Acta Cybernetica **23** (2018) 1017–1038.

Sector Based Linear Regression, a New Robust Method for the Multiple Linear Regression

Gábor Nagy[a]

Abstract

This paper describes a new robust multiple linear regression method, which based on the segmentation of the N dimensional space to N+1 sector. An N dimensional regression plane is located so that the half (or other) part of the points are under this plane in each sector. This article also presents a simple algorithm to calculate the parameters of this regression plane. This algorithm is scalable well by the dimension and the count of the points, and capable to calculation with other (not 0.5) quantiles. This paper also contains some studies about the described method, which analyze the result with different datasets and compares to the linear least squares regression.

Sector Based Linear Regression (SBLR) is the multidimensional generalization of the mathematical background of a point cloud processing algorithm called Fitting Disc method, which has been already used in practice to process LiDAR data. A robust regression method can be used also in many other fields.

Keywords: linear regression, robust regression, quantile regression

1 Introduction

The linear regression is an important component in a lot of calculation in the science and the engineering practice. This tool makes a relationship between one or more independent and one dependent variables by a linear function according to a given dataset.

The most popular method of the linear regression uses the least squares approach for fitting a line (or a plane in higher dimensions) to the given dataset. The outlier points makes remarkable impact in the result of the least squares based regression method.

There are some robust method of the linear regression [18, 21, 17, 22], for example, the Random Sample Consensus (RANSAC) method [6, 4, 7] and the Theil-Sen estimator [19, 23].The complexity of the RANSAC method is increased

[a]Óbuda University, Alba Regia Technical Faculty, Institute of Geoinformatics, E-mail: nagy.gabor@amk.uni-obuda.hu

DOI: 10.14232/actacyb.23.4.2018.3

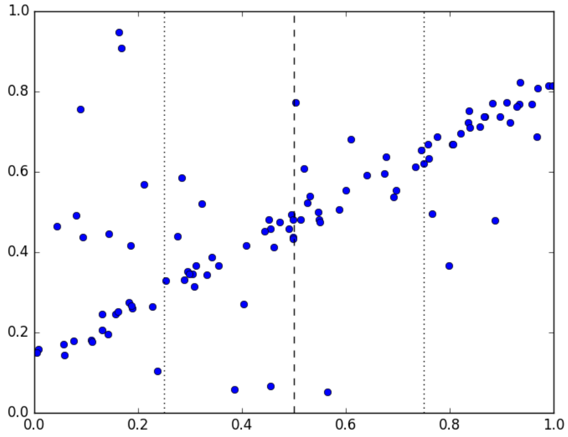

Figure 1: The sectors in case of $N = 1$. (The $N = 1$ is the number of the independent values, the total dimension of the space is $N + 1 = 2$, because the dependent value increases the dimension.) The area is divided to two parts (by the dashed line). The centres of this sectors are displayed by dotted lines.

highly with the dimension in the multiple linear regression, because $\binom{N}{M}$ different planes can be fitted to M given points in an N dimensional space. Both of these methods are not suitable for using with different quantiles.

This article describes the Sector Based Linear Regression (SBLR), a new robust method for the multiple linear regression. The SBLR method runs $O\left(MN^3\right)$ time, where M the number of the points of the dataset, and N is the number of the independent variables. The dimension of the space will be $N + 1$ with the one dependent variable. The SBLR can be used with different quantiles, for example a regression line over the 10 percent ($q = 0.1$) of the points, as other quantile regression methods [12, 23, 1].

2 Principles of the method

In the simple linear regression (one independent variable and one dependent variable, $N = 1$), the regression line has two parameters, for example the a and the b in the $y = ax + b$ equation. The plane can be divided into two parts (in the following: sectors) by a line parallel to the y axis (Figure 1.). A regression line are searched where the half ($q = 0.5$) or the other portion of the points are under the line in both sectors (Figure 2.).

In case of the regression planes (two independent variables and one dependent variable, $N = 2$), the plane of the two independent variable can be divided to three 120 degrees angles as sectors (see Figure 3.). The division can be performed

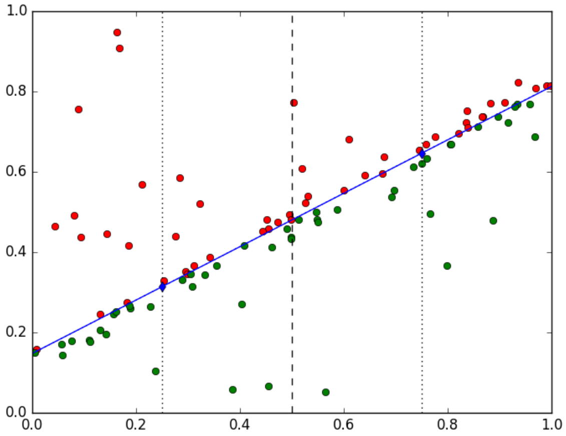

Figure 2: The principle of the SBLR method in case of $N = 1$ (one independent and one dependent values). The half of the points (displayed by green dots) are under, and the other half (displayed by red dots) are over the regression line in both sectors.

by the azimuth, which can be calculate from the two independent variables. (The `atan2()` function can calculate the azimuth in many programming languages.) The determined portion of the points are under the regression plane in all of these three sectors.

3 Extension to N independent variables

The method can be extended to any independent variables, the number of these variables is denoted by N. The dimension of space will be $N+1$ with the dependent variable.

The division of the sectors can be performed by the distances from the centres of the sectors, the points are classified to the sector, whose centre is the closest to the point. (The coordinates of the point are the independent variables of the regression.) This method is usable in any dimension, if the centres of the sectors are known.

The $N + 1$ centres of the sectors are the vertices of a regular N dimensional hyper-tetrahedron (N-simplex), whose centre is the origin of the N dimensional Cartesian coordinate system. The coordinates of the vertices (denoted $v^N_{i,j}$, where i is the index of the vertex from 0 to $N + 1$, j is the index of the coordinate from 1 to N, and N is the dimension of the space) can be calculated by the following recursive function:

- if $N = 0$, the result is `[[]]` (a list which contains an empty list)

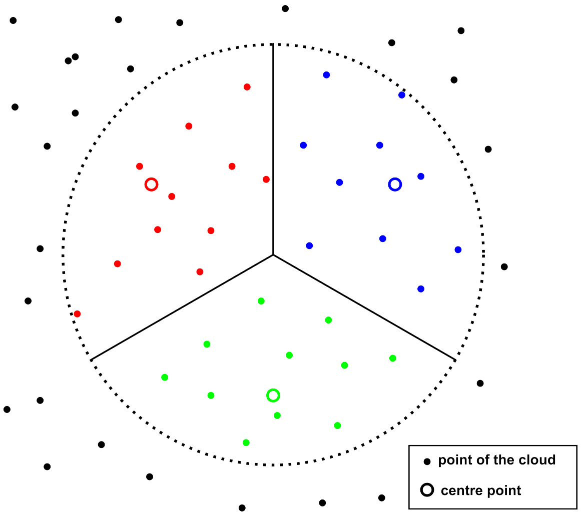

Figure 3: The sectors in case of $N = 2$. This figure represents the plane of the two independent variables, the coordinate of the dependent variable is perpendiculat to this plane. The half (or other quantile) of the points are under the regression plane in all sectors. (The points of the different sectors are displayed by different colors) This case is used in the LiDAR data processing where the points are the points of the LiDAR point cloud, the independent values are the horizontal coordinates of the points and the dependent coordinate is the vertical coordinate.

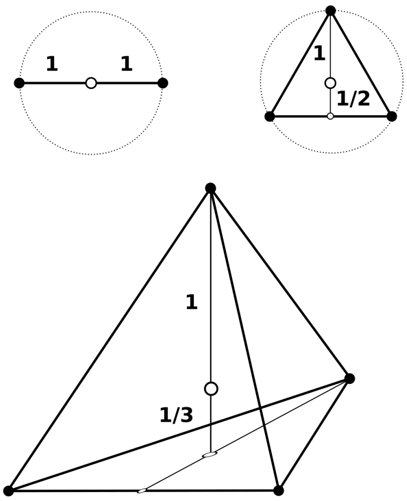

Figure 4: Calculate the coordinates of the vertices of an N-dimensional hyper-tetrahedron. (where $1 \leq N \leq 3$)

- if $N > 0$, the coordinates of the vertices are calculated by this expression:

$$v^N_{i,j} = \begin{cases} v^{N-1}_{i,j}\sqrt{1-\frac{1}{N}} & \text{if } i < N+1 \,\text{and}\, j < N \\ -\frac{1}{N} & \text{if } i < N+1 \,\text{and}\, j = N \\ 0 & \text{if } i = N+1 \,\text{and}\, j < N \\ 1 & \text{if } i = N+1 \,\text{and}\, j = N \end{cases} \tag{1}$$

If $N = 1$ then $v^1_{1,1} = -1$ and $v^1_{2,1} = 1$. If $N = 2$ then $v^2_{1,1} = -\frac{\sqrt{2}}{2}$, $v^2_{1,2} = -\frac{1}{2}$, $v^2_{1,1} = \frac{\sqrt{2}}{2}$, $v^2_{1,2} = -\frac{1}{2}$, $v^2_{1,1} = 0$ and $v^2_{1,2} = 1$. (Figure 4.)

These vertices are at 1 unit distance from the origin of the coordinate system. The sectors centres are $\frac{N}{N+1}$ units from the origin, because this point is the nearest to the centres of the sector. The sectors are indexed from 0 to N. The coordinates of the sector centres are:

$$s^N_{i,j} = \frac{N}{N+1} v^N_{i+1,j} \tag{2}$$

The $N+1$ dimensional regression hyperplane can be specified by $N+1$ value in two ways. One of them is a linear expression:

$$h = l_0 + l_1 x_1 + l_2 x_2 + \cdots + l_j x_j + \cdots + l_N x_N \tag{3}$$

where x_j is the coordinates of the position (the independent values, j indexed from 1 to N), and l_j is the $N+1$ coefficients of the N dimensional hyperplane (j indexed from 0 to N) in a $N+1$ dimensional space.

The other way to define the independent values (the elevations of the plane) in the $N+1$ centres of the sectors (the vertices of the N dimensional regular hyper-tetrahedron), which are denoted c_i, where i is the index of the vertex from 0 to N. The vector of c_i values (denoted $\underline{c}$) can be calculated simply from the vector of l_j values (denoted $\underline{l}$):

$$\underline{c} = \underline{\underline{Q}} \cdot \underline{l} \tag{4}$$

And the $\underline{l}$ can be calculated from the $\underline{c}$, if both sides of (4) are multipled left-hand side by $\underline{\underline{Q}}^{-1}$:

$$\underline{l} = \underline{\underline{Q}}^{-1} \cdot \underline{c} \tag{5}$$

The 4 and the 5 link between heights of sector's centres and coefficients of the linear equation of the hyperplane.

The $\underline{\underline{Q}}$ is an $N+1 \times N+1$ size matrix:

$$\underline{\underline{Q}} = \begin{bmatrix} 1 & s^N_{0,1} & \cdots & s^N_{0,j} & \cdots & s^N_{0,N-1} & s^N_{0,N} \\ 1 & s^N_{1,1} & \cdots & s^N_{1,j} & \cdots & s^N_{1,N-1} & s^N_{1,N} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots \\ 1 & s^N_{i,1} & \cdots & s^N_{i,j} & \cdots & s^N_{i,N-1} & s^N_{i,N} \\ \vdots & \vdots & \ddots & \vdots & \ddots & \vdots & \vdots \\ 1 & s^N_{N-1,1} & \cdots & s^N_{N-1,j} & \cdots & s^N_{N-1,N-1} & s^N_{N-1,N} \\ 1 & s^N_{N,1} & \cdots & s^N_{N,j} & \cdots & s^N_{N,N-1} & s^N_{N,N} \end{bmatrix}$$

The $\underline{\underline{Q}}^{-1}$ is the inverse of $\underline{\underline{Q}}$, and can be calculated in $O\left(N^3\right)$ time. Because the $\underline{\underline{Q}}$ contains only constant values (the coordinates of the sector centres, and 1 values), the program has to calculate the matrix inversion only once. The multiplications (in (4) and (5)) need $O\left(N^2\right)$ time.

4 The calculation method

There is a given dataset, which contains M points. Each point contains N independent values (the coordinates in an N dimensional space) and one dependent value (which is an extra dimension). In the following, $p_{k,j}$ notation is used for the independent variables of the points, where k is the index of the point from 0 to $M-1$, and the j is the index of the coordinates from 1 to N. The $p_{k,0}$ values are the dependent variables.

4.1 Normalization

The first step is the normalization of the coordinates to the $[-1,+1]$ interval by the $x_j = a_j X + b_j$ expression. If one regression will be calculated for all points, calculate the normalized coordinates with $a_j = \frac{2}{\max(x_j)-\min(x_j)}$ and $b_j = -1 - \min(x_j)\, a_j$.

In another case, the regression will be calculated a selected part of the dataset. The points, which are nearest to a specified position (specified an $\underline{r}$ vector, whose elements are r_j) than a defined R radius ($R^2 \leq \sum_{j=1}^{N} (x_j - r_j)^2$). In this case, the $a_j = \frac{1}{R}$ and the $b_j = -r_j$.

In the following steps, the program uses these normalized coordinates.

4.2 Separating into sectors

In the next step, the points will be separated into the sectors, and calculate the initial value of the sector centres (c_i). Each points put the sector whose centre is the closest to the point. I use $p_{i,k,j}$ notation in the separated dataset, where i is the index of the sector (from 0 to N) and k is the number of the point in the sector from 1 to m_i.

All of the sectors have to contain at least one point ($\forall\, i\ m_i > 0$). If any sector does not contain any point ($\exists\, i\ m_i = 0$), the method can not work. This can happen, when the number or the dispersion of the points is not suitable. The probability of the any empty sector, when the dispersion is random (the $P\,(\text{point in the sector}) = \frac{1}{N+1}$ in all of the sectors) is $P\,(\text{any empty sector}) = 1 - \left(1 - \left(\frac{N}{N+1}\right)^M\right)^{N+1}$.

The initial values of the sector's centres (c_i) are the defined quantile (q) of the dependent variables of the sector's points:

$$c_i = \text{quantile}\left(\left[p_{i,1,0}, p_{i,2,0}, \ldots, p_{i,m_i,0}\right], q\right) \tag{6}$$

These values determines the initial regression plane. (See the Figure 5. in case of $N = 1$.)

4.3 The iteration steps

The key element of the method is an iteration step. The program goes from sector to sector and calculates the new values of the sector's centre.

Many $N+1$ dimensional hyperplanes can be calculated, which are fitted to the centres of the other sectors and each points of the sector. The row of the sector's centre in the $\underline{\underline{Q}}$ matrix has to be changed to the coordinates of the point ($[p_{i,k,1}, p_{i,k,2}, \ldots, p_{i,k,N}]$), and the c_i value has to be changed to the $p_{i,k,0}$ (k is the index of the point in the sector) in the $\underline{c}$ vector, and use this modified (5) to calculate the parameters of the hyperplane. After calculating of the hyperplane parameters (l_j), calculate and store the the elevation of this plane in the sector centre by the (3):

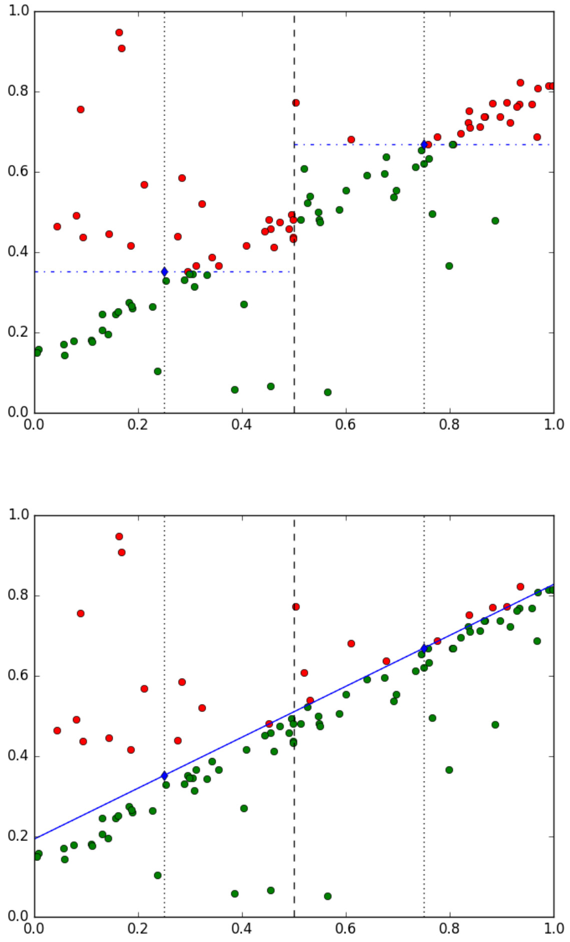

Figure 5: The initial step in case of $N = 1$. The median values are calculated for both sectors. (upper figure) These values (displayed by diamonds) will be the height of the initial regression line in the center of the sectors (dotted line). The points are displayed by red dots over and green dots under the lines (the height of the median, and the initial regression line). The initial regression line is fitted to the centre points. (lower figure)

$$h_k = l_0 + l_1 p_{i,k,1} + l_2 p_{i,k,2} + \cdots + l_j p_{i,k,j} + \cdots + l_N p_{i,k,N} \tag{7}$$

The new value of the sector's centre is the defined quantile (q) of these values:

$$c_i^{new} = \text{quantile}\left(\left[h_1, h_2, \ldots, h_{m_i}\right], q\right)$$

The program continues this process in the sector number $(i+1) \mod N+1$, and check the difference between the new and the old c_i values. If the difference less than a specified value ($\left|c_i^{old} - c_i^{new}\right| < \varepsilon$), a counter is increased one, otherwise the counter set to zero. The iteration loop is repeated while this counter is less than $N+1$. (The first two step in case of $N = 1$ is presented in Figure 6.)

The changes of the heights of the sector's centres typically will be less in the iterations. This ensures convergence.

4.4 Completion

Finally, the parameters of the regression plane are calculated by the (5) from the centres of the sectors. The received parameters are in a normalized coordinate system. (See 4.1)

If only the elevation of the plane is needed in the origin of the normalized coordinate system, the l_0 is this. If the plane equation is needed in the original coordinate system, the $l_i a_i$ expression can be used.

5 Studying the SBLR algorithm

Some simple Python [20, 16, 14] programs were made to test the SBLR algorithm. The `sblr.py` module is a simple implementation of the SBLR method. The test programs use this module.

The test programs use random datasets, which are created by the `random` Python module. This module can generate random numbers with several distribution. In the following studies the test programs use the $y = 3x - 5$ linear base function. The independent values (x) are generated by a uniform random value between 0 and 10 (`random.uniform(0,10)`). The dependent values are calculated by the $y = 3x - 5 + error$ equation. The *error* is various random number with 1 standard deviation and 0 median. A specific part of the points are outlier; the dependent variable of this points is a uniform random value between -7 and 27.

The test programs use different random numbers for the *error* value based on the `random` Python module. The uniform distribution error is a random number between $-\sqrt{3}$ and $\sqrt{3}$ by the `random.uniform(-1,1)*1.7320508075688772` expression. The normal distribution uses `random.normalvariate(1,0)`, the lognormal distribution uses `random.lognormalvariate(1,0)-1` and the exponential distribution uses `random.expovariate(1)-0.6931471805599453`. The minus 1 and minus $0.6931471805599453 \simeq \ln(2)$ need for the 0 median.

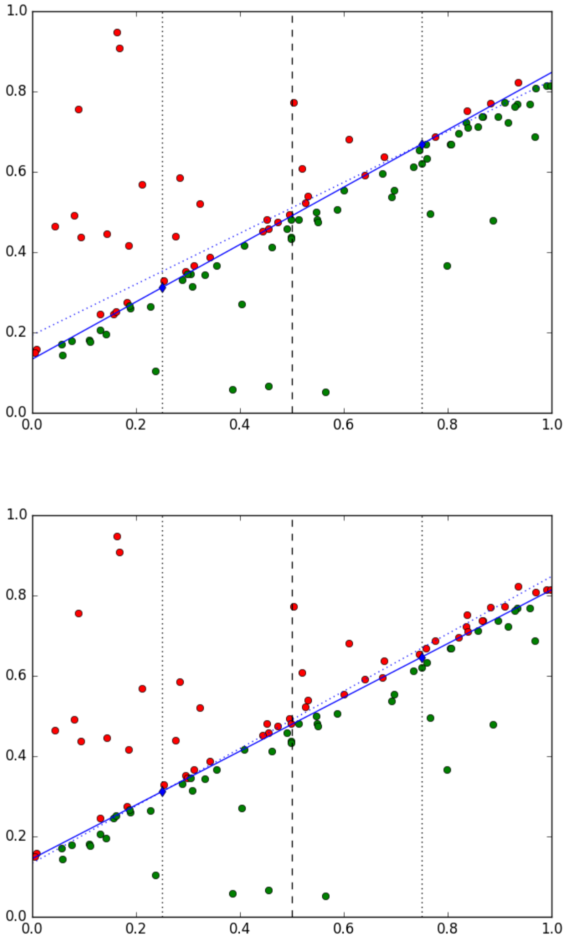

Figure 6: The iteration steps in case of $N = 1$. The new values of the sector's centres are determined so that the half (or other quantile) of the sector's points will be under the line, which is fitted the new centre of this sector and the other sector's centre. The new line is continuous, the line of the last iteration is dotted. The iteration is repeated until the change of the values are less than a limit (denoted ε) in both sectors.

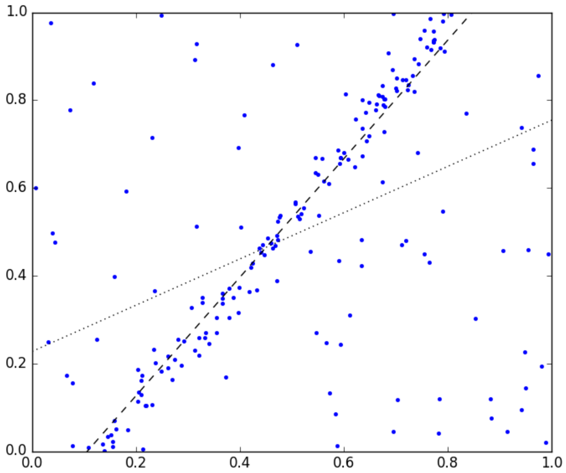

Figure 7: The result of the SBLR method (dashed line) and the least squares linear regression (dotted line) in a dataset with many outlier points.

5.1 Comparsion to the least squares linear regression

The least squares method is the most common regression tool, but the any outlier measurements can indicate significant difference in the result. (Figure 7.) A regression line can be calculated by the least squares method, the sum of squares of the differences between the points and the regression line will be the smallest with this regression line.

A test program generated random datasets with different portion of outlier points (from 0 to 75 percent). The test program generated 5000 datasets in all outlier portion (0%, 1%, 2%, ... 75%) and calculated regression lines in each dataset by the SBLR and the least squares methods. The two regression lines were compared to the original line, and calculate the averages of the distance from this line in the $[0, 10]$ interval. This number was the metrics of the fitting in these studies.

In each outlier portion, the test program stored 5000 fitting value; and another program calculated the averages of these values in both methods in each outlier portion. The Figure 8. shows the result of these studies with different number of points.

Another studies compare the average distance with different count of the points and different distribution of errors. The point numbers were the elements from an arithmetic sequence from 50 to 2000 with step of 50. The studies made with different errors (normal, uniform, lognormal and exponential) and different percentage of outliers (0 and 2). The program generates 5000 random datasets in each case. The result of these studies are seen in the Figure 9. and Figure 10.

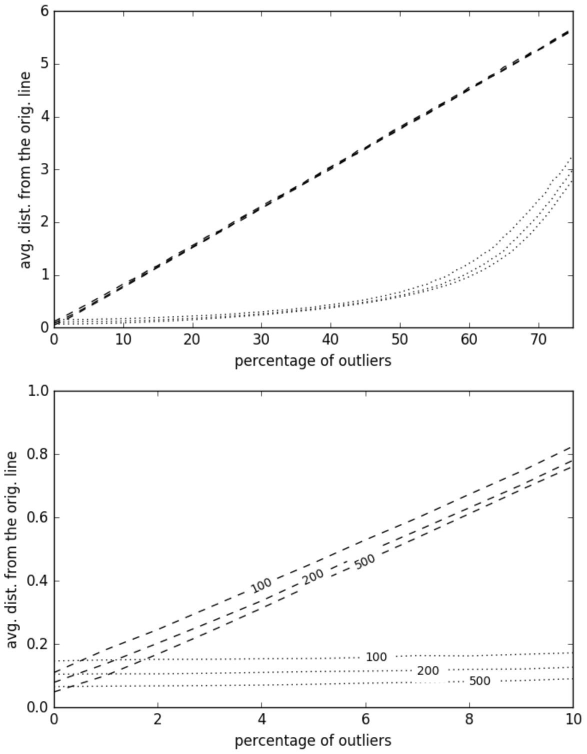

Figure 8: The average distance from the original line and the percentage of outliers with different number of points (100, 200 and 500) by least squares (dashed line) and SBLR method (dotted line). The range between 0 and 10 percent is zoom in on the lower figure.

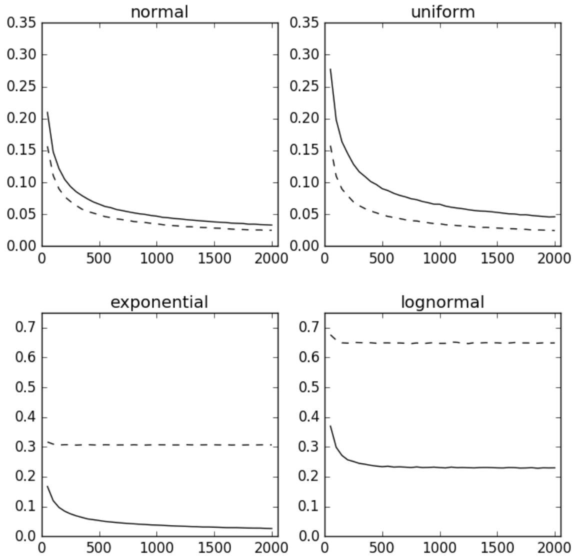

Figure 9: The average distance from the original line (vertical axes) with different number of points (horizontal axes) by least squares (dashed line) and SBLR (continuous line) methods. The datasets do not contain outlier points.

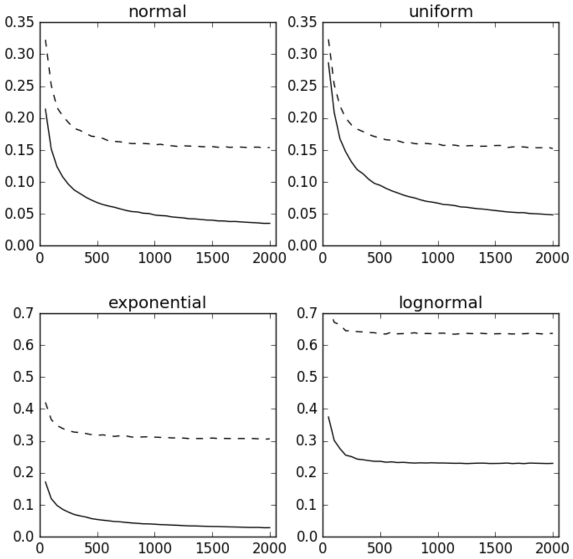

Figure 10: The average distance from the original line (vertical axes) with different number of points (horizontal axes) by least squares (dashed line) and SBLR (continuous line) methods. The datasets contain 2 percent of outlier points.

In the asymmetric error distributions (exponential and lognormal), the SBLR created better result than the least squares method without outlier points. If the dataset has 2 percent of outlier points, the SBLR made better result in all of the examined error distributions.

5.2 Examining the iteration steps

The computation time of the SBLR method grows linearly with the count of the points (denoted M in this article). This computation time may be increased if the iteration steps of the method grows with M.

Some test programs were created to study the correlation between the number of the points and the iteration steps. The M was different values according to a geometric sequence. The initial value of this sequence is 100 and the common ratio is $\sqrt[4]{2} \simeq 1.1892$ (the result was rounded). The largest datasets had 102400 points.

The test programs created 1000 different random datasets with each M and each distributions, calculated the regression lines and store the number of the iteration steps with $\varepsilon = 10^{-5}$. Another program analyzed the stored data and calculate the means of the iteration steps. (Table 1.)

The number of the iteration step does not grow, moreover a little decrease, when the M increased. The computation time of the SBLR method is $O\left(MN^3\right)$. The result is same when the 30 percent of the points are outlier. Other parameters were not changed. (Table 2.)

5.3 The limits and possible errors of the SBLR method

The SBLR method can calculate only linear regression, and only one regression in a dataset. The Ref. [10] presents a method, which can be found more linear regression from one dataset.

The method can work if all sectors have at least one point. The good result needs more points in all sectors to eliminate the impact of the outliers.

An outlier point may result wrong sector layout. The normalization step (see in 4.1) create a wrong result, where the outlier point is in a sector, and all the rest in the other sector. This problem can be avoided, if the sector centre is determined as the median of the values. In the practical applications (in the author's practice), this mistake has not occurred, because the points are selected from a bigger dataset (see in 6.1), therefore it did not have far points in the independent coordinates.

6 Application possibilities

The SBLR has a lot of application possibilities. This method may be used in projects, where need a robust linear regression. The SBLR may be useful, when quantile regression are needed in any dimension spaces.

Table 1: The average number of iteration steps with different distribution of errors and different number of points (without outliers)

Count of points	distribution of the errors			
	normal	lognormal	exponential	uniform
100	9.369	9.561	9.453	9.473
119	9.305	9.394	9.273	9.281
141	9.148	9.299	9.293	9.237
168	9.252	9.255	9.176	9.265
200	9.177	9.374	9.234	9.305
238	9.010	9.199	8.909	9.087
283	8.893	9.098	8.879	9.004
336	8.891	9.011	8.869	9.022
400	8.718	9.030	8.764	8.829
476	8.526	8.887	8.624	8.842
566	8.572	8.833	8.566	8.772
673	8.351	8.773	8.459	8.609
800	8.394	8.725	8.346	8.564
951	8.221	8.578	8.319	8.485
1131	8.164	8.477	8.239	8.280
1345	8.056	8.391	8.145	8.272
1600	8.028	8.437	8.149	8.188
1903	7.921	8.317	8.008	8.023
2263	7.842	8.206	7.949	7.982
2691	7.748	8.230	7.863	7.855
3200	7.730	8.152	7.868	7.898
3805	7.600	8.071	7.763	7.634
4525	7.474	8.003	7.627	7.593
5382	7.407	7.966	7.615	7.632
6400	7.371	7.888	7.560	7.511
7611	7.255	7.868	7.544	7.382
9051	7.059	7.768	7.453	7.301
10763	7.070	7.811	7.363	7.170
12800	6.997	7.730	7.307	7.127
15222	6.952	7.654	7.175	7.034
18102	6.833	7.521	7.174	7.008
21527	6.720	7.501	7.152	6.921
25600	6.629	7.468	7.067	6.828
30444	6.611	7.441	6.973	6.704
36204	6.553	7.329	6.926	6.603
43054	6.433	7.311	6.929	6.618
51200	6.294	7.234	6.815	6.481
60887	6.215	7.146	6.827	6.384
72408	6.201	7.079	6.752	6.275
86108	6.093	7.046	6.662	6.172
102400	6.002	6.990	6.685	6.128

Table 2: The average number of iteration steps with different distribution of errors and different number of points (with 30 percent outliers)

Count of points	distribution of the errors			
	normal	lognormal	exponential	uniform
100	9.738	9.776	9.740	9.729
119	9.678	9.601	9.583	9.614
141	9.484	9.502	9.472	9.492
168	9.570	9.557	9.439	9.611
200	9.409	9.498	9.338	9.482
238	9.275	9.349	9.438	9.453
283	9.264	9.277	9.195	9.305
336	9.157	9.146	9.184	9.251
400	9.142	9.068	9.279	9.262
476	9.118	9.121	9.001	9.203
566	8.919	8.926	9.012	9.096
673	8.908	8.907	8.865	9.005
800	8.841	8.755	8.932	9.019
951	8.800	8.781	8.815	8.804
1131	8.744	8.786	8.848	8.787
1345	8.617	8.725	8.794	8.816
1600	8.732	8.547	8.827	8.736
1903	8.545	8.552	8.690	8.570
2263	8.494	8.517	8.670	8.481
2691	8.524	8.525	8.648	8.417
3200	8.395	8.527	8.640	8.328
3805	8.290	8.357	8.572	8.265
4525	8.221	8.352	8.439	8.179
5382	8.210	8.355	8.429	8.236
6400	8.116	8.359	8.334	8.065
7611	8.016	8.191	8.327	7.996
9051	7.988	8.209	8.253	7.844
10763	7.959	8.147	8.114	7.873
12800	7.817	8.087	8.141	7.705
15222	7.887	8.051	8.134	7.715
18102	7.804	8.005	8.057	7.586
21527	7.681	7.909	8.056	7.523
25600	7.727	7.925	7.938	7.451
30444	7.680	7.912	7.948	7.373
36204	7.506	7.843	7.880	7.291
43054	7.509	7.786	7.851	7.218
51200	7.354	7.733	7.800	7.107
60887	7.336	7.746	7.707	7.076
72408	7.320	7.646	7.712	6.973
86108	7.257	7.631	7.652	6.918
102400	7.144	7.549	7.622	6.894

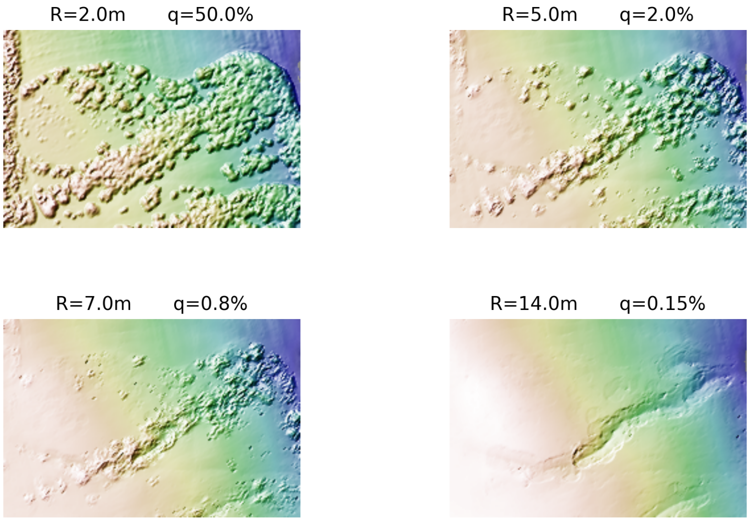

Figure 11: The application of the SBLR method in LiDAR data processing with different R and q values.

6.1 LiDAR data processing

SBLR can be used in any application, where a robust linear regression method is required. If the distribution of the measurement error is skewed, the method can use a different q value than 0.5.

This method has been used for processing the LiDAR point clouds. In this case ($N = 2$), the two independent value are the horizontal coordinates, the dependent variable is the elevation, and the measurements are the points of the LiDAR point cloud. (See the Figure 3.) The classical X, Y, and Z coordinates of the points are denoted x_1, x_2 and h in this case in the equation of a fitting plane, and $p_{k,1}$, $p_{k,2}$ and $p_{k,0}$ in the point of the cloud.

The regression plane is fitted to a part of the total LiDAR cloud, which is cut by a circle shape with R radius. The regression plane fits to this part of the cloud, because this method is called "Fitting Disc" method. [15] This principle may be used in other cases, where the connection is not linear between the independent and the dependent values: select the points, which are nearest than a radius (R) from an examined position, and fit a linear, N dimensional plane to this part, which is approximately linear. (See in the Figure 12., in a two-dimensional illustration.) The Fitting Disc method is a local application of the Sector Based Linear Regression.

Digital Elevation Models can be created, if the SBLR based Fitting Disc method is applied in each point of the DEM grid. The result depends from R and q values, for example the Figure 11. In the forest areas the appropriate result needs very low

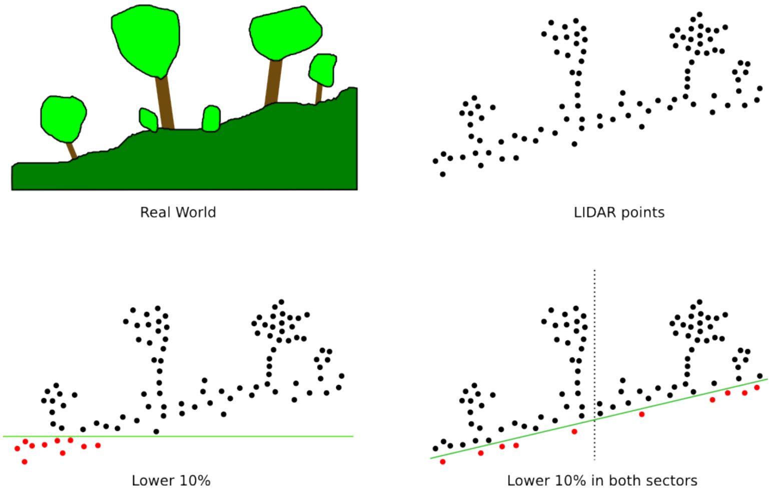

Figure 12: The LiDAR data processing with SBLR in a two-dimensional illustration. The ground surface is evaluated by $q = 0.1$ parameter, because the majority of the points are in the trees and bushes, over the ground surface.

q values; and the very low q values need long R radius, because some points must be under the plane. If the intention is at least on average n points under the plane in each sectors, the radius is $R \geqq \sqrt{\frac{3n}{qd\pi}}$, where d is the density of the LiDAR point cloud in ${}^{\text{points}}/_{\text{m}^2}$.

The SBLR based Fitting Disc method can be applied to recognize planes in a point cloud, for example the roofs of the buildings. In these cases the plane of the detected object (for example a roof) can be calculated by SBLR from a segment of the point cloud.

6.2 Other possibilities

A linear regression plane can be fitted to the data of the pixels of a picture near a position (like the LiDAR data processing) and calculated a filtered color by this regression plane. This filtering method is same as the Two-Dimensional Median Filtering Algorithm [8].

The SBLR can be used for any data processing task, where a linear regression is needed in an N-dimensional space. This method can be used well with a lot of outlier data or a random error with asymmetric distribution.

The SBLR is a linear case of the quantile regression [13, 12, 11]. The quantile regression is used in different disciplines, for example ecology [3] or economy [2, 5].

A robust linear regression method can provide a robust method to determine the parameters an affine transformation by control points. This calculation needs two independent linear regression for the two coordinates (in case of the two-dimensional affine transformation), because each equations of the affine transformation are a linear regression, where the independent variables are the coordinates of the reference system one, and the dependent variable is a coordinate of the reference system two.

7 Conclusions and future work

The Sector Based Linear Regression is a robust method for fitting an N dimensional hyperplane to a dataset which has N independent and 1 dependent variables. The studies of this article focused to the simple $N = 1$ case, and the practical application (LiDAR data processing) uses the $N = 2$ case, but the method can be applied in any dimension. This method provides quantile regression, it is useful in some cases (for example the LiDAR data processing, when the majority of the points are over the ground surface).

The processing time of the SBLR method is increased only linear with the size of the input data (the number of the points, denoted by M in this article). This advantage makes it ideal for big data processing applications.

This article presents the principle of the method, an algorithm for the SBLR, and some studies and application possibilities of the method. A simple implementation of the SBLR method has been made. The source code of this Python 3 module is attached to this article. In the future, i would like to implement the method in other programming languages, and improve the efficiency of the program.

The principle of the Sector Based Linear Regression can be adapted to non-linear regressions. The area must be divided more sectors in these cases, because the non-linear curves need more parameters.

8 Acknowledgement

This research was supported by the project number TÁMOP-4.2.2.B-15/1/KONV-2015-0010, titled “Tudományos képzés műhelyeinek fejlesztése az Alba Regia Műszaki Karon” (in English: Developing workshops of the scientific education in the Alba Regia Technical Faculty).

The Figure 1., Figure 2., Figure 5., Figure 6, Figure 7., Figure 8., Figure 9., Figure 10. and Figure 11. were created by Matplotlib [9].

9 Additional files

This article contains two animated GIF files. The `sblr.gif` shows the SBLR method during operation in case of $N = 1$. The `fitdisc.gif` presents the test area of the Figure 11. in many other cases of R and q parameters of the Fitting Disc method.

The implemented SBLR algorithm is already attached `sblr.py` Python 3 module. This module provides the SBLR calculations in any Python 3 program.

References

[1] Bertsimas, Dimitris and Mazumder, Rahul. Least quantile regression via modern optimization. *The Annals of Statistics*, pages 2494–2525, 2014.

[2] Buchinsky, Moshe. Changes in the us wage structure 1963-1987: Application of quantile regression. *Econometrica: Journal of the Econometric Society*, pages 405–458, 1994.

[3] Cade, Brian S and Noon, Barry R. A gentle introduction to quantile regression for ecologists. *Frontiers in Ecology and the Environment*, 1(8):412–420, 2003.

[4] Choi, Sunglok, Kim, Taemin, and Yu, Wonpil. Performance evaluation of ransac family. *Journal of Computer Vision*, 24(3):271–300, 1997.

[5] Coad, Alex and Rao, Rekha. Innovation and firm growth in high-tech sectors: A quantile regression approach. *Research policy*, 37(4):633–648, 2008.

[6] Fischler, Martin A and Bolles, Robert C. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[7] Hast, Anders, Nysjö, Johan, and Marchetti, Andrea. Optimal ransac-towards a repeatable algorithm for finding the optimal set. 2013.

[8] Huang, T, Yang, G, and Tang, G. A fast two-dimensional median filtering algorithm. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 27(1):13–18, 1979.

[9] Hunter, J. D. Matplotlib: A 2d graphics environment. *Computing In Science & Engineering*, 9(3):90–95, 2007.

[10] Isack, Hossam and Boykov, Yuri. Energy-based geometric multi-model fitting. *International journal of computer vision*, 97(2):123–147, 2012.

[11] Jurečková, Jana. Robust quantile regression. *Encyclopedia of Environmetrics*, 2006.

[12] Koenker, Roger. *Quantile regression.* Number 38. Cambridge university press, 2005.

[13] Koenker, Roger and Bassett Jr, Gilbert. Regression quantiles. *Econometrica: journal of the Econometric Society*, pages 33–50, 1978.

[14] Millman, K Jarrod and Aivazis, Michael. Python for scientists and engineers. *Computing in Science & Engineering*, 13(2):9–12, 2011.

[15] Nagy, Gábor, Tamás, Jancsó, and Chen, Chongcheng. The fitting disc method, a new robust algorithm of the point cloud processing. *ACTA POLYTECHNICA HUNGARICA*, 14(6):59–73, 2017.

[16] Oliphant, Travis E. Python for scientific computing. *Computing in Science & Engineering*, 9(3), 2007.

[17] Rousseeuw, Peter J and Hubert, Mia. Regression depth. *Journal of the American Statistical Association*, 94(446):388–402, 1999.

[18] Rousseeuw, Peter J and Leroy, Annick M. *Robust regression and outlier detection*, volume 589. John Wiley & Sons, 2005.

[19] Theil, Henri. A rank-invariant method of linear and polynomial regression analysis. In *Henri Theils Contributions to Economics and Econometrics*, pages 345–381. Springer, 1992.

[20] Van Rossum, Guido et al. Python Programming Language. In *USENIX Annual Technical Conference*, volume 41, 2007.

[21] Wilcox, Rand R. *Introduction to robust estimation and hypothesis testing.* Academic press, 2011.

[22] Wilcox, Rand R and Keselman, HJ. Modern regression methods that can substantially increase power and provide a more accurate understanding of associations. *European journal of personality*, 26(3):165–174, 2012.

[23] Zhou, Weihua and Serfling, Robert. Multivariate spatial u-quantiles: A bahadur–kiefer representation, a theil–sen estimator for multiple regression, and a robust dispersion estimator. *Journal of Statistical Planning and Inference*, 138(6):1660–1678, 2008.

Received 11th November 2017