ISSN 2395-8618

Exploration, Exploitation Phenomena and Regression Analysis: Propensity Metric, Anomaly Reduction, Dimensionality Reduction

Chaman Lal Sabharwal

***Abstract*--The classical Ordinary linear Least Square approximation (OLS) model has been used as the best fit regression for linear trend data. In data analysis, the accuracy of analysis depends on the model as well as the metric used to measure the error. Singular Value Decomposition (SVD) is also applied for Normal linear Least Square (NLS) approximation along the perpendicular to the approximating line. The OLS line is not sensitive to temporal variation in time variables whereas SVD is sensitive, it renders OLS less suitable for time sensitive data. Both OLS and SVD use quantitative metric for regression analysis, and SVD has inherent constraints. Propensity score analysis is an innovative class new technique for qualitative error analysis. Propensity score method is easier to communicate to non-expert audience. Moreover, propensity score estimates are often more robust than the percent error estimates of predicted values over the true values. Herein we present a hybrid algorithm that achieves a balance between quantitative and qualitative approximation accuracy of both OLS and NLS (SVD). This metric has also proved useful for evaluating the effects of treatments in real patient data. This technique is also suitable for anomaly removal. Visualization is a preferred way to ascertain the quality of a new algorithm and is used to demonstrate the hybrid algorithm. We have applied this criteria for comparison with other existing methods. We have found that this technique is reliable and preferable to explain to the expert as well as non-expert. The empirical tests show the accuracy improvements over conventional methods.**

***Index Terms*--least square regression, singular value decomposition, propensity, anomaly, accuracy, precision, learning management systems**

I. INTRODUCTION

The regression analysis is used to determine the correlation between variables and predicting value of dependent variables. The linear regression is a reliable model to predict the value of a dependent variable[1]. This assumes that only one of the variables has error. In empirical data, sometimes the error permeates both the variables [2]. There may be other ways also to establish relation between independent and dependent variables, then we have to distinguish between different universally accepted minimum-error algorithms [3]. For data analysis, most of the time raw data is not directly applicable to analysis algorithms directly. It is mandatary to cleanse data for reliable and accurate regression analysis. Thus, it is expected that the data is accurate otherwise the prediction analysis will be unreliable. If the data is correlated and noisy, it is indispensable to transform the data into uncorrelated and noise free data to prevent overfitting. In such cases, anomaly reduction is mandatory. Some smoothing operation is performed to bring data in line with the approximation concept before applying the learning algorithm. Cleansing is a natural phenomenon, e.g. the physicians use sharp blades to perform incisions, we wash edibles before eating to stay healthy. Data smoothing may be performed via filtering with some kernel or via data noise reduction. Furthermore, the numerical data may be standardized by mean-centering and unit standard deviation etc. Some related algorithms are not equivalent [4]. The approximation error measurement depends on the metric applied to analyze approximation. Cognitive modelling is one of the representative research methods in cognitive sciences. For cognitive model to be viable, it must be verifiable by using well thought metric [5]. We leverage these techniques to devise a cognitively acceptable minimum-error scheme based on propensity metric in conjunction with Euclidean metric. Data dimensionality reduction can be effectively done via SVD, SVD uses dimension reduction operation in the latent space. This step yields noise reduction when the data is transformed back to original space. Such algorithms use relaxation technique to obtain improved hybrid approximation algorithm.

For multivariable data, (x,y), x, y are vectors, most of the time y is scaler valued. The simplest case occurs in 2D when both x, y are scalar valued, it is easy to comprehend. In the simplest case, linear least square regression is a straight line.

Manuscript received on January 10, 2018, accepted for publication on May 7, 2018, published on June 30, 2018.

Chaman Lal Sabharwal is with the Missouri University of Science and Technology Rolla, MO-65409, USA (e-mail: chaman@mst.edu).

ISSN 2395-8618

This linear representation model approximates the non-parametric data points (x_i,y_i) with points (x_p,y_p) on a parametric line. Since a line is uniquely defined by two points, it has two parameters, intercept, a, and slope or elevation, b. The line is a parametric representation of data. One of the models, measures error along the y-axis. In other words, $x_i = x_{ip}$, $y_{ip} = a + b\ x_{ip}$ such that the sum of squares of errors is minimum, error $E_1 = \sum_{k=1,n} (x_k - x_{kp})^2 + (y_k - y_{kp})^2 = \sum_{k=1,n} (y_k - a - b\ x_k)^2$.

In statistical analysis, the accuracy of approximation depends on several parameters. One such parameter is the metric used to measure the approximation error. Each metric has its own merits. For OLS, there are several issues. For least square approximation, it is in fact approximation in y direction, not min distance perpendicular to the approximation line [6], [7], [8]. In order to correct this, we devise a true line at min-distance from the input data, normal distance least square fit line, NLS [9]. We call it normal linear least square approximation (NLS) similar to ordinary linear least square approximation (OLS). NLS may become complicated for multiple dimensions, we also show that linear algebra SVD can be leveraged to achieve NLS more easily [10]. Finally, we see that OLS is not sensitive to data spread, NLS will also correct this deficiency of OLS. We also define a new metric, propensity scoring metric (PSM) for OLS, NLS and hybrid algorithms for pairwise comparison. Propensity score has been used in other areas for estimating the effect of a treatment, policy or other causal effects [11]. We will show the effect of new metric as compared to OLS and NLS metrics. We show that the hybrid algorithm achieves a balance between quantitative and qualitative approximation accuracy of both OLS and SVD. Also, it will be shown that it can be used for noise and anomaly reduction. Thus, there are several approaches to approximate data linearly: ordinary linear least square regression (OLS), (new) normal linear least square regression (NLS), singular value decomposition linear least square regression (SVD), (new) hybrid linear least square regression (HLA). To measure the accuracy of approximation, there are several measures: quantitative and qualitative. Knowing what technique and metric to use makes all the difference in analysis and makes most out of data. That way one spends less time on justifying the conclusions. The challenge is the decision making on the metric used to approximate. The intent of this paper is the design a greedy(hybrid) algorithm that yields better approximation than the OLS and NLS/SVD approximation algorithms, also a way to detect and remove anomalies in the training data.

The paper is organized as follows: Section II gives background and justification for the work. It describes OLS, NLS in R^n and computation by mean-centering data, Section III derives new NLS formulation, Section IV describes SVD and it connection to NLS, Section V gives new hybrid greedy algorithm and its implementation, error analysis of OLS, NLS/SVD, and Hybrid algorithms is provided with respect to both metrics, it introduces propensity score metric (PSM) and anomaly reduction, Section VI is conclusion, Section VII is an appendix giving all the necessary details about linear algebra.

II. BACKGROUND

Here data is represented as a matrix of real values. It is easier to work with data if it is standardized. Simple example of standardization is mean-centering data with unit standard deviation. Ordinarily the reference point of data is the origin, mean-centering implies that the centroid of data is translated to the origin. We will soon see how mean-centering simplifies the computations.

Let the data be represented by an m×n real matrix A, i.e., m rows of n-vectors or n columns of m-vectors. If **x** is column of A, the mean of **x** is denoted by $\bar{\mathbf{x}}$, where $\bar{\mathbf{x}} = \frac{\sum_{i=1,m} x_i}{m}$. To centralized **x,** it is translated to $\mathbf{x} - \bar{\mathbf{x}}$. Similarly, if **y** is row of A, it is centralized as $\mathbf{y} - \bar{\mathbf{y}}$, where the mean of **y** is $\bar{\mathbf{y}} = \frac{\sum_{i=1,n} y_i}{n}$. Further, if **x** and **y** are both rows (or both columns), the mean of dot product of **x** and **y** is denoted by $\overline{\mathbf{xy}} = \frac{\mathbf{x \bullet y}}{n} = \frac{\sum_{i=1,n} x_i y_i}{n}$, for **x** = **y**, it is denoted by $\overline{\mathbf{x}^2} = \frac{\mathbf{x \bullet x}}{n} = \frac{\sum_{i=1,n} {x_i}^2}{n}$. Most of the linear transformations are performed by means of matrix multiplication, for example, centralization is a linear transformation for mean-centering a matrix [12]. There is an immaculate transformation T_m to mean-center the columns of A as follows. Let I_m be m*x*m identity matrix, $\mathbf{e_m}$ be a column m-vector of ones, and $T_m = I_m - \mathbf{e_m e_m}^T/m$. This T_m is called the column centralizer. For example, if **x** is a column vector then

$T_m\mathbf{x} = I_m\mathbf{x} - \mathbf{e_m e_m}^T\mathbf{x}/m = \mathbf{x} - \mathbf{e_m e_m \bullet x}/m = \mathbf{x} - \bar{\mathbf{x}}e_m$

or in short $\mathbf{x} - \bar{\mathbf{x}}$ where $\bar{\mathbf{x}}$ is the mean of **x**. This T_m applied on the left of A, it centralizes columns of the matrix. Similarly, if T_n is multiplied on the right of A, the AT_n mean-centers the rows of A. For example, for row vector **y**:

$\mathbf{y}T_n = \mathbf{y}\ I_n - \mathbf{y}\ \mathbf{e_n e_n}^T/n = \mathbf{y} - \mathbf{y} \bullet \mathbf{e_n}\ \mathbf{e_n}^T/n = \mathbf{y} - \bar{\mathbf{y}}\mathbf{e_n}^T$

Centralizing data simplifies computations by reducing the number of parameters to be computed. After preforming analysis on mean-centered data, data origin is translated back to the centroid. This is a standard technique used for computational simplification and for visualization in graphics [13], [14].

A.1 Linear Least Square Approximation

There are two ways to compute linear least squares approximation. It depends on the concept of approximation. One way is to find line at shortest distance perpendicular to the desired line. Another way is to minimize the distance along a vertical coordinate axis, e.g, y -axis. Both methods accomplish specific tasks and the corresponding approximation errors are different. A hybrid approach is doubly robust estimator at increased cost and reduced error, propensity metric shows a remarkable improvement. The hybrid technique generalizes the line to polygonal line that effectively improves the pointwise

ISSN 2395-8618

accuracy without the risk of overfitting data. This method is qualitative for measuring the accuracy of points are closer to the approximation rather than the quantitative distance error. We focus on more qualitative accuracy in data approximation rather than absolute error, that may be attributed to anomalies/outliers.

A.2 Conventional Ordinary Linear Least Square Approximation (OLS)

For n× (m+1) input data, the rows are of the matrix are composite **x** (**x** is m-vector) and y coordinates of data points, that is, m-vector **x** elements are attributes, and scalar y is an associated value. For notation, $\mathbf{x_i}$ refers to the ith row, x_{ij}, refers to the element in the i-th row, j-th column. There is short-cut notation x_{*k} represents a column of the k-th element of all rows, and $\overline{x_k}$ is the mean of the column of kth elements x_{*k} of all row vectors. For clarity, note that $\mathbf{x_k}$ refers to the kth row/ attribute vector of vectors, whereas $\overline{x_k}$ represents mean of the k-th attribute. We want to find a linear least square approximation hyperplane. First, for hyperplane

y = a + $\mathbf{b}^{\mathrm{T}}$ **x** = a + $\sum_{k=1,m} b_k x_k$, we need to calculate *parameters* a and **b** that minimize the function

f(a, **b**)=$\sum_{i=1,n}(y_i - a - \boldsymbol{b}^T\mathbf{x_i})^2$.

That leads to two equations

$\frac{\partial f(a,\mathbf{b})}{\partial a} = \sum_{i=1,n}(y_i - a - \boldsymbol{b}^T\mathbf{x}_i) = 0$ (1)

and

$\frac{\partial f(a,\mathbf{b})}{\partial b_k} = \sum_{i=1,n}(y_i - a - \boldsymbol{b}^T\mathbf{x}_i)x_{ik} = 0$ (2)

Let $\overline{x_k} = \frac{\sum_{i=1,n} x_{ik}}{n}$, $\bar{y} = \frac{\sum_{i=1,n} y_i}{n}$, $\overline{x_k y} = \frac{\sum_{i=1,n} x_{ik} y_i}{n}$, $\overline{x_k{}^2} = \frac{\sum_{i=1,n} x_{ik}{}^2}{n}$,

the first equation (1) becomes

$\bar{y} = a + \boldsymbol{b}^T \bar{\mathbf{x}}$ which implies that the regression plane passes through the centroid ($\bar{\mathbf{x}}$, $\bar{y}$).

The second equation (2) implies that for k=1,m these m equations are

$\sum_{i=1,n}(y_i - a - \sum_{j=1,m} b_j x_{ij})x_{ik} = 0$

$\sum_{i=1,n}(x_{ik}y_i - a\mathbf{x}_{ik} - \sum_{j=1,m} b_j x_{ij}\, x_{ik}) = 0$

which means

$\overline{x_k y} = a\overline{x_k} + \sum_{j=1,m} b_j \overline{x_j x_k}$

$\overline{x_k y} = a\overline{x_k} + \boldsymbol{b}^T\, \overline{\mathbf{x}\, x_k}$.

Now $\bar{y} = a + \boldsymbol{b}^T \bar{\mathbf{x}}$ and

$\overline{x_k y} = a\overline{x_k} + \sum_{j=1,m} \mathbf{b}_j \overline{x_j \mathbf{x}_k}$ for k=1,m

That is

$\sum_{j=1,m} b_j \overline{x_j x_k} = \overline{x_k y} - a\overline{x_k}$ (1)

Also $\bar{y} = a + \boldsymbol{b}^T \bar{\mathbf{x}}$ can be expanded as

$\bar{y} = a + \sum_{j=1,m} b_j \overline{x_j}$

Multiply by $\overline{x_k}$

$\overline{x_k}\ \bar{y} = a\overline{x_k} + \sum_{j=1,m} \overline{x_k} b_j \overline{x_j}$

Or

$\sum_{j=1,m} \overline{x_k} b_j \overline{x_j} = \overline{x_k}\ \bar{y} - a\overline{x_k}$ (2)

Subtracting (2) from (1)

We get $\sum_{j=1,m} b_j \overline{x_j x_k} - b_j \overline{x_j}\ \overline{x_k} = \overline{x_k y} - \overline{x_k}\ \bar{y}$

or $\sum_{j=1,m}(\overline{x_k \boldsymbol{x_j}} - \overline{x_k}\ \overline{x_j}) b_j = \overline{x_k y} - \overline{x_k}\ \bar{y}$

in m unknowns b_j.

These equations can be rewritten in terms of symmetric coefficient matrix is

$[\ \overline{x_i \boldsymbol{x_j}} - \overline{x_i}\ \overline{x_j}\ ]\ \mathbf{b} = [\ \overline{x_i y} - \overline{x_i}\ \bar{y}\ ]$

This gives **b**. However since $\bar{y} = a + \boldsymbol{b}^T \bar{\mathbf{x}}$, once **b** is known, the offset/bias term a can be efficiently computed from

$a = \bar{y} - \boldsymbol{b}^T \bar{\mathbf{x}}$

In the special case, m=1, then k=1, **x** has only one component say x_1 = x

We can solve for a and b to yield [15]

$b = \frac{\overline{xy} - \bar{x}\bar{y}}{\overline{x^2} - \bar{x}^2}$ and $a = \frac{\overline{x^2}\bar{y} - \bar{x}\,\overline{xy}}{\overline{x^2} - \bar{x}^2}$

It may be noted that for mean-centered data, $\bar{x} = 0$, $\bar{y} = 0$, it results in a=0.

Briefly, for input data n×2 matrix, columns are x, y coordinates of data points, we find a linear least square approximation line. Before exploiting any approximation, it is assumed that data is accurate, else prediction will also be inaccurate. For linear approximation line y = a + bx, we need to calculate *two parameters, also called regression coefficients*, a and b for minimizing of

f(a, b)=$\sum_{i=1,n}(y_i - a - bx_i)^2$.

Using calculus criteria based on derivatives, it leads to two equations

$\frac{\partial f(a,b)}{\partial a} = \sum_{i=1,n}(y_i - a - bx_i) = 0$

$\bar{y} - a - b\,\bar{x} = 0$ (1)

and

$\frac{\partial f(a,b)}{\partial b} = \sum_{i=1,n}(y_i - a - bx_i)x_i = 0$

$\overline{xy} - a\bar{x} - b\,\overline{x^2} = 0$ (2)

The first equation (1) becomes $\bar{y} = a + b\,\bar{x}$, which implies that the regression line, y = a + bx, passes through the centroid ($\bar{x}$, $\bar{y}$). The two equations are

$\bar{y} = a + b\,\bar{x}$ and $\overline{xy} = a\bar{x} + b\,\overline{x^2}$

can be solved for a and b to yield

$b = \frac{\overline{xy} - \bar{x}\bar{y}}{\overline{x^2} - \bar{x}^2}$ and $a = \frac{\overline{x^2}\bar{y} - \overline{xy}\,\bar{x}}{\overline{x^2} - \bar{x}^2}$

However since $\bar{y} = a + b\,\bar{x}$, once b is known, the offset/bias term a can be efficiently computed from $a = \bar{y} - b\,\bar{x}$.

A.3 Mean-Centered data formulation

Continuing in R^2, mean-centering allows us to consider regression line through the origin because centroid is translated to the origin. The bias term *a* becomes zero automatically and the data becomes unbiased. To take advantage of standardization, The OLS can be simplified for mean-centered data, we need to compute *only one* regression coefficient b for minimizing f(b)=1/n$\sum_{i=1,n}(y_i - bx_i)^2$

ISSN 2395-8618

or

$$f(b) = 1/n \sum_{i=1,n}(y_i - bx_i)^2$$
$$= 1/n \sum_{i=1,n} (y_i^2 - 2by_ix_i + b^2 x_i^2)$$
$$= \overline{y^2} - 2b\overline{xy} + b^2\,\overline{x^2}$$

That is

$$f(b) = \overline{y^2} - 2\overline{xy}\,b + \overline{x^2}\,b^2$$

For calculus based critical values, see [16]. *Calculus* based critical value criteria requires that f'(b) = 0. This leads to $-2\overline{xy} + \overline{x^2}\,2b = 0$ or

$$b = \frac{\overline{xy}}{\overline{x^2}}$$

So, for mean-centered data, OLS line is

$$y = bx, \text{ with } b = \frac{\overline{xy}}{\overline{x^2}}$$

which is a simpler expression than the raw data computations. Since f''(b) = $2\,\overline{x^2}$ is positive, the critical value is minimum.

However, if we want to go to the original frame, original reference point, we may translate the origin back to the centroid, then line translate into original coordinates

$$y - \bar{y} = b(x - \bar{x}) \text{ or } y = \bar{y} - b\bar{x} + b\,x$$

that is

$$y = a + b\,x \text{ where } a = \bar{y} - b\,\bar{x}$$

In this case, *only* b is to be computed, a is an automatic byproduct.

This gives a line through (0,a) and along the direction $\frac{(1,b)}{\sqrt{(1+b^2)}}$

Non-Calculus (algebraic) approach proceeds as follows.

$$f(b) = \overline{y^2} - 2\overline{xy}\,b + \overline{x^2}\,b^2$$

Since it is a quadratic (convex) function and $\overline{x^2} \geq 0$, Figure 1, there is *only one* minima. This expression simplifies to

$$f(b) = \overline{y^2} - 2\overline{xy}\,b + \overline{x^2}\,b^2$$
$$= \overline{x^2}\left(b - \frac{\overline{xy}}{\overline{x^2}}\right)^2 + \frac{\overline{x^2}\,\overline{y^2} - \overline{xy}^2}{\overline{x^2}}$$

Since $\overline{x^2}\,\overline{y^2} - \overline{xy}^2 \geq 0$, f(b) is min when $b = \frac{\overline{xy}}{\overline{x^2}}$. This is what we got above using calculus.

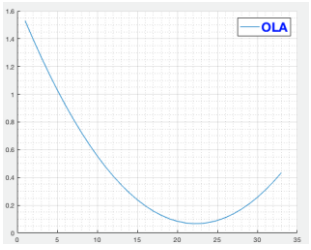


Figure 1. The convex function f(b) has only one global minima giving the slope b for OLS line.

In essence, this is a common sense three step approach to find the OLS line. The three steps are, (1) mean-center the data, translate the centroid $(\bar{x}, \bar{y})$ to the origin (0,0), (2) calculate the direction of least square error approximating line through the origin, (3) translate data origin back to centroid $(\bar{x}, \bar{y}\ )$ for original frame of reference. The computations using mean-centered data are simpler. In, Figure 2, Cyan dots are the raw training data, solid red line is the approximation line, and red dotted lines are errors between the training data and corresponding predicted approximations. In Figure 3, there the black dotted lines are normal(perpendicular, orthogonal) to the regression line where as red dotted lines are vertical, along the y-axis direction. Clearly the normal lines are shorter than vertical line.

We will explore and exploit further whether there are some other lines whose normal distance error is even smaller than this line error. That leads us to next section.

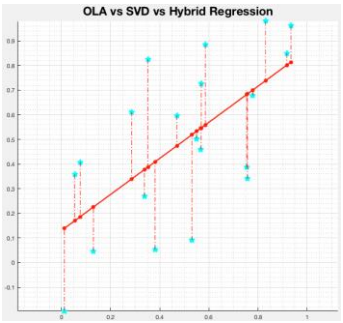


Figure 2. Data points, regression line, approximation errors

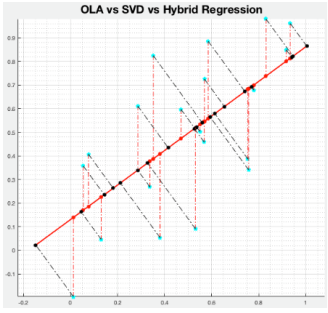


Figure 3. The red vertical dotted lines are error from OLS line along y-axis, the black orthogonal dotted lines are error from OLS line along the normal. Normal distance error is smaller than vertical distance error.

III. NORMAL LINEAR LEAST SQUARE APPROXIMATION (NLS)

NLS has not been used in social sciences because of its complexity [17]. The ordinary linear approximation (OLS) line is not as close to the data points as expected because distances/errors are measured along the y-axis. If distances are measured along the normal (perpendicular) to the approximation line, then line is more representative of data. The normal (perpendicular, orthogonal) distance problem is formulated below. SVD is a method that accomplishes the same goals, without resorting to calculus of extrema computations.

For the reasons stated above, we assume that the data (x_i, y_i), i=1, 2, …, n is mean-centered, otherwise we can use centralizer transformation to mean-center the data. The problem becomes that of finding the value of *only one parameter b* that minimizes f(b) where

ISSN 2395-8618

$$f(b) = 1/n \sum_{i=1,n} \left(\frac{y_i - bx_i}{\sqrt{1+b^2}}\right)^2 \quad \text{or}$$

$$f(b) = 1/n \sum_{i=1,n} \frac{(y_i{}^2 + b^2 x_i{}^2 - 2bx_i y_i)}{1+b^2}$$

$$= \frac{\overline{y^2} + b^2\,\overline{x^2} - 2b\overline{xy}}{1+b^2} \qquad (1)$$

Thus, for local minima of $f(b) = \frac{\overline{y^2} + b^2\,\overline{x^2} - 2b\overline{xy}}{1+b^2}$

$$f(b) = \frac{b^2\,\overline{x^2} - 2b\overline{xy} + \overline{y^2}}{1+b^2} = \frac{b^2\,\overline{x^2} - 2b\overline{xy} + \frac{\overline{xy}^2}{\overline{x^2}} - \frac{\overline{xy}^2}{\overline{x^2}} + \overline{y^2}}{1+b^2}$$

$$= \frac{b^2\,\overline{x^2} - 2b\overline{xy} + \frac{\overline{xy}^2}{\overline{x^2}} - \frac{\overline{xy}^2}{\overline{x^2}} + \overline{y^2}}{1+b^2}$$

$$= \frac{\overline{x^2}\left(b - \frac{\overline{xy}}{\overline{x^2}}\right)^2 - \frac{\overline{xy}^2}{\overline{x^2}} + \overline{y^2}}{1+b^2}$$

$$= \frac{\overline{x^2}\left(b - \frac{\overline{xy}}{\overline{x^2}}\right)^2 + \frac{\overline{x^2}\,\overline{y^2} - \overline{xy}^2}{\overline{x^2}}}{1+b^2}$$

Note that $\overline{x^2}\,\overline{y^2} - \overline{xy}^2$ always ≥ 0. It is equivalent to standard result $|\mathbf{x}\bullet\mathbf{y}| \leq |\mathbf{x}||\mathbf{y}|$ which can be quickly derived from triangle inequality or geometric definition of dot product.

We saw that in the unnormalized case, f(b) is minimum when

$$b - \frac{\overline{xy}}{\overline{x^2}} = 0 \text{ or } b = \frac{\overline{xy}}{\overline{x^2}}$$

This is *not true* in this case, see Figure 4. For OLS, f(b) is quadratic, convex and has only one extreme/minima blue curve. For NLS, f(b) is not convex, not quadratic red curve. It has two extrema, one maxima and one minima. In both OLS and NLS cases, the minima are close to each other, but not identical.

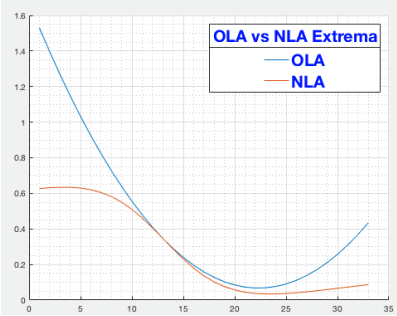


Figure 4 f(b) is convex for OLS case, f(b) is not convex for NLS case.

For NLS, f(b) is never negative. As b approaches zero, f(b) becomes $\overline{y^2}$ and as b approaches infinity, f(b) becomes $\overline{x^2}$.

To calculate the minimum, setting the first derivative of f(b) w.r.t b to zero, f'(b)=0, we get quadratic

$$\overline{xy}\,b^2 + \left(\overline{x^2} - \overline{y^2}\right) b - \overline{xy} = 0 \qquad (2)$$

Since it is a quadratic, it has two critical values, b_1, b_2

$$b = \frac{-(\overline{x^2} - \overline{y^2}) \pm \sqrt{(\overline{x^2} - \overline{y^2})^2 + 4\,\overline{xy}^2}}{2\,\overline{xy}} \qquad (3)$$

f(b) can't have both local minima, see Figure 4. If f"(b_1)>0, the b_1 is a local minima else f"(b_2)>0, then b_2 is a local minima.

However, from the Figure 4, it is clear the minimum occurs at larger of b1 and b2.

Once $b = \frac{-(\overline{x^2} - \overline{y^2}) + \sqrt{(\overline{x^2} - \overline{y^2})^2 + 4\,\overline{xy}^2}}{2\,\overline{xy}}$ is computed, we have a line through the origin (0,0) along the *direction* $\frac{(1,b)}{\sqrt{(1+b^2)}}$

The normal least square line (NLS) is shown in Figure 5. This is not the same as OLS regression line seen in Figures 2 and 3.

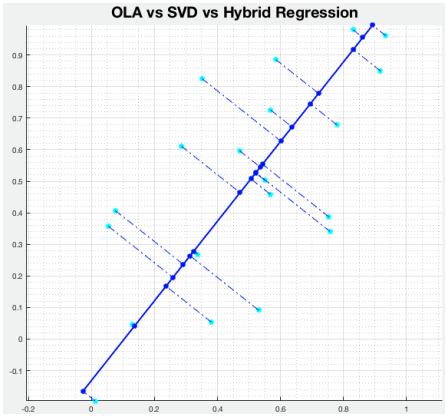


Figure 5. Cyan dots are the data points blue line is NLS line. Blue dots are the approximation, Blue dotted lines are normal errors from NLS line.

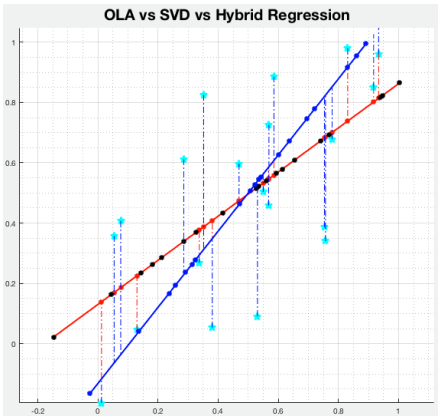


Figure 6 Red line is OLS, Blue line is NLS. Red dotted lines and Blue dotted lines are vertical errors form the Cyan data points. NLS vertical error from Blue line is *more* than OLS error from red line.

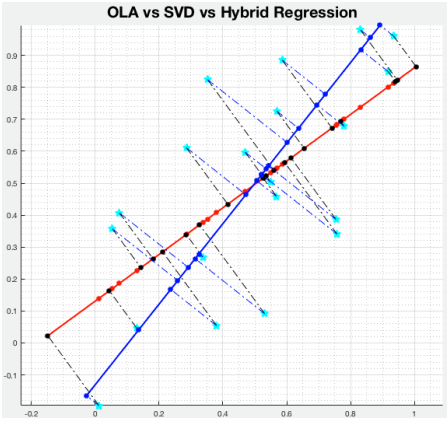


Figure 7 Red line is OLS, Blue line is NLS. Red dotted lines and Blue dotted lines are orthogonal errors form the Cyan data points. NLS normal error from Blue line is *less* than OLS error from red line.

ISSN 2395-8618

Further, the approximation error in both cases (OLS and NLS) is minimum depending on how the error is measured. Visual inspection shows that *majority* of the cyan dots are *closer* to blue line dots than the cyan dots to red line dots, see Figure 6, Figure 7. This visualization justifies, to some extent, to prefer NLS over OLS. Note when overall vertical error is larger for NLS line where as overall normal error is larger for OLS line. This confusion needs some resolution. We will give formal justification later in section V. Since NLS is based on calculus, its derivative is complex, the second derivative is quite complex, we explore an easier implementation of this idea by means of exploiting singular value decomposition (SVD).

In some applications, error is measured along vertical line y-direction, while in some application error is measured along the normal to the Least Squares Approximation line. When there is no other algorithm to compare, a technical indicator is used to measure to quality of approximation. Bollinger technique uses band of 1,2, 3 standard deviation bands to test the goodness of the model,

The third type of error is never used in numerical least square approximation. The propensity metric has been used for non-numerical data. Our goal is to blend the two algorithms and the corresponding measure of error into uniform error metric and compare the performance of two methods. The optimal approximation is better represented by propensity metric, no matter which method of error computation is used.

IV SINGULAR VALUE DECOMPOSITION (SVD)

Today, singular value decomposition is used in many theoretical and applied fields: computer science and engineering, psychology and sociology, atmospheric science and astronomy, health and medicine etc. [18], [19], [16], [20], [21]. It is also extremely useful in machine learning and in both descriptive and predictive statistics. There is no unique basis function for R^n. The goal is to determine a suitable basis function so that A can be expressed in response to the application. The normal least square approximation (NLS) hyperplane can also be obtained directly by using linear algebra singular value decomposition (SVD). Before we discuss the connection between NLS and SVD, we may note that SVD is important on its own right due to applications in various areas. For the sake of completeness, we give brief description of SVD.

Singular Value Decomposition (SVD) is a matrix factorization technique generalizing eigen-decomposition and principal component analysis. Every positive semi-definite real matrix can be decomposed into three matrix factors: left singular vectors matrix, right singular vectors matrix and a diagonal matrix of singular values in descending order on main diagonal. The goal is not to recreate the matrix, but to create the *best linear least square approximation* [22], [23]. There are various advantages of SVD. First, 150 years old *Principal Component Analysis* (PCA) is a specialization of eigen-decomposition to symmetric matrices with orthogonal eigenvectors such that $A = VDV^{-1} = VDV^T$. In case, A is not a square data matrix, PCA does not apply. However, A^TA is a symmetric square positive semi-definite matrix, then $A^TA = VDV^T$, [24], [25], [26]. Besides other benefits of this factorization, we are interested in *direction vector* only for least square approximation. The columns of V are eigenvectors of A^TA corresponding to eigenvalues arranged in descending order. Since eigenvectors correspond to directions of approximation lines, we show that direction vector of NLS corresponds to first eigenvector of SVD [27], [28], [16]. The following table, describes the distinction between eigen decomposition (ED), PCA and SVD. Briefly, for eigen decomposition of A, U is the matrix of eigenvectors of A, D is diagonal matrix of eigenvalues of A, conveniently eigenpairs are arranged on descending order of eigenvalues.

$$\text{ED} \rightarrow A = UDU^{-1}$$

For PCA and SVD, U and V are matrices of eigenvectors of symmetric matrices AA^T and A^TA, S is the matrix of singular values of positive semi-definite matrix and D is the matrix of eigenvalues of A such that

$$\text{PCA} \rightarrow A = UDU^T$$

and

$$\text{SVD} \rightarrow A = USV^T$$

The following Table 1 shows a summary of different aspects to express A in terms of ED, PCA, SVD using eigenvectors as basis vectors. Examples show the case where the matrix is (1) symmetric and positive semidefinite,(2) matrix is symmetric, but not positive semi-definite, (3) matrix is not symmetric, but is positive semi-definite, and (4) where matrix is not symmetric, and no positive semidefinite.

A. Connection between NLS and SVD

For simplicity, A is n×2, of data points in the xy-plane. To minimize the error between observed **P** and estimated direction **v**. Since $\mathbf{P} = \mathbf{P}\bullet\mathbf{v}\ \mathbf{v} + (\mathbf{v}\text{x}\mathbf{P})\text{x}\mathbf{v}$, minimizing $|(\mathbf{v}\text{x}\mathbf{P})\text{x}\mathbf{v}|$ means maximizing the distance $|\mathbf{P}\bullet\mathbf{v}\ \mathbf{v}|$ or $|\mathbf{P}\bullet\mathbf{v}|$ because **v** is a unit vector [16].

We derive the direction **v** so that sum of squares of distances from training data points to predicted direction vector **v** is least. Note, **v** passes through the origin because the data is mean-centered. Since data is mean-centered, the approximation line passes through the origin. By default, vectors **P** are column vectors in linear algebra, thus rows of A are position vectors $[x,y] = \mathbf{P}^\mathbf{T}$. As seen above, the vector **P** can be written as the sum of a vector along unit vector **v** and a unit vector **w** orthogonal to **v**, that is, using vector notation $\mathbf{P} = \mathbf{P}\bullet\mathbf{v}\ \mathbf{v} + (\mathbf{P} - \mathbf{P}\bullet\mathbf{v}\ \mathbf{v}) = v\ \mathbf{v} + w\ \mathbf{w}$. This means that minimizing the distance w amounts to maximizing v. We are to maximize over all data points $\mathbf{P}_i$. The problem becomes that of maximizing

$$\sum_{i=1,n} |\mathbf{P}_i\bullet\mathbf{v}|^2$$

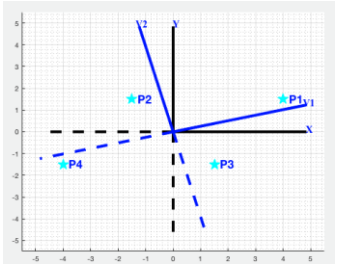
for *all* P_i for *some* vector **v** to be determined. Now

$$\begin{aligned}\sum_{i=1,n} |\mathbf{P}_i\bullet\mathbf{v}|^2 &= \sum_{i=1,n} \mathbf{P}_i\bullet\mathbf{v}\ \mathbf{P}i\bullet\mathbf{v} = \sum_{i=1,n} \mathbf{v}\bullet\mathbf{P}_i\ \mathbf{P}_i\bullet\mathbf{v} \\ &= \sum_{i=1,n} \mathbf{v}^\mathbf{T}\mathbf{P}_i\ \mathbf{P}_i^\mathbf{T}\mathbf{v} = \mathbf{v}^\mathbf{T}\left(\sum_{i=1,n} \mathbf{P}_i\ \mathbf{P}_i^\mathbf{T}\right)\mathbf{v} \\ &= \mathbf{v}^\mathbf{T}\ (A^TA)\mathbf{v}.\end{aligned}$$

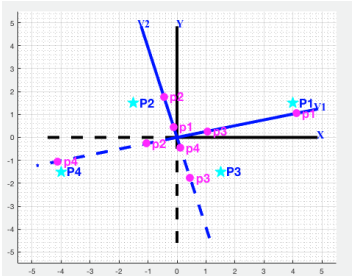
ISSN 2395-8618

TABLE 1 MATRIX TYPES AND THEIR ED, PCA, SVD

Matrix A	A-Symmetric	A-Pos Semi- Definite	ED matches A	PC matches A	SVD matches A
$\begin{bmatrix}0 & 0\\0 & 1\end{bmatrix}$	√	√	√	√	√
$\begin{bmatrix}0 & 0\\0 & -1\end{bmatrix}$	√	x	√	√	x
$\begin{bmatrix}0 & 0\\1 & 0\end{bmatrix}$	x	√	x	x	√
$\begin{bmatrix}1 & 0\\1 & -1\end{bmatrix}$	x	x	√	x	x



(a)



(b)

Figure 8. (a) Data points, standard x-,y-axes, v1-,v2- eignevectors axes, (b) Projection of Data points on v1-,v2- eignevector axes, Data points are closer to eigenvectors than the standard axes.

This means that $\sum_{i=1,n} |\mathbf{P}i \bullet \mathbf{v}|^2$ is maximum if **v** is an eigenvector of A^TA and corresponds to largest eigenvalue of A^TA. Similarly, all the other eigenvectors can be obtained incrementally one at a time, constraining each vector orthogonal to the previous eigenvectors. Thus, SVD is computed iteratively in descending order of eigenvalues and corresponding eigenvectors orthogonal to the previously computed eigenvectors. From this analysis, it is clear that the largest eigenvalue amounts to the largest spread of data along the corresponding eigenvector. The spread of projections of data on $\mathbf{v_1}$ is larger than that on $\mathbf{v_2}$, see Figure 8(b).

For example, $\mathbf{P^T}$'s are data points in 2D, $\mathbf{v_1}$, $\mathbf{v_2}$ are eigenvectors corresponding to largest eigenvalues of A^TA . For this consideration, the NLS requires only $\mathbf{v_1}$, the direction with largest eigenvalue, and with largest data spread.

Uniqueness of Eigenvectors. As a side remark, for the matrix, any non-zero multiple of an eigenvector is again an eigenvector. To make the eigenvectors unique, they are normalized to unit vectors. But if **u** is unit eigenvector, then – **u** is also a unit vector, see Figure A in appendix for MATLAB[30], svd computed eigenvectors [27], [28]. In the literature, it is an accepted convention to make the first non-zero component positive in the eigenvector, see Figure [see appendix]. Since eigenvectors are ordered, we use ordering to make the k-th element of k-th vector to be positive, see Figure A [see Appendix] that makes the vectors look more natural like a right- handed system. In case, the kth element is zero, then the first non-zero element is made positive. This is the approach we prefer to use [16]. Incidentally, recall that the direction vectors in OLS and NLS had first component as positive in the figure.

For example, consider the matrix $A = \begin{bmatrix}1 & 0\\0 & -1\end{bmatrix}$, then $AA^T = A^TA = A^2 = \begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}$, PCA uses AA^T and A^TA (A, A^2 are symmetric; A^2 is a positive semi-definite matrix) for computing the eigen-pairs. In this example, except for signs, the eigenvalues of A are square roots of the eigenvalues of A^2 that are 1 and 1, the corresponding eigenvectors are eigenvectors of A^2 are $\begin{bmatrix}0\\1\end{bmatrix}, \begin{bmatrix}1\\0\end{bmatrix}$. In particular, the eigenvector of A corresponding to eigenvalue -1 and eigenvector of A^2 pertaining the eigenvalue 1 are identical. Matlab svd function does not reconstruct eigenvalues of A accurately. The vectors in V are superficially adjusted to match A. In our algorithm, we include the proper signs. Matlab R2017b computes SVD resurrects A as

$$\begin{bmatrix}1 & 0\\0 & -1\end{bmatrix} = \begin{bmatrix}0 & 1\\1 & 0\end{bmatrix}\begin{bmatrix}1 & 0\\0 & 1\end{bmatrix}\begin{bmatrix}0 & -1\\1 & 0\end{bmatrix}.$$

This is inaccurate as $\begin{bmatrix}0\\-1\end{bmatrix}$ is not an eigenvector of A or A^2. Also, it may be noted that $\begin{bmatrix}0 & -1\\1 & 0\end{bmatrix}$ is not the transpose or inverse of $\begin{bmatrix}0 & 1\\1 & 0\end{bmatrix}$.

Here we used the proper sign for *square root of 1 to -1*, because -1 is eigenvalue of A. Consistent with the definition of SVD with correct sign of eigenvalue [28], [15], the correct eigen-decomposition $A = VDV^{-1} = VDV^T$ is

$$\begin{bmatrix}1 & 0\\0 & -1\end{bmatrix} = \begin{bmatrix}0 & 1\\1 & 0\end{bmatrix}\begin{bmatrix}-1 & 0\\0 & 1\end{bmatrix}\begin{bmatrix}0 & 1\\1 & 0\end{bmatrix}.$$

Recall, Singular Value Decomposition (SVD) is a generalization of PCA to include (1) non-square rectangular

ISSN 2395-8618

and (2) positive semi-definite matrices [21]. However, PCA and SVD are equivalent for symmetric positive semi-definite matrices. SVD uses covariance matrices AA^T and A^TA to determine two orthogonal matrices U, V of eigenvectors and a diagonal matrix S for singular values such that the eigenvectors in U, (and V) are (1) pairwise orthogonal, (2) normalized to unit vectors and (3) arranged in the descending order of singular values. A singular value of A is square root of the eigenvalue of A^TA and AA^T. Then SVD decomposes A into three factors U, V and S such that $A = USV^T$. By dropping the least significant singular values and corresponding singular vectors, best approximation of data matrix can be reconstructed, quantitative error can be estimated simply by using the discarded eigenvalues. The examples where A is not both symmetric and positive semi-definite are shown in the Table 1 to confirm SVD and PCA are not equivalent in general. In our work, A is symmetric positive semi-definite, consequently AA^T and A^TA turn out to be symmetric positive semi-definite [22], [10],[127],[26].

Example. To accommodate both PCA and SVD, we generalize the previous example matrix to symmetric, positive semi-definite (PSD) matrix $A=\begin{bmatrix}1 & 0\\ 0 & 2\end{bmatrix}$. Now $AA^T=A^TA=A^2=\begin{bmatrix}1 & 0\\ 0 & 4\end{bmatrix}$ has eigen values 1,4. Thus the singular values for A are 2,1; so, $D=S=\begin{bmatrix}2 & 0\\ 0 & 1\end{bmatrix}$ which is same as S obtained from SVD. Thus, for PCA/SVD of A, the eigenvectors of AA^T, A^TA form orthogonal matrices $U = \begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix}$, $V=\begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix}$, and singular values become the diagonal entries of $S = \begin{bmatrix}2 & 0\\ 0 & 1\end{bmatrix}$. Now PCA as well as SVD factorization is

$$\begin{bmatrix}1 & 0\\ 0 & 2\end{bmatrix} = \begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix}\begin{bmatrix}2 & 0\\ 0 & 1\end{bmatrix}\begin{bmatrix}0 & 1\\ 1 & 0\end{bmatrix}$$

which implies that $A = USV^T = VSV^T$. $= USU^T$.

Summarizing this discussion in Table 2, we see that several possibilities exist for an arbitrary matrix A. For example, in Table 2, there are some cases where A is (1) symmetric and (1.1) has PCA SVD decomposition equivalent on PSD matrix (1.2) has PCA, but SVD does not exist as A is not SPD matrix, and (2) not symmetric and (2.1) PCA does not exist because A is not square matrix, SVD decomposition exists on PSD matrix (2.2) has no PCA, no valid natural SVD decomposition for non-square non-PSD matrix. Also refer to Table 1 for square matrices.

TABLE 2. FOUR EXHAUSTIVE CASES

Data Matrix	Positive Semi-Definite	Not Positive Semi-Definite
Symmetric	$\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}$ PCA, SVD	$\begin{bmatrix}1 & 0\\ 0 & -1\end{bmatrix}$ PCA, ~~SVD~~
Not Symmetric	$\begin{bmatrix}1 & 0\\ 0 & 1\\ 0 & 0\end{bmatrix}$ ~~PCA~~, SVD	$\begin{bmatrix}-1 & 0\\ 0 & -1\\ 0 & 0\end{bmatrix}$ ~~PCA~~, ~~SVD~~

V. HYBRID GREEDY ALGORITHM DESIGN

The idea of hybrid algorithm is not just to amalgamate these two algorithms in such a way that the new algorithm accuracy supersedes the accuracy of the base algorithm, but to extract the best features of both and improve on them with propensity metric and double SVD. The error analysis is metrically and cognitively appealing to humans. Interval of error, also known as Bollinger band of uncertainty, quantifies the range of uncertainty in a value and propensity score is a frequency metric used for comparing them pairwise to determine the best algorithm. Hybrid algorithm uses a doubly robust balancing method. it is responsive to treatment for data dealing with patient treatments. Hybrid algorithm is designed to overcome the limitations of traditional parametric methods.

We design a hybrid greedy algorithm leveraging the best of OLS and NLS/SVD approximation lines in two ways: non-parametric polygonal possibly overfitting; and parametric line in general. For each observed point, (x_0,y_0), we have seen in Figure 6 and Figure 7 that there is a corresponding predicted point (x_R, y_R) on regression line and a predicted point (x_S, y_S) on SVD line. If (x_0,y_0) is an observed value, (x_R, y_R) is predicted point value corresponding to the OLS line $y = a + bx$. The vertical distance is along y direction. The distance between (x_0,y_0) and (x_R,y_R) is the y-distance, the OLS regression error $e_R = |y_0-y_R|$. For normal distance from NLS or SVD approximation line, it is along perpendicular to the line, it turns out that $x_S \neq x_0$ in (x_S, y_S) , the distance between (x_0,y_0) and (x_S,y_S) is Euclidian normal distance $e_S = \sqrt{(x_0 - x_S)^2 + (y_0 - y_S)^2}$.

It is clear from Figure 6 and Figure 7 that for some points in observed data, $e_R < e_S$ while for some other points $e_S < e_R$. In each method, the total error E is sum of squares of pointwise distances (errors) for all data points, question arises which one (E_R for OLS and E_S for SVD) is acceptable due to the dual nature on error computation. There is no denying the fact if vertical distances are used for *both* lines, then $E_R<E_S$ and if normal distances are used for *both* lines, then $E_S < E_R$. Then how does the user determine which one preferable to use: OLS or NLS/SVD? For greedy algorithm, define the approximation point (x_H, y_H) to be that point which is closer to the observed point (x_0,y_0) in both ways. Euclidean distance is used to measure closeness. For each input, we will determine approximate line that represents the input data no matter how the error is computed, see Figure 10 for green color dots, these are closer to cyan dots than red line dots or blue line dots. Instead of measuring the quantitative distance we define a qualitative metric that is more useful in visualization and is cognitively acceptable. Non-parametric algorithm uses to regression coefficients of OLS and NLS, whereas the parametric version computes its own regression coefficients.

ISSN 2395-8618

A. Non-Parametric Hybrid Greedy Algorithm

Algorithm A:

Input: array of x and y mean-centered data values

Output: hybrid greedy approximation points (x_H,y_H),where $(x_R\ y_R)$ is on OLS, $(x_S\ y_S)$ is on SVD line

1. Calculate regression coefficients a and b for OLS regression from observed x,y
   Calculate predicted values by linear regression $y_R = a+bx$
   Calculate approximation error E_R
   Test Goodness of the regression line
2. Calculate A=[x ,y], x, y are columns of matrix A.
   Calculate SVD [U S V] = svd(A)
   Use first column of V to get b. a is automatic
   Calculate x_S,y_S of projected points $[x_S,y_S]$ on column vectors of V that is AVV'
   Calculate approximation error E_S
   Test Goodness of the NLS line
   Compare error E_R and E_S
3. Calculate greedy hybrid x_H, y_H using a variation of relaxation method
   for all point pairs $[x_R, y_R]$, $[x_S, y_S]$
   if d($[x_S, y_S]$, $[x_0, y_0]$)<= d((x_R, y_R), (x_0, y_0))
   $(x_H, y_H) = (x_S, y_S)$;

else

   $(x_H, y_H) = (x_R, y_R)$;

   end

end

   Calculate error E_H from pointwise e_H
   Test Goodness of the hybrid line
   Compare error E_S, E_R, E_H
   Compare by propensity values
4. x_H, y_H are arrays of predicted coordinates on hybrid polygonal line, cognistically appealing and lower metric error.

This algorithm gives non-parametric polygonal approximation and possibly overfitting. The next algorithm parametrizes it by using SVD on polygonal approximation, see Table 2. The hybrid representation (x_H,y_H) is closer to the input training data (x,y) that the OLS approximation (x_R,y_R) points and NLS approximation (x_S,y_S) points. Note that in practice we do not need to store the polygonal approximation values, it is more efficient to retain the regression coefficients of OLS and NLS/SVD for real time calculations.

B. Parametric Hybrid Algorithm

This algorithm is of theoretical interest and for visualization, Algorithm A is sufficient for practical use. The non-parametric *polygonal* approximation algorithm gives insight for improving the accuracy, Figure 11. It has two shortcomings it does conserve space, and it is subject to overfitting the input training data. Here we explore double approximation to design a general algorithm which conserves space as well as it is parametric, see Figure 11.

Algorithm B

Input: array of x and y mean-centered data values

Output: hybrid approximation line parameters for points (x_H,y_H),where $(x_R\ y_R)$ is on OLS, $(x_S\ y_S)$ is on SVD line

- As in algorithm A, polygonal greedy approximation is (x_H,y_H)
- Use SVD to fit computed points (x_H,y_H) with SVD algorithm to derive parameters for the direction of the line
- Use direction of this double SVD line to compute approximation (x_D, y_D)
- Test Goodness of the based on this double NLS line
- Compare E_R,E_S, E_H, E_D, and propensity metric values

Now almost all observed points are closer to greedy line than OLS and NLS/SVD approximation lines. It satisfies the general parametric and space conservation requirements, see Table2.

Note over the entire data set, *red dots have smallest error* from cyan dots when distances are measured along y, while *blue dots have smallest error* from cyan dots when distances are measured along the normal to the line, see Figure 9. Each green dot is at a smaller of the two distances from cyan dot, interestingly, it *does not mean* that green dots have *overall* smaller error than the two, in fact it will be bigger than each. The green dots can be connected by a polygonal line see Figure 10 or an SVD straight line approximation, see Figure 7. We have seen that NLS is better than OLS. We may use SVD to approximate data (x_H,y_H) to (x_D,y_D) line, see Figure 11.

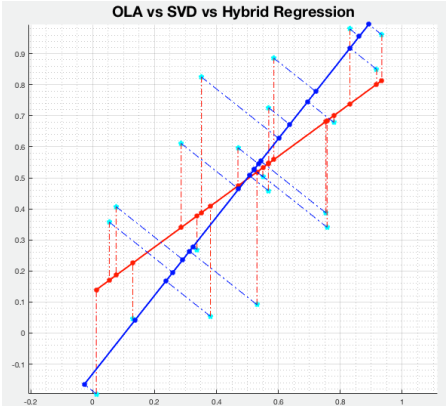


Figure 9. Cyan dots are data points, Red line is OLS line, Blue line is NLS/SVD line, Green dots are hybrid approximation dots

ISSN 2395-8618

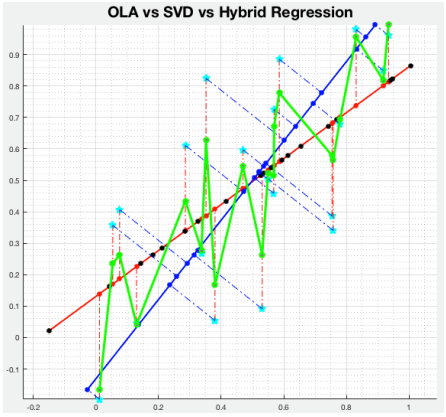


Figure 10 Non-Parametric polygonal Hybrid data points, Cyan dots are points which are closer to green dots than red or blue dots. Hybrid polygonal line, green polygonal line connects the green hybrid points (x_H,y_H).

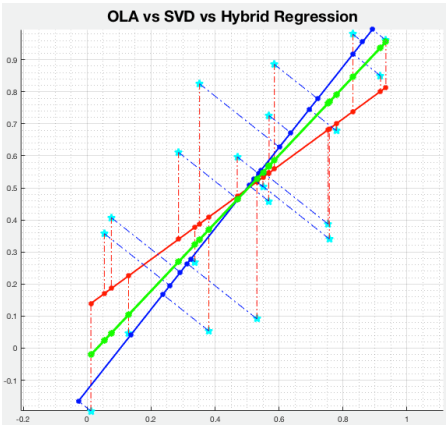


Figure 11 Non-Parametric polygonal line Green dots in Figure 10 are not shown here for clarity. SVD line is created to corresponding green points into the green Hybrid parametric line.

C. Precision and Propensity

We have seen three ways to process data. OLS is best when error is measured along y-axis. SVD is best when error is measured long normal is measured. Propensity is best when the frequency of nearness is used. The question arises which one is preferable. The propensity method is cognitively and visually preferable. The linear least square approximation error is *quantitative* measure. The precision and propensity are a *qualitative* measure of accuracy [10], [31], [11]. Quantitative error is a function of the location of data points, propensity depends on percentage of data points for pointwise binary outcome from comparing error due to a pair of methods. This is similar to precision metric used in data mining community confusion matrix. For percentage of data truly closer to OLS, SVD lines, Hybrid line pairwise, see Table 2 and Table 5. From Figure 10, it is clear that green construction is preferable, but the quantitative error comparison is inconclusive. However, we use propensity metric to determine the level of accuracy that hybrid line has as compared to OLS and SVD. When errors are measured in the respective methods, we can calculate the propensity value for one line relative to the other line to conclude the preference irrespective of which method is used to calculate errors. It is determined that overall SVD/NLS approximation is better approximation than OLS, see Figure 10. Similarly, propensity metric shows, that hybrid line is preferable to both OLS and SVD lines, see Figure 11, Table 2. Table 5.

D. Anomaly Detection and Removal

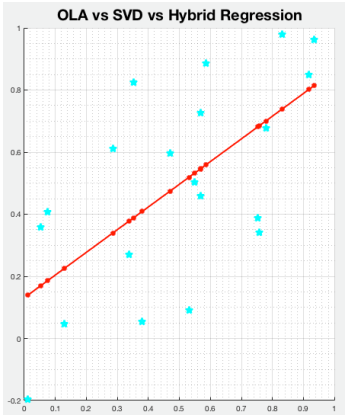
It is clear that pointwise vertical distance error, e_R, is always greater than normal distance error, e_S, from any line. Since sum of squares of errors for OLS line, E_R is smallest in the vertical distance metric, the regression error from any other line is bound to be larger than error, E_R, from OLS line. Pointwise error in OLS and NLS is inconclusive. Propensity score metric (PSM) is a qualitative measure to differentiate for better approximation line, where the distance metric fails. This will give insight to error measurement modeling to the algorithm designers. PSM can also leveraged identify the anomalies. To detect anomalies accurately, we create a confusion matrix for frequency of points within one standard deviation of both the lines, see Table 3 of confusion matrix for noise reduction using Bollinger band about OLS and NLS approximation lines.. Any point which is not within this Bollinger band about any of the two lines, is probably an anomaly (FF). Such is point is candidate for further scrutiny. After analyzing it with the hybrid line, it determined that hybrid line is a better differentiator for noisy data. After clipping suspicious points for the data, we reapplied our algorithm to ascertain that reduced data set gives better accuracy, see Table 4, and Table 5.

Table 3 Confusion Matrix for Anomaly

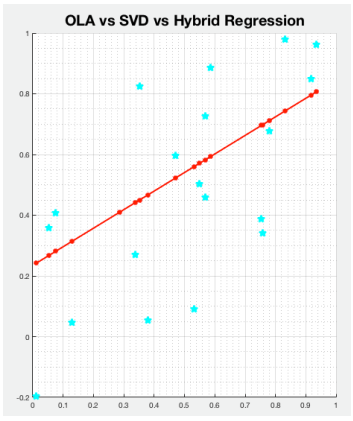
	OLS within	OLS outside
SVD within	TT	TF
SVD outside	FT	FF

Example: Noisy data, vertical distances error not realistic. In the Figure 12(b), we can see that if fifth point is noisy, it has affected the entire approximation line. In particular for the *neighboring* points, there is glaring offset. Experiments show that one outlier point can adversely affect the approximation line in the immediate neighborhood of noisy point, see Figure 12. Red line is least square regression line on raw data of 20 points. This regression line is noise sensitive, see Figure 12(a),(b). If one of data points is an outlier, it can create a large adverse effect on the outcome. Figure 12(c) shows the improvement on this shortcoming after removing noise.

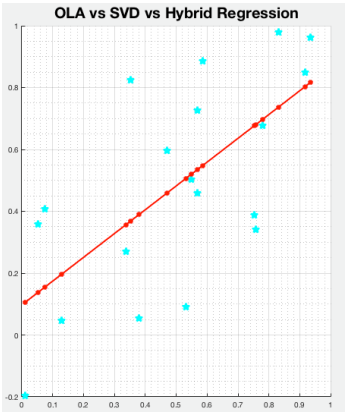
ISSN 2395-8618



(a)



(b)



(c)

Figure 12, (a) No noise, (b) Noise introduced in position 5, direction of line changes , (c) Noise removal, position 5 removed from the data, data has one less point.

Figure 12 (a) has no noise, (b) has noise in position 5, as a result the regression lines are different, (c) here noise is removed, now (a) and (c) are same, but (c) has one less points as point 5 has been removed. We do not see any major difference in the regression lines.

The goal of the new algorithm is to improve the prediction capability rather than numeric value of approximation error. Numeric error is a measure of divergence from the true value. The hybrid algorithm achieves a balance between quantitative and qualitative approximation accuracy of both OLS and NLS/SVD. We use STD-standard deviation for confidence interval about the approximation lines. If A is the set of points outside the confidence interval and B is the set of points where $e_R > e_S$, the $A \cap B$ is a candidate set of anomalies.

Table 4 Comparison of Algorithms

	OLA	SVD	Hybrid
Approximation Line Directi	[0.81, 0.59]	[0.62, 0.78]	[0.68, 0.73]
Approximation Error	7.06%	5.09%	3.32%
Confidence in one std Interv	75.00%	100.00%	100.00%
closeness OLA vs SVD	25.00%	75.00%	
closeness OLA vs Hybrid	0.00%		100.00%
closeness SVD vs Hybrid		15.00%	85.00%

Table 5 Comparison of Algorithms 5% Noise Removal

	OLA	SVD	Hybrid
Approximation Line Directi	[0.79, 0.62]	[0.66, 0.75]	[0.68, 0.73]
Approximation Error	6.46%	4.71%	3.32%
Confidence in one std Interv	88.89%	100.00%	100.00%
closeness OLA vs SVD	0.00%	100.00%	
closeness OLA vs Hybrid	0.00%		100.00%
closeness SVD vs Hybrid		16.70%	83.30%

E. Temporal Sensitivity

In health care environment, if the time interval for a treatment is changed, we expect to see the temporal change in response to a treatment. Using OLS, we see that there is no change in response to temporal change, that is, the computed error remains unchanged, see Figures 9-12. Figure 13 is the visual summary of quantitative and qualitative error in the methods. Using the same data set, on scaling the time interval, the NLS/SVD and Hybrid algorithms respond positively to the changes. This suggests that OLS is not suitable for such temporal applications. In the example, we also notice that as the slope of the hybrid line increase, the error decreases. Experiments confirm that the slope of 45 degrees is brake-even point with maximum error. Slope below or above 45 degrees accounts for reduction in error. For comparison of the three algorithms, see Table 2, Table 3. It shows the computed direction vectors of the approximation lines, approximation error in the Euclidean distance metric, and propensity how close is training data to one algorithm vs the other formulation, see Figures 13-17.

ISSN 2395-8618

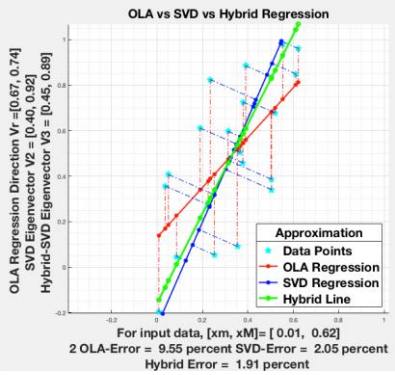


Figure 13 Relative errors one time interval [0.01,0.62]

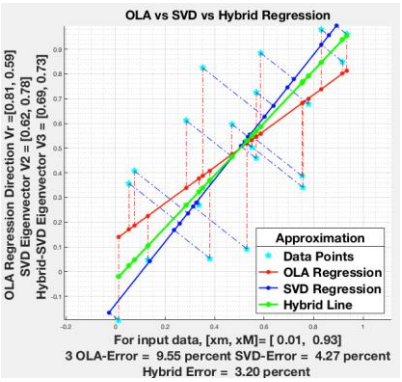


Figure 14 Relative errors on time interval [0.01,0.93]

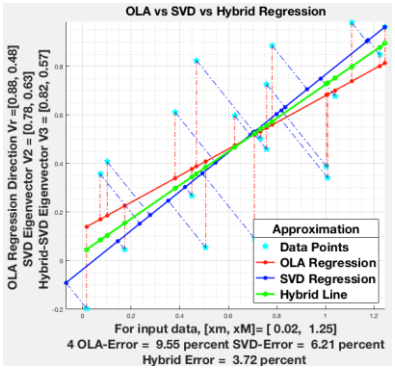


Figure 15 Relative errors on time interval [0.02,1.25]

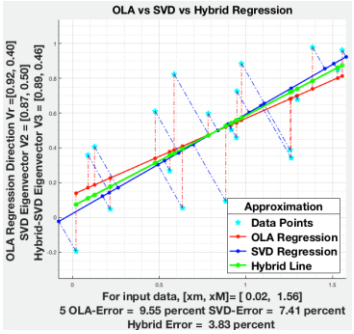


Figure 16 Relative errors on time interval [0.02,1.56]

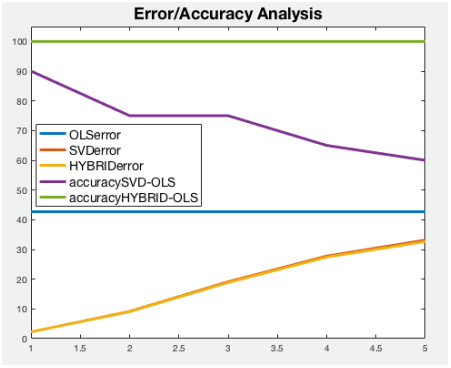


Figure 17. Green line shows percentage of Hybrid points closer to data points as compared to OLS. Purple line shows percentage of SVD points closer to data points as compared to OLS. Blue line shows percentage of error in OLS. Yellow and red (on top of each other) percentage of error in SVD and Hybrid algorithms.

VI. CONCLUSION

In the paper, we have explored several algorithms and several metrics to determine cognitively and visually acceptable criteria for least square regression. The algorithms are ordinary least squares regression (OLS), orthogonal least square regression(NLS) and Singular value decomposition (SVD). We explored these algorithms along with our hybrid algorithm. We exploited them using the quantitative and qualitative metrics. We explored 1. various ways to approximate numerical data, 2. Temporal versions of prediction, 3. how to reduce noise. Here we first removed noise by virtually using OLS and NLS. The hybrid data is then approximated by leveraging NLS/SVD, double approximation. It is determined that hybrid algorithm outperforms the other algorithms when applied and compared pairwise. This will give insight for error measurement modeling to the researchers. They will benefit from the hybrid linear least approximation algorithm.

OLS was found to be insensitive to temporal data spread, whereas SVD was implicitly modifying the independent (temporal) variables of the original input in pursuit of lower error. We designed a hybrid algorithm that overcomes these shortcomings and supersedes the accuracy of the existing algorithms. From the experiments, it follows that error is least for lines that are almost horizontal or vertical, the breakeven point occurs as the slope of the line becomes closer to 45 degrees. No matter what the slope is, the new hybrid regression line error is always bounded above by the error in OLS regression line. It is interesting to note that OLS remains unchanged while new regression line approximation error responds to the slope variation. We also showed how to improve svd algorithm of MATLAB[30] with correct directions of eigenvectors, a natural technique. The algorithm was implemented on MAC OS Mojave v 10.14, IntelCire i5, 8GB 1600MHZ using Matlab R1700b [30]. We have described the error measurement methods and propensity metric that is preferable for exploitation and visualization.

ISSN 2395-8618

VII. APPENDIX

A. Principal Component Analysis, and Singular Value Decomposition

This section is self-contained tutorial on PCA/SVD. The linear algebra concepts of *vector*, *transpose* of a vector, *scalar* product of a vector, Euclidean *norm*, *length* of a vector, *unit* vector, *sum* of two vectors, *dot product* of two vectors (analytical, geometric, matrix forms), *orthogonal* vectors, *Gram-Schmidt* orthogonalization, *matrix*, *square* matrix, *identity* matrix, *diagonal* matrix, *transpose* of matrix, *symmetric* matrix, *sum* of matrices, *scalar product* of a matrix, *determinant* of a matrix, *rank* of a matrix, *inverse* of a matrix, *norm* of a matrix, *orthogonal* matrix, *rotation* matrix, *rank* of a matrix, *determinant* of a matrix, *product* of matrices, *vector space*, and *basis* of a vector space, are standard terms in linear algebra. Additional terms that we use are an *eigenvector*, and an *eigenvalue*. All vectors are column vectors unless specifically stated. All matrices and vectors are real in this discussion. For details on linear algebra, reader may consult references [13, Jolliffe1995].

All the required transformations are built in the toolboxes of modern languages, Java, C++, Matlab, R, and Python. Weka contains tools for data pre-processing, classification, regression, clustering, association rules, and visualization. Herein modeling tools are eigenvalues and eigenvectors of covariance matrix. MatLab and Python automatically rank the eigenvalues in descending order, and orders the eigen vectors accordingly. Descending order is more natural because eigenpairs are important for further analysis in dimension reduction. In statistics, Principal Component Analysis (PCA) [Jolliffe1995] is also known as discrete Karhunen-Lo`eve (KL) transform which is used for extracting patterns from complex data sets by reducing the dimensionality of complex data set.

B. Definitions and properties of Vectors

Definition. A *vector* is an ordered set of finite number of elements and is denoted by a *column vector* **u**. Almost all the time we encounter vectors with numeric values for elements. In fact, they can be of any valid type.

The vector *notation*: a vector is denoted by a *bold lowercase* character. The elements of a vector are *italic lowercase*. For example, $\mathbf{u} = [u_1, u_2, \ldots, u_n]$ is a row vector, $\mathbf{v} = \begin{bmatrix} v_1 \\ \ldots \\ v_n \end{bmatrix}$ is a column vector.

Definition. The *transpose* of a vector **u** is denoted by $\mathbf{u}^T$. Transpose of a column vector is a row vector, and transpose of row vector is column vector. The transpose of a column vector u is written as $\mathbf{u}^T = [u_1, u_2, \ldots, u_n]$ or u may also be written as $\mathbf{u} = [u_1, u_2, \ldots, u_n]^T$.

Definition. The scalar multiple of a vector **u** by a scalar s is denoted by s**u** and is obtained by multiplying each component of **u** by s: $s\mathbf{u} = [su_1, su_2, \ldots, su_n]^T$.

Definition. For any vector **u**, the *norm or length* of **u** is denoted by $|\mathbf{u}|$ and is the square root of the sum of squares of its components:

$$|\mathbf{u}| = \sqrt{(u_1^2 + \ldots + u_n^2)}, \quad |\mathbf{u}|^2 = u_1^2 + \ldots + u_n^2 = \mathbf{u} \bullet \mathbf{u} = \mathbf{u}^T\mathbf{u}.$$

Definition. A vector **u** is a *unit vector* if it is of unit length, $|\mathbf{u}|^2 = 1$, $\mathbf{u} \bullet \mathbf{u} = \mathbf{u}^T\mathbf{u} = 1$.

Definition. The *sum* of two vectors **u** and **v** is written as **u+v** and is defined as vector whose elements are sums of respective components of **u** and **v**, e.g., $\mathbf{u+v} = [u_1 + v_1, u_2 + v_2, \ldots, u_n + v_n]^T$.

Definition. The *dot product* of **u** and **v** is denoted by **u•v**. It is defined in several different equivalent ways.

Analytically dot product **u•v** is the sum of products of components of **u** and **v**: $\mathbf{u} \bullet \mathbf{v} = u_1v_1 + \ldots + u_nv_n$.

Geometrically dot product is $\mathbf{u} \bullet \mathbf{v} = |\mathbf{u}||\mathbf{v}|\cos(\theta)$ where θ is the angle between the directions **u** and **v**.

Matrix product form the dot product is expressible as a row-column matrix multiplication $\mathbf{u} \bullet \mathbf{v} = \mathbf{u}^T\mathbf{v} = [u_1 \quad \ldots \quad u_n] \begin{bmatrix} v_1 \\ \ldots \\ v_n \end{bmatrix}$.

Property: If **u** and **v** and two vectors, it is true that $|\mathbf{u} \bullet \mathbf{v}| \leq |\mathbf{u}||\mathbf{v}|$. It follows trivially from geometric definition of dot product and cosine an angle that is less than or equal to 1.

Property. Dot product is *commutative*.

$\mathbf{u} \bullet \mathbf{v} = u_1v_1 + \ldots + u_nv_n = v_1u_1 + \ldots + v_nu_n = \mathbf{v} \bullet \mathbf{u}$, or $\mathbf{u}^T\mathbf{v} = \mathbf{v}^T\mathbf{u}$.

Definition. The vectors **u** and **v** are *orthogonal* if

$$\mathbf{u} \bullet \mathbf{v} = \mathbf{u}^T\mathbf{v} = 0.$$

Definition. A set of vectors is *orthonormal* if each vector is a unit vector, and any two different vectors are mutually orthogonal.

Matrices are used to represent data elegantly and efficiently for visual inspection. All the knowledge is hidden in the tables. Rows can be interpreted as classification rules with attribute values. One of the attributes can be a classification attribute.

The matrix *notation*: an mxn matrix is denote by $A = [a_{ij}]$ where the ij-th element of matrix A is denoted by a_{ij}. For example, any matrix can be represented systematically by using corresponding elements: $A = [a_{ij}]$, $U = [u_{ij}]$, $V = [v_{ij}]$, $S = [s_{ij}]$. If A is a matrix, $\mathbf{a_{i\bullet}}$ is a row vector representing the i-th row of matrix A, and $\mathbf{a_{\bullet j}}$ is a column vector representing the j-th column of A**.** Thus ith row is $\mathbf{a_{i\bullet}} = [a_{i1}, a_{i2}, \ldots, a_{in}]$. Similarly jth column is $\mathbf{a_{\bullet j}} = \begin{bmatrix} a_{1j} \\ a_{2j} \\ \ldots \\ a_{mj} \end{bmatrix}$.

Definition. If m=n , then mxn matrix is called a *square* matrix.

Definition. If every entry of a square matrix D is zero except for the diagonal entries d_{ii}, then the matrix D is called *diagonal* matrix. *In general*, for mxn matrix, if every entry except d_{ii}, is zero, it is also diagonal matrix.

ISSN 2395-8618

Definition. If every entry of a square matrix I is zero except for diagonal entries, I_{ii} , which are unity, then the matrix I is called *identity* matrix. A diagonal matrix with diagonal entries 1 becomes identity.

Definition. The *transpose* of a matrix A is denoted by A^T and is defined by interchanging rows to columns or interchanging columns to rows. If A =$[a_{ij}]$, then $A^T = [a_{ji}]$.

Definition. The trace of a matrix A is defined as the sum of entries on its main diagonal. If A = $[a_{ij}]$ then trace(A) = $\sum_i a_{ii}$

Property. For a square matrix A,

$\text{trace}(AA^T) = \text{trace}(A^TA) = \text{trace}(A^2) = |A|^2$

Proof. $\text{trace}(AA^T) = \sum_i \boldsymbol{a_{i\bullet}}\, a_{i\bullet}{}^T = \sum_i \boldsymbol{a_{i\bullet}} \bullet a_{i\bullet}$

$\text{trace}(A^TA) = \sum_i \boldsymbol{a_{\bullet i}}{}^T\, a_{\bullet I} = \sum_i \boldsymbol{a_{\bullet i}} \bullet a_{\bullet i}$

$= \sum_i \sum_k a_{ik}\, a_{ik,} = \sum_i \sum_k a_{ki}\, a_{ki,}$

In either case $= \sum_i \sum_k a_{ik}{}^2 = |A|^2$

It is the sum of squares of all the entries in A,

$|A| = \sqrt{\text{trace}(AA^T)} = \sqrt{\text{trace}(A^TA)} = \sqrt{\text{trace}(A^2)}$

Proposition. For a symmetric matrix A, trace(A) = $\text{trace}(UDU^T)$ = trace(D) = sum of eigenvalues of A, that is, $\text{trace}(A) = \sum_i \lambda_i$

Proof.

$\text{trace}(A) = \text{trace}(UDU^T) =$

$= \sum_i \boldsymbol{u_{i\bullet}}\, D \boldsymbol{u_{i\bullet}}{}^T$

$= \sum_i [\, u_{ik} \lambda_k] \boldsymbol{u_{i\bullet}}{}^T$

where $[\, u_{ik} \lambda_k]$ is a row vector with index k

$= \sum_i \sum_k \; u_{ik} \lambda_k u_{ik}$

$= \sum_k \; \lambda_k \sum_i u_{ik} u_{ik}$

$= \sum_k \; \lambda_k \boldsymbol{u_{\bullet k}} \bullet \boldsymbol{u_{\bullet k}}$ $\quad \mathbf{u}_{\bullet k}$ is a unit column vector.

$= \sum_k \; \lambda_k$

Thus it shows that trace(A) is the sum of eigenvalues of A.

Definition. The mxn matrix A and pxq matrix B are compatible for addition if m=p, and n=q. The *sum* is denoted by A+B and is defined by A+B==$[a_{ij} + b_{ij}]$

Definition. The mxn matrix A and pxq matrix B are compatible for multiplication AB if n=p. If matrices A, B are compatible for multiplication, then *matrix product* AB =$[\sum_{k=1,n} a_{ik}b_{kj}]$ = $[\, r_i c_j]$ = $[\, r_i{}^T \bullet c_j]$ = $[\, r_i \bullet c_j{}^T]$ where r_i is the ith row of A and c_j is the jth column of B.

For a matrix A, the product AA^T and A^TA is called *covariance* of matrix A.

Definition. A square matrix A is *invertible* if there is a matrix B such that AB = I, B is called the *inverse* of A. The *inverse* of invertible matrix A is denoted by A^{-1} so that $AA^{-1} = I$.

Definition. The matrix A is *orthogonal*, if the rows and columns are pairwise orthogonal, and $AA^T = I$, identity matrix.

Property. The *transpose of a product* is the product of transposes in reverse order: $(AB)^T = B^TA^T$.

Definition. The matrix A is *symmetric* if $A = A^T$. The matrix A is self-inverse.

Property. For any matrix A, the product A^TA is a *symmetric* matrix: $(A^TA)^T = A^TA^{TT} = A^TA$.

Definition. The Euclidean *norm* of a matrix A is denoted by |A| and defined by $|A| = \sqrt{\sum_{i,j} (a_{ij}{}^2)}$.

Property: If A and B and two matrices, it is true that $|AB| \le |A||B|$.

Proof. Let r_i be i-th row of A, $r_i{}^T$ be i-th column of A^T. Let c_j be j-th column of B, $c_j{}^T$ be j-th row of B^T. Then

$|AB|^2 = \sum_{i,j} (r_i{}^T \bullet c_j)^2 \le \sum_{i,j} \; |r_i{}^T|^2 |c_j|^2 \le \sum_i |r_i|^2 \sum_j |c_j|^2 \le |A|^2 |B|^2$

Definition. The *rank* of a matrix A is the number of linearly independent rows/columns in a matrix.

Property. The row rank and column rank of a matrix are the same.

Proof. Orthogonal transformation does not change the rank. Since $A = USV^T$, the rank of A is the same as rank of USV^T. It is the same as rank of S. Since S is a diagonal matrix, the row rank and columns rank of a diagonal matrix are same.

Definition. The *determinant* of a matrix A is denoted by det(A). The determinant is computed recursively in terms of row or column and it cofactors.

C. Eigenvalues and Eigenvectors

Definition. Let A be nxn matrix. If there exists a non-zero vector **u** and a number λ such that $A\mathbf{u} = \lambda\, \mathbf{u}$, then λ is called an eigenvalue and **u** is called a corresponding eigenvector.

The equation $A\mathbf{u} = \lambda\, \mathbf{u}$ is called the characteristic equation. If A is an nxn matrix, an eigenvalue of A is a solution of determinant$(A-\lambda I) = 0$. It is a polynomial of degree n, and has n solutions called eigenvalues. The eigenvalues are called eigen (proper, latent, characteristic, singular) values. The eigenvectors are also known as eigen (proper, latent, characteristic, singular) vectors. The eigenvectors and eigenvalues in tandem are referred to as eigenpairs. The coordinate system defined by eigenvectors is called the eigenspace or eigenframe. The transformation matrix is called *rotation* matrix.

Note. An eigenvector is not unique, if **u** is an eigenvector, then any non-zero multiple of **u** is also an eigenvector. To make it *unique*, it is a convention to normalize it to a unit vector, **u**. But **u** and – **u** are unit vectors. Many researchers make first non-zero element in the unit vector positive[Leskovec2014]. This is not satisfactory in some cases: (1) it requires search for the nonzero element and (2) it does not bring about a natural right handed tradition. For example, in Figure A (c), we show a better way to make eigen vectors unique.

ISSN 2395-8618

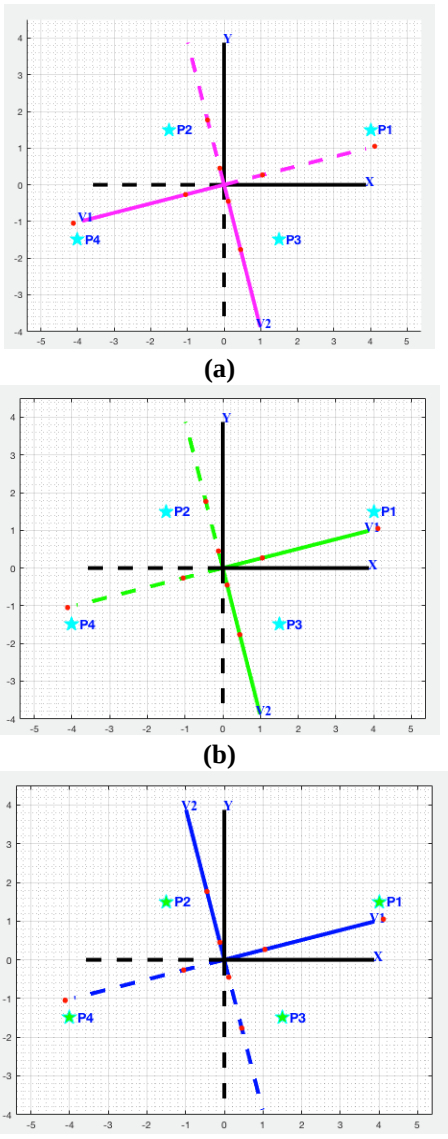


Figure A. (a) Eigenvectors as computed by MATLAB svd, (b) by convention, each vector has first non-zero element positive, (c) our approach, first eigenvector has first non-zero element positive, second eigenvector has second non-zero element positive by using ordering of eigenvectors so the eigenvectors form a right handed system.

Vector space basis is the set of vectors so that every other vector in the space can be expressed as a linear combination of the basis vectors. For R^n, $\mathbf{e_k} = (e_{kj})$ for k=1,n where $e_{kk} = 1$ and e_{kj} is zero for j≠k, $\{ \mathbf{e_k} \}$ is a basis of vectors. In fact any linearly independent set $\{ \mathbf{u_k} \}$ of n vectors can be a basis of R^n. Any n linearly independent, orthonormal unit vectors is an orthogonal basis of R^n.

For SVD, we use these special matrices A^TA and AA^T for calculating the eigenvectors of A^TA and AA^T. Herein, we elaborate the details of some results that we take for guaranteed. For an arbitrary non-symmetric rectangular mxn matrix A, the matrix AA^T is mxm and the matrix A^TA is nxn. Both are symmetric and square matrices.

Proposition The eigenvalues of a real symmetric matrix are real.

Proof. The complex conjugate of u is denoted by $\bar{u}$. Let λ be an eigenvalue of A, then Au = λ u, where u is a unit vector and $\bar{A}\,\bar{u} = \bar{\lambda}\,\bar{u}$. A is real, $A\,\bar{u} = \bar{\lambda}\,\bar{u}$.

$\lambda = \lambda\ \bar{u}^T u = \bar{u}^T \lambda u = \bar{u}^T A u = \bar{u}^T \bar{A}^T u$ for real symmetric A

$= (\bar{A}\,\bar{u})^T u = \bar{\lambda}\ \bar{u}^T u = \bar{\lambda}$

therefore $\lambda = \bar{\lambda}$. Hence λ is a *real* number.

Example. If the matrix is not symmetric, eigenvalues are not necessarily real. For example, let $A = \begin{bmatrix} 1 & 1 \\ -1 & 1 \end{bmatrix}$, it is non-symmetric, its eigenvalues are complex: $1 \pm \sqrt{-1}$.

Example. If A is matrix, it may have repeated eigenvalues. Let $A = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}$, it is symmetric, its eigenvalues are 1. The eigenvectors form a basis of the transformed space. Let $A = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$, it is non-symmetric, its eigenvalues are 1. The eigenvectors do not form a basis of the transformed space.

In PCA, for any matrix A, we calculate eigenvalues and eigenvectors of covariance matrices A^TA (and AA^T) which form the basis of vector space of rows (and columns) of matrix A.

Proposition. The eigenvalues of special real symmetric matrices, A^TA and AA^T, are real and non-negative. Not all symmetric matrices have this property.

Proof. if λ is an eigenvalue and **v** is a unit eigenvector of A^TA, then

$A^TA\,\mathbf{v} = \lambda\,\mathbf{v}$ and $\mathbf{v} \bullet \mathbf{v} = 1$.

Now $\lambda = \lambda\,\mathbf{v} \bullet \mathbf{v} = \lambda\,\mathbf{v}^T\mathbf{v}$

$= \mathbf{v}^T \lambda \mathbf{v} = \mathbf{v}^T(A^TA\mathbf{v})$

$= (\mathbf{v}^TA^T)A\mathbf{v}) = (A\mathbf{v})^T(A\mathbf{v})$

$= (A\mathbf{v}) \bullet (A\mathbf{v}) \geq 0$.

That is non-zero eigenvalues of AA^T and A^TA are positive.

Example. Every symmetric matrix does not have this property.

Let $A = \begin{bmatrix} 1 & 2 \\ 2 & 1 \end{bmatrix}$, it is symmetric, its eigenvalues are -1 and 3. Thus all the eigenvalues of symmetric matrix are *not* always non-negative.

If $A = \begin{bmatrix} 1 & 2 \\ 2 & 4 \end{bmatrix}$, it is symmetric, all the eigenvalues are non-negative: 0, 5.

If $A = \begin{bmatrix} 2 & 1 \\ 1 & 2 \end{bmatrix}$, it is symmetric, all the eigenvalues are positive: 1,3.

Proposition The eigenvectors corresponding to different eigenvalues of a matrix A are linearly independent.

ISSN 2395-8618

Proof. Let $\mathbf{u_1}$ and $\mathbf{u_2}$ be eigenvectors for distinct eigenvalues λ_1 and λ_2. We show that they are linearly independent. Let

$x\mathbf{u_1} + y\mathbf{u_2} = 0$,

then $A(x\mathbf{u_1} + y\mathbf{u_2}) = 0$ or

$x\lambda_1\mathbf{u_1} + y\lambda_2\mathbf{u_2} = 0$, eliminating y we get $x(\lambda_1 - \lambda_2)\mathbf{u}_1 = 0$, since $(\lambda_1 - \lambda_2)\mathbf{u}_1 \neq 0$, $x = 0$ similarly $y = 0$, hence they are linearly independent.

This is true for any number of eigenvectors corresponding to different eigenvalues. It is a useful method of solution of n linear equations.

Proposition The eigenvectors corresponding to different eigenvalues of a real symmetric matrix A are orthogonal.

Proof. Let $\mathbf{u}$ and $\mathbf{v}$ be eigenvectors for eigenvalues λ and μ where $\lambda \neq \mu$.

$$\text{Then } \lambda \mathbf{u^T v} = (\lambda \mathbf{u^T})\mathbf{v} = (A\mathbf{u})^T \mathbf{v} = \mathbf{v}^T(A\mathbf{u}) = (\mathbf{v}^T A^T)\mathbf{u}$$
$$= (A\mathbf{v})^T \mathbf{u} = (\mu v)^T \mathbf{u} = \mu \mathbf{v}^T\mathbf{u}$$
$$= \mu \mathbf{u}^T\mathbf{v}$$

Now $\lambda \mathbf{u^T v} = \mu \mathbf{u^T v}$ or $(\lambda - \mu)\mathbf{u^T v} = 0$.

Since $\lambda \neq \mu$, $\mathbf{u^T v} = 0$ or $\mathbf{u \cdot v} = 0$, therefore $\mathbf{u}$ and $\mathbf{v}$ are orthogonal.

In SVD, we use AA^T and A^TA which are naturally symmetric.

If the eigenvectors are not orthogonal, it will defeat the purpose of simplicity and efficiency. It is possible that an eigenvalue of a matrix is of multiplicity greater than one, that is, corresponding to an eigenvalue there may be several eigenvectors, not necessarily orthogonal. In that case, we can use Gram-Schmit orthogonalization process to create orthogonal set of eigenvectors.

Property [22] . Any real symmetric matrix A can be written as $A = UDU^T = UDU^{-1}$ for some invertible matrix U. Here U is the matrix of eigenvectors of A whereas D is the diagonal matrix of eigenvalues of matrix A.

Proof. Let U be matrix of eigenvectors of matrix A. If $\mathbf{u}_k$, λ_k is an eigenpair of A, the $A\mathbf{u_k} = \lambda_k \mathbf{u_k}$ or

$$A\mathbf{u_k} = \mathbf{u_k}\lambda_k.$$

Then

$$AU = A[\mathbf{u_k}] = [A\mathbf{u_k}] = [\lambda_k \mathbf{u_k}] = [\mathbf{u_k}\lambda_k] = [\mathbf{u_k}]D = UD$$

Since U is invertible matrix, we have

$$A = UDU^{-1}$$

The eigenvectors may be orthogonal, U is orthogonal matrix.

Thus $A = UDU^{-1} = UDU^T$

The eigenvalues may not be positive, except for signs, they are square roots of the eigenvalues of A^2.

Corrolary. Since AA^T is symmetric, therefore $AA^T = UDU^T$, where U is the eigenvector matrix and D is the eigenvalue matrix of AA^T.

Proposition For matrix AA^T, let u be an eigenvector corresponding to non-zero eigenvalue λ. Then A^Tu is an eigenvector of A^TA with the eigenvalue λ.

Proof. Let $\mathbf{u}$ be an eigenvector of AA^T and λ be the corresponding non-zero eigenvalue. Then

$$AA^T \mathbf{u} = \lambda \mathbf{u}$$

Since eigenvalue $\lambda \neq 0$ $\mathbf{u} \neq 0$, therefore $A^T\mathbf{u}$ is a non zero eigenvector and now

$$A^TA(A^T\mathbf{u}) = A^T(AA^T\mathbf{u})$$
$$= A^T\lambda\mathbf{u}$$
$$= \lambda A^T\mathbf{u}$$
$$= \lambda(A^T\mathbf{u})$$

Therefore $A^T\mathbf{u}$ is an eigenvector of A^TA with eigenvalue $\lambda \neq 0$.

Similarly if $\mathbf{v}$ is an eigenvector of A^TA and λ be a corresponding non-zero eigenvalue of A^TA, $A\mathbf{v}$ is an eigenvector of AA^T.

Proposition Let $\mathbf{v}$ be an eigenvector of A^TA and λ be a corresponding non-zero eigenvalue. Then $A\mathbf{v}$ is an eigenvector of AA^T.

Proof. Let $\mathbf{v}$ be an eigenvector of A^TA and λ be the corresponding non-zero eigenvalue. Then

$$A^TA\mathbf{v} = \lambda \mathbf{v}$$

Since eigenvalue $\lambda \neq 0$, $\mathbf{v} \neq 0$, therefore $A\mathbf{v}$ is non zero and now

$$AA^T(A\mathbf{v}) = A(A^TA\mathbf{v})$$
$$= A\lambda\mathbf{v}$$
$$= \lambda A\mathbf{v}$$
$$= \lambda(A\mathbf{v})$$

Therefore $A\mathbf{v}$ is an eigenvector of AA^T with eigenvalue $\lambda \neq 0$.

Proposition. Let $\mathbf{v}_k$ be an eigenvector of A^TA for non-zero eigenvalue λ_k. Then $A\mathbf{v}_k$ is an eigenvector of AA^T, say, $\mathbf{u}_k$, and that $A\mathbf{v}_k = \sigma_K \mathbf{u}_k$ or $\mathbf{u}_k = (1/\sigma_K) A\mathbf{v}_k$ where σ_K is the square root of the corresponding eigenvalue λ_k of A^TA.

Proof. Since $\mathbf{u}_k$ are unit vectors eigenvectors of AA^T , and $\mathbf{v}_k$ are unit vectors eigenvectors of A^TA, $A\mathbf{v}_k$ is some scalar multiple of $\mathbf{u}_k$.

Let $A\mathbf{v}_k = \sigma_k u_k$ for some non-zero σ_K. Since u_k is a unit vector,

$$\sigma_k^2 = \sigma_k \mathbf{u}_k \cdot \sigma_k \mathbf{u}_k = A\mathbf{v}_k \cdot A\mathbf{v}_k$$
$$= \mathbf{v}_k \cdot A^TA\mathbf{v}_k = \mathbf{v}_k \cdot \lambda_k \mathbf{v}_k = \lambda_k$$

or

$$\lambda_k = \lambda_k \mathbf{v}_k \cdot \mathbf{v}_k = A^TA\mathbf{v}_k \cdot \mathbf{v}_k$$
$$= A\mathbf{v}_k \cdot A\mathbf{v}_k = \sigma_k \mathbf{u}_k \cdot \sigma_k \mathbf{u}_k = \sigma_k * \sigma_k = \sigma_k^2$$

Therefore $\sigma_k^2 = \lambda_k$ or $\sigma_k = \sqrt{\lambda_k}$

Hence σ_k is a square root of eigenvalue λ_k .

D. Singular Value Decomposition

Any symmetric positive semi-definite matrix A can be represented as the product of three matrices U, S, V^T where U

ISSN 2395-8618

and V are orthogonal matrices of eigenvectors of AA^T and A^TA; and S is a matrix whose diagonal entries are square roots of eigenvalues of AA^T and A^TA.

Proposition The eigenvalues of AA^T and A^TA are identical except for the zero eigenvalues. Here λ is a non-zero eigenvalue of AA^T if and only if it is eigenvalue of A^TA.

Proof. λ is an eigenvalue of AA^T implies there is a non-zero vector $\mathbf{u}$ such that $AA^T\mathbf{u} = \lambda\mathbf{u}$

$AA^T\mathbf{u} = \lambda\mathbf{u}$ implies $A^TAA^T\mathbf{u} = \lambda A^t\mathbf{u}$

or $A^TA(A^T\mathbf{u}) = \lambda(A^T\mathbf{u})$

which means λ is an eigenvalue of A^TA.

Similarly if λ is an eigenvalue of A^TA, there is a non-zero vector $\mathbf{v}$ such that $A^TA\mathbf{v} = \lambda\mathbf{v}$ implies $AA^TA\mathbf{v} = \lambda A\mathbf{v}$

or $AA^T(A\mathbf{v}) = \lambda(A\mathbf{v})$

which means λ is an eigenvalue of AA^T.

Proposition. If $A = USV^T$ where matrices U and V are orthogonal then U is matrix of eigenvectors of AA^T, V is a matrix of eigenvectors of A^TA and S is diagonal matrix of square roots of non-zero eigenvalues, and conversely.

Proof.

Since $A = USV^T$

then $AA^T = USV^T(USV^T)^T$

$= USV^T(V^{TT}S^TU^T)$

$= USV^T(V^{TT}S^TU^T)$

$= USV^TVS^TU^T$

$= US^2U^T$

Therefore $AA^TU = US^2$.

That is $AA^T\mathbf{u_k} = \mathbf{u_k}s_k^2$ for vectors $\mathbf{u_k}$.

Thus U is matrix of eigenvectors $\mathbf{u}_k$ of AA^T. The diagonal entries s_k^2 of S^2 are eigenvalues of AA^T. Thus the entries s_k of S are square roots of eigenvalues of AA^T.

Similarly we can verify that V is the matrix of eigenvectors of A^TA.

Conversely, to prove the converse, let $\lambda_k, \mathbf{v}_k$ be eigenpair for A^TA, then $A^TA\mathbf{v}_k = \lambda_k\mathbf{v}_k$

We seen above that the eigenvalues of AA^T and A^TA are identical. Now as seen above that for non-zero eigenvalues, the relation between eigenvectors of A^TA and AA^T is $A\mathbf{v}_k = \sqrt{\lambda_k}\,\mathbf{u}_k$ where $\mathbf{v}_k$ is an eigenvector of A^TA and $\mathbf{u}_k$ is an eigenvector of AA^T

For any n-vector x, it can be expressed as linear combination of $\mathbf{v}_k$'s

$\mathbf{x} = \mathbf{x}\bullet\mathbf{v_1}\mathbf{v_1} + \ldots + \mathbf{x}\bullet\mathbf{v_n}\mathbf{v_n}$

$A\mathbf{x} = \mathbf{x}\bullet\mathbf{v_1}A\mathbf{v_1} + \ldots + \mathbf{x}\bullet\mathbf{v_n}A\mathbf{v_n}$

$A\mathbf{x} = A\mathbf{v_1}\mathbf{x}\bullet\mathbf{v_1} + \ldots + A\mathbf{v_n}\mathbf{x}\bullet\mathbf{v_n}$

$A\mathbf{x} = A\mathbf{v_1}\mathbf{v_1}\bullet\mathbf{x} + \ldots + A\mathbf{v_n}\mathbf{v_n}\bullet\mathbf{x}$

From $A\mathbf{v_k} = \sqrt{\lambda_k}\,\mathbf{u_k}$ for k=1,n

$A\mathbf{x} = \sqrt{\lambda_1}\,\mathbf{u_1}\mathbf{v_1}\bullet\mathbf{x} + \ldots + \sqrt{\lambda_n}\,\mathbf{u_n}\mathbf{v_n}\bullet\mathbf{x}$

$A\mathbf{x} = \sqrt{\lambda_1}\,\mathbf{u_1}\mathbf{v_1}^T\mathbf{x} + \ldots + \sqrt{\lambda_n}\,\mathbf{u_n}\mathbf{v_n}^T\mathbf{x}$

$A\mathbf{x} = (\mathbf{u_1}\sqrt{\lambda_1}\mathbf{v_1}^T + \ldots + \mathbf{u_n}\sqrt{\lambda_n}\,\mathbf{v_n}^T)\mathbf{x}$

Since this true for any vector x,

$\mathbf{A} = (\mathbf{u_1}\sqrt{\lambda_1}\mathbf{v_1}^T + \ldots + \mathbf{u_n}\sqrt{\lambda_n}\,\mathbf{v_n}^T)$

Therefore we have proved that $A=USV^T$

E. Calculating PCA from SVD.

We prove that for a *symmetric* matrix A with non-negative eigenvalues, PCA can be derived from SVD. If the columns of U are eigenvectors of AA^T, the columns of V are eigenvectors of A^TA, the diagonal entries of S square root of eigenvalues of A^TA, then SVD of A be $A=USV^T$.

Proposition. Let A be a symmetric matrix positive semi-definite, the $A=USU^T$, with columns of U are eigenvectors of $A^TA = A^2$ if and only if columns of U are eigenvectors of A.

Proof. By SVD algorithm

$A = USV^T$

where The columns of U are eigenvectors of A^2; the diagonal entries of S, square roots of eigenvalues of A^2.

Since A is a symmetric square matrix,

U=V and consequently $A=USU^T$

However,

$A = USU^T$ implies $AU = US$

It means that the columns of U are eigenvectors of A and the diagonal entries of S are eigenvalues of A.

Thus the columns of U are eigenvectors of A^2 if and only if eigenvectors of A, the diagonal entries of S are square roots of eigenvalues of A^2 iff the eigenvalue of A are non-negative.

Note. If A is not positive semi-definite, D can have negative entries corresponding to negative eigenvalues. In this case, PCA cannot be derived for SVD, see example below.

Example. The matrix $A=\begin{bmatrix}1 & 2\\ 2 & 1\end{bmatrix}$ is symmetric so is $AA^T = A^TA = A^2 = \begin{bmatrix}5 & 4\\ 4 & 5\end{bmatrix}$. The eigen values of A are 3, -1, singular value of A are 3,1.

Eigenvectors of $A=\begin{bmatrix}1 & 2\\ 2 & 1\end{bmatrix}$ are $\frac{\begin{bmatrix}1\\1\end{bmatrix}}{\sqrt{2}}$ and $\frac{\begin{bmatrix}1\\-1\end{bmatrix}}{\sqrt{2}}$ corresponding to eigenvalues 3and -1.

Eigenvectors of $A=\begin{bmatrix}5 & 4\\ 4 & 5\end{bmatrix}$ are $\frac{\begin{bmatrix}1\\1\end{bmatrix}}{\sqrt{2}}$ and $\frac{\begin{bmatrix}1\\-1\end{bmatrix}}{\sqrt{2}}$ corresponding to eigenvalues 9 and 1.

Eigenvectors are same, PCA and SVD are not some..

PCA: $UDU^T = \frac{\begin{bmatrix}1 & 1\\1 & -1\end{bmatrix}}{\sqrt{2}}\begin{bmatrix}3 & 0\\0 & -1\end{bmatrix}\frac{\begin{bmatrix}1 & 1\\1 & -1\end{bmatrix}}{\sqrt{2}} = \frac{\begin{bmatrix}3 & -1\\3 & 1\end{bmatrix}}{\sqrt{2}}\frac{\begin{bmatrix}1 & 1\\1 & -1\end{bmatrix}}{\sqrt{2}} = \frac{\begin{bmatrix}2 & 4\\4 & 2\end{bmatrix}}{2} = \begin{bmatrix}1 & 2\\2 & 1\end{bmatrix} = A.$

However, SVD: $USU^T = \frac{\begin{bmatrix}1 & 1\\1 & -1\end{bmatrix}}{\sqrt{2}}\begin{bmatrix}3 & 0\\0 & 1\end{bmatrix}\frac{\begin{bmatrix}1 & 1\\1 & -1\end{bmatrix}}{\sqrt{2}} = \frac{\begin{bmatrix}3 & 1\\3 & -1\end{bmatrix}}{\sqrt{2}}\frac{\begin{bmatrix}1 & 1\\1 & -1\end{bmatrix}}{\sqrt{2}} = \frac{\begin{bmatrix}4 & 2\\2 & 4\end{bmatrix}}{2} = \begin{bmatrix}2 & 1\\1 & 2\end{bmatrix}$ is not the same as A.

Hence PCA cannot be derived from SVD if A is not postitive

ISSN 2395-8618

semi-definite.

Orthogonal matrix is also called a *rotation* matrix, because this matrix rotates the original coordinate axes. Rotation does not change lengths and relative angles as seen below.

Property. If R is orthogonal matrix and **u** is a vector, then

$$|R\mathbf{u}|=|\mathbf{u}|$$

Proof.

$|R\mathbf{u}|^2=(R\mathbf{u})^T R\mathbf{u} = \mathbf{u}^T R^T R\mathbf{u}$,
since R is orthogonal $R^TR=I$
$|R\mathbf{u}|^2 = \mathbf{u}^T I\mathbf{u} = \mathbf{u}^T\mathbf{u} = |\mathbf{u}|^2$
Therefore $|R\mathbf{u}|=|\mathbf{u}|$

Property. If R is orthogonal matrix and A is matrix, then

$$|RA|=|A|$$

Proof. Let $a_{\bullet j}$ be j-th column of A. Using the rotation property of vectors,

$$|RA|^2 = \sum_{j=1,n} |R\, a_{\bullet j}|^2 = \sum_{j=1,n} |a_{\bullet j}|^2 = |A|^2$$

Property. If U and V are orthogonal and S is a diagonal matrix, then

$$|USV^T|=|D|$$

Proof. Using the rotation property of matrices,

$$|USV^T| = |SV^T| = |VS^T| = |S^T| = |S| = |D|$$

Property. If rows/columns corresponding to smaller variation are deleted, there is smaller loss of information. If rows/columns corresponding to zero eigenvalues only are deleted, then there is no loss of information in the reduced dimensionality.

Proof. By SVD, there exist U, V, S such that $A = USV^T$. Now $A' = U(:,1:k)\ S(1:k,1:k)\ V(1:k,:)^T$ by deleting m-k columns after first k columns in U and n-k columns after first k columns in V, after deleting all rows and all columns after first k rows and k columns in S. Let S_{new} be S corresponding to dimension reduction, by zeroing all eigenvalues except first k diagonal entries. Let S_{new} correspond to dimension reduction. The A' is the same as $B= US_{new}V^T$. In this reduction, loss of information is |A-B|, whereas A and B have the same size mxn.

Now $A=USV^T$ $B=US_{new}V^T$

$$|A-B| = |USV^T - US_{new}V^T|$$
$$= |U(S - S_{new})V^T|$$

using orthonormality of column vectors of U and V we have

$$|A-B| = |S - S_{new}|$$
$$= |S - S_{new}|$$
$$= \sqrt{(\sum_{p>k} s_{pp}^2)}$$
$$= \sqrt{(\sum_{p>k} \lambda_p)}$$

This shows that the smaller the value of $\sqrt{(\sum_{p>k} \lambda_p)}$, the smaller the norm |A-B|, the closer A and B. If all eigenvalues λ_p with p>k are zero, then there is no loss of information.

Here are two interesting result.

Property. If A is symmetric positive semi definite, the $A= B^TB$ for some symmetric positive semi definite B.

Proof. By SVD, we have $A= USU^T$. The entries if S are non-negative. Let $B =U\sqrt{S}U^T$. Since A is symmetric, B is symmetric.

$$BB^T = U\sqrt{S}U^T(U\sqrt{S}U^T)^T$$
$$= U\sqrt{S}U^T\, U^{TT}\sqrt{S}^TU^T$$
$$= U\sqrt{S}U^T\, U\sqrt{S}U^T$$
$$= U\sqrt{S}\sqrt{S}U^T$$
$$= USU^T$$
$$= A$$

This property show that SVD transforms correlated data into uncorrelated data.

Property. If A is symmetric positive semi definite, there is a transformation M such that covariance matrix $MA(MA)^T$ is diagonal.

Proof. By SVD, we have $A= USU^T$.

Let $M=U^T$

Then $MA= SU^T$

Now $MA(MA)^T = SU^T (SU^T)^T$
$= SU^T US^T$
$= SS^T$
$= S^2$

Note. Let A is mxn, U is mxm, V is nxn, $Av_k = \sigma_k u_k$ and if $AA^t u_k = \lambda_k u_k$, then $A^t u_k = \sigma_k v_k$ and $\sigma_k = \sqrt{\lambda_k}$.

If m<n, we compute V first and then U*S=AV. If m>n, then we compute U first and then $V*S^I = A^TU$.

Since S is diagonal, its inverse is reciprocal of the diagonal entries, except for zero entries which are left unchanged. In case of zero entries, it becomes Pseudo inverse denoted by S^I. Pseudo inverse is left inverse if m>n otherwise it is left inverse.

Eitherway $U = AV\ S^I$ or $V^T = S^I\ A^TU$ or $V = U^TA\ S^I$ where S^I is pseudo inverse. This is computationally more stable.

Once U, and V are computed, S can be quickly verified from $S = U^TAV$.

REFERENCES

[1] Saraçlı, S., Yılmaz, V., & Doğan, İ. (2009b). Simple linear regression techniques in measurement error models. *Anadolu University Journal of Science and Technology*, 10(2), 335-342.

[2] Stefanski, L.A. (2000). Measurement error models. *Journal of the American Statistical Association*, 95(452), 1353-1358.

[3] McCartin, B. J, (2003). A geometric characterization of linear regression. *Statistics*, 37(2), 101–117. http://dx.doi.org/10.1080=0223188031000112881

[4] Ding, G., Chu, B., Jin, Y., & Zhu, C. (2013). Comparison of orthogonal regression and least squares in measurement error modeling for prediction of material property. *Nanotechnology and Material Engineering Research, Advanced Materials Research*, 661, 166-170.

ISSN 2395-8618

http://dx.doi.org/10.4028/
www.scientific.net/AMR.661.166

[5] Leng, L., Zhang, T. Kleinman, L., & Zhu, W. (2007). Ordinary least square regression, orthogonal regression, geometric mean regression and their applications in aerosol science. *Journal of Physics, Conference Series* 78(1), 1-5. Retrieved from http://iopscience.iop.org/1742-6596/78/1/012084

[6] Steven C Chapra and Raymond P Canale, Numerical Methods for Engineers, 7th Edition, ISBN: 978 0073397924, McGraw-Hill Publishers, 2015.

[7] Cohen, J., Cohen P., West, S.G., & Aiken, L.S. (2003). Applied multiple regression/correlation analysis for the behavioral sciences. (2nd ed.) Hillsdale, NJ: Lawrence Erlbaum Associates.

[8] Draper, N.R.; Smith, H. (1998). Applied Regression Analysis (3rd ed.). John Wiley. ISBN 0-471-17082-8.

[9] Taliha Keles, Comparison of Classical Least Squares and Orthogonal Regression in measurement error models, International Online Journal of Educational Sciences, 10(3), 200-20014.

[10] Yehuda Koren, Robert Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. Computer 42, 8 (2009).

[11] Stephen Vaisey, Treatment Effects Analysis, https://statisticalhorizons.com/seminars/public-seminars/treatment-effects-analysis-spring17.

[12] Gwowen Shieh, Clarifying the role of mean centering in multicollinearity of interaction effects, British Journal of Mathematical and Statistical Psychology (2011), 64, 462–477.

[13] Jim Hefferon, Linear Algebra, Free Book, http://joshua.smcvt.edu/linearalgebra, 2014.

[14] John F. Hughes, AndriesVan Dam,Morgan McGuire, David F. Sklar, James D. Foley, Steven K. Feiner, Kurt Akler Computer Graphics: principle and Practice, 3rd edition , Addison Wesley, 2014.

[15] Chaman Sabharwal, Hybrid Linear Least Square and Singular Value Decomposition Approximation, International Journal of Trend in Research and Development, Volume 5(3), ISSN: 2394-9333 www.ijtrd.com May-Jun 2018, pp. 1-8.

[16] Sabharwal, Chaman Lal, SVD Adaptive Algorithm for Linear Least Square Regression and Anomaly Reduction, IOSR Journal of Computer Engineering (IOSR-JCE) e-ISSN: 2278-0661,p-ISSN: 2278-8727, Volume 20, Issue 5, Ver. III (Sep - Oct 2018), PP 33-48, www.iosrjournals.org

[17] Carr, J.R. (2012). Orthogonal regression: A Teaching perspective. *International Journal of Mathematical Education in Science and Technology*, 43(1), 134-143. http://dx.doi.org/10.1080/0020739X.2011.573876

[18] P. Groves, B. Kayyali, D. Knott, S. V. Kuiken, "The 'Big Data' Revolution in Healthcare", *Center of US Health System Reform Business Technology Office*, pp. 1-20, 2013.

[19] C. C. Yang, L. Jiang, H. Yang, M. Zhang, "Social Media Mining for Drug Safety Signal Detection" *ACM SHB'12*, October 29, 2012, Maui, Hawaii, USA.

[20] Jure Leskovec, Anand Rajaraman, Jeffrey D Ullman, Datamining of Massive Datasets, 2014.

[21] Patrick J.F. Groenen, Michel van de Velden, Multidimensional Scaling, Econometric Institute EI 2004-I5, Erasmus University Rotterdam, Netherlands, 2015.

[22] Jonthan Shlens A Tutorial on Principal Component Analysis, arXiv:1404.1100 [cs.LG], pp. 1-15,2014

[23] Sebastian Raschka Principal Component Analysis in 3 Simple Steps LSA-Least Squares Approximation http://sebastianraschka.com/Articles/2015_pca_in_3_steps.html, 2015.

[24] Abdi, Hervé, Beaton, Derek, Principal Component and Correspondence Analyses Using R, Springer, ISBN 978-3-319-09256-0, Digitally watermarked, DRM-free, 2017.

[25] Caroline J Anderson, Psychology Lecture Notes: Principal Component Analysis, 2017.

[26] H. Y. Chen, R. LiÅLegeois, J. R. de Bruyn,and A. Soddu, "Principal Component Analysis of Particle Motion", *Phys. Rev.* E 91, 042308 - 15 April 2015.

[27] Karen Bandeen-Roche Nov 28, 2007, An Introduction to Latent variable Models, http://www.biostat.jhsph.edu/~kbroche/Aging/Intro to Latent VariableModels.pdf.

[28] Yusuke Ariyoshi and Junzo Kamahara. 2010. A hybrid recommendation method with double SVD reduction. In International Conference on Database Systems forAdvanced Applications. Springer, 365–373.

[29] Chaman Sabharwal, Principal Component Analysis and Qualitative Spatial Reasoning, 28th International Conference on Computer Applications in Industry and Engineering, CAINE 2015, October 12-14, 2015, San Diego, California, USA pp.23-28.

[30] Matlab, https://www.mathworks.com/downloads/

[31] Mark Tygert, Regression-aware decompositions, arXiv1710.04238v2, 12 Feb 2018.