Journal of Mathematics and Modeling in Finance (JMMF)
Vol. 1, No. 2, Summer & Autumn 2021
Research paper

Spectral graph embedding for dimension reduction in financial risk assessment

Hossein Teimoori Faal[1], **Meyssam Bagheri**[2]

[1] Department of Mathematics and Computer Science, Allameh Tabataba'i University, Tehran, Iran
hossein.teimoori@atu.ac.ir

[2] Department of Mathematics and Computer Science, Allameh Tabataba'i University, Tehran, Iran
m_bagheri98@atu.ac.ir

Abstract:
The economic downturn in recent years has had a significant negative impact on corporates performance. In the last two years, as in the last years of 2010s, many companies have been influenced by the economic conditions and some have gone bankrupt. This has led to an increase in companies' financial risk. One of the significant branches of financial risk is the *company's credit risk*. Lenders and investors attach great importance to determining a company's credit risk when granting a credit facility. Credit risk means the possibility of default on repayment of facilities received by a company. There are various models for assessing credit risk using statistical models or machine learning.
In this paper, we will investigate the machine learning task of the binary classification of firms into bankrupt and healthy based on the *spectral graph theory*. We first construct an *adjacency graph* from a list of firms with their corresponding *feature vectors*. Next, we first embed this graph into a one-dimensional Euclidean space and then into a two-dimensional Euclidean space to obtain two lower-dimensional representations of the original data points. Finally, we apply the *support vector machine* and the *multi-layer perceptron* neural network techniques to proceed binary *node classification*. The results of the proposed method on the given dataset (selected firms of Tehran stock exchange market) show a comparative advantage over PCA method of *dimension reduction*. Finally, we conclude the paper with some discussions on further research directions.

Keywords: Spectral graph embedding, Principle component analysis, Financial risk assessment, Affinity matrix, Bankruptcy.
Classification: 05C50, 15A21, 90C27.

1 Introduction

A challenging problem in data mining is to understand, interpret and analyze a dataset which lies in a high-dimensional space where each instance is represented by many features. The key to this phenomenon is the fact that only a few features are *crucial* to analyze the given machine learning task or indeed our data points actually lie on a lower-dimensional manifold. We recall that a *credit risk* is an as-

[1]Corresponding author
Received: 2021-06-29 Accepted: 2021-08-06
http://dx.doi.org/10.22054/jmmf.2021.13841

sessment of the likelihood that a borrower, whether a firm or an individual, might not be able to pay back the money loaned. However, it's not only people that have credit risk, investments such as bonds, also bear it.
The credit risk tells investors how risky it is to invest in any particular asset. The higher the risk, the higher the chances of losing money on the investment, and vice-versa. When you get a loan, your credit risk is calculated, but when you are thinking of investing, you need to calculate the credit risk of the investment itself. Here our main goal is to study *financial risk assessment* of firms in stock markets and to classify firms into the bankrupt and healthy ones based on a *dimension reduction* of the high-dimensional space of feature vectors associated with each firm. Now to reduce the dimension of the dataset, one can either select a subset of features as the set of important features using *feature selection* techniques like wrapper and filter methods or we can transform our dataset into a lower-dimensional space while preserving meaningful structure of the original dataset. There are several interesting mathematical models for the given row data. The appropriate choices of the model is one of the key points in any machine learning task. One of the important discrete structures that forms the basis of formulating many real-life problems into abstract language of discrete mathematics is a *graph*.
A finite and simple graph is a suitable discrete model to represent binary relationship among a set of data points. Such a graph data will usually have a high dimension. Features associated to each firm obtained based on its financial ratios can be used to evaluate the financial situation of that firm. Hence, one can use these features to classify companies as bankrupt or healthy . The assessment of financial risk here will be focused on spectral methods based on graph embedding into the real line or the plane.
Investors usually can use a number of financial risk ratios to assess a firms' prospects. There are several different models of *credit risk evaluation* using statistical or machine learning models. Perhaps the first model of evaluating the financial risk is the *Fitzpatrick model* [14] which is used as a classical model. Thus, we can associate a feature vector with any firm i ($i = 1, 2, \ldots, n$), as follows

$$\mathbf{x}_i = (r_{i,1}, r_{i,2}, \ldots, r_{i,m}).$$

For example, a firm in Tehran stock exchange market may have the following five financial ratios such as net profit margin, return on asset, inventory turnover, receivable collection period, and debt ratio. Then we have

$$\mathbf{x}_i = (0.4522, 0.7565, 0.2895, 0.2171, 0.5989).$$

From now on, for the simplicity of the notation, we will denote the set of numbers $\{1, 2, \ldots, n\}$ by $[n]$. We next very briefly review the basic transform-based techniques, including principle component analysis (PCA for short) and four classic nonlinear data reduction techniques: isomap, locally linear embedding (LLE for

short), Laplacian eigenmap or Laplacian embedding and local tangent space alignment (LTSA for short). In all of these methods an eigendecomposition technique is used to get a lower-dimensional embedding of the original dataset that is guaranteed to achieve a *global optimality*.

We recall that PCA [10] is a *linear* technique that preserves the largest variance in the dataset while decorrelating the reduced dataset. To be more precise, an eigenvalue problem related to the *covariance matrix* $\mathbf{C}$ of the dataset is obtained as $\mathbf{Cu} = \lambda\mathbf{u}$. Indeed, the eigenvectors, $\mathbf{u}$, correspond to the significant eigenvalues, λ , form a *basis* for a linear transformation that optimally maximizes variance in the original dataset. Then, the lower dimensional representation of the dataset is expressed by the matrix equation $\mathbf{Y} = \mathbf{Xu}$ and the eigenvalues can be used to determine the dimension d of the lower-dimensional representation of the original data points.

Next, we recall that the Isomap technique [5] preserves pairwise *geodesic distances* between data points. It begins by constructing a graph so-called the *adjacency graph of the dataset* G based on the *neighborhoods information*. It is important to note that those neighborhoods can be either the k-nearest neighbors or points which lie within ϵ-neighbors. Next, the geodesic distances [6] between all pairs of points are estimated by calculating their shortest path distances over the (adjacency) graph. More precisely, if we assume that $D_G = \{d_G(i,j)\}_{i,j\in[n]}$ is the matrix of geodesic distances, where $d_G(i,j)$ is the distance between points i and j. Then Isomap constructs an embedding into a d-dimensional Euclidean space such that the pairwise Euclidean distances between points in this space approximate the geodesic distances in the input space. Let $D_Y = \{d_Y(i,j)\}_{i,j\in[n]}$ be the Euclidean distance matrix and $d_Y(i,j) = \parallel Y_i - Y_j \parallel_2$. Now the ultimate goal is to minimize the cost function $\parallel \tau(D_G) - \tau(D_Y) \parallel_2$, where the function performs double centering on the matrix to support efficient optimization. The optimal solution is found by solving the eigendecomposition of (D_G). Then, the Y coordinates are computed based on the d largest eigenvalues and their corresponding eigenvectors.

The LLE method [3] preserves the reconstruction $\omega_{i,j}$ that are used to describe a data point X_i as as a linear combination of its neighborhoods X_j, $j \in N(i)$, where $N(i)$ stands for the set of neighbors of the point with index i. The optimal weights for each i are obtained by solving the following minimizing problem

$$\min_{\omega} \Big\{ \parallel X_i - \sum_{j\in N(i)} \omega_{i,j} X_j \parallel_2^2 \;\Big|\; \sum_{j\in N(i)} \omega_{i,j} = 1 \Big\}.$$

The LLE method assumes that the manifold is *locally linear* and hence the reconstruction weights are *invariant* in the low-dimensional space. The embedding Y of LLE is the solution of the minimizing the cost function $\sum_i \parallel Y_i - \sum_j w_{i,j} Y_j \parallel$, in which the matrix $\mathbf{W} = (w_{i,j})_{i,j\in[n]}$ is called the *reconstruction weight matrix* with entries $w_{i,j} = 0$ if $j \notin N(i)$ and $w_{i,j} = \omega_{i,j}$ otherwise. The key point is that Y can be obtained from the minimization of the eigenvalue-eigenvector problem for the

matrix $\mathbf{M} = (\mathbf{I} - \mathbf{W})^t(\mathbf{I} - \mathbf{W})$.
The LTSA method [7] applies the method of PCA on the neighbors of each data point which results in a local tangent space that represent the local geometry of the space. The space is denoted by local coordinates $\theta^i_j,\ j = 1, 2, \ldots, k$ that are k-nearest neighborhoods of the point with index i ($i = 1, 2, \ldots, n$). Then, the local tangent spaces are aligned to make a *global coordinate system* of the underlying manifold. The local geometry based on local coordinates must be preserved by the global coordinate. Thus, for building the global coordinates $Y_i,\ i = 1, 2, \ldots, n$ in low-dimensional feature space, LTSA seeks to find the local affine transformations L_i , to minimize the *reconstruction errors*, $\sum_k \parallel E \parallel^2 = \sum_k \parallel Y_i(I - \frac{1}{k}ee^t) - L_i\Theta_i \parallel^2$, in which I is the identity matrix, e is a column vector of ones and $\Theta_i = [\theta^i_1, \theta^i_2, \ldots, \theta^i_k]$.
We finally recall the method of *Laplacian eigenmaps* or Laplacian embedding which is the main approach of this paper. This method provides a lower-dimensional representation of the original data points based on the idea of minimizing the *weighted distances* between a data point and other data points within an ϵ-neighborhood of that data point (or k-nearest neighborhoods) [15]. Indeed, this interesting spectral approach can be regarded as *graph-theoretical* version of *Least square* method in engineering disciplines. The main idea is to assign a weight $w_{i,j}$ between each pairs of connected vertices $\{i, j\} \in E_G$ which shows that how these two vertices are correlated. Then, we build a weighted version of *adjacency matrix* which we denote it by $\mathbf{W}_G = (w_{i,j})_{i,j\in[n]}$, corresponding to the adjacency graph of the set of original data points simply by letting

$$w_{i,j} = e^{-\frac{\parallel \mathbf{x}_i - \mathbf{x}_j \parallel_2^2}{t}},$$

where $t = 2\sigma^2$ and σ is the standard deviation of the *Gaussian kernel*; that is a parameter controlling the correlation between pairs of vertices.
We also define a (weighted) degree matrix $D_G = (d_{i,j})$ by assigning only a non-zero value to ith diagonal ($i \in [n]$) by putting $d_{i,i} = \sum_{j=1}^{n} w_{i,j}$. Finally, the (combinatorial) Laplacian of the corresponding dataset is defined as $L_G = D_G - W_G$.
For data on graphs, by the embedding of a graph G with associated vertex-signal $\mathbf{u} = (u_1, u_2, \ldots, u_n) \in \mathbb{R}^n$ into the real line $\mathbb{R}$, we mean a task of finding a real-valued function $f : V(G) \mapsto \mathbb{R}$ such that each vertex $v \in V(G)$ maps into a (unique) real number $f(v)$ (see Fig 1). Indeed, this function can be a Laplacian *eigenvector* corresponding to the second smallest Laplacian *eigenvalue* of the graph G.

One can similarly embed a graph data (a graph with its vertex-signal) into a plane, that is an Euclidean space $\mathbb{R}^2$ of dimension two. In this case, we need to consider the pair of the eigenvectors corresponding to the second and the third smallest Laplacian eigenvalues of G. Thus, the ith coordinate of the second and the third Laplacian eigenvectors of G can be considered as the first and the second coordinates of the embedded ith vector.

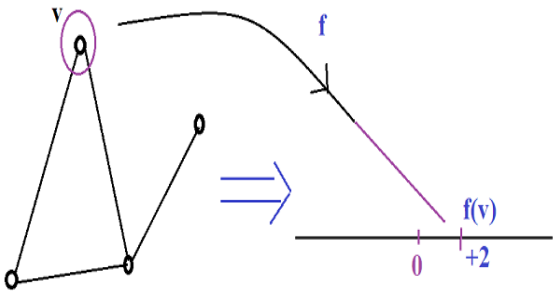

Figure 1: The Vertex-Embedding in Dimension One

In this respect, we will use the *spectral embedding* besides the *principle component analysis* to obtain a *binary classification* of firms into two main classes of *bankrupt* and *healthy*. To do this, we first build a similarity graph based on the k-nearest neighborhood algorithm from the set of n firms and their feature vectors. Then, this similarity graph G is embedded into the plane based on the idea of the eigenvectors corresponding to the second and the third smallest Laplacian eigenvalues of the graph G. This method is called the *Linear extension embedding* of the graph G.

The paper will be organized, as follows. In the next section, we review selected important papers related to *financial risk assessments* based on the *node-classification* of graphs associated with financial datasets. Next, in Section 3, we quickly review the basics of the *spectral graph embedding* in general as an applied tool for node-classification problems in machine learning. Then, in Section 4, we specifically concentrate on the proposed method of this paper for embedding the adjacency graph of a collection of firms into the real line and the plane. In Section 5, we obtain the results of the implementation of the proposed model for both cases of one and two-dimensional embeddings. We also show a compararative advantage of the *spectral graph embedding* over *principle component analysis* as the classical dimension reduction method. Finally, we conclude the paper with some discussions about future research works.

2 Literature Review

In this section, we quickly review the most important references in the area of spectral methods for binary classification of firms into bankrupt and healthy.
He and Niyogi in 2003 [10] invented a linear method of dimension reduction which is called the *linear preserving projection* where a graph is built containing neighboring vertices information. Then, based on the Laplacian matrix of this graph, its mapping matrix to lower-dimensional spaces is calculated. Deng Cayi et al. in 2008 [11] for recognizing images of human faces used a smoothing discrete Laplacian model for reducing the dimension of images. They first construct a graph related to the whole set of images in which each image represented by a vertex of that graph. Then, using the LPP method, they calculated its affinity matrix. Next, based on an approximation of Neumann's discrete Laplacian they tried to smooth the underlying graph and then by solving the corresponding eigenvalue problem they found the optimal eigenvector that can be used to embed the graph of images into the real line.
Kouki and Elkhaldi in 2011 [12] have shown the preference of MDA and RA model in compare with neural networks using the data of Tunisian firms. They used a three layer perceptron for implementing the research model and came to the conclusion that these neural networks for short-time periods have better performances while having worst performance for long-term periods of MDA and LR models.
In 2016 Brédart and Cultrera [13], designated a binary logical model based on financial ratios for predicting the bankruptcy of firms. They used financial ratios as independent variables in this logical model to optimally determine the bankruptcy and healthiness of firms. They showed that the financial ratios *portability* and *liquidity* with a good accuracy determine the bankruptcy of firms.
Most of the intelligent methods that are used in the area of non-parametric models are neural networks, fuzzy set theory, decision tree, case-based reasoning, support vector machine and so on. The research study of Blanco et al. in 2013 [15] showed that the multi-layer perceptron can be performed better than *statistical learning* models such as LDA, LR and QAD for reducing the expenses of wrong classification of firms into bankrupt and healthy. Rafiei et al. in 2011 [16] showed the *multi-layer perceptron* model in preference to *multivariate discriminant analysis* model for predicting the bankruptcy of Iranian firms.

3 Spectral Graph Embedding

In this section, we are going through the details of the main approach of this paper based on the idea of Laplacian eigenmaps method.
Now, we assume that n firms in a stock exchange market are considered where we associate an m-dimensional feature vector to each of them. Therefore, we have a rectangular matrix $\mathbf{X} \in \mathbb{R}^{m\times n}$ which has the property that its rows represent the

feature vectors and its columns indicates the firms.
There exist many methods to embed a graph into a low-dimensional space like line, plane and even Euclidean 3-dimensional space, but here we will concentrate on a method which is called the *spectral graph embedding*. Next, we explain this method in details.
We let $G = (V, E)$ be a finite, simple and undirected graph with n vertices in which each vertex v_i represent the firm indexed by i. We also assume that our graph is a *vertex-weighted* graph in which the weight of a vertex i corresponds to the feature vector $\mathbf{x}_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,m})$.
There are several standard ways to build such a graph on our original data points like *ϵ-neighborhood*, *k-nearest neighborhood* and *Delanoy* graphs to name a few. Here, we will concentrate on the method of k-nearest neighborhood of a graph G which is based on the well-known *KNN* algorithm [8] .
In this method of construction, we first obtain a list of pairwise Euclidean distances among vertices; that is, the distance between their corresponding feature vectors. Next, there are two ways to build such a k-nearest neighborhood graph for a given k. In the first approach, we connect the ith vertex to the jth vertex whenever this vertex is among the first k nearest vertices to the jth one *or* the jth vertex is among the first k nearest vertices to the ith vertex. The graph obtained this way is called the (standard) k nearest neighborhood graph of the original data set. In the second method, the ith vertex is connected to the jth vertex whenever this vertex is among first k nearsest vertices to the jth one *and* the jth vertex is among the first k nearest vertices to the ith vertex. This latter graph is called the *mutual* k-nearest neighborhood graph of the original data set.
In this paper, we will use the first method and we take k to be the number 5. There are several original ideas to compute the affinity matrix which we are going to describe it in the next subsection.

3.1 Computing Affinity Matrices

Here is a list of some interesting and important affinity matrices that are used in graph mining literatures.

(i) **Linear Discriminant Analysis:** In this method it is supposed that we have m data points which belong to t classes $\mathcal{C}_1, \mathcal{C}_2, \ldots, \mathcal{C}_t$. Hence, if we assume that the class $\mathcal{C}_t$ has m_t data points, $m_1 + m_2 + \ldots + m_t = m$, then the affinity matrix is defined, as follows

$$W_{i,j}^{LDA} = \begin{cases} \frac{1}{m_t}, & if \quad \mathbf{x}_i, \mathbf{x}_j \in \mathcal{C}_t, \\ 0, & otherwise. \end{cases}$$

(ii) **Linear Projection Preserving:** If we define $N_k(\mathbf{x}_i)$ as the set of k nearest neighborhoods of the ith vertex, having the feature vector $\mathbf{x}_i$, then one can

define the corresponding affinity matrix by

$$W_{i,j}^{LPP} = \begin{cases} e^{-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|_2^2}{2\sigma^2}}, & if \quad \mathbf{x}_i \in N_k(\mathbf{x}_j), \mathbf{x}_j \in N_k(\mathbf{x}_i), \\ 0, & otherwise. \end{cases}$$

Note that in the above formulation, the parameter σ is called the *kernel width*.

4 Graph Spectral Embedding in dimension one and two

The main goal of *graph embedding* is to represent each vertex with high-dimensional feature vector (or weight) in lower-dimensional spaces like line, plane or even three-dimensional Euclidean space so that the similarity between vertices will preserve. Indeed, the similarity between pairs of vertices will be measured based on their feature vectors.

Now, we will assume that the vector $\mathbf{y} = (y_1, y_2, \ldots, y_m)$ is a vector that maps the graph into the real line. The optimal vector for $\mathbf{y}$, is obtained by the minimization of the following quadratic form

$$\sum_{\{i,j\} \in E(G)} W_{i,j} \Big(y_i - y_j \Big)^2,$$

which can be considered as the *graph least square* problem.

Now, it is an easy problem in *spectral graph theory* to show that

$$\sum_{\{i,j\} \in E(G)} W_{i,j} \Big(y_i - y_j \Big)^2 = \mathbf{y}^t L_G \mathbf{y}. \tag{1}$$

Here, the matrix $L_G = D_G - W_G$ is the *combinatorial Laplacian* matrix of the graph G. We recall that the matrix W_G is called the *weighted adjacency* matrix of the graph G. The diagonal matrix D_G is called the *weighted degree* matrix of G. For more details see the paper [4].

Considering the above discussions, our optimization problem can finally be formulated as follows.

$$\mathbf{y}^\star = \arg \min_{\mathbf{y} D_G \mathbf{y} = 1} (\mathbf{y} L_G \mathbf{y}) = \arg \min_{\mathbf{y}=1} \frac{\mathbf{y} L_G \mathbf{y}}{\mathbf{y} D_G \mathbf{y}}. \tag{2}$$

It is noteworthy to mention that the constraint $\mathbf{y} D_G \mathbf{y} = 1$ has been added for the sake of *scalability* and to avoid the *degenerate* cases.

The difficulty of solving the optimization problem 2 is relay on the fact that for a very large number of firms (large values of n) the problem may be *impratical*.

One simple way to resolve this problem, based on the definition of $L_G = D_G - W_G$, is rather solving the following dual optimization problem

$$\mathbf{y}^{\star} = \arg \min_{\mathbf{y} D_G \mathbf{y}=1} (\mathbf{y} W_G \mathbf{y}) = \arg \max_{\mathbf{y}=1} \frac{\mathbf{y} W_G \mathbf{y}}{\mathbf{y} D_G \mathbf{y}}. \tag{3}$$

Now, we use the fact that the maximization problem of the *Rayleigh Quotient* $R(G, \mathbf{y}) = \frac{\mathbf{y} W_G \mathbf{y}}{\mathbf{y} D_G \mathbf{y}}$ in 3 is equivalent to finding the eigenvector corresponding to the maximum eigenvalue of the *generalized eigenvalue problem* $W_G \mathbf{y} = \lambda D_G \mathbf{y}$ based on the well-known *Courant-Fisher* well-known theorem.
Unfortunately, we still have the problem of the *curse of dimnensionality* for the cases that n is a very large number. To overcome this difficulty, we note that the number of features in this kind of problems is usually very small in compare with n. Hence, the method of *linearly preserving projection* (LPP for short) can be employed to finally solve the problem. To do so, we define the linear transformation $f : \mathbb{R}^n \mapsto \mathbb{R}^m$ by $y_i = f(\mathbf{x}_i) = \mathbf{u}^t \mathbf{x}_i$. where the vector $\mathbf{u} \in \mathbb{R}^m$ should be determined.
The pointwise relations $y_i = \mathbf{u}^t \mathbf{x}_i$ can be put together to find a new change of variable (or dimension reduction) $\mathbf{y} = \mathbf{X}^t \mathbf{u}$ form the space of n-dimensional vectors to the space of m-dimensional vectors. Thus, finally the problem of finding an optimal *embeddign vector* $\mathbf{y} \in \mathbb{R}^n$ reduces to the problem of finding another optimal vector $\mathbf{u} \in \mathbb{R}^m$, as the solution of the following *maximization* problem

$$\mathbf{u}^{\star} = \arg \max_{\|\mathbf{y}\|_2=1} \frac{\mathbf{y} W_G \mathbf{y}}{\mathbf{y} D_G \mathbf{y}} = \arg \max_{\|\mathbf{u}\|_2=1} \frac{\mathbf{u}^t \mathbf{X} W_G \mathbf{X}^t \mathbf{u}}{\mathbf{u}^t \mathbf{X} D_G \mathbf{X}^t \mathbf{u}}. \tag{4}$$

Clearly, the above problem again is equivalent to compute the eigenvector corresponding to the maximum eigenvalue of the following the generalized eigenvalue problem

$$\mathbf{X} W_G \mathbf{X}^t \mathbf{u} = \mu \mathbf{X} D_G \mathbf{X}^t \mathbf{u}. \tag{5}$$

4.1 Discretized Laplacian Smoothing

One of the main issues regarding *machine learning* tasks in computer science is the problem of depedencies of the output of the learning model on changes in *initial values* . For example in *image processing* when we have an *edge detection* task, the lack of smoothing the small changes will result in disability to detect the edges of your images.
In our model of the evaluation of the financial risks of firms based on their feature vectors, the lack of recognizing small changes in features may result in detecting changes in the main characteristics of firms and finally will impose not enough *precision* in classification into bankrupt firms and *healthy* firms. Thus, we are going to add *smoothing analysis* to our case study to make our model more *robust* to small changes of important features of each firm in our senario.
Here, we assume that a smooth function f is defined on the features of a given firm and the *Laplacian operator* $\mathcal{L}$ acts on f as follows

$$\mathcal{L}f = \sum_{i=1}^{n} \frac{\partial^2 f}{\partial x^2}. \tag{6}$$

Therefore, using the definition of the *second derivative* of a function, one can use the formula (6) to find some good approximation of the linear differential operator $\mathcal{L}$.

Indeed, we have several interesting options to choose a *discretized version* of the above Laplacian operator. Our choice here will be the *Neumann discretized Laplacian* with normalized factor 4 which is widely used in mechanical engineering and machine learning tasks, as follows

$$\Delta = \frac{1}{4}\begin{bmatrix} 1 & -1 & 0 & \cdots & 0 & 0 & 0 \\ 1 & -2 & 1 & \cdots & 0 & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & -2 & 1 \\ 0 & 0 & 0 & \cdots & 0 & -1 & 1 \end{bmatrix}. \tag{7}$$

Thus, considering the smoothing by discretized Laplacian in formula (7), we can smooth the optimization problem (4) by

$$\mathbf{u}^{\star} = \arg \max_{\|\mathbf{u}\|_2=1} \frac{\mathbf{u}^t \mathbf{X} W_G \mathbf{X}^t \mathbf{u}}{(1-\beta)\mathbf{u}^t \mathbf{X} D_G \mathbf{X}^t \mathbf{u} + \beta \mathbf{u}^t \boldsymbol{\Delta}^t \boldsymbol{\Delta} \mathbf{u}}. \tag{8}$$

Here the *parameter* β is a controlling parameter of the model which attains its value on the open interval $(0, 1)$.

It is obvious now that our main *smoothed formulation* (8) is equivalent to the *generalized eigenvalue-eigenvector* problem

$$\mathbf{X} W_G \mathbf{X}^t \mathbf{u} = \theta_G \Big((1-\beta)\mathbf{X} D_G \mathbf{X}^t + \beta \boldsymbol{\Delta}^t \boldsymbol{\Delta} \Big) \mathbf{u}. \tag{9}$$

4.2 Embedding a graph into the plane

In this section, we briefly explain how to embed a graph into the two-dimensional Euclidean space; that is, the plane $\mathbb{R}^2$.

By embedding a graph $G = (V, E)$ into the plane $\mathbb{R}^2$, we mean an injective mapping form the vertex set V into the plane such that each vertex $v \in V$ maps injectively to a (unique) point $(\mathbf{x}(v), \mathbf{y}(v))$.

Similar to what we did in the embedding of G into the one-dimensional real line, it is enough to find a solution to the following *minimization* problem

$$\arg\min \sum_{\{u,v\} \in E(G)} \left|\left| \begin{pmatrix} \mathbf{x}(u) \\ \mathbf{y}(v) \end{pmatrix} - \begin{pmatrix} \mathbf{x}(u) \\ \mathbf{y}(v) \end{pmatrix} \right|\right|_2^2. \tag{10}$$

Indeed, one can simply show that this problem can be decomposed into two one-dimensional cases. To be more precise, the problem (10) is equivalent to

$$\begin{aligned}\arg\min \quad & \sum_{\{u,v\}\in E(G)} \left(\Big(\mathbf{x}(u) - \mathbf{x}(v)\Big)^2 + \Big(\mathbf{y}(u) - \mathbf{y}(v)\Big)^2 \right) \\ & = \arg\min \mathbf{x}^t L_G \mathbf{x} + \arg\min \mathbf{y}^t L_G \mathbf{y},\end{aligned}$$

subjected to the following constraints

$$\| \mathbf{x} \|_2^2 = 1, \qquad \| \mathbf{y} \|_2^2 = 1. \tag{11}$$

and

$$\mathbf{1}^t.\mathbf{x} = 0, \qquad \mathbf{1}^t.\mathbf{y} = 0. \tag{12}$$

But still we have the problem of *degeneracy*; that is,

$$\mathbf{x} = \mathbf{y} = \mathbf{v}_2,$$

where $\mathbf{v}_2$ stands for the eigenvector corresponding to the second smallest Laplacian eigenvalue. To avoid this, we only need to impose the condition that both solutions $\mathbf{x}$ and $\mathbf{y}$ must be *perpendicular* to each other.

5 The Implementation and Evaluation of the Model

As we discussed earlier, one of the important issues in investment or *lending a loan* from lenders viewpoint is to consider the risk (or predicting) that a firm is bankrupt or healthy in the time of receiving the credits or loans. Thus, we can define the problem as follows. Let us assume that we have n firms with their associating feature vectors (extracted from their financial ratios) and we are going to predict the classification of them into two categories of bankrupt and healthy firms. Hence, we will use two methods of evaluating these firms.

In the first method, the feature vector of each firm which is an m-dimensional vector is embedded into the real line (or into the plane) using the *principle component analysis*. Then, by using *support vector machine* (SVM for short) and *Multi-layer perceptron* (MLP for short) methods we will classify all the firms into bankrupt and healthy.

In the second approach, by associating a graph (k-nearest neighborhood graph) to the collection of firms, we first build the *adjacency* matrix corresponding to that graph (for $k = 5$) and then we construct its *affinity matrix* based on the *LPP* method. We note that in this step, one can consider several *kernels* in which we choose the LPP formula in the previous section.

The next step is to employ the graph embedding technique besides *discretized laplacian smoothing* technique to reduce the dimension of each feature vector of

each firm. We can use any *smoothing-value* $\beta \in (0,1)$. Here, we decide to use the value $\beta = 0.6$. We finally use the same techniques of SVM and MLP from the first method to classify the firms into two groups of bankrupt and healthy.
To implement our algorithm on a suitable *data set*, we use the latest *financial statements* of selected firms in Tehran stock exchange market related to different industries. Our important assumption is that if the chosen firm has been bankrupt in the past years, the financial statments of that firm has also been added to our data set. Based on the article 141 of *Business law*, a firm is assumed to be *bankrupt* if the *accumulated losses* of the firm is equal to 50% of the *registered capital* of that firm.
Considering the fact that based on the above definition of *bankruptcy*, the number of *bankrupt samples* are nearly 10% of the whole data, we also include those firms that their *retained earning* are about 30% of their registered capitals. Thus, at the begining we had a collection 585 firms and finally considering 30% of retained earning as mentioned above, we obtained a set of 677 samples. After the implementation of our algorithms, we were able to classify the samples into 171 bankrupt samples and 506 healthy samples. The final point is that for investigating the financial statements of these firms, here we considered 8 *financial ratios* which are *net profit margin*, *return on assets*, *return on equity*, *current ratio*, *inventory turnover ratio*, *receivables collection period*, *debt ratio* and *total asset turnover ratio*.

5.1 Embedding into the real line

In this subsection, we discuss about our results regarding the embedding of our graph into the real line.
After data preparation (data preprocessing) and implementation of our model on the data set based on both methods of PCA and graph embedding using the well-known libraries *Sklearn* and *Scipy*, the original data will be embedded into the real line. The figure 2 shows the implementation for the PCA algorithm in one-dimensional case.

The next figure 3 shows the implementation of spectral graph embedding algorithm in dimension one.

After dimension reduction based on the two above methods, we have used 80% of the whole data to train SVM model and the rest of data was used to test the model. We have also calculated the *accuracy* of the model for classification of firms into bankrupt and healthy.
It is important to note that in the embedding of our graph into one dimension using the MLP method, since the dimension of input data is equal to one we can not achieve the suitable accuracy. Hence, for having a correct evaluation of the effectiveness of both methods of the PCA and the graph embedding in the binary classification of firms, we have employed the SVM approach. Furthermore, for relying on confidential results we have *randomly* chosen training and test subsets of data form the original dataset.

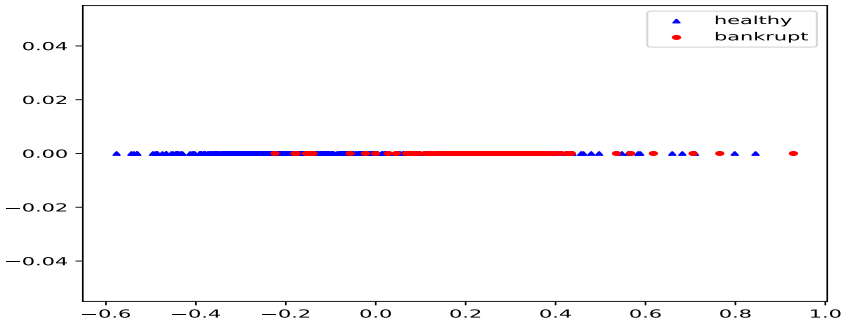

Figure 2: The Implementation of One-dimensional PCA Algorithm.

To be more precise, in each step of our 100 times testing, we have chosen two new subsets of data as the training and the test datasets. The result of these experiments has shown in figure 4.

As you can clearly see from the figure 4, the results of the graph embedding method is better (85% of cases) than the PCA method.

5.2 Embedding into the plane

In this subsection, we discuss about our results regarding the embedding of our graph into the plane.
We first note that the implementation of plane embedding is easy, considering the previous case of the line embedding. The results of our implementations have shown in figure 5 and figure 6.

After dimension reduction based on both methods of the PCA and the spectral

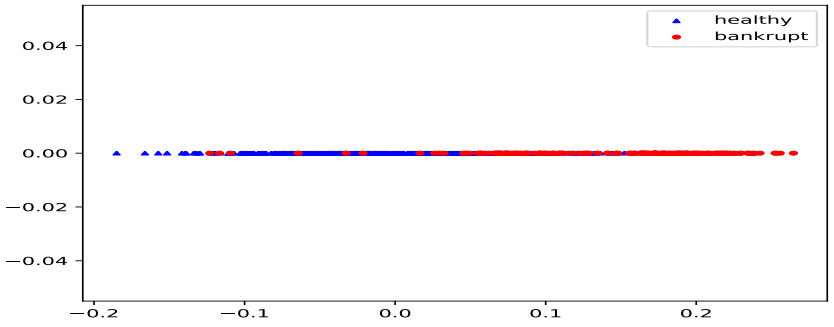

Figure 3: The Implementation of One-dimensional Graph Embedding Algorithm.

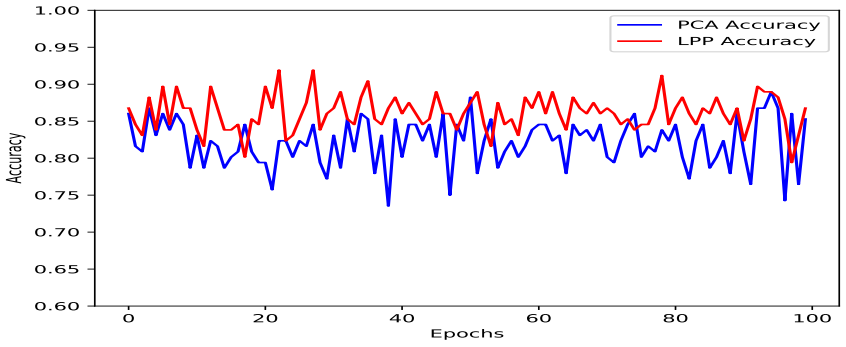

Figure 4: Comparing the accuracy of PCA and Spectral Graph Embedding in Dimension One.

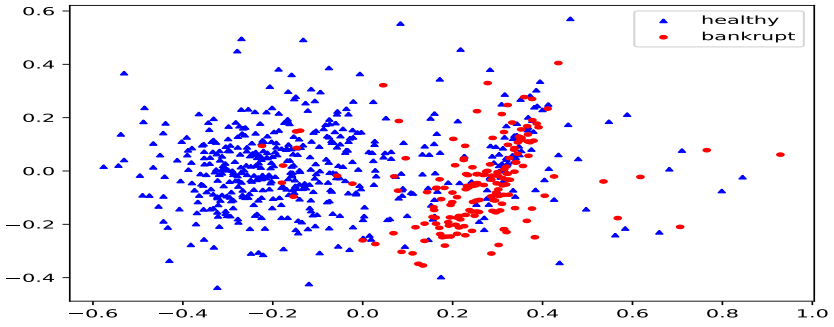

Figure 5: The Implementation of Two-dimensional PCA Algorithm.

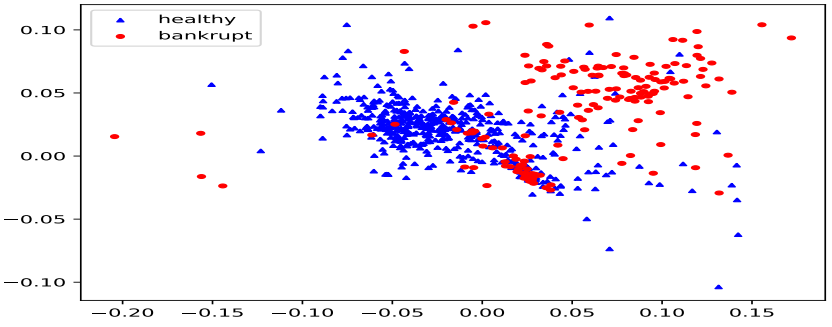

Figure 6: The Implementation of Two-dimensional Graph Embedding Algorithm.

embedding into the two-dimensional case, once again we have used 80% of the whole data to train SVM model and the rest of data was used to test the model. Moreover, We have calculated the *accuracy* of the model for classification of firms into bankrupt and healthy. We also have randomly chosen the training and the test subsets of data form the original dataset similar to that of the line embedding scenario.
Again, similar to the line embedding case, in each step of our 100 times testing, we have chosen two new subsets of data as the training and the test datasets. The result of these experiments has shown in figure 7.

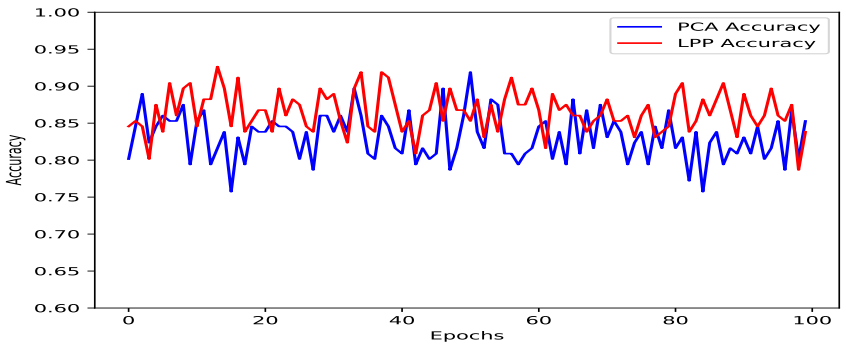

Figure 7: Comparing the accuracy of PCA and Spectral Graph Embedding in Dimension Two.

6 Concluding Remarks and Future Works

In this paper, we introduce the method of *spectral graph embedding* into the line and the plane as a dimension reduction approach to transform an original higher-dimensional financial data with interdependencies to a lower-dimensional data in one or two diemsions to classify a collection of firms in Tehran stock exchange market into bankrupt and healthy. Then, we use the well-known method of data dimension reduction; that is, the PCA approach as an alternative method for comparing our results of the spectral graph embedding algorithm with the PCA method and to evaluate the *risk assessment* of those firms.
To investigate and describe the financial statements of our firms, we chose 8 important features for each firm, so we had a collection of feature vectors of dimension 8. The output of reduced datasets in the first and the second cases was then given to the SVM and the Multi-layer perceptron models, respectively. In both cases, the results of implementations showed a good improvement in the case of the graph spectral embedding in compare with the other two learning models. Considering the fact that the spectral methods do not need any prior assumptions on the shape

of the original dataset, we can apply this method to other scenarios of similar types. The two parameters σ and β which indicates the kernel width and smoothing coefficient, respectively, are very *crucial* in the accuracy of the proposed model. Therefore, one potential suggestion for the future work is to investigate the importance of these two parameters in the accuracy, performance and the effectiveness of the proposed model. Furthermore, the *optimality* of these parameters can be a challenging problem for further research works. We also like to mention the fact that having more refined and high-volume data may have a *significant impact* on the *accuracy* of the presented model. Finally, considering the small number of firms in Tehran stock exchange market in compare with other international stock exchange markets around the world, the result of applying the presented model on more data along with more refined and suitable data may imply much better accuracy and performance of the system.

Bibliography

[1] B. RIBEIRO, N. CHEN, A. CHEN, *Financial credit risk assessment: a recent review*, Artificial Intelligence Review, 45 (2016), 1-23.

[2] K.PEARSON, *On lines and planes of closest fit to systems of points in space*, Philos. Phenomenol., 2(6) (1901), 559-572.

[3] S.T. ROWEIS, L.K. SAUL, *Nonlinear dimensionality reduction by locally linear embedding*, Science, 290 (2000), 2323-2326.

[4] D. A. SPIELMAN, *Graphs, vectors, and matrices*, Bull. Amer. Math. Soc. 54 (2017), 45-61.

[5] L.K. SAUL, S.T. ROWEIS, Y. SINGER, *Think globally, fit locally: unsupervised learning of low dimensional manifolds*, J. Machine Learn. Res., 4 (2003), 119-155.

[6] E. W. DIJKSTRA, *A note on two problems in connexion with graphs*, Numer. Math., 1 (1959), 269-271.

[7] Z. ZHANG, H. ZHA, *Principal manifolds and nonlinear dimension reduction via local tangent space alignment*, SIAM J. Sci. Comput., 26 (2002), 313-338.

[8] E. FIX, J.L. HODGES, *Discriminatory analysis. nonparametric discrimination: consistency properties*, USAF school of aviation medicine, Randolph Field, Texas, 1951.

[9] M. BELKIN P. NIYOGI, *Laplacian eigenmaps for dimensionality reduction and data representation*, Neural Comput., 15(6) (2003), 1373-1396.

[10] XIAOFEI HE, PARTHA NIYOGI, *Locality preserving projections (LPP)*, Proceedings of the 16th international conference on neural information processing systems., (2003), 153-160.

[11] DENG CAI, XIAOFEI HE, YUXIAO HU, JIAWEI HAN, THOMAS HUANG, *Learning a spatially smooth subspace for face recognition*, IEEE, (June 2007), 17-22.

[12] KOUKI M, ELKHALDI, *A toward a predicting model of firm bankruptcy: evidence from the Tunisian context*, Middle East Finance Economics., 14 (2011), 26-43.

[13] LOREDANA CULTRERA, XAVIER *Brédart*, *Bankruptcy prediction: the case of Belgian SMEs*, Review of Accounting & Finance., 15 (2016), 101-119.

[14] BERNARDETE RIBEIRO, NING CHEN, ALEXANDER KOVACEC, *Shaping graph pattern mining for financial risk*, Neurocomputing., 326327 (2019), 123-131.

[15] A. BLANCO, R. PINO-MEJIAS, J. LARA, S. RAYO, *Credit scoring models for the microfinance industry using neural networks: evidence from Peru*, Expert Syst Appl., 40(2013), 356-364.

[16] FM. RAFIEI, S. MANZARI, S. BOSTANIAN, *Financial health prediction models using artificial neural networks, genetic algorithm and multivariate discriminant analysis: Iranian evidence*, Expert Syst Appl., 38 (2011), 10210-10217.

How to Cite: Hossein Teimoori Faal[1], Meyssam Bagheri[2], *Spectral graph embedding for dimension reduction in financial risk assessment*, Journal of Mathematics and Modeling in Finance (JMMF), Vol. 1, No. 2, Pages:63–78, (2021).