 1, \;\; 1 \leq i, j \leq s$ — entries are probabilities.

3. $\sum_{j=1}^{s} P_{ij} = 1, \quad i \in 1 \ldots s$ — row sums must be 1.

4. $P_{is} \leq P_{i+1,s}, \quad i = 1 \ldots s-1$ — the "lower" the rating, the higher the probability of default.

5. $P_{i,j+1} \leq P_{ij}$ if $1 \leq i \leq j \leq s-1$
   $P_{i,j+1} \geq P_{ij}$ if $1 \leq j < i \leq s-1$

6. $P_{i+1,j} \leq P_{ij}$ if $1 \leq j \leq i \leq s-2$
   $P_{i+1,j} \geq P_{ij}$ if $1 \leq i < j \leq s-1$

In the sequel, we will denote by $\mathcal{P}$ the set of matrices P that satisfy the constraints 1-6.

Constraints 1-3. require P to be a transition probability matrix, while constraints 4-6. reflect the "prior" expectations about the matrix because the states are ordered. Constraints 5. and 6. express the so-called "ordering" properties of the rating states. For example, we wish P to reflect the property that the probability of a company going from state C to state B should be higher than the probability of going from state C to state A, etc. For more on the subject of the desired properties of transition probability matrices, see [6].

In order for the relation (1) or (2) to hold, we must have "true" probabilities of default vectors C_k, $k = 1, \ldots, n$. Now, of course, we don't have such knowledge; instead we estimate the vectors C_k from the historical data (the estimation procedure is described in the following section). Since the vectors C_k that we use are estimates which are likely to contain errors, the relation (2) will no longer hold exactly. So we take it as the "soft" constraint of our model. To be more precise, we try to find the best-fit vector x (matrix P), that satisfies the relation (2) in the least-squares sense. This is equivalent to replacing the right-hand-side vector f in (2) with the vector containing the errors:

$$Fx = f + \Delta f$$

and trying to find x that minimizes the (Euclidean) norm of the error vector Δf. This is a pretty crude approach; it means that we account for the uncertanties in data only in the right-hand-side vector f and not in the matrix F, but we use this as the basis for our model.

To summarize the basic model: we solve the following constrained ordinary least-squares problem.

$$\begin{aligned} min \quad & \| Fx - f \|_2^2 \\ s.t. \quad & \text{Constraints } 1-6. \end{aligned} \tag{5}$$

where x is the vectorized form of P and F and f are as defined in (3) and (4). This is a convex quadratic optimization problem and thus the global optimum can be found in polynomial time. Even for problems of large size (involving many states and many periods) the solution can be found very fast. For more on the computational issues, see Section 2.3.

2.2 Description of the data

The data file we used contained the Moody's ratings of 5890 firms from various industries over a 60 months period. The data file was somewhat incomplete for our purposes; so we dealt with some of the shortcomings in data as follows.

Moody's uses the symbol rating system (Aaa, Aa1, etc., see [1] for a complete description). We have replaced the Moody's class names with the numeric codes: 1=Aaa, 2=Aa1, etc. We have also removed the companies that had either a blank or a NR (not rated) in one of the periods.

2.2.1 Sparsity problem

In the data set that was sent to us, there was significant disbalance in the number of transitions to and from some of the rating classes. The sparsity of the data for some of the classes contributes significantly to the errors in the estimation. To alleviate this problem we have merged a few of the rating classes together. We did this in the following way: we computed the total number of transitions to and from each of the rating classes in successive periods. Then we kept merging the neighbouring classes with the fewest number of transitions together into groups until the number of transitions across groups was in equilibrium. As a result we ended up with a total of 15 classes or states: Aaa and Aa1 have been merged together in one group which we call class 1; and Caa, Ca1, Ca2, Ca3 and C have been merged together in one group which we call class 15.

The biggest problem the data has presented was that there were no instances of firms defaulting. We must have received the rating history data of some "elite" companies, when for our purposes we needed the history of ratings of defaulting companies as well. The mathematical model we use is based on the histories of default, therefore it was imperative that we have some default data. Since we didn't have any histories of default, we had to call our state 15 (Caa, Ca1, etc.) the "default" state. As the default must be an absorbing state, we forced class 15 to be absorbing, i.e. we set the ratings of all the companies that had reached state 15 to stay in state 15 for all the following periods. This represents a serious change, since in practice the company that, for example, is rated Caa may get a higher rating in a later period. In our case we have forced such a company to stay rated Caa.

Having applied all the changes described above to the data, we have ended up with 15 rating states (1,...,15=default) for 2268 companies over a 60 month period. The following describes the procedure we used to estimate the default frequencies.

2.2.2 Estimation of default probabilities

As it has been mentioned in Section 2, our model requires the knowledge or good estimates of $C_k,\ k = 1 \dots n$, the vectors of the probabilities of default over k periods. Since we don't know the true vectors $C_k,\ k = 1, \dots, n$, we compute empirical relative frequencies of default over n periods from the historical data. This is done as follows: Let N_{ik} be the number of companies in the historical data set that started in state i and ended up in default over k time periods. Let NT_i be the total number of companies in the historical data set that started in state i and remained rated over k time periods. Then we estimate C_{ik} to be:

$$C_{ik} = \frac{N_{ik}}{NT_i}, \quad i = 1, \dots, s-1; \ k = 1, \dots, n$$

Note that according to this definintion $N_{i,k+1} \geq N_{ik}$. This is because default is an absorbing state, i.e each time we count the company as "defaulting over k periods" we also count it as "defaulting over periods $k+1,\ k+2, \dots$". See file `frequencies.m` in the Appendix A for the details on the computation of empirical frequencies.

After we have estima ted $C_k,\ k = 0 \dots n$, we use these estimates to construct the matrix F and the vector f for problem (5). Given all the above problems with the data, the transition probability matrix obtained by solving (5) may not be too trustworthy. However, we believe that the approach we have taken is useful and, given a good data set, it may be used to find good estimates of the transition probability matrices.

2.3 Computational procedure

This section describes the algorithm and the software that we have used to solve problem (5), which we rewrite here as follows:

$$\begin{aligned} min \quad & \tfrac{1}{2} \parallel Fx - f \parallel_2^2 \\ s.t. \quad & l_a \leq Ax \leq u_a \\ & 0 \leq x \leq 1 \end{aligned} \tag{6}$$

In this formulation we have introduced the matrix A along with the lower and upper bounds on the constraints and on the variables. This is a compact way of describing constraints 1-6 of subsection 2.1.3 in terms of the vector x. In fact, the relation $l_a \leq Ax \leq u_a$ describes constraints 3-6; constraint 1. was made explicit in the definition of x and constraint 2. is written as bounds on the variables $0 \leq x \leq 1$. The explicit construction of the matrix A and vectors l_a, u_a is very tedious. The reader interested in such technicalities can find the details of the construction in the file *dual.m* in the Appendix A.

Now rewrite the problem (6) as:

$$\begin{aligned} min \quad & \tfrac{1}{2} z^T z \\ s.t. \quad & Fx - z = f \\ & l_a \leq Ax \leq u_a \\ & 0 \leq x \leq 1 \end{aligned} \tag{7}$$

by introducing a vector of slack variables z. As it has been mentioned before, this problem is a convex quadratic optimization problem with linear constraints and as such, it can be solved very efficiently using one of the existing software packages for quadratic convex optimization (see, for example [3]). We have chosen the MOSEK optimization toolbox for MATLAB, which uses a primal interior-point algorithm to solve the problem, see [2]. Actually, the formulation of the problem which we fed to MOSEK was the Lagrangean dual of (7), which turns out to be:

$$\begin{aligned} max \quad & f^T z - e^T v - e^T t - \tfrac{1}{2} z^T z \\ s.t. \quad & A^T \begin{bmatrix} v \\ w \end{bmatrix} - F^T z + t \geq 0 \\ & w, t \geq 0 \\ & z, v\text{–free} \end{aligned} \tag{8}$$

In this formulation of the problem, $e = (1, \ldots, 1)^T$ and the variables are the vectors v, w, t, z. The implementation of the above formulation, based on the commercial software package MOSEK [2], is given in Appendix A (file `dual.m`). There is no duality gap between optimal solutions to the problems (7) and (8). The optimal values of the primal variables x can be recovered from the optimal values of the Lagrangean multipliers in the solution to the dual problem. For more on the duality theory see, for example, [4].

The advantage of using the dual formulation (8) is that it usually has a lot fewer constraints than the primal formulation (7), thus the dual problem can be solved a lot faster using a primal interior-point method. Indeed, on the test problem the solution was obtained very fast: 15 iterations of the algorithm, 1.8 seconds CPU time on the problem with 15 states and 60 periods using a personal computer with a Pentium II processor. The resulting transition probability matrix is presented in the Appendix B. It should be mentioned once again though, that one should not put much faith in this matrix because of all the significant shortcomings of the data set used. However, it can be seen that the resulting matrix satisfies all the required properties of the transition probability matrix.

3 Accounting for Non-Stationarity

In practice, the assumption of stationarity, that is, that the transition probability matrix is constant over time, is unrealistic. In particular, business cycles might change this matrix. In this section, we still assume that the data is "perfect", and address the stationarity problem. We will revisit this problem in the context of data uncertainty in section 4.

An approach to the business cycle problem is outlined in the report [5]. There, the main assumption is that there is a single factor reflecting "systemic" changes across every industry segment. This approach leads to a non-convex optimization problem, which is easy to solve via line search in the case of a single factor.

Our goal here is to devise an efficient approach that can handle multi-factor models. Another goal is to find *bounds* on the transition matrix that are consistent with the historical data. These bounds are interesting in their own right, for example in the context of predicting future defaults and bounds on their probabilities.

We first outline an efficient linear programming approach to this problem, then explore some variations on this approach.

3.1 A linear programming approach

Our basic approach is to assume that there are as many factors explaining variations of the transition matrix, as there are elements in the matrix.

Let $\mathcal{P}$ denote the set of matrices P that satisfy the "hard" constraints 1-6. of the problem. The consistency relations (1) we had before,

$$C_k = PC_{k-1}, \;\; k = 1, \ldots, n$$

rely on the assumption that P is constant, and lead to a set of overdetermined linear equations. We replace the above "soft" constraints by the set of undetermined linear equations

$$C_k = P_k C_{k-1}, \;\; k = 1, \ldots, n, \tag{9}$$

where P_k is the transition probability matrix at time k.

We now seek optimal bounds on the matrices P_k, $k = 1, \ldots, n$, that satisfy the time-varying consistency relations (9) above exactly. That is, we seek matrices P_- and P_+, both in $\mathcal{P}$, such that:

- $P_+ \geq P_-$, where the inequality is understood componentwise (that is, $P_+(i,j) \geq P_-(i,j)$ for every (i,j), $1 \leq i,j \leq s$.

- For each time instant k, there exists *some* matrix P_k in the interval $[P_-, P_+]$ that satisfies the consistency relation (9) at time k.

The interpretation of the above requirements is that the actual process is non-stationary, but its degree of non-stationarity (measured by the distance between P_+ and P_-) is bounded. The only assumption we make about this process is that P_k are bounded above and below, and can otherwise "jump" freely within its bounds. The number of "factors" explaining the variations in P_k is thus equal to the number of elements in that matrix, s^2.

Of course, there are many matrices P_-, P_+, P_k that satisfy the above requirements. Our aim is to select optimal bounds, by minimizing a measure of distance between the two matrices P_+, P_-. One way to measure this distance is the maximum componentwise norm. Since $P_+ \geq P_-$, the distance is simply

$$\max_{1 \leq i,j \leq s} P_+(i,j) - P_-(i,j).$$

Our optimization problem is thus formulated as follows.

$$\begin{array}{lll} \textit{minimize} & t & \\ \textit{subject to} & t \geq P_+(i,j) - P_-(i,j), & 1 \leq i,j \leq s \\ & C_k = P_k C_{k-1}, & k = 1, \ldots, n, \\ & P_-(i,j) \leq P_k(i,j) \leq P_+(i,j), & 1 \leq i,j \leq s \\ & P, P_+, P_- \in \mathcal{P}. & \end{array} \tag{10}$$

The above is a linear programming (LP) problem in a scalar variable t and matrix variables P_-, P_+ and $P_1, \ldots, P_n$. This means that there are $n+2$ matrix variables, each of size s^2. The above problem can be very efficiently solved using state-of-the-art commercial packages such as MOSEK [2] or CPLEX. (Typical run times for LPs with millions of variables and thousands of constraints are of the order of one hour on a workstation, see [3].)

3.2 Variations

We may introduce several variations on the LP formulation above, in order to account for possible prior knowledge on the time-varying process governing the transition probability matrix. Some of these variations retain the attractive feature of being solvable efficiently, but not all of them. In practice, one must make a trade-off between ease of solution and the accuracy of the model.

Bounded variation. The previous setting imposes no bound on the speed at which the transition probability matrices P_k can "jump" within the interval $[P_-, P_+]$ from a time instant to the next. This might not be realistic. One may impose instead a bounded variation on the P_k's with a constraint of the form

$$-\alpha \leq P_k(i,j) - P_{k-1}(i,j) \leq \alpha, \;\; 2 \leq k \leq n, \;\; 1 \leq i,j \leq s,$$

where $\alpha \geq 0$ is a bound on the absolute variation on P_k. (One may also consider relative variations.) The above constraints are easily incorporated as additional linear constraints in the linear programming framework.

In practice, the bound α is not known but it is easy to compute a trade-off curve showing for every α, the best achievable objective $\max_{1 \leq i,j \leq s} P_+(i,j) - P_-(i,j)$ of the resulting LP.

Factor models. The above LP framework implicitely assumes that there are "hidden variables" affecting the process and changing the transition probability matrix. In this sense, it accounts for business cycles, only that the business cycle is not described via a single factor but many. In fact, it accounts for a total of s^2 such factors—as many as there are elements in the transition probability matrix. Thus, the above approach does account for business cycles, albeit in a non-traditional manner, with many business cycle variables, as opposed to, say, one or two.

At the other extreme end, we might be interested in the case when the changes in the transition probability matrix are the result of changes in just a few factors, say one. In our setting, enforcing this knowledge would result in a non-convex problem (in the variables $P_{\pm}$ and the time-varying factors).

However, preliminary experiments indicate that, for the data set we used, the actual number of factors was much less than s^2 when using the above LP formalism. In other words, we find that the optimal "worst-case sequence" P_k, as well as the optimal bounds $P_{\pm}$, all lie in a subspace of dimension less than s^2. Our hope is to show, at least experimentally, that this is the case. In this sense, we would avoid the computational burden resulting from imposing a few factors directly; these few factors would come "for free". We will investigate this topic in further detail later.

Prediction of the time-varying transition matrix. In the context of estimating future defaults, we would need an estimate of the future values of the transition matrix, P_{n+1}, P_{n+2}, etc. The LP approach provides bounds on these estimates, P_+ and P_-. Futhermore, it also provides the sequence of past transition matrices $P_1, \ldots, P_n$. Therefore, it is possible to use time-series analysis, based on this sequence, to obtain a prediction of the future values of the transition matrix. For details on time-series analysis, see [5].

4 Accounting for Data Errors

In this section, we seek to examine the impact of errors in the data on the formulation of the problem and its solution. We assume here that the data, namely, the estimated vectors of frequencies of transition to default C_k, $k = 1, \ldots, n$, are only known within given intervals. These intervals could be given by standard statistical procedures that are based on sample means. To simplify notation, we collect the C_k's in one matrix $C = [C_1 \ldots C_n]$ and assume that $C_- \leq C \leq C_+$, where C_-, C_+ are given, and the inequalities are understood componentwise.

4.1 A robust least-squares approach

A first approach is to start from the constrained least-squares problem arising from the "basic" formulation of section 2. The matrices F, f defined in (3,4), that appear in the least-squares objective (5) depend (linearly) on C; let us denote this dependence by $F(C), f(C)$. We can seek a robust solution P to the problem by asking that the worst-case residual

$$\max_{C_- \leq C \leq C_+} \|F(C)x - f(C)\|_2$$

be minimized (in the above, as in section 2, the vector x contains all the variables in the matrix P). Such a problem is studied in the context of (unconstrained) least-squares in [8]. The approach can be easily extended to the case when additional constraints on the variables are imposed. (In our case, we impose the linear constraints that $P \in \mathcal{P}$.)

The resulting optimization is complex, due to the combinatorial nature of the problem of computing the worst-case residual. However, an efficient relaxation procedure can be used. This procedure relies on a semidefinite programming solution, that gives an *upper bound* on the optimal worst-case residual (within a factor of $\pi/2$, see [11, 10]). Semidefinite programs are nonlinear, convex extensions of linear programs that can be solved using similar interior-point techniques [12, 15].

4.2 An ellipsoidal approach

The above approach does provide an estimate of the transition probability matrix that is robust (insensitive) to errors in the data, at the expense of a higher discrepancy in the consistency relations. The resulting solution is very efficiently (albeit approximately) computed.

However, in practice, the norm residual is only an artifact of the "basic" approach outlined in section 2. Hence, it is not clear why we should concentrate on this measure to find the matrix P; we elaborate on this point below. Further, the robust least-squares approach provides only a *single* estimate, and no bounds on this estimate. Such bounds would be very useful in the context of estimating bounds on probabilities of future defaults.

Let us consider again the "basic" framework. Choosing the least-squares objective amounts to assuming that the "true" consistency relations are of the form

$$PC_{k-1} = C_k + \Delta C_k, \quad k = 1, \ldots, n,$$

where ΔC_k contains "errors" in the right-hand side. The least-squares optimization problem seeks to minimize the squared sum of all the error norms, $\sum_k \|\Delta C_k\|_2^2$. We see here why this approach is, in fact, inconsistent: it implicitely assumes errors in C_k and no errors in C_{k-1} at time k.

A more consistent model is to assume that there are errors in $C_1, \ldots, C_n$, and that these errors are present in every consistency relation. The equations above are replaced with

$$P(C_{k-1} + \Delta C_{k-1}) = C_k + \Delta C_k, \quad k = 1, \ldots, n.$$

Assuming that we know bounds on the errors ΔC_k, $k = 1, \ldots, n$, we see that the above relations really define a *set* of possible values of (or bounds on) P (this set can be empty if the bounds are too stringent, of course). The idea in the ellipsoidal approach is to compute bounds on P based on ellipsoidal approximations.

Let us revert to the notations in section 2, with $F(C)$ and $f(C)$ are the data-dependent matrix and vector arising in the least-squares problem. Our consistency relation now has the form

$$F(C)x = f(C) \text{ for some } C, \quad C_- \leq C \leq C_+.$$

The set of such x's is in general very complicated. But we can find an *outer* ellipsoidal approximation to it, using semidefinite programming relaxation techniques [7]. These techniques result in an ellipsoid of confidence for x, that is, for the vector formed with the elements of the transition probability matrix P. Let us denote this ellipsoid by $\mathcal{E}$.

Once the ellipsoid of confidence $\mathcal{E}$ is computed, we may compute an estimate x such that:

1. the hard constraints of the problem are satisfied;
2. $x \in \mathcal{E}$.

This is a quadratic convex feasibility problem. In general, there will be many solutions, but we may select the analytic center [15] (a kind of center of gravity that can be computed efficiently via interior-point methods) as our "central" estimate; the ellipsoidal bounds provide guaranteed bounds on the impact of data errors, on our central estimate.

4.3 Revisiting the non-stationary framework

We now outline how we could account for non-stationarity in the spirit of section 3, while taking data uncertainty into account. Let us denote by $C_k^{\pm}$ vectors containing componentwise upper and lower bounds on the probability vectors C_k, which are otherwise unknown.

The consistency relations are of the form

$$C_k = P_k C_{k-1}, \quad k = 1, \ldots, n,$$

and they should hold *for some* choice of the vectors C_k, C_{k-1} within their respective upper and lower bounds.

We again seek optimal upper and lower bounds P_+, P_- on the time-varying transition matrices P_k, $k = 1, \ldots, T$. Now however, the vectors C_k are *variables,* and the problem is not an LP anymore.

Using an ellipsoidal technique, we may compute a number of ellipsoids $\mathcal{E}_k$ (in the space of $s \times s$ matrices) such that

$$C_k = P_k C_{k-1}, \quad C_k^- \leq C_k \leq C_k^-, \quad C_{k-1}^- \leq C_{k-1} \leq C_{k-1}^-$$

implies $P_k \in \mathcal{E}_k$. Then we address the following problem:

$$\begin{array}{lll} \text{minimize} & t & \\ \text{subject to} & t \geq P_+(i,j) - P_-(i,j), & 1 \leq i,j \leq s \\ & P_k \in \mathcal{E}_k, & k = 1, \ldots, n, \\ & P_-(i,j) \leq P_k(i,j) \leq P_+(i,j), & 1 \leq i,j \leq s \\ & P, P_+, P_- \in \mathcal{P} & \end{array} \tag{11}$$

The above is a quadratic convex optimization problem, which can be efficiently solved.

The interpretation of the overall approach is as follows. The consistency relations with data uncertainty imply some error bounds on the time-varying transition probability matrix. We encapsulate this information in a single (time-varying) ellipsoid that is guaranteed to contain all transition probability matrices consistent with the available bounds on the data. We then compute the best upper and lower bounds on the transition probability matrix that are consistent with those elipsoidal bounds, as well as with the priori ("hard") constraints.

There is another, more direct, route that we will explore. We may start with the problem

$$\begin{array}{lll} \text{minimize} & t & \\ \text{subject to} & t \geq P_+(i,j) - P_-(i,j), & 1 \leq i,j \leq s \\ & C_k = P_k C_{k-1}, \;\; C_k^- \leq C_k \leq C_k^-, \;\; C_{k-1}^- \leq C_{k-1} \leq C_{k-1}^- & k = 1, \ldots, n, \\ & P_-(i,j) \leq P_k(i,j) \leq P_+(i,j), & 1 \leq i,j \leq s \\ & P, P_+, P_- \in \mathcal{P}, & \end{array} \tag{12}$$

in which P_-, P_+, P_k, as well as the C_k's, are variables. The above is a quadratic, but non-convex, problem. However, we may apply recent relaxation techniques [14] in order to come up with approximate (that is, upper bounds) solution to the problem. These would result in upper and lower bound matrices P_+, P_- that are respectively above and below the true bounds.

5 Conclusions and Further Work

In this report, we have revisited the "basic" estimation problem outlined in the earlier report [13]. We have derived an efficient, unsupervised algorithm that relies on recent interior-point mtehods for convex optimization. We have shown how to account for rare events in the estimation of the data needed to solve the problem. In a second part, we have begun to address the problems of non-stationarity and data uncertainty. We have devised a simple solution for the non-stationarity in the case when the data is deemed to be known exactly. We have outlined several possible approaches to the data uncertainty problem, one of which allows for a non-stationary model.

Several challenges lie ahead. One is to test the various approaches to robustness, with a special attention to the trade-off between numerical efficiency and conservatism (pessimism) of the bounds on the transition probability matrix. Another important topic is to compute bounds on the data itself—a crucial input to the robust estimation procedure. We will explore this topic in the context of a novel approach to estimation known as graphical models [9].

Acknowledgments

We would like to thank Mr. Ménagé, manager of the RECORD team, as well as the RECORD team members, for their helpful comments and discussions during this research.

References

[1] Moody's investors service. Available at *http://www.moodys.com.*

[2] Mosek optimization software. Available at *http://www.mosek.com.*

[3] Neos guide. Available at *http://www-fp.mcs.anl.gov/otc/Guide/index.html.*

[4] S. P. Boyd and L. Vandenberghe. Introduction to convex optimization with engineering applications. Lecture notes for EE364, Stanford University, 1996. Available via anonymous ftp into `isl.stanford.edu`, under `/pub/boyd`.

[5] S. Chesnel and C. Daleux. Impact du cycle économique sur les probabilités de défault. Mémoire de projet personnel en laboratoire, ENSTA, 1999.

[6] JP Morgan Company. Creditmetrics: technical document. Technical report, JP Morgan, 1997. Available at *http://www.riskmetrics.com/corp/pubs.*

[7] L. El Ghaoui and G. Calafiore. Confidence ellipsoids for uncertain linear equations with structure. In *Proc. IEEE Conf. on Decision and Control*, 1999.

[8] L. El Ghaoui and H. Lebret. Robust solutions to least-squares problems with uncertain data. *SIAM J. Matrix Anal. Appl.*, 18(4):1035–1064, October 1997.

[9] M. I. Jordan. *Learning in Graphical Models.* Kluwer Academic Press, 1998.

[10] Y. E. Nesterov. Quality of semidefinite relaxation for nonconvex quadratic optimization. Technical report, CORE, Universite Catholique de Louvain, Belgium, 1997.

[11] Y. E. Nesterov. Semidefinite relaxation and non-convex quadratic optimization. *Optimization methods and software*, 12:1–20, 1997.

[12] Yu. Nesterov and A. Nemirovsky. *Interior point polynomial methods in convex programming: Theory and applications.* SIAM, Philadelphia, PA, 1994.

[13] Equipe RECORD. Calcul de la matrice de transition dans RECORD. Technical report, DREI, BNP-Paribas Group, 1999.

[14] R. Saigal, L. Vandenberghe, and H. Wolkowicz, editors. *Handbook on Semidefinite Programming and Applications.* Kluwer Academic Publishers, Dordrecht, The Netherlands, 2000.

[15] L. Vandenberghe and S. Boyd. Semidefinite programming. *SIAM Review*, 38(1):49–95, March 1996.

A Matlab Files

A.1 frequencies.m

```
function C=frequencies(A,states)
% takes a matrix, A, of transitions data and calculates
% the relative frequencies of default.
% Cik=relative frequency of companies
% starting in state i and defaulting over k periods
% states = number of states
% C=frequencies(A,states)

[m,n]=size(A);
n=n-1;
% the matrix that will finally contain the relative frequencies
C(states,n)=0;
% helper matrix that stores the total number of occurences of a
% given group (used for calculating the relative frequencies)
D(states,n)=0;

% this loop calculates how many many companies go from a given
% state to default over a period of j units, j=1:n
for i=1:states
  for j=1:n
    for k=0:n-j
      C(i,j)=sum((A(:,1+k)==i)&(A(:,j+k+1)==states))+C(i,j);
      D(i,j)=sum((A(:,1+k)==i))+D(i,j);
    end
  end
end

% this takes into account boundary effects .. suppose
% there are 60 periods. at period=59, a company is in state
% Baa and then in period=60, it goes to default.
% This occurence will add to the number of companies that
% are in state Baa and after one period go to default ..
% but it should also add to the statistics of the
% number of companies in state Baa that default after
% 2,3,4 .. period otherwise we might get frequencies that
% look illogical .. ie. we might have a case where the
% frequnecy of transitioning from Baa to
% default after 1 period is higher than the frequency of
% transitioning from Baa to default after 10 periods
% basically, i am using the
% fact that once a company defaults, it stays in default.
for i=1:m
  if A(i,n+1)==states
```

```
    for j=2:n
      for k=n-j+2:n
        C(A(i,j),k)=C(A(i,j),k)+1;
                  D(A(i,j),k)=D(A(i,j),k)+1;
      end
    end
  end
end

% calculates the relative frequencies of default for i=1:states
  for j=1:n
    if D(i,j)==0
      C(i,j)=0;
    else
      C(i,j)=C(i,j)/D(i,j);
    end
  end
end
```

A.2 dual.m

```
function P = dual(C)
% DUAL estimates a transition probability matrix P from the empirical
% default frequency matrix C.  It solves the least squares problem:
%
% min 0.5||Fx-f||^2  s.t la <= Ax <= ua, l <= x <= u
%
% where F is a certain matrix, dependent on C and f is a certain vector
% dependent on C and x is a vectorized form of P.
% The problem is being solved by solving the DUAL of the problem:
%
% min 0.5 z'z, s.t. Fx-z=f;  la <= Ax <= ua, l <= x <= u
%
% The dual simplifies to:  max f'z - e'v - e's - 0.5z'z
%                s.t. A'[v; w] - F'z + s >=0, s,w>=0, v,z-free.
%%
% This approach is beneficial when the number of rows of F >> number of columns
% or equivalently, when the number of periods is large compared to the
% number of states (see the description of C below).
%
% input:
% C:    a sxn matrix, [C(1) ... C(n)], containing the historical
%        default frequencies over periods 1...n.
%        s is the number of states, n is the number of periods.
%        Each vector C(k) consists of entries C(i,k) = relative frequency of
%        companies starting in state i and defaulting over k periods.
%        Since default is an absorbing state, C(s,k) must be 1 for all k=1..n
```

```
%
% outputs:
%       P       a sxs matrix--estimated matrix of transitions probabilities.
%
% see also: direct.m, dual.m
% M. Oks  6-10-00


% read the number of states and the number of periods
[s,n]=size(C);

% Check that C is valid
if ~(isequal(C(s,:), ones(1,n)))
        disp(' C is not a valid matrix; Exiting');
        return;
end

% setting up data for the problem
clear prob;

% create a vector f
f=reshape(C(1:s-1,:), n*(s-1), 1);

%  Change C a bit, it will be usefull later;  Now C is s by (n+1)
C=[zeros(s,1), C];
C(s,1)=1;

% Create a matrix F
F=[];
I=eye(s-1);
for i=1:n
        F=[F; kron(I, C(:, i)')];
end

% Create the constraint matrix A

% sum of rows of P must be 1
A=kron(I, ones(1,s));

% probabilities of default decrease with the starting state:
t=zeros(1,s);
t(1,s)=1;
temp=kron(eye(s-2), t);
A=[A; [temp, zeros(s-2,s)]-[zeros(s-2,s), temp]];

% Ordering rules:
% order in rows of P
```

```
temp=zeros((s-1)^2, s*(s-1));
k=0;
for i=1:s-1,  for j=i:s-1
                k=k+1;
                temp(k, s*(i-1)+j+1)=1;
                temp(k, s*(i-1)+j)=-1;
              end
              for j=1:i-1
                k=k+1;
                temp(k, s*(i-1)+j)=1;
                temp(k, s*(i-1)+j+1)=-1;
              end
end
A=[A; temp];

% order in columns of P
temp=zeros((s-1)*(s-2), s*(s-1));
k=0;
for i=1:s-2,  for j=i+1:s-1
                k=k+1;
                temp(k, s*(i-1)+j)=1;
                temp(k, s*i+j)=-1;
              end
              for j=1:i
                k=k+1;
                temp(k, s*(i-1)+j)=-1;
                temp(k, s*i+j)=1;
              end
end
A=[A; temp];
[m, k]=size(A);
k=n*(s-1);      % will be useful later

% Setup the data for mosek; the vector of variables is [z; s; v; w]
% (in that order) for convinience.

prob.qosubi = 1:k;
prob.qosubj = 1:k;
prob.qoval = -ones(k, 1);

prob.c= [f; -ones(s^2-1, 1); sparse(m+1-s, 1)];

prob.a = [-F', speye(s*(s-1)), A'];

% Lower and upper bounds on constraints
prob.blc = sparse(s*(s-1), 1);
prob.buc = inf*ones(s*(s-1), 1);
```

```
% Lower and upper bounds on variables
prob.blx = [-inf*ones(k,1); sparse(s*(s-1),1); -inf*ones(s-1, 1); sparse(m+1-s,1)];
prob.bux = [];

% Solve the problem using mosek
[r, res]=mosekopt('maximize', prob);

% Display return code
fprintf(' Return code: %d\n', r);

% x solution is the vector of the Lagrange multipliers of the constraints
x=res.sol.y;

% Convert into P
P=reshape(x, s, s-1)';
P=[P; zeros(1,s)];
P(s,s)=1;

% Report objective value (just for comparison purposes)
fprintf('Objective value: %-6e\n', 0.5*norm(F*x-f)^2);
```

B Estimated Transition Probability Matrix

The following estimate for the transition probability matrix was obtained using the data set BaseFinale.xls sent to us. The data set has been modified as described in section 2.2, and the matrix was computed using the procedure described in section 2.3.

```
Pdual =

  Columns 1 through 7

    0.5110    0.2492    0.1335    0.0734    0.0327    0.0002    0.0000
    0.1541    0.4193    0.2306    0.1335    0.0621    0.0003    0.0000
    0.0968    0.2019    0.4022    0.2095    0.0893    0.0003    0.0000
    0.0592    0.1312    0.2246    0.4198    0.1647    0.0004    0.0000
    0.0305    0.0828    0.1461    0.2477    0.4923    0.0006    0.0000
    0.0000    0.0000    0.0000    0.0000    0.0000    0.9707    0.0098
    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.8716
    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.4486
    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.2102
    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000
    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000
    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000
    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000
    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000
         0         0         0         0         0         0         0

  Columns 8 through 14

    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000
    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000
    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000
    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000
    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000
    0.0098    0.0098    0.0000    0.0000    0.0000    0.0000    0.0000
    0.0257    0.0257    0.0257    0.0257    0.0257    0.0000    0.0000
    0.4486    0.0257    0.0257    0.0257    0.0257    0.0000    0.0000
    0.2102    0.5026    0.0257    0.0257    0.0257    0.0000    0.0000
    0.0000    0.0000    0.8556    0.0722    0.0722    0.0000    0.0000
    0.0000    0.0000    0.0000    0.8427    0.0787    0.0787    0.0000
    0.0000    0.0000    0.0000    0.4607    0.4607    0.0787    0.0000
    0.0000    0.0000    0.0000    0.0000    0.0000    0.9148    0.0852
    0.0000    0.0000    0.0000    0.0000    0.0000    0.0000    0.9776
         0         0         0         0         0         0         0

  Column 15

    0.0000
```

0.0000
0.0000
0.0000
0.0000
0.0000
0.0000
0.0000
0.0000
0.0000
0.0000
0.0000
0.0000
0.0224
1.0000