ISOMAP AS DIMENSION REDUCTION ALGORITHM

MARIUSZ BUDZIŃSKI

Abstract

Central object of interest of Data Analysis are data. Being more precise I should rather say: data sets, which I will understand as observations made from a statistical population. Data sets can be obviously characterizesed by dimension. I trick this concept as basic without giving the definition. It turns out that too big dimensionality of a data set might imply negative consequences for general quality of our analysis. The goal of this report is to introduce with applications Isomap algorytm, which is one of the Dimension Reduction methods. Before doing that I will introduce very short motivation for Dimension Reduction methods in general.

1. Introduction

At the beginning we are given problem to solve, whatever it is, we are also given data set $\mathbf{X} = \{\mathbf{X}_{np}\}_{n,p\in\mathbb{N}}$ in order to do that. Our goal is to introduce the best algorythm $\mathcal{A}_{\{\mathbf{X}_{np}\}}(\cdot)$, which for every observation $\mathbf{y}$ of $\{\mathbf{X}_{np}\}$ type will give answear $\mathcal{A}_{\{\mathbf{X}_{np}\}}(\mathbf{y})$. The existance of the best algorythm means that we are given an order to compare algorythms. Natural question arises how to find the best one? Let's make an important observation.

Observation 1. *In order to find the best solution of our problem we can manipulate with data set* $\mathbf{X}$*.*

That is why we might consider collection of best algorythms $\{\mathcal{A}_{\{\mathbf{X}_{\mu,\nu}\}}\}_{\mu\leq n,\nu\in\mathbb{N}}$, based on the $\{\mathbf{X}_{\mu,\nu}\}_{\mu\leq n,\nu\in\mathbb{N}}$ which is new data set created from $\mathbf{X}$ which was given at the beginning. To simplyfy our consideration we can also assume that we are not generating observation by ourself, that is why $\mu \leq n$. Notice that the number of features $\nu \in \mathbb{N}$ might be larger than p because we can easly generate new variables, for example by using transformation made on the old one's. We can also assume that our initial data set $\mathbf{X}$ is without systematic errors so *a priori* the best algorythm is with $\mu = n$. What about ν? To answear this question consider phenomena called *Curse of dimensionality.*

Curse of dimensionality refers to various phenomena that arise when analyzing data in high-dimensional spaces. Intuitively when dimensionality of data set growths, more and more observations have to be included in data set to be sure that random sample represents all properties of the population. It is very often that number of observation needed to maintain a statistical properties of given sollution growths exponentially with dimensionality. There are plenty of other issues like duplication of information through several features so field of interest of feature selection but I will skip this part. For now I would like to consider cases when *Curse of dimensionality* arises purly because of growing dimensionality of independent features. Consider two following examples.

Example 1. *Let's consider* $p \in \mathbb{N}$ *binary independent variables which are included in every single observation. Fix also number of given observations and denote it as* $n \in \mathbb{N}$*. For sure there are* 2^p *unique observations. So to have chances of having at least one of any individual case in data set we need that* $n \geq 2^p$*.*

So number of needed observation growths very rapidly with number of features. Now consider so called *peaking phenomena* which is phenomenon of decreasing and then increasing of error made by specific classificaton algorythm with groth of number of features p for any but fixed number of observations n. To illustrate this phenomena I will present TRUNK'S example (1979).

Example 2. *On common probability space are given random variables:* $\mathbf{X}, \mathbf{Y}$*:*

$$\mathbb{P}(\mathbf{Y} = y_1) = \mathbb{P}(\mathbf{Y} = y_2) = \frac{1}{2}$$
$$(\mathbf{X}|\mathbf{Y} = \{y_1, y_2\}) \sim \mathcal{N}(\{\mu, -\mu\}, \mathbb{I}_p)$$

where $\mu = (\frac{1}{i}^{\frac{1}{2}})_{i \le p}$*. We are given an independent n-sample matrix* $\mathbf{X}_{np}$ *and vector* $\mathbf{Y}_n$ *of its classes. Our task is to predict for every observation* $x = (x_j)_{j \in \overline{1,p}}$ *its class. Solution will be based on Bayes classifier:*

For given x its class is of index

$$j = \operatorname*{argmax}_{i \in 1,2} \mathbb{P}(\mathbf{Y} = y_i | \mathbf{X} = x) \tag{1}$$

Now let's apply Bayes theorem:

$$\mathbb{P}(\mathbf{Y} = y_i | \mathbf{X} = x) = \frac{p(\mathbf{X} = x)|\mathbf{Y} = y_i)\mathbb{P}(\mathbf{Y} = y_i)}{\mathbb{P}(\mathbf{X} = x)}.$$

That will give us that $j = 1$ in (1) if the following condition hold:

$$\mathbb{P}(x|\mathbf{Y} = y_2) \le \mathbb{P}(x|\mathbf{Y} = y_1) \iff \mu^T x > 0.$$

Now cosider error of this classifier.

$$\begin{aligned} P_{error} =& \mathbb{P}(\mu^T X > 0 | \mathbf{Y} = y_2)\mathbb{P}(\mathbf{Y} = y_2) + \mathbb{P}(\mu^T X < 0 | \mathbf{Y} = y_1)\mathbb{P}(\mathbf{Y} = y_1) = \\ & \mathbb{P}(\mu^T X > 0 | \mathbf{Y} = y_2) \\ P_{error} =& \int_0^\infty \frac{1}{\sqrt{2\pi\mu^T\mu}} e^{-\left(x + \mu^T\mu\right)^2/2\mu^T\mu} \, \mathrm{d}x = \frac{1}{\sqrt{2\pi}} \int_{\sqrt{\mu^T\mu}}^\infty e^{-x^2/2} \, \mathrm{d}x = \\ & \frac{1}{\sqrt{2\pi}} \int_{\sqrt{\sum_{i=1}^p \frac{1}{i}}}^\infty e^{-x^2/2} \, \mathrm{d}x \xrightarrow{p \to \infty} 0 \end{aligned}$$

Everything works if we know the value of μ. If it is unknown, which is always the case, we can use empirical estimator $\hat{\mu}$ and to simplify calculation Central Limit Theorem as well.

$$P_{error} = \mathbb{P}(\hat{\mu}^T \mathbf{X} \ge 0 | Y = y_2) = \mathbb{P}(\frac{\hat{\mu}^T \mathbf{X} - \mathbb{E}(\hat{\mu}^T \mathbf{X} | Y = y_2)}{\sqrt{\operatorname{Var}(\hat{\mu}^T \mathbf{X} | Y = y_2)}} \ge \frac{-\mathbb{E}(\hat{\mu}^T \mathbf{X} | Y = y_2)}{\sqrt{\operatorname{Var}(\hat{\mu}^T \mathbf{X} | Y = y_2)}} | Y = y_2)$$

Remark

$$\begin{aligned} \mathbb{E}(\hat{\mu}^T \mathbf{X} | Y = y_2) =& - \mu^T\mu = -\sum_{i=1}^p \frac{1}{i}, \\ \operatorname{Var}(\hat{\mu}^T \mathbf{X} | Y = y_2) =& \frac{p}{n} + (1 + \frac{1}{n}) \sum_{i=1}^p \frac{1}{i}. \end{aligned}$$

Finally we have

$$P_{error} = 1 - \Phi(\frac{\sum_{i=1}^p \frac{1}{i}}{\sqrt{\frac{p}{n} + (1 + \frac{1}{n}) \sum_{i=1}^p \frac{1}{i}}}),$$

where $\Phi(x)$ is distribution function of $\mathcal{N}(0, 1)$. Notice that:

(1)
$$\frac{\sum_{i=1}^p \frac{1}{i}}{\sqrt{\frac{p}{n} + (1 + \frac{1}{n}) \sum_{i=1}^p \frac{1}{i}}} \xrightarrow{p \to \infty} 0 \rightsquigarrow P_{\text{error}} = \frac{1}{2}$$

(2) *By taking n larger and larger $\rightsquigarrow$ decreased speed of convergence*

This observations are visible at the following Figure.1 also as mentioned earlier peaking pheomena. We see that probability of wrong classification at first decrease with number of features and then increases very rapidly. Also the interval where it is indeed worth to increase number of dimension is very short.

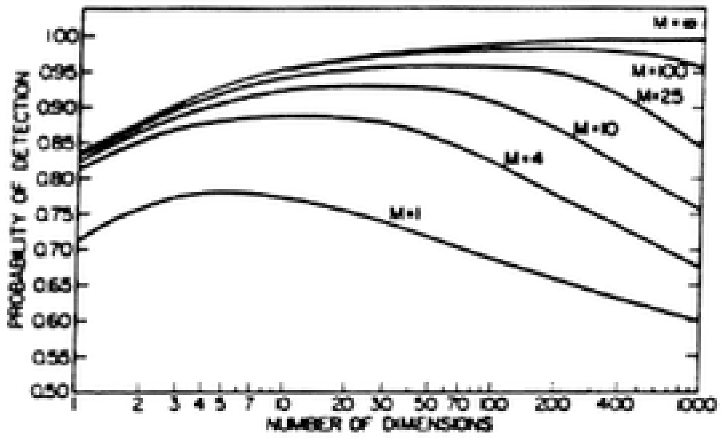

FIGURE 1. Figure taken from Trunk, G. V. (July 1979). "A Problem of Dimensionality: A Simple Example". IEEE Transactions on Pattern Analysis and Machine Intelligence. PAMI-1 (3): 306–307. M refers to the number of observations.

That is why it is better to have less dimensions than more which is the same conclusion that folow the first example. That is why dimensionality reduction is believed to might be useful. To end the motivation part look at other advantages of dimensionality reduction:

(1) droping correlation of features
(2) easier interpretability of our model and saving memory
(3) less chances of overfitting and easier data visualization.

Focus now on Isomap.

2. Isomap algorithm

Algorithm was introduced in [1]. At the begining we need to have defined some dissimilarity function between observations which is the distance between points. Note that it can be defined not only for numerical data. So Isomap algorithm might be used in various of cases, also for not numerical data sets, and it might be summarised in the following steps:

(1) Create weighted graph $\mathbb{G}$ where nodes are observations and the edges of the graph $\mathbb{G}$ represent that two points are neighbours. Weights are distances between neighbours. We can consider two cases:
   (a) ϵ Isomap, where neighbours of particular observation are those observations which are closer to it than ϵ
   (b) k Isomap, where neighbours of particular observation are k the closest observations to it

   We need to choose at the begining which version of Isomap algorithm we will use. Dependent on versions we do have to choose ϵ neighbourhood range or k number of neighbours, both of them are hyperparameters.
(2) Using distances between neighbours create A_{ij} which is the distance matrix between **all** points, where

$$\mathbf{A}_{ij} = \begin{cases} \mathbb{G}_{ij} & \text{for } x_i \text{ and } x_j \text{ being neighbours} \\ \text{geodesic distance} & \text{for points not being neighbours} \end{cases}$$

   Where $\mathbb{G}_{ij}$ are weights in graph $\mathbb{G}$. Geodesic distance between two points which are not neighbours is the length of the shortest path in graph $\mathbb{G}$ that connects this points. By length of the path I understand sum of weights of the edges in given path. To do that you can use for instance *Floyd Algorithm*:

(a) Start with

$$\mathbf{A}_{ij} = \begin{cases} \mathbb{G}_{ij} & \text{for } x_i \text{ and } x_j \text{ being neighbours} \\ \infty & \text{for points not being neighbours} \end{cases}$$

(b) Denote $NS(x_i, x_j, \{x_l\}_{l\in\overline{1,k}})$ as the shortest path between x_i and x_j going through $\{x_l\}_{l\in\overline{1,k}}$, where of course $NS(x_i, x_j, \{x_l\}_{l\in\overline{1,1}}) = \mathbf{A}_{ij}$

(c)

$$NS(x_i, x_j, \{x_l\}_{l\in\overline{1,k+1}}) = min(NS(x_i, x_j, \{x_l\}_{l\in\overline{1,k}}), \\ NS(x_i, x_{k+1}, \{x_l\}_{l\in\overline{1,k}}) + NS(x_{k+1}, x_j, \{x_l\}_{l\in\overline{1,k}}))$$

(d) Repeat Step c) ending when $k = n$ then $\mathbf{A}_{ij} = NS(x_i, x_j, \{x_l\}_{l\in\overline{1,n}})$ so we have used all observations and that is why it is desired distance matrix.

(3) Perform *Metric Multidimensional scaling* (MDS) on geodesic distance matrix $\mathbf{A}$

Now I will describe with details MDS starting with general considerations which will be crucial for MDS. Our goal for now is to find points in $\mathbb{R}^d$, $d \in \mathbb{N}$, with a L_2 distance matrix equal to $\mathbf{A}$ from 2). Let's also make an important observation.

Observation

Distance matrix $\mathbf{A}$ from 2) is symmetric matrix having $\mathbf{A}_{ii} \equiv 0$ and $\mathbf{A}_{ij} \geq 0$. This motivates the following theorem.

Theorem 2.1. *Let* $\mathbf{H} = \mathbb{I} - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T$ *and* $\mathbf{A}$ *be symmetric matrix having* $\mathbf{A}_{ii} \equiv 0$ *and* $\mathbf{A}_{ij} \geq 0$. *Then* $\mathbf{A}$ *is matrix of euclidean distances* $\iff$ $-\mathbf{HAH}$ *is positive defined.*

Proof. $\Rightarrow$

We have assumed that $\mathbf{A}_{ij} = \langle a_i - a_j, a_i - a_j\rangle = \|a_i\| + \|a_j\| - 2\langle a_i, a_j\rangle$ for some $(a_i)_{1,n} \in \mathbb{R}^s$, for some not specified s. Which can be also rewriten as follows :

$$\mathbf{A}_{ij} = \mathbf{C}_{ij} + \mathbf{C}_{ij}^T - 2(\mathbf{BB}^T)_{ij},$$

where the definition of $\mathbf{C}_{ij}$ is obvious and

$$\mathbf{B} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ \vdots \\ a_n \end{bmatrix}. \tag{2}$$

Notice also that due to the definition on $\mathbf{C}$ we have that $\mathbf{HCH} = 0$ and that is why

$$-\mathbf{HAH} = 2\mathbf{HBB}^T\mathbf{H} = 2(\mathbf{B}^T\mathbf{H})^T\mathbf{B}^T\mathbf{H}$$

To check if $-\mathbf{HAH}$ is positive defined consider the following

$$-v^T\mathbf{HAH}v = 2\langle\mathbf{B}^T\mathbf{H}v, \mathbf{B}^T\mathbf{H}v\rangle \geq 0.$$

The last inequality follow straightforward the definition of the inner product.

$\Leftarrow$ Because $-\mathbf{HAH}$ is positively defined and it is symmetric matrix it can be seen as scalar product of some collection of vectors, for instance $-\mathbf{HAH} = (\mathbf{B}^T\mathbf{H})^T\mathbf{B}^T\mathbf{H}$. Using notation as in (2) we should notice that

$$\mathbf{B}^T\mathbf{H} = (a_i - \frac{1}{n}\sum_{j=1}^{n} a_j)_{i\in 1,n}.$$

so the matrix of scalar product of the above collection of vectors gives us the $-\mathbf{HAH}$. Then also because

$$\left\|(a_i - \frac{1}{n}\sum_{j=1}^{n} a_j) - (a_j - \frac{1}{n}\sum_{j=1}^{n} a_j)\right\|_2 = \|a_i - a_j\|_2 ,$$

we have that euclidean distances of $\{\frac{1}{\sqrt{2}}(a_i - \frac{1}{n}\sum_{j=1}^{n} a_j)\}_{i\in 1,n}$ gives us $\mathbf{A}$ matrix. □

Now using notation of Theorem 2.1 if $-\mathbf{HAH}$ is positive defined and we know that it is symmetric we can perform eigendecomposition:

$$-\mathbf{HAH} = \mathbf{W}\Lambda\mathbf{W}^T = (\mathbf{W}\Lambda^{\frac{1}{2}})(\Lambda^{\frac{1}{2}}\mathbf{W}^T),$$

where Λ is matrix of eigen values of $-\mathbf{HAH}$ and $\mathbf{W}$ matrix of columns as eigen vectors. So finally also using Theorem. 2.1 and it's proof we have that

$$\begin{aligned} -\mathbf{HAH}_{ij} &= \langle \beta_i, \beta_j \rangle, \text{ where} \\ \beta_i &:= (\sqrt{\lambda_j}\mathbf{W}_{ij})_{j\in\overline{1,n}} \text{ and} \\ \mathbf{A}_{ij} &= \left\| \frac{1}{\sqrt{2}}\beta_i - \frac{1}{\sqrt{2}}\beta_j \right\|_2 . \end{aligned} \tag{3}$$

So for given dissimilarity matrix $\mathbf{A}$ we have found vectors $\{\frac{1}{\sqrt{2}}\beta_i\}_{i\in 1,n}$ in $\mathbf{R}^n$ **for which matrix of euiclidean distances** gives $\mathbf{A}$. Also λ_j are ordered with decreasing order. Because if the higher is the value of eigenvalue, the more information is related to it's eigenvector. Now we are ready to introduce steps of MDS:

(1) Calculate the matrix of geodesic distances $\mathbf{A}$
(2) Diagonalise $\frac{-1}{2}\mathbf{HAH}$ and find eigen vectors matrix $\mathbf{W}$ also as eigen values matrix $\mathbf{\Lambda}$
(3) Negative eigen values change to 0
(4) Select $s \prec n$ columns of $\mathbf{W}\Lambda^{\frac{1}{2}}$
(5) Points after dimension reduction are rows of cuted $\mathbf{W}\Lambda^{\frac{1}{2}}$

First two points refers to the (3) where $\beta_i \in \mathbb{R}^n$ are defined and their distance matrix is equal to $\mathbf{A}$. But it could be also the case that $-\mathbf{HAH}$ is not positively defined so after performing igendecomposition it might appear that some of eigen values λ_i are negative. Then from (3) follows that β_i will be complex. To fix this problem we have step (3). So to perform scaling we basically leaving first s dimension of β_i as in (3). Because we have ordered eigenvalues in decreasing order so information lost due to forgeting about the last dimensions of β_i will be the smallest. The MDS is now complete. The very important thing about Isomap algorythm is that distance matrix is made with usage of geodesic lines. That means we are trying to preserve local structure of 'manifold' on which data lie while doing dimension reduction. That is why Isomap is often categorised as nonlinear method or 'manifold learning' method. Now to complete our report let's look at applications.

3. Applications

Examples were made in *Python* with usage of *sckit-learn* library. In the first example I want to show usefulness of the Isomap algorithm. Let's consider data set which is called *swiss roll.* For those not knowing *swiss roll* is a kind of cake and it is attained on the following Figure 2. Aditionally on figure are distinguished different classes by different colors. It will be very helpful to evaluate usability of Isomap algorithm performing dimensionality reduction for classification tasks.

We can easly see that structure of this data set is not linear, that is why I have choosen this data set. I will also compare Isomap algorithm which is thought to be non linear with the most popular linear technique, PCA algorithm. Also to show usefulness of using geodesic distances I will perform MDS which is used in Isomap.

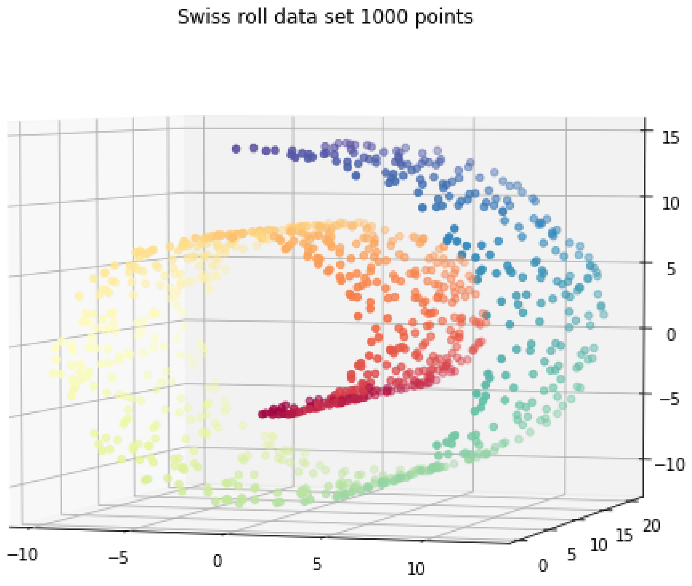

FIGURE 2. Swiss roll data set.

In the following Figure 3 there are attained results of performing dimesion reduction to two dimensions. We have used k Isomap with 10 neighbours with L2-metric dissimilarity matrix. The dissimilarity matrix with L_3 distances was used for MDS.

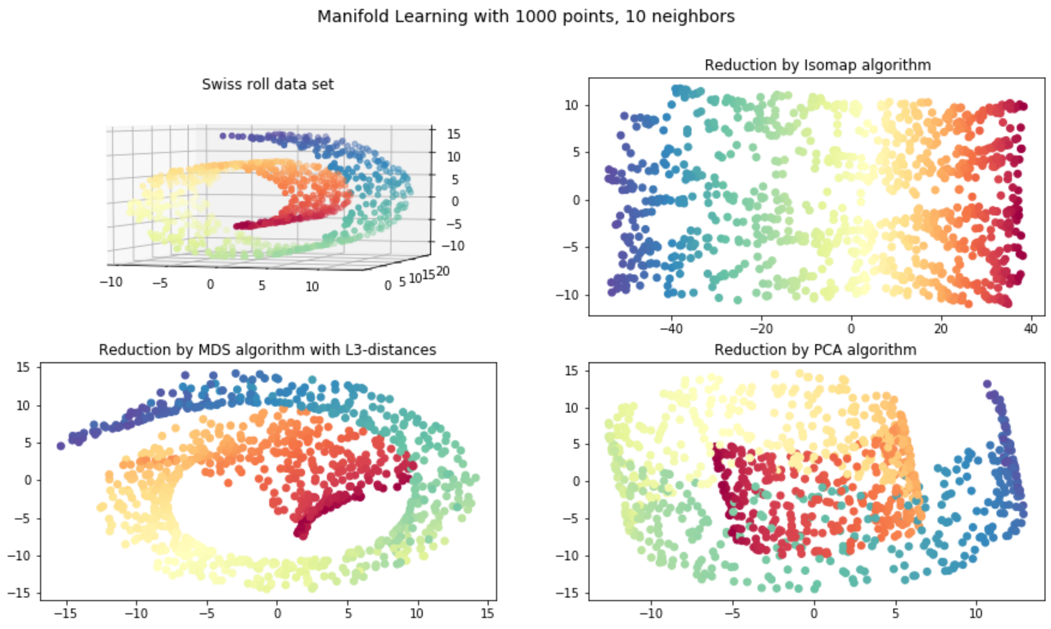

FIGURE 3. Dimension reduction for Swiss roll data set with usage of different algorithms.

We should make important remark that in case of L_2 distances MDS is equivalent to the PCA method that is why I have used L_3 norm. It is worth mantioning that with L_1 distance results for MDS were not satisfactory and subjectively worse than for PCA . MDS and PCA algorithms behave very similarly and the utility of both methods still leave something to be desired. The mentioned classification task is imposibble to perform in satisfactory way without further processing. It's a little bit different for Isomap. We see that classes are seperated very well and original *swiss roll* is unrolled. Comparing MDS with Isomap we can see big advantage of using geodesic distances between points. Because k number of neighours used in Isomap algorithm is hyperparameter I have run algorith with different k, results are attaind in the following Figure 4. Where *reconstruction error* is defined as follows:

$$\text{Reconstruction error} = \text{frobenius norm}\ \{\tfrac{[K(D)-K(D_{\text{fit}})]}{n_{\text{samples}}}\}.$$

Where D is the matrix of distances for the input data X, D_{fit} is the matrix of distances for the output embedding before reduction and K is the isomap kernel:

$$K(D) = -0.5(I - \tfrac{1}{n_{\text{samples}}})D^2(I - \tfrac{1}{n_{\text{samples}}}).$$

Details can be found at *sklearn library* documentation of *Isomap* function. As we see hyperpa-

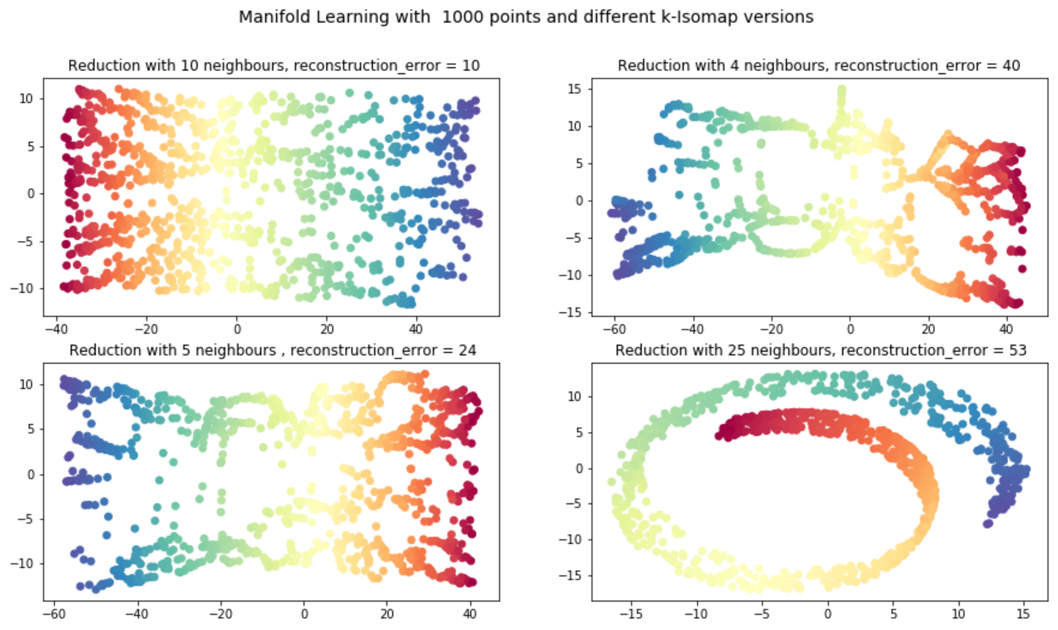

FIGURE 4. Dimension reduction for Swiss roll data set with usage of k Isomap with different number of neighbours.

rameter k have big impact on algorithm behaviour. When it is getting smaller and smaller the structure after reduction is becoming 'denser' and 'denser'. The very interesting case is when we are taking more neighbours. Dimension reduction with 25 neighbours gives us very similar result as MDS and PCA gave and as more neighbours we are using then the more simmilar the result is. It is not surpraseing because geodesic distance are getting closer to the simple euclidean distances between points. That is why Isomap might be found as extantion of MDS and in particular PCA. One more important observation should be made. The algorithm is very sensitive on k! Because it is hyperparameter there is no strict rule to fix it except for instance comparison of *reconstruction error*.

Another example is the real word example naimly *MNIST* data set which is database of 70.000 handwritten digits of size 28x28 pixel. For my purposes at the beginning I have choosen 1000 random digits. In the following Figure 5 are attained few examples of them:

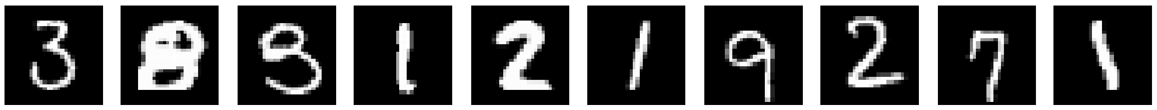

FIGURE 5. Examples of elements of the MNIST data set

Now we would like to test usability of the Isomap algorithm in task of classification of MNIST digits. For prior choosen examples I have performed dimension reduction with Isomap. To find the number k of neighbours and the number of dimension for reduction I have used *grid search* and the obtained results are attached at the following Figure 6

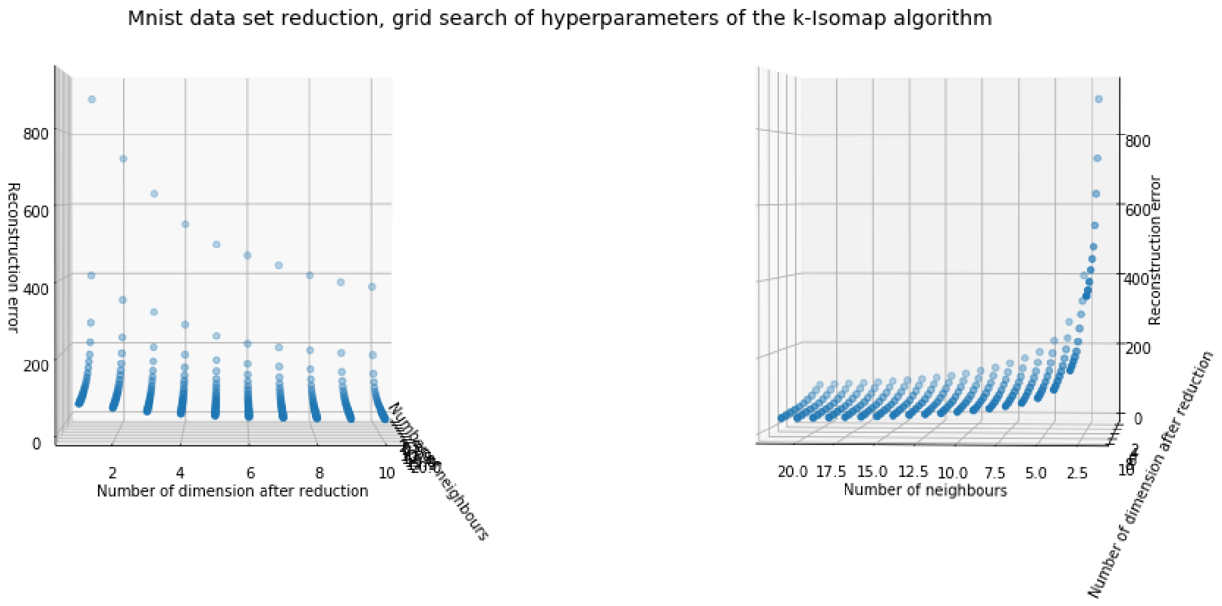

FIGURE 6. Grid search of hyperparameters for MNIST data set and k Isomap algorithm.

It is clearly seeable that *Reconstruction error* decreases with number of dimensions after reduction and with increasing number of neighbours used in k-Isomap. As we see from the right plot *Reconstruction error* reaches the plateau around 15 neighbours. Left plot suggest that we should rather include more dimensions than 10. Simmilar conclusions we might draw when instead of *Reconstruction error* we will use *accuracy score* of the multiclass Logistic regression classifier which is shown at the following Figure 7.

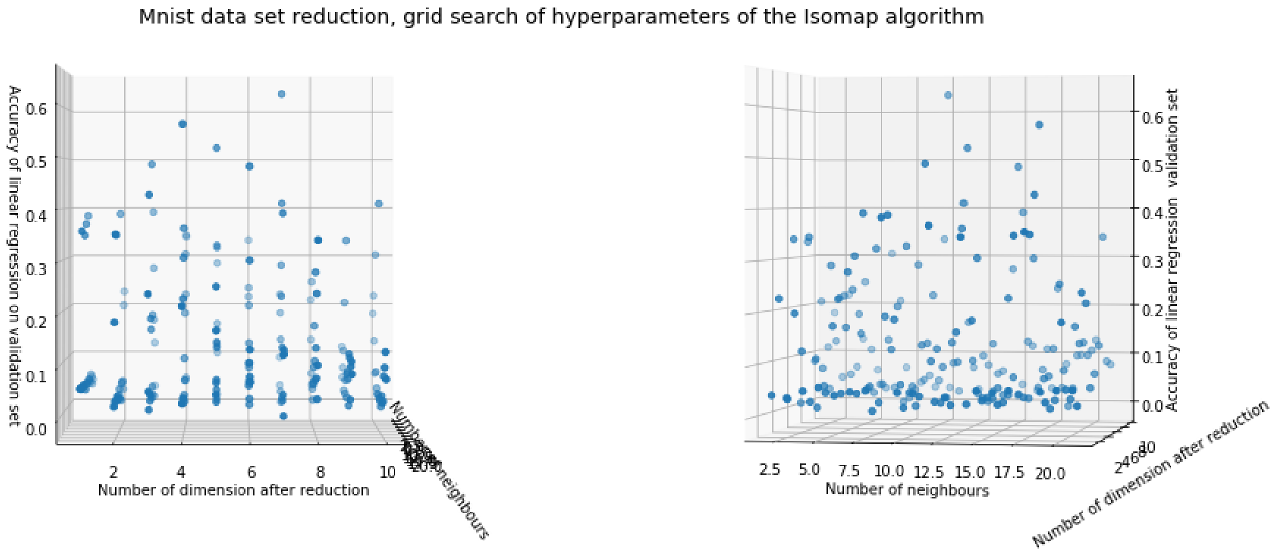

Figure 7. Grid search of hyperparameters for MNIST data set and k Isomap algorithm.. Accuracy score was determined for validation set (1000 observation randomly choosen).

From the above plots from Figure 7 it is hard to draw any conclusion except that for those tested values of k and dimension after reduction we are getting not meaningful values of accuracy, without any monotone relation. The same procedure of grid search was made for Number of dimension in the following ranges: 100-110, 200-210, 300-310, 400-410. There was no improvement in results. There was made also attempt with reduction to two dimension because of course digits are written on 2-D piece of paper. Grid search was conduct for k in range from 2 to 20, also number of examples in the training set was inreased to 2000 and then to 3000 and 5000. There situation was the same. On contrary when we use all features without any prior manipulation, Logistic regression give quite good classification, on test set 92,6% when we use Isomap algorithm I recieved values much smaller approximately a dozen or so percent. So in this case Isomap turned out to be not useful. Nevetheless Isomap algorithm can be very useful tool, for istance authors of this algorithm Tenenbaum, de Silva, Langford in [1] showed interesting example with face images. With this example I would like to end my report.

References

[1] Joshua B. Tenenbaum, Vin de Silva, and John C. Langford. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319, 2000.