Fall 2011

Credit Rating and Assignment of Naics Codes Using Lsi Method

Jerome Ouedraogo

Version: December 7, 2011

CREDIT RATING AND ASSIGNMENT OF NAICS CODES USING LSI METHOD

by

JEROME OUEDRAOGO

(Under the Direction of Patricia Humphrey)

ABSTRACT

The objective here is first, to improve automatic assignment of industry codes using LSI (lexical processing) by increasing the algorithm efficiency (both computationally and in term of input requirements), then quantify the lender's risk as "distance to default" (higher distance to default indicates default is less likely to occur), estimate the distance to default for each company and combine the results to obtain an estimate of the distance to default for each NAICS code.

Index Words: LSI, NAICS code, distance to default

CREDIT RATING AND ASSIGNMENT OF NAICS CODES USING LSI METHOD

by

JEROME OUEDRAOGO

B.S. in Statistics

DEUG in Mathematics and Physics

A Thesis Submitted to the Graduate Faculty of Georgia Southern University

in Partial Fulfillment of the Requirements for the Degree

MASTER OF SCIENCE

STATESBORO, GEORGIA

2011

CREDIT RATING AND ASSIGNMENT OF NAICS CODES USING LSI METHOD

by

JEROME OUEDRAOGO

Major Professor: Patricia Humphrey

Committee: Charles Champ

John Barkoulas

Electronic Version Approved:

December 2011

TABLE OF CONTENTS

Page

LIST OF TABLES

Table Page

LIST OF FIGURES

Figure Page

CHAPTER 1

INTRODUCTION

The goals of this paper are to improve the LSI (Latent Semantic Indexing) approach to find NAICS (North American Industry Classification System) codes for each company's establishments (subsidiaries) using its description, construct a credit rating for each sector of NAICS codes.

Risk from credit is the deviation of the performance of a portfolio of loans from its expected value. Credit risk is diversifiable, but it is difficult to eliminate completely. This is because portions of default risk result from exposure to systematic risks (market risks). In addition, the idiosyncratic nature of some portion of these losses remains a problem for creditors in spite of the beneficial effect of diversification. This is particularly true for banks that lend in local markets. Credit risk arises due to uncertainty in a counterparty's ability to meet its obligations in accordance with agreed upon terms. Banks are required to maintain capital that will cover an amount predicted by a Value-at-Risk calculation (prescribed by the Federal Reserve). Quantification of credit risk helps banks manage diversification, and also helps in the development of adequate controls over risk. Analytical techniques, such as those built into automated credit scoring, are designed to assign a risk rating to each debtor. Clustering companies into groups may facilitate the assignment of a risk rating. An example of credit ratings for groups of companies appears in Table 1 of [Santomero][8]. A suggested approach to grouping companies uses NAICS codes, where the NAICS code may be identified using Latent Semantic Indexing (LSI).

The North American Industry Classification System (NAICS) is used by business and government to classify businesses according to type of economic activity (service

and process of production) in Canada, Mexico and the United States. It has largely replaced the older Standard Industrial Classification (SIC) system. However, certain government departments and agencies, such as the U.S. Securities and Exchange Commission (SEC), still use the SIC code. BB&T, a commercial bank, is interested in using this new classification to rate their credit risk.

For their masters' theses, NCSU students (Han Liu and Mbaga Nzabakurana)[5] worked on the LSI but their approach did not perform well for some companies like TMobile, Nike, and WalMart. For this reason, this problem was assigned as a project during the Industrial Math/Stat Modeling Workshop for Graduate Students - July 7-15, 2011 - organized by SAMSI (Statistical and Applied Mathematical Sciences Institute) at Raleigh, NC. As part of the group that worked on this project, I modified the algorithm and changed the approach to increase the accuracy of the existing algorithm (on the LSI); it performed well with the precedent companies. However, there are 19,720 NAICS six digit codes with some repetition. For each NAICS sector, we also estimated a credit rating using distance to default, but some distances were negative.

The challenge is to narrow those NAICS code to 1,175 unique codes, and perform the algorithm. Further, we will investigate more on the method used to compute distance to default and try to understand the negative distance.

CHAPTER 2

UNDERSTANDING NAICS CODES

The North American Industry Classification System (NAICS, pronounced Nakes) was developed under the direction and guidance of the Office of Management and Budget (OMB) as the standard for use by Federal statistical agencies in classifying business establishments for the collection, tabulation, presentation, and analysis of statistical data describing the U.S. economy. Use of the standard provides uniformity and comparability in the presentation of these statistical data. NAICS is based on a production-oriented concept, meaning that it groups establishments into industries according to similarity in the processes used to produce goods or services. NAICS replaced the Standard Industrial Classification (SIC) system in 1997.[12]

NAICS was initially developed and subsequently revised by Mexico's INEGI, Statistics Canada, and the U.S. Economic Classification Policy Committee (the latter acting on behalf of OMB). The goal of this collaboration was to produce common industry definitions for Canada, Mexico, and the United States. These common definitions facilitate economic analyses of the economies of the three North American countries. The statistical agencies in the three countries produce information on inputs and outputs, industrial performance, productivity, unit labor costs, and employment. NAICS, which is based on a production-oriented concept, ensures maximum usefulness of industrial statistics for these and similar purposes.[12]

NAICS in the United States was designed for statistical purposes. However, NAICS is frequently used for various administrative, regulatory, contracting, taxation, and other non-statistical purposes. For example, some state governments offer tax incentives to businesses classified in specified NAICS industries. Some contracting

authorities require businesses to register their NAICS codes, which are used to determine eligibility to bid on certain contracts. The requirements for these non-statistical purposes played no role in the initial development of NAICS or its later revisions.[12]

An establishment is generally a single physical location where business is conducted or where services or industrial operations are performed (e.g., factory, mill, store, hotel, movie theater, mine, farm, airline terminal, sales office, warehouse, or central administrative office). An enterprise, on the other hand, may consist of more than one location performing the same or different types of economic activities. Each establishment of that enterprise is assigned a NAICS code based on its own primary business activity.[12]

The NAICS numbering system employs a two through six-digit code at the most detailed industry level. The first five digits are generally (although not always strictly) the same in all three countries (Canada, US and Mexico). Each digit in the code is part of a series of progressively narrower categories, and more digits signify greater classification detail. The last digit designates national industries. The first two digits designate the largest business sector, the third digit designates the subsector, the fourth digit designates the industry group, and the fifth digit designates particular industries. For example looking at the NAICS code 111110, it can be broken down as:

11 Sector –Agriculture, Forestry, Fishing and Hunting

111 Crop Production : Establishments are classified to the crop production subsector when crop production (i.e., value of crops for market) accounts for one-half or more of the establishment's total agricultural production.

1111 Oilseed and Grain Farming

11111 Soybean Farming

111110 Soybean farming, field and seed production.

CHAPTER 3

LATENT SEMANTICS INDEXING (LSI)

3.1 Algorithm Description

Latent Semantic Indexing (LSI) also called latent semantic analysis (LSA) is an indexing and retrieval method that uses a mathematical technique (singular value decomposition) to identify patterns in the relationships between the terms and concepts contained in an unstructured collection of text. LSI is based on the principle that words that are used in the same contexts tend to have similar meanings.

To perform a LSI analysis, we first construct a Term-Document Matrix (TDM) to identify the occurrences of the m unique terms within n documents. In other words, a TDM is a mathematical matrix that describes the frequency of terms that occur in a collection of documents.

$$A = \begin{pmatrix} a_{11} & \dots & a_{1n} \\ . & \dots & . \\ . & \dots & . \\ . & \dots & . \\ a_{m1} & \dots & a_{mn} \end{pmatrix}$$

In our study, each column is one NAICS code; rows represent key words appearing in NAICS code descriptions. A full list of all 19,720 NAICS codes can be obtained from http://www.census.gov/cgi-bin/sssd/naics/naicsrch?chart=2007. Each element represents the number of occurrence of each word (row) in a code (column). The full matrix A using all 19,720 NAICS codes is 6489x19720.

As an example, we choose at random 7 NAICS codes and represent the coresponding

TDM. Each code corresponds to one "document" as follows:

Document1	221310	Water Supply and Irrigation Systems
Document2	238320	Painting and Wall Covering Contractors
Document3	311230	Breakfast Cereal Manufacturing
Document4	315221	Men's and Boys' Cut and Sew Underwear and Nightwear Manufacturing
Document5	315233	Women's and Girls' Cut and Sew Dress Manufacturing
Document6	316213	Men's Footwear (except Athletic) Manufacturing
Document7	316214	Women's Footwear (except Athletic) Manufacturing

The corresponding TDM is:

Document	D1	D2	D3	D4	D5	D6	D7
athletic	0	0	0	0	0	1	1
boys	0	0	0	1	0	0	0
breakfast	0	0	1	0	0	0	0
cereal	0	0	1	0	0	0	0
contractors	0	1	0	0	0	0	0
covering	0	1	0	0	0	0	0
cut	0	0	0	1	1	0	0
dress	0	0	0	0	1	0	0
footwear	0	0	0	0	0	1	1
girls	0	0	0	0	1	0	0
irrigation	1	0	0	0	0	0	0
manufacturing	0	0	1	1	1	1	1
men	0	0	0	1	0	1	0
nightwear	0	0	0	1	0	0	0
painting	0	1	0	0	0	0	0
sew	0	0	0	1	1	0	0
supply	1	0	0	0	0	0	0
systems	1	0	0	0	0	0	0
underwear	0	0	0	1	0	0	0
wall	0	1	0	0	0	0	0
water	1	0	0	0	0	0	0
women	0	0	0	0	1	0	1

Table 3.1: TDM

After creating the TDM, we then create a query matrix. The query matrix is a column vector created using the key words in the TDM, and the information we have (the description of a given company in our case).

Suppose a fictitious company called XYZ has the following activities as described by the company itself:

XYZ currently make shoes, jerseys, shorts, etc. for a wide range of sports. XYZ sells an assortment of products, including shoes and apparel for sports activities like association football,basketball, running, combat sports, tennis, American football, athletics, golf, and cross training for men, women, and children.

The corresponding query matrix using the dictionnary of the TDM on the precedent page is:

	Query
athletic	0
boys	0
breakfast	0
cereal	0
contractors	0
covering	0
cut	0
dress	0
footwear	0
girls	0
irrigation	0
manufacturing	0
men	1
nightwear	0
painting	0
sew	0
supply	0
systems	0
underwear	0
wall	0
water	0
women	1

Table 3.2: Query

Finally, we find the smilarities (between query and documents) through angle comparison. In this step, we rank documents in decreasing order of query-document cosine similarities. The best codes are those with the largest cosine.

$$sim(C_i, q) = cosine(C_i, Q) = \frac{(C_i.Q)}{\|C_i\|\|Q\|}$$

where C_i for $i = 1$ to $n =$ number of documents (codes).

3.2 Text-to-Matrix Generator (TMG)

The TMG is a MATLAB Toolbox for Generating Term-Document Matrices from text collections. TMG is written entirely in MATLAB and runs on any computer system that supports that environment. This can be downloaded from http://CRAN.R-project.org/package=quantmod.

We used the TMG (Text-to-Matrix Generator) toolbox to construct our TDM. TMG is constructed to perform preprocessing and filtering steps that are typically performed on text documents. Its interface presents the following options:

- Create the TDM corresponding to a set of documents (matrix A);
- Create the query vectors from user input (Q);
- Update existing TDM by incorporation of new documents;
- Broaden existing TDM results by deletion of specified documents.

This is accomplished in the following steps:

Step 1

Use TMG to build the NAICS codes TDM. We use 19,720 NAICS code descriptions (6 digits) to build the TDM (matrix A).

The input file is a text file, and NAICS code is separated from other by a blank line. In fact, in TMG, document must be separated by a blank line. Otherwise, it will understood as as one document.

Step 2

- Get the company description, format it as a text file.
- Use TMG to create a query matrix (using the TDM key words (6,482 key words) created in step 1) based on the company description'

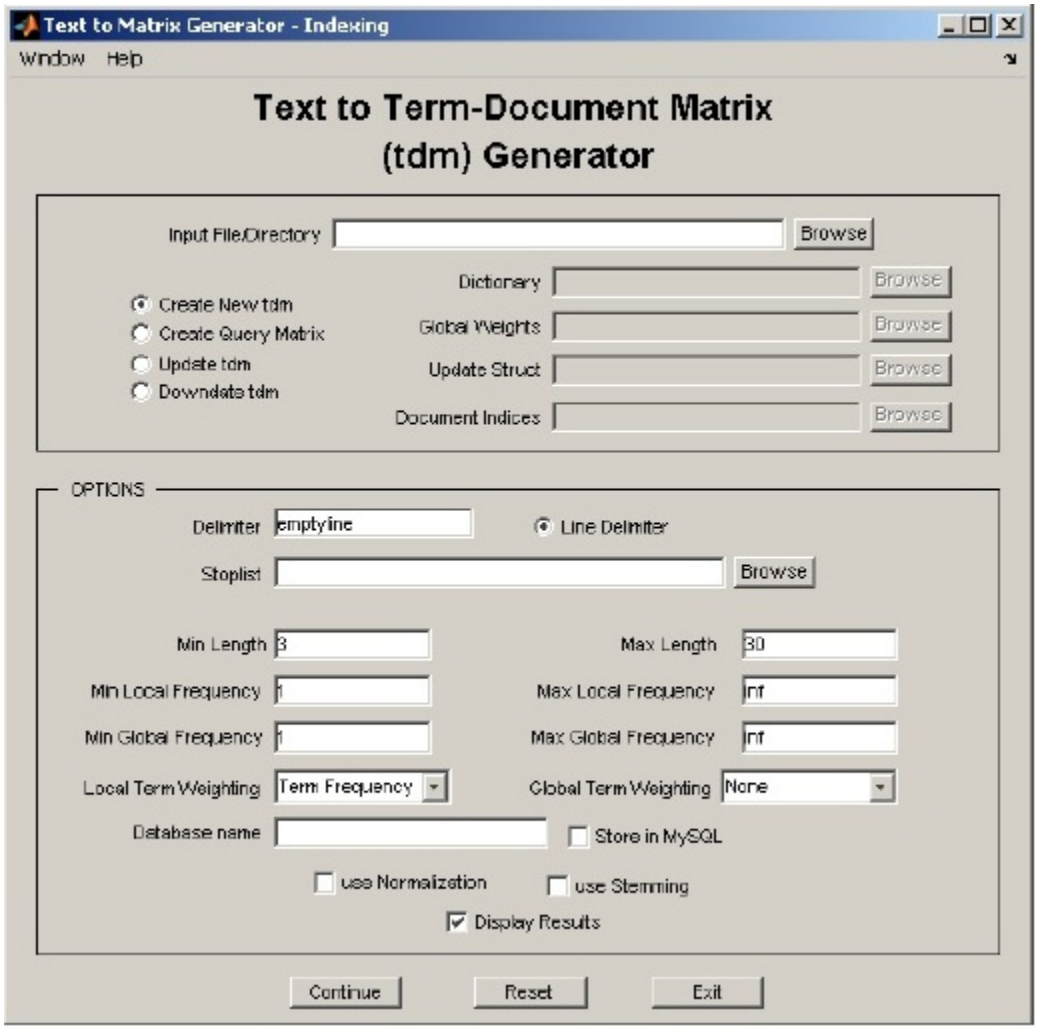

Figure 3.1: TMG

Step 3

Display the best NAICS codes (columns) matching the company description through angle comparison. In this step, we rank documents in decreasing order of query-document cosine similarities. The best codes are those with the largest cosine,

$$sim(C_i, q) = cosine(C_i, Q) = \frac{(C_i.Q)}{\|C_i\|\|Q\|}$$

where C_i for $i = 1$ to $n = 19,720$ are the columns of the matrix A.

Depending on how it is used, the results can be different. The stop words (words ignored in creating both the TDM and query) play an important role. The difference between the first approach (used by NCSU students) and the second approach (the one I used during the workshop) is how we create the query matrix using TMG (step 2).

The NCSU students' approach used the "Update existing TDM by incorporating new documents" option to create the query matrix. This approach is wrong because using this option can add new key words, and consequently may change our TDM matrix. In our second approach, we use the "Create the query matrix vectors from user input (Q)" in the second step.

A third approach was also used in this study. The first two approaches used the 19,720 NAICS codes which are not unique. For example the code 111120 has 8 different descriptions.

111120 Canola farming, field and seed production

111120 Flaxseed farming, field and seed production

111120 Mustard seed farming, field and seed production

111120 Oilseed farming (except soybean), field and seed production

111120 Rapeseed farming, field and seed production

111120 Safflower farming, field and seed production

111120 Sesame farming, field and seed production

111120 Sunflower farming, field and seed production

This third approach will modify the second (which perfomed well) by using 1,175 codes which are unique in lieu of 19,720 codes which are not. The resulting matrix A here was 1,454x1,175 instead of 6,489x19,720.

3.3 Results

The first method suggested these NAICS codes for T-Mobile.

Code	Description
518210	Data entry services
484220	Mobile home towing services, local
484230	Mobile home towing services, long-distance
712110	Mobile museums
722330	Canteens, mobile

Table 3.3: Results for T-Mobile, using the first method

By comparison, the second approach suggested the following NAICS codes:

Cosine ∡	Code	Description
0.4000	517210	wireless data communication carriers except satellite
0.2582	454112	auctions internet retail
0.2582	515111	broadcasting networks radio
0.2582	515120	broadcasting networks television
0.2582	515210	cable broadcasting networks
0.2582	515210	subscription television networks
0.2582	517110	cable tv providers except networks
0.2582	518210	data entry services

Table 3.4: Results for T-Mobile, using the second approach

And the third approach, the following NAICS codes:

Cosine ∡	Code	Description
0.5556	722330	Mobile food service
0.3443	518210	data processing hosting and related services
0.2887	453930	manufactured mobile home dealers
0.2722	323122	prepress services
0.2722	519190	all other information services
0.2722	541199	all other legal services
0.2722	541214	payroll services
0.2722	541219	other accounting services
0.2722	541310	architectural services
0.2722	541330	engineering services
0.2722	541340	drafting services
0.2722	541940	veterinary services
0.2722	561491	repossession services
0.2722	561611	investigation services
0.2722	561720	janitorial services
0.2722	561730	landscaping services
0.2722	561990	all other support services
0.2722	562910	remediation services

Table 3.5: Results for T-Mobile, using a third approach

T-mobile was not the only company examined using the three appraoches. We ran also the search for other companies like Nike, Bed, Bath & Beyond (BBB), CVS, Duke Energy, Greenbrier Companies, Home Depot, Macy, Old Dominion Freight, Rent A Center, and Shenandoah Telecom. As with T-mobile, the results were best for the second approach.

	Best matches
339920	Track and field athletic equipment (except apparel, footwear) manufacturing
423910	Athletic goods (except apparel, footwear, nonspecialty) merchant wholesalers
339920	Hockey equipment (except apparel) manufacturing
339920	Squash equipment (except apparel) manufacturing
533110	Industrial design licensing
448150	Apparel accessory stores
811490	Sporting equipment repair and maintenance without retailing new sports equipment
316213	Leather footwear, men's (except athletic, slippers), manufacturing
316219	Leather upper athletic footwear manufacturing
316219	Vinyl upper athletic footwear manufacturing
339932	Balls, rubber (except athletic equipment), manufacturing
424340	Athletic footwear merchant wholesalers
316214	Leather footwear, women's (except athletic, slippers), manufacturing
339920	Football equipment and supplies (except footwear, uniforms) manufacturing
315292	Apparel, fur (except apparel contractors), manufacturing

Table 3.6: Results for Nike using a first approach

Cosine	Best matches
0.4554	339920 track and field athletic equipment except apparel footwear manufacturing
0.3339	423910 athletic goods except apparel footwear nonspecialty merchant wholesalers
0.3213	339920 hockey equipment except apparel manufacturing
0.3213	339920 squash equipment except apparel manufacturing
0.3014	448150 apparel accessory stores
0.3014	533110 industrial design licensing
0.2906	811490 sporting equipment repair and maintenance without retailing new sports equipm
0.2874	316213 leather footwear men's except athletic slippers manufacturing
0.2874	316219 leather upper athletic footwear manufacturing
0.2874	316219 vinyl upper athletic footwear manufacturing
0.2874	339932 balls rubber except athletic equipment manufacturing
0.2811	424340 athletic footwear merchant wholesalers
0.2787	316214 leather footwear women s except athletic slippers manufacturing
0.2787	339920 football equipment and supplies except footwear uniforms manufacturing
0.2732	315292 apparel fur except apparel contractors manufacturing

Table 3.7: Results for Nike using a second approach

Cosine	Best matches
0.4423	316213 men's footwear except athletic manufacturing
0.4244	315999 other apparel accessories and other apparel manufacturing
0.4104	316214 women s footwear except athletic manufacturing
0.2708	316219 other footwear manufacturing

Table 3.8: Results for Nike using a third approach

Cosine	Best matches
0.5103	453310 used merchandise stores
0.5000	442110 bed stores retail
0.4593	442299 housewares stores
0.4593	442299 linen stores
0.4593	452990 general stores
0.4593	453220 christmas stores
0.4167	445299 dairy product stores
0.4167	448150 furnishings stores men s and boys
0.4167	453310 second hand merchandise stores

Table 3.9: Results for B, B & B using a second approach

CHAPTER 4

SINGULAR VALUE DECOMPOSITION

For completeness, we will present why the LSI uses the Singular Value Decomposition and how.

A fundamental deficiency of current information retrieval methods is that the words searchers use often are not the same as those by which the information they seek has been indexed. There are actually two sides to the issue: synonymy and polysemy (Deerwester S. et al.)[3] We use synonymy in a very general sense to describe the fact that there are many ways to refer to the same object. By polysemy we refer to the general fact that most words have more than one distinct meaning (homography). In different contexts or when used by different people the same term (e.g. "chip") takes on varying referential significance. Thus the use of a term in a search query does not necessarily mean that a document containing or labeled by the same term is of interest. Polysemy is one factor underlying poor "precision".

To deal with such problems (synonymy, polysemy, and dependence of documents), Deewester et al. [3] proposed the Singular Value Decomposition (SVD) method. In the following, we will define the SVD, give some properties and then show how it is impleted in LSI.

4.1 Theory of Singular Value Decomposition (SVD)

The Singular Value Decomposition (SVD) is a widely used technique to decompose a matrix into several component matrices, exposing many of the useful and interesting properties of the original matrix.

To get to the SVD, we start by matrix diagonalization. Recall that if A is a symmetric real nxn matrix, there is an orthogonal matrix V and a diagonal D such that $A = VDV^T$. Here the columns of V are eigenvectors for A and form an orthonormal basis for R^n; the diagonal entries of D are the eigenvalues of A.

Suppose that we have now an arbitrary real mxn matrix A. We can still find orthogonal matrices U and V and a diagonal matrix, $\sum$; such that $A = U \sum V^T$. Note that, U is mxm and V is nxn, so that $\sum$ is rectangular with the same dimensions as A. The diagonal entries of $\sum$; that is the $\sum_{ii} = \sigma_i$, can be arranged to be nonnegative and in order of decreasing magnitude. These σ_i are called singular values of A. The columns of U and V are called left and right singular vectors, for A.

Now let's look at the SVD for an mxn matrix A. Here the transformation takes R^n to a different space, R^m, so it is reasonable to ask for a natural basis for each of domain and range. The columns of V and U provide these bases.

How do we choose the bases $v_1, v_2, ..., v_n$ and $u_1, u_2, ..., u_m$ for the domain and range? We can choose those bases as follows:

Let $A^T A = VDV^T$, with the diagonal entries λ_i of D arranged in nonincreasing order, and let the columns of V (which are eigenvectors of $A^T A$) be the orthonormal basis $v_1, v_2, ..., v_n$.

$$\begin{aligned}
(Av_i)(Av_j) &= (Avi)^T(Avj) \\
&= v_i^T A^T A v_j \\
&= v_i^T(\lambda_j v_j) \\
&= \lambda_j(v_i v_j)
\end{aligned}$$

The set $Av_1, Av_2, ..., Av_n$ is orthogonal, and the nonzero vectors in this set form a basis for the range of A. Thus, the eigenvectors of A^TA and their images under A provide orthogonal bases allowing A to be expressed in a diagonal form.
Lastly, we must normalize the eigenvectors to have length 1.
For $i = j$, we have:

$$\begin{aligned}(Av_i)(Av_i) &= (Avi)^2 \\ &= \lambda_i(v_iv_i) \\ &= \lambda_i\end{aligned}$$

Which means $\lambda_i \geq 0$.

Since these eigenvalues were assumed to be arranged in nonincreasing order, we conclude that $\lambda_1 \geq \lambda_2 \geq ...\lambda_k \geq 0$ and, since the rank of A equals k; $\lambda_i = 0$ for $i > k$. The orthonormal basis for the range is therefore defined by:
$u_i = \frac{Av_i}{|Av_i|} = \frac{Av_i}{\sqrt{\lambda_i}}$ for $1 \leq i \leq k$.
If $k \leq m$ we extend this to an orthonormal basis of R^m. Letting $\sigma_i = \sqrt{\lambda_i}$, we have $Av_i = \sigma_iu_i$ for all $1 \leq i \leq k$.

Assembling the v_i as the columns of a matrix V and the u_i to form U; we have $AV = U\sum$, where $\sum$ has the same dimensions as A with entries σ_i along the main diagonal and all other entries equal to zero. Hence, $A = U\sum V^T$, which is the singular value decomposition of A. When viewed in a purely algebraic sense, any zero rows and columns of the matrix $\sum$ are superfluous. They can be eliminated if the matrix product $A = U\sum V^T$ is expressed using partitioned matrices as follows:

$$A = [u_1 ... u_k | u_{k+1} ..., u_m] \left[\begin{array}{ccc|c} \sigma_1 & & & \\ & \ddots & & \\ & & \sigma_k & \\ \hline & & & \\ & 0 & & 0 \\ & & & \end{array}\right] \left[\begin{array}{c} v_1^T \\ \vdots \\ v_t^T \\ \hline v_{t+1}^T \\ \vdots \\ v_n^T \end{array}\right]$$

Although these partitions assume that k is strickly less than m and n, it should be clear how to modify the arguments if k is equal to m or n. When the partitioned matrices are multiplied, the result is:

$$A = [u_1 ... u_k] \begin{bmatrix} \sigma_1 & & \\ & \ddots & \\ & & \sigma_k \end{bmatrix} \begin{bmatrix} v_1^T \\ \vdots \\ v_k^T \end{bmatrix} + [u_{k+1} ..., u_m] \begin{bmatrix} \\ 0 \\ \\ \end{bmatrix} \begin{bmatrix} v_{k+1}^T \\ \vdots \\ v_n^T \end{bmatrix}$$

From this last equation, we can see that only the first k u's and v's make any contribution to A. Indeed, we may write:

$$A = [u_1 ... u_k] \begin{bmatrix} \sigma_1 & & & \\ & . & & \\ & & . & \\ & & & \sigma_k \end{bmatrix} \begin{bmatrix} v_1^T \\ . \\ . \\ . \\ v_k^T \end{bmatrix}$$

In this last form, the matrices U and V are no longer square (mxk and kxn respectively), and the diagonal matrix is square. This is the alternative version of the SVD that is taken as the definition in some expositions: Any mxn matrix A of rank k can be expressed in the form $A = U \sum V^T$ where U is an mxk matrix such that $U^T U = I$, $\sum$ is a kxk diagonal matrix with positive entries in decreasing order on the diagonal, and V is an nxk matrix such that $V^T V = I$.

4.2 How does the SVD works with LSI?

One of the most fundamental problems of information retrieval (IR) is how to find the unique documents that are semantically close to a given query. Our early approaches used exact keyword matching techniques to identify relevant documents (results are likely to be unsatisfactory). We already pointed out the two main reasons which are:

- Synonymy: two different words may refer to the same concept. For example: car, and automobile. If a document contains just the word car and the query just the word automobile then exact keyword matching techniques will fail to retrieve this document. We can see that in the example above for words manufaturing in the TDM and make in the query matrix. We have also footwear, and shoes.

- Polysemy: the same word may refer to different concepts, depending on the context. For example the word bank (river bank or money bank).

The SVD in the Information Retrival is also called two-mode factor analysis.

How does the SVD handle the two fundamental problems (synonymy and polysemy)? Synonymy is captured by the dimensionality reduction of Principal Component Analysis. Words which are highly correlated will be mapped in close proximity in the lower dimensional space.

The different steps of the SVD for LSI are:

Step 1: Construct the term-document matrix A and query matrix.

Step 2: Decompose matrix A by singular value decomposition (SVD) and find the U, S and V matrices, where $A = USV^T$

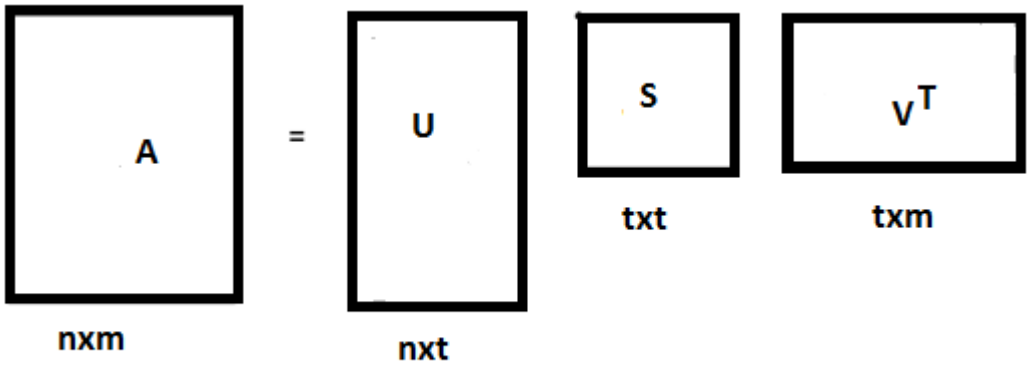

Figure 4.1: SVD of TDM

Where :

U has orthogonal, unit-length columns ($U^T U = I$)

V has orthogonal, unit-length columns ($V^T V = I$)

S is the diagonal matrix of singular values

n is the number of rows of A

m is the number of columns of A

t is the rank of A ($\leq min(n, m)$)

Step 3: Implement a Rank k Approximation by keeping the first columns of U and V and the first columns and rows of S.

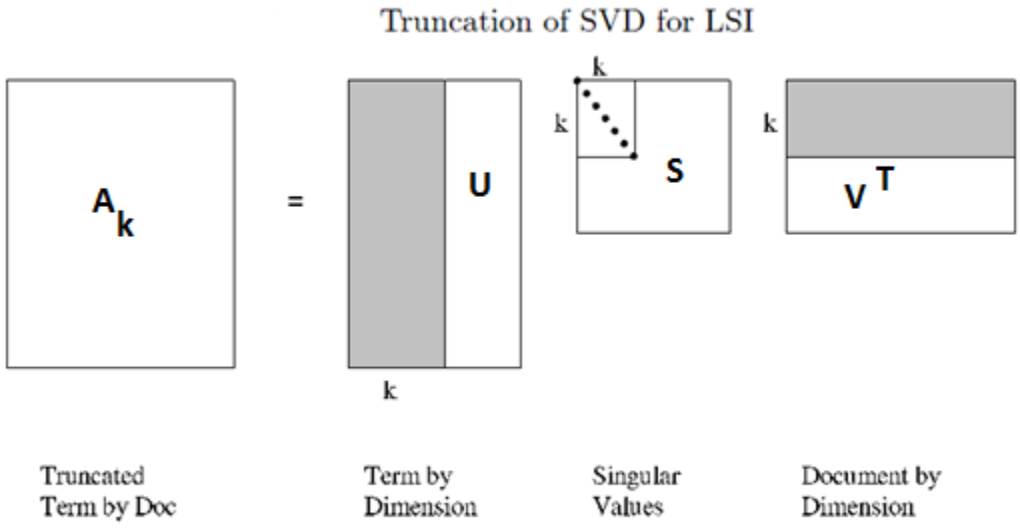

Figure 4.2: SVD of reduced TDM

Step 4: Find the new document vector coordinates in this reduced k-dimensional space. $A_k = U_k S_k V_k^T$

Step 5: Find the new query vector coordinates in the reduced k-dimensional space. $q_{new} = q^T U_k S_k^{-1}$

Note: These are the new coordinates of the query vector in k dimensions and are now different from the original query matrix q given in Step 1.

Step 6: Rank documents in decreasing order of query-document cosine similarities.

SVD can be viewed as a technique for deriving a set of uncorrelated indexing variables or factors; each term and document is represented by its vector of factor values. It is possible for documents with somewhat different profiles of term usage to be mapped into the same vector of factor values. This is the main reason that this method performs well. In fact, the SVD representation, by replacing individual

terms with derived orthogonal factor values, can help to solve the two problems cited precedently.

The amount of dimension reduction, i.e., the choice of k , is critical to our work. Ideally, we want a value of k that is large enough to fit all the real structure in the data, but small enough so that we do not also fit the sampling error or unimportant details. The proper way to make such a choice is an open issue in the factor analytic literature. In practice, we currently use an operational criterion - a value of k which yields good retrieval performance.

If the singular values in S are ordered by size, the first k largest may be kept and the remaining smaller ones set to zero. The product of the resulting matrices is a matrix A_k which is only approximately equal to A , and is of rank k . It can be shown that the new matrix A_k is the matrix of rank k which is closest in the least squares sense to A.

In our case, we use k = 200. The first 10 singular value are 42.8124; 26.2782, 20.7500, 18.9361, 14.5628, 14.4206, 13.6613, 13.2115, 11.8536, 11.4940. 9.32, 6.7355, 5.1338, 3.9262 are the 20th, 50th, 100th, and 200th singular values. We got the following results for T-Mobile:

Cosine ∡	Code & Description
0.7456	519130 internet broadcasting
0.6846	519130 broadcasting exclusively on internet audio
0.6679	454112 auctions internet retail
0.6679	454112 internet auctions retail
0.6498	519130 web broadcasting
0.6229	561422 order taking for clients over the internet
0.6140	519130 search portals internet
0.6091	519130 broadcasting exclusively on internet video
0.6091	519130 video broadcasting exclusively on internet
0.5830	517210 wireless internet service providers except satellite
0.5728	515210 cable broadcasting networks

Table 4.1: Results for T-Mobile, using SVD

Note that here the cosines are bigger than in table 3.4. Also, the LSI values are real numbers while the original term frequencies are integers, which may give us negative cosines suggesting negative correlation.

However, the storage and computaion cost a lot. Using SVD, we can no longer take advantage of the fact that each term occurs in a limited number of documents, which can be exploited by storing only nonzero elements of the sparse term-document matrix. With recent advances in electronic storage media, the storage requirements of LSI are not a critical problem, but the loss of sparseness has other, more serious implications. For example, it took about 5 minutes to run the algorithm when using SVD, whereas it took the time for a click to run when not using SVD.

CHAPTER 5

LOGNORMALITY AND BLACK-SCHOLES FORMULA

After assigning a NAICS code to a company, the objective is now to assign a credit risk rating to this company according to its NAICS code. The credit rating we will compute here is the distance to default. To understand this concept, a review of the lognormality of the stock price, and the Black Scholes Formula is a must.

In the following, we will present the asumption of lognormality of the stock price, then derive some important probabilities and conditional expectations which are the basis of the Black-Scholes formula, and finally derive the Black-Scholes formula.

5.1 Lognormality of the stock price

The lognormal distribution is the probability distribution that arises from the assumption that continuously compounded returns of a stock ($r_t = \ln \frac{S_t}{S_{t-1}}$) are normally distributed.

Let t (denominated in years) be time to expiration of an option to purchase (or sell) a stock with value S_t at time t,
S_0 the stock price at time 0 (starting time),
α the annual mean growth of the stock,
σ standard deviation of the stock price, and
δ the annual dividend yield on the stock.

Assuming that the continuously compounded capital gain from 0 to t, $\ln(\frac{S_t}{S_0})$, is

normally distributed with mean $(\alpha - \delta - \frac{1}{2}\sigma^2)t$ and variance $\sigma^2 t$, :

$$\ln\left(\frac{S_t}{S_0}\right) \sim N[(\alpha - \delta - \frac{1}{2}\sigma^2)t, \sigma^2 t]$$

We can also write

$$\ln\left(\frac{S_t}{S_0}\right) = (\alpha - \delta - \frac{1}{2}\sigma^2)t + \sigma\sqrt{t}Z$$

Or equivalently,

$$S_t = S_0 e^{(\alpha-\delta-\frac{1}{2}\sigma^2)t+\sigma\sqrt{t}Z} \tag{5.1}$$

where $Z \sim N(0,1)$.

One may ask the question why we have the term $-\frac{1}{2}\sigma^2$ in the mean of the lognormal distribution. This follows from the fact that the stock price follows a geometric brownian motion (which is equivalent to the fact that the stock price follows the lognormal distribution).

The derivation above is equivalent to assuming that the stock price (S_t), follows a geometric brownian motion:

$$\frac{dS_t}{S_t} = (\alpha - \delta)dt + \sigma dZ(t) \tag{5.2}$$

where Z(t) is Brownian motion

A stochastic process is a random process that is a function of time. Brownian motion is a stochastic process that is a random walk occurring in continuous time, with movements that are continous rather than discrete. A random walk can be generated by flipping a coin each period and moving one step, with the direction determined by whether the coin is heads or tails. To generate a Brownian motion, we would flip the coins infinitely fast and take infinitesimally small steps at each point. Since all steps are infinitely small, movements are essentially continous.

Brownian motion is a continuous stochastic process, with the following characteristics:

- $Z(0) = 0$
- $Z(t+s) - Z(t)$ is normally distributed with mean 0 and variance s.
- $Z(t+s_1) - Z(t)$ is independant of $Z(t) - Z(t-s_2)$, where $s_1, s_2 > 0$. In other words, nonoverlapping increments are independently distributed.
- Z(t) is continous.

Now that we know what is a Brownian motion, we still need one more important lemma called $Itô$'s Lemma.

Proposition 3.1.1 $Itô$'s Lemma (as given by McDonald [7])
Let the change in the stock price be given by $dS_t = [\alpha(S(t),t) - \delta(S(t),t)]dt + \sigma(S(t),t)dZ(t)$. If C(S(t),t) is twice-differentiable function of S(t), then the change in C is:

$$dC(S(t),t) = C_S dS + \frac{1}{2}C_{SS}(dS)^2 + C_t dt \tag{5.3}$$

Where we use the notation
$C_S = \frac{\partial C}{\partial S}$, $C_{SS} = \frac{\partial^2 C}{\partial S^2}$, $C_t = \frac{\partial C}{\partial t}$,

Let $C(S(t),t) = \ln(S_t)$ in our case. We have:
$C_S = \frac{\partial C}{\partial S} = \frac{1}{S_t}$,
$C_{SS} = \frac{\partial^2 C}{\partial S^2} = \frac{-1}{S_t^2}$,
$C_t = \frac{\partial C}{\partial t} = 0$.

$$\begin{aligned} d\ln(S_t) &= \frac{1}{S_t}dS_t - \frac{1}{2S_t^2}(dS_t)^2 \\ &= \frac{1}{S_t}S_t((\alpha-\delta)dt + \sigma dZ(t)) - \frac{1}{2S_t^2}(\sigma^2 S_t^2 dt) \\ &= (\alpha-\delta)dt + \sigma dZ(t) - \frac{1}{2}\sigma^2 dt \\ &= (\alpha-\delta-\frac{1}{2}\sigma^2)dt + \sigma dZ(t) \end{aligned}$$

This shows why $-\frac{1}{2}\sigma^2$ appears in the mean.

5.2 Probabilities

In this section we will compute some important probabilities which will be usefull later.

If the stock price today is S_0, we can compute the probability that $S_t < K$, where K is some arbitrary number.

$$\begin{aligned} P(S_t < K) &= P(\ln(S_t) < ln(K)) \\ &= P\left(\frac{\ln(S_t) - \ln(S_0) - (\alpha-\delta-\frac{1}{2}\sigma^2)t}{\sigma\sqrt{t}} < \frac{\ln(K) - \ln(S_0) - (\alpha-\delta-\frac{1}{2}\sigma^2)t}{\sigma\sqrt{t}}\right) \\ &= P\left(z < \frac{\ln(K) - \ln(S_0) - (\alpha-\delta-\frac{1}{2}\sigma^2)t}{\sigma\sqrt{t}}\right) \\ &= N\left(\frac{\ln(K) - \ln(S_0) - (\alpha-\delta-\frac{1}{2}\sigma^2)t}{\sigma\sqrt{t}}\right) \end{aligned}$$

Where N is the cumulative distribution function (cdf) of the standard normal distribution.

This can also be written as:

$$P(S_t < K) = N(-\hat{d}_2) \tag{5.4}$$

where $\hat{d}_2 = \frac{\ln(S_0)-\ln(K)+(\alpha-\delta-\frac{1}{2}\sigma^2)t}{\sigma\sqrt{t}}$

From the formula above, one can deduce $P(S_t > K)$ as follows:

$P(S_t > K) = 1 - P(S_t < K)$ or

$$P(S_t > K) = N(\hat{d}_2) \tag{5.5}$$

from symmetry of the normal distribution.

5.3 The Conditional Expected Price

As we said earlier, conditional expectation is one of the basics in the Balck-Scholes formula.

To understand this concept, let us first give some definitions.

Call option: A call option is a contract where the buyer has the right to buy, but not the obligation to do so.

Strike price (K): The strike price, or exercise price, of a call option is what the buyer pays for the asset, if exercised.

Exercise: The exercise of a call option is the act of paying the strike price to receive the asset.

Expiration: The expiration of the option is the date by which the option must either be exercised or it becomes worthless.

Exercise style: the exercise style of the option governs the time at which exercise can occur.

- **European-style option:** Exercise could occur only at expiration.
- **American-style option:** Exercise could occur any time during the life time of the option.
- **Bermudan-style option:** Exercise could occur only during specified periods, but not for the entire life of the option.

The buyer is not obligated to buy the stock, and hence will only exercise if the option payoff is greater than zero.

Purchased call payoff=$\max(0, S_t - K)$.

The seller is said to be the option writer.

Written call payoff = $-\max(0, S_t - K)$.

Put option: A put option is a contract where the seller has the right to sell, but not the obligation.

Purchased put payoff=$\max(0, K - S_t)$.

The seller is said to be the option writer.

Written put payoff = $-\max(0, K - S_t)$.

Now that we are familiar with the terms, we can start computation of conditional expectations.

Given that an option expires in the money (which has a positive payoff), what is the expected stock price? The answer to this question is the conditional expected stock price. For a call option with strike price K, we want to calculate $E(S_t|S_t < K)$, the expected stock price conditional on $S_t < K$

The probability density function of a lognormal with parameters m and v is: $f(x; m; v) = \frac{e^{-(\ln x - m)^2/2v^2}}{xv\sqrt{2\pi}}$

Let first compute the expectation of X, conditional on $X < k$.

we have:

$\frac{1}{v\sqrt{2\pi}N(-\hat{d_2})} \int_0^k e^{-(\ln x - m)^2/2v^2} dx$

$$\begin{aligned} E(X|X<k) &= \int_0^k \frac{xf(x;m,v)}{P(X<k)}dx \\ &= \int_0^k \frac{xf(x;m,v)}{P(X<k)}dx \\ &= \int_0^k \frac{xe^{-(\ln x - m)^2/2v^2}}{xv\sqrt{2\pi}P(X<k)}dx \\ &= \frac{1}{v\sqrt{2\pi}P(X<k)} \int_0^k e^{-(\ln x - m)^2/2v^2} dx \end{aligned}$$

Substitute $y = \ln x - m$:

$x = e^{y+m}$

$dx = e^{y+m}dy$

The bounds of the integral from x to $y = \ln x - m$ are:

$0 \to -\infty$

$k \to \ln k - m$

The expression becomes:

$$\begin{aligned} E(X|X<k) &= \frac{1}{v\sqrt{2\pi}P(X<k)} \int_{-\infty}^{\ln k - m} e^{-(y)^2/2v^2 e^{y+m}} dx \\ &= \frac{e^m}{v\sqrt{2\pi}P(X<k)} \int_{-\infty}^{\ln k - m} e^{-(y^2 - 2v^2 y)/2v^2} dx \end{aligned}$$

Using the fact that $y^2 - 2v^2 y = (y - v^2)^2 - v^4$, we have:

$$\begin{aligned}
E(X|X<k) &= \frac{e^m}{v\sqrt{2\pi}P(X<k)}\int_{-\infty}^{\ln k-m} e^{-((y-v^2)^2-v^4)/2v^2}dx \\
&= \frac{e^m}{v\sqrt{2\pi})P(X<k)}\int_{-\infty}^{\ln k-m} e^{-((y-v^2)^2-v^4)/2v^2}dx \\
&= \frac{e^m}{v\sqrt{2\pi}P(X<k)}\int_{-\infty}^{\ln k-m} e^{-(y-v^2)^2/2v^2}e^{0.5v^2}dx \\
&= \frac{e^{m+0.5v^2}}{P(X<k)}\int_{-\infty}^{\ln k-m}\frac{e^{-(y-v^2)^2/2v^2}}{v\sqrt{2\pi}}dx \\
&= \frac{e^{m+0.5v^2}}{P(X<k)}\int_{-\infty}^{\ln k-m}\frac{e^{-(y-v^2)^2/2v^2}}{v\sqrt{2\pi}}dx \\
&= \frac{e^{m+0.5v^2}N(\frac{\ln k-m-v^2}{v})}{P(X<k)}
\end{aligned}$$

Going back to our initial conditional expectation:

$S_t < K \Leftrightarrow \frac{S_t}{S_0} < \frac{K}{S_0}$ and we can let $k = \frac{K}{S_0}$,

$$\begin{aligned}
\frac{\ln k - m - v^2}{v} &= \frac{\ln K - \ln S_0 - (\alpha - \delta - 0.5\sigma^2)t - \sigma^2 t}{\sigma\sqrt{t}} \\
&= \frac{\ln K - \ln S_0 - (\alpha - \delta + 0.5\sigma^2)t}{\sigma\sqrt{t}} \\
&= \frac{-(\ln S_0 - \ln K + (\alpha - \delta + 0.5\sigma^2)t)}{\sigma\sqrt{t}} \\
&= -\hat{d}_1
\end{aligned}$$

We can now write the expected stock price conditional on $S_t < K$ as follow:

$$E(S_t|S_t<K) = Se^{(\alpha-\delta)t}\frac{N(-\hat{d}_1)}{N(-\hat{d}_2)} \tag{5.6}$$

and

$$E(S_t|S_t>K) = Se^{(\alpha-\delta)t}\frac{N(\hat{d}_1)}{N(\hat{d}_2)} \tag{5.7}$$

We can also write: $\hat{d}_2 = \hat{d}_1 - \sigma\sqrt{t}$

5.4 The Black-Scholes formula

Using the equations we just derived, If we let E^* denote the expectation taken with respect to risk-neutral probabilities (the stock's probability of growth, computed using the risk free interest rate r, in lieu of the expected growth α), and P^* denote those probabilities, the price of a European call option on a stock will be

$$\begin{aligned} C(S_0, K, \sigma, r, t, \delta) &= e^{-rt} E^*(S_t - K | S_t > K) P^*(S_t > K) \\ &= e^{-rt} E^*(S_t | S_t > K) P^*(S_t > K) - e^{-rt} E^*(K | S_t > K) P^*(S_t > K) \\ &= S_0 e^{-\delta t} N(d_1) - K e^{-rt} N(d_2). \end{aligned}$$

The call price is:

$$C(S_0, K, \sigma, r, t, \delta) = S_0 e^{-\delta t} N(d_1) - K e^{-rt} N(d_2). \tag{5.8}$$

Similarly the price of a European put option on a stock will be

$$P(S_0, K, \sigma, r, t, \delta) = K e^{-rt} N(d_2) - S_0 e^{-\delta t} N(d_1). \tag{5.9}$$

CHAPTER 6

HOW TO COMPUTE THE DISTANCE TO DEFAULT (THE MERTON DEFAULT MODEL)

In this section, we try to evaluate credit risk of companies (grouped by sector using NAICS codes) based on their distance-to-default. The distance-to-default measure is defined as the number of standard deviations the asset value is away from the default barrier. The concept of distance-to-default is straightforward. Default of a company occurs if its assets fail to meet liability payments. In general a company will default when its asset value reaches the book value of its total debts, that is, when its market net value reaches zero. Thus the higher the value of the firm's assets, relative to the default barrier, the company would be farther away from default.

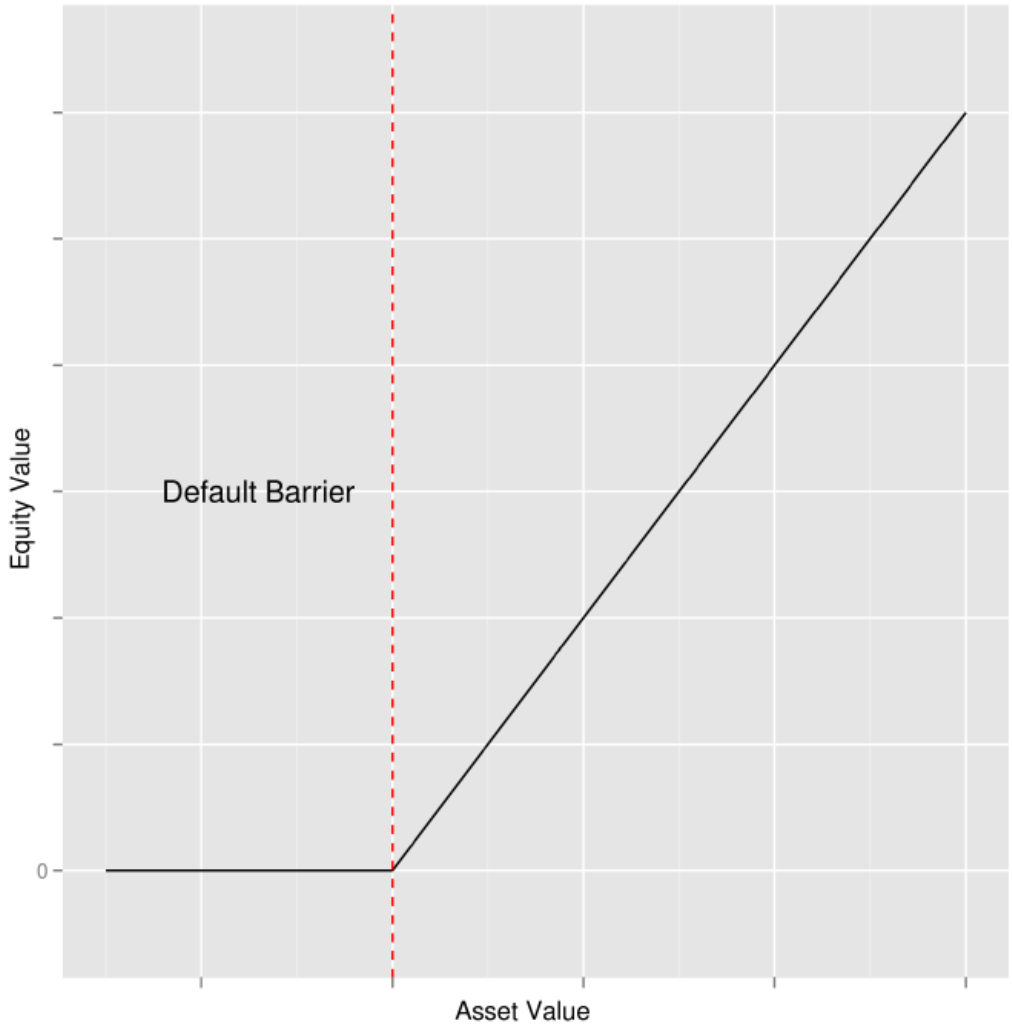

Figure 6.1: How default occurs (plot of $\max(0, A_T - B)$

6.1 Pricing a zero-coupon bond

In this section we introduce basic concepts and terminology related to default in the context of pricing a zero-coupon bond. Suppose that a firm with asset value A_0 issues a zero-coupon bond (a bond that does not pay interest, and pays only the par-value at maturity) maturing at time T, with a promised payment of B. Let B_T denote the market value of the bond at time T. At time T, there are two possibile outcomes:

•$A_T > B$. Since assets are worth more than the repayment owed to bondholders, shareholders will repay bondholders in full, so $B_T = B$. Shareholders' equity will then be worth $A_T - B > 0$

•$A_T < B$. Shareholders will walk away from the firm, surrendering it to bondholders. The value of the bonds at time T is then $B_T = A_T$

Therefore the value of the shareholders' equity at time T, E_T, is

$E_T = \max(0, A_T - B)$

Thus the value of the debt is

$$\begin{aligned} B_T &= \min(A_T, B) \\ &= A_T + \min(0, B - A_T) \\ &= A_T - \max(0, A_T - B) \end{aligned}$$

This says that the bondholders own the firm, but have written a call option to the equity-holders.

In fact, at expiration if the asset value (A_T) is greater than the strike price (B), the equity-holders will exercise the call; then the bondholders receive B and give up the asset which worth more. A different way to write this equation is the following:

$$\begin{aligned} B_T &= \min(A_T, B) \\ &= B + \min(0, A_T - B) \\ &= B - \max(0, B - A_T) \end{aligned}$$

The interpretation of this last equation is that the bondholders own risk-free debt with a payoff equal to B, but have written a put option on the assets with strike price B.

In fact, at expiration if the asset value (A_T) is less than the strike price (B), the equity-holders will exercise the put. They sell the asset (which worth less) to the bondholders.

6.2 Default at Maturity

If we assume that the assets of the firm are lognormally distributed, then we can use the lognormal probabilities in section 3.1 to compute either the risk-neutral or the actual probability that the firm will default. This approach of default modeling is called the Merton model since Merton (1974) used continuous-time methods to provide a model of the credit spread. The Merton default model has in recent years been the basis for credit risk analysis provided by Moody's KMV[1] Assume that the

[1] An analytical tool which provides the Expected Default Frequency (EDF) measure, available on public firms and sovereigns and is the market standard credit risk measure used by financial professionals around the world to assess credit risk. (https://www.creditedge.com)

assets of the firm (noted A), follow the process:

$$\frac{dA}{A} = (\alpha - \delta)dt + \sigma dZ$$

where

α is the expected return on the firm assets and δ is the cash payout made to the claim holders of the firm, σ the volatility and Z is a Wiener process.

Suppose the firm has issued a single zero-coupon bond with promised payment B, that matures at time T and no makes no interim payouts. Default occurs at time T if $A_T < B$. The probability of default at time T, conditional on the value of assets at time t ($t < T$), is

$$\begin{aligned} P(A_T < B|A_t) &= N\left(-\frac{\ln\left(\frac{A_t}{B}\right) + (\alpha - \delta - \frac{1}{2}\sigma^2)(T-t)}{\sigma\sqrt{T-t}}\right) \\ &= N(-\hat{d}_2) \end{aligned}$$

Using the risk-neutral pricing,

$$\begin{aligned} P(A_T < B|A_t) &= N\left(-\frac{\ln\left(\frac{A_t}{B}\right) + (r - \delta - \frac{1}{2}\sigma^2)(T-t)}{\sigma\sqrt{T-t}}\right) \\ &= N(-d_2) \end{aligned}$$

Where r is the risk-free interest rate.

The expression $\hat{d}_2$ is called the distance to default, and measures the size (in standard deviations) of the random shock required to induce default. To understand this interpretation, recall that when assets are lognormally-distributed, the expected log asset value at time T is

$E[\ln(A_T)] = \ln(A_t) + (\alpha - \delta - 0.5\sigma^2)(T-t)$,

Thus, the distance to default is the difference between $E[\ln(A_T)]$ and the default level B, normalized by the standard deviation.

Distance to default (DD) is:

$$\begin{aligned} DD &= \frac{E[\ln(A_T)] - \ln(B)}{\sigma\sqrt{(T-t)}} \\ &= \left(\frac{\ln\left(\frac{A_t}{B}\right) + (\alpha - \delta - \frac{1}{2}\sigma^2)(T-t)}{\sigma\sqrt{T-t}} \right) \\ &= \hat{d_2} \end{aligned}$$

The default probability which measure how likely the company will default can be computed as $N(-DD) = 1 - N(DD)$.
As we may expect, this probability will tend to zero when we are far from default.

For computational purposes, we will use the risk-neutral pricing method.
Distance to default (DD) is:

$$\begin{aligned} DD &= \frac{E[\ln(A_T)] - \ln(B)}{\sigma\sqrt{(T-t)}} \\ &= \frac{\ln\left(\frac{A_t}{B}\right) + (r - \delta - \frac{1}{2}\sigma^2)(T-t)}{\sigma\sqrt{T-t}} \\ &= d_2 \end{aligned}$$

Letting t=0, we have:

$$\begin{aligned} DD &= \frac{E[\ln(A_T)] - \ln(B)}{\sigma\sqrt{(T)}} \\ &= \left(\frac{\ln\left(\frac{A_0}{B}\right) + (r - \delta - \frac{1}{2}\sigma^2)(T)}{\sigma\sqrt{T}} \right) \\ &= d_2 \end{aligned}$$

Now we can understand the negative distance. If

$$
\begin{aligned}
DD \leq 0 & \\
& \Leftrightarrow \quad \ln\left(\frac{A_0}{B}\right) + (r - \delta - \frac{1}{2}\sigma^2)(T) \leq 0 \\
& \Leftrightarrow \quad \ln A_0 \leq \ln B - (r - \delta - \frac{1}{2}\sigma^2)(T) \\
& \Leftrightarrow \quad A_0 \leq Be^{-(r-\delta-\frac{1}{2}\sigma^2)(T)}
\end{aligned}
$$

That means at the time t = 0 (or at some other specified time t), we already know that the company will default. For such company, it will be difficult for them to borrow money unless they have recently invested in major capital improvement.

Using the formula above, we computed DD using data from Yahoo Finance (3600 publicly traded companies with stock price and its standard deviation, number of shares, and total debt) we calculated the distance to default for each company using the three-month Treasury Bill rate as the risk free interest rate.

We then obtained distance-to-default amongst companies with in two digit NAICS sector codes, and used the median as an indicator for each sector, as shown below.

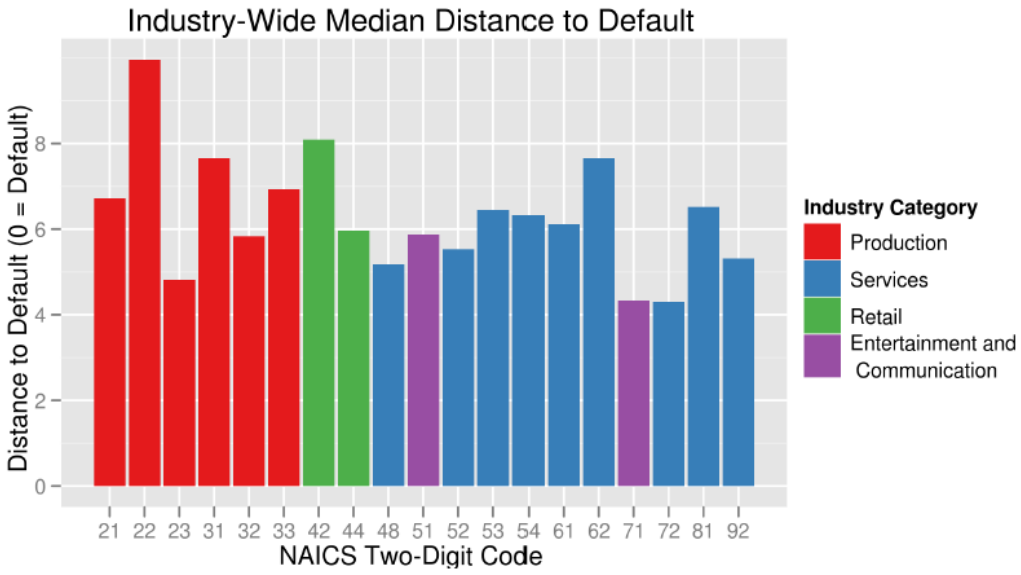

Figure 6.2: DD for company using NAICS 2-digit code

21 Mining, Quarrying, and Oil and Gas Extraction

22 Utilities

23 Construction

31, 32 & 33 Manufacturing

42 Wholesale Trade

44 Retail Trade

48 Transportation and Warehousing

51 Information

52 Finance and Insurance

53 Real Estate and Rental and Leasing

54 Professional, Scientific, and Technical Services

61 Educational Services

62 Health Care and Social Assistance

71 Arts, Entertainment, and Recreation

72 Accommodation and Food Services

81 Other Services (except Public Administration)

92 Public Administration

Table 6.1: Sectors codes

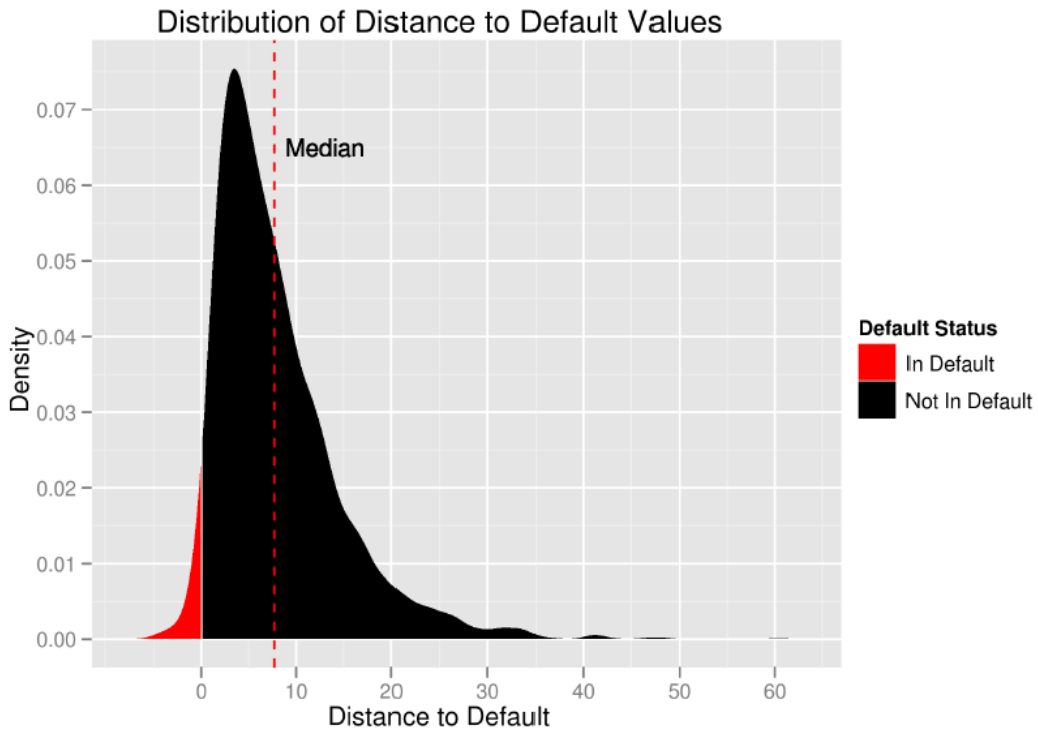

Figure 6.3: Distribution of DD for the same sectors as fig 6.3.2

This graph shows that some companies are in default. They won't be able to pay they debt.

CHAPTER 7

CONCLUSION AND RECOMMENDATIONS FOR FUTURE WORK

We conclude that the second approach is the best among all three because the our results showed that.

The advantages of the LSI are:

As output, it provides a list of potential NAICS code matches which could be all valid because a firm can have multiple NAICS codes (e.g., Georgia Southern University can be a school, an athletic center, etc.)

It is a powerful and automatic searching technique in the sense that it only needs the company description to be entered to perform its code searching procedure.

However, it required:

A more accurate description of the company should be done. In other words, the key words in the description should match the key words in the Term to Document Matrix.

For a company to use this system, it will also be necessary to hire an analyst who has a bit of knowledge in MatLab, and also a MatLab license.

Notice that TMG has a module that can compute and retrieve the NAICS but the output is an html file (showing the rank of column of matrix A and the angle). It also has the flexibility of choosing the SVD or not.

Further work might focus on the third approach. This can be done by describing the 1175 codes using the key words used to describe the same code in the 19,720 codes.

This work may include how to omit some irrelevant words. In fact, in our example, the word athletic shouldn't have been included as a key word because in the two

descriptions, the phrase is “except Athletics”.

Some study may focus on how to compute the correlation between the NAICS codes.This can be done using SVD.

To compute the Distance to Default, we used the implied volatility. A comparative study may use stochastic volatility.

Appendix A

MATLAB CODE FOR COMPUTING SIMILARITIES

load Q.mat;

load A.mat;

load titles.mat;

L = titles;

$[m\ n] = size(A)$;

cos_tol = 0.25;

j=1;

% compute the cosine or similarities

for i=1:n

C_1 = A(:,i);

cos = (C_1'*Q)/(norm(C_1)*norm(Q));

if cos $\geq$ cos_tol

T(j)=i;

C(j) = cos;

j=j+1;

end

end

$[R_1\ R_2]$ = sort(C, 'descend');

ii = T(R_2);

tit_list = titles(ii);

R1

% Build description list

```
sprintf('The ordered list of NAICS suggestions for BedBathBeyond are:')
for i = 1:length(ii)
dd=tit_listi;
disp(dd(2,:))
end
```

Appendix B

MATLAB CODE FOR COMPUTING SIMILARITIES USING SVD

```
load Query-tmb.mat;
load A.mat;
load titles.mat;
load Ssvd.mat;
load Usvd.mat;
load Vsvd.mat;
S = Ssvd;
U = Usvd;
V = Vsvd;
L = titles;
cos_tol = 0.60;
j=1;
B = S*V';
n = 19720;
Q = Q'*U*inv(S);
Q=Q';
for i=1:n
C1 = (U*B(:,i));
D = C1'*U*inv(S);
D=D';
cos = (D'*Q)/(norm(D)*norm(Q));
if cos ¿= cos_tol
T(j)=i;
```

```
C(j) = cos;
j=j+1;
end
end
% [R1 R2]= sort(T);
% R3 = find(R2,10, 'last');
% R4 = L(R3);


    [R1 R2] = sort(C, 'descend');
ii = T(R2);
tit_list = titles(ii);
R1
% build description list


    for i = 1:length(ii)
dd=tit_listi;
disp(dd(2,:))
end
```

REFERENCES

[1] Bodie Z. et al., *Investments,* 8e edition, Mc Graw Hill, 2009

[2] Crosbie P. , J. Bohn, *Modeling Default Risk,* Moody's KMV Company, 2003.

[3] Deerwester S. and al., *Indexing by Latent Semantic Analysis,* Journal of the American Society of Information Science 41 (6), 391-407, 1990.

[4] Garcia E., *Latent Semantic Indexing (LSI), A Fast Track Tutorial*, 2006.

[5] Han Liu & Mbaga Nzabakurana, *NAICS Code Assignation*, unpliblished Master report, 2011.

[6] Kontostathis A. and Pottenger W. M., *A Framework for Understanding Latent Semantic Indexing (LSI) Performance*

[7] McDonald R. L., *Derivatives Markets, second edition,* Pearson Addison Wesley, 2006.

[8] Santomero M. Anthony, *Commercial Bank Risk Management: an Analysis of the Process,* The Wharton School, 1997.

[9] Han X., Ouedraogo J., VanderPlas S., Wang A., Zeng Y. *IMSM 2011: Credit Risk Quantification*, July 8, 2011.

[10] Zeimpekis D., Gallopoulos E., *TMG: A MATLAB Toolbox for Generating Term-Document Matrices from Text Collections.*

[11] Zeimpekis D., Gallopoulos E., *Text to Matrix Generator: Users Guide*, 2008.

[12] *NAICS Main page*, August 17, 2011. US Census Bureau retrieved from http://www.census.gov/eos/www/naics.