Geometric Approach to Stepwise Regression

Barbara Żogała-Siudem[1] and Szymon Jaroszewicz[2,3]

[1] Systems Research Institute, Polish Academy of Sciences,
ul. Newelska 6, 01-447 Warsaw, Poland
[2] Institute of Computer Science, Polish Academy of Sciences,
ul. Jana Kazimierza 5, 01-248 Warsaw, Poland
[3] National Institute of Telecommunications,
ul. Szachowa 1, 04-894 Warsaw, Poland

Abstract. Stepwise feature selection is one of the most popular variable selection techniques for linear models. The procedure, however, is computationally demanding, especially when the number of potential variables is large. In our previous work we proposed a way to speed up stepwise algorithm on large data, based on multidimensional indices and a bound based on correlations between variables. This paper presents an alternative proof of the bound and shows that it cannot be improved.

1 Introduction

Nowadays, many sophisticated methods for data analysis are available. However, a very important issue is not only the modeling itself, but also finding relevant variables to include in the model. Unfortunately, there are currently no methods to assist the researcher (except his/her own intuition) in finding external sources of relevant data such as public datasets available on the web.

Potentially, an answer to this problem could be Linked Open Data (LOD): a project to make statistical data collected by various organizations, government statistical offices, etc. publicly available on the internet in a way which is well suited for automated access. The movement has recently gained momentum and huge amounts of data became available online from sources such as Eurostat [1], United Nations, International Monetary Fund, etc. The current state of Linked Open Data can be seen in the diagram [2] which shows data sources and links between them. More information on Linked Open Data can be found e.g. in [3–6].

We believe however, that in its current form Linked Open Data is not suitable for statistical practice. Linking new datasets is based on purely syntactic criteria, which can easily result in huge amount of unrelated data being downloaded to researcher's computer. Building models on such data would then be extremely time consuming and prone to overfitting.

A solution, in our opinion, is a linking procedure based on statistical, not syntactic properties. One example of such a solution (and at the same time the

most relevant previous work) is Google Correlate [7–9], which given a *query dataset* finds the most correlated Google query. The service has several limitations: it is restricted to finding correlated Google queries and does not include other publicly available datasets. Moreover it is only able to find single most correlated variables, while in practice we are interested in building complete statistical models.

In [10] we presented a method to build fast stepwise linear regression models by using a multidimensional indexes to search for relevant variables. As the multidimensional index we used FLANN (Fast Library for Approximate Nearest Neighbors) [11, 12]. Since the indices only allow for finding single most correlated variables, the stepwise procedure had to be rewritten using only this operation. The method is based on forward stepwise regression (see e.g. [13]) but during each step a spatial index is used to search for candidate variables; only those candidates are then used for classical stepwise selection.

The paper is organized as follows. In Section 2 we give a brief summary of [10], introduce the necessary notation, followed by a short introduction to multidimensional indexing. Forward stepwise feature selection is explained in Section 2.2 together with the theorem guaranteeing its correctness. Later, in Section 3 a geometric proof of Theorem 1 is described, and a theorem is given proving that the bound cannot be improved. Finally, in Section 4 we conclude the paper.

2 Fast stepwise regression

The main idea of our approach to speed up the stepwise regression procedure is based on Theorem 1, which was proved and discussed shortly in [10]. To present this theorem let us start with introducing some notation and explaining the stepwise regression procedure.

2.1 Notation

Lowercase letters will denote n-dimensional vectors. In particular, $y \in \mathbb{R}^n$ will be the response variable of a linear model, $r \in \mathbb{R}^n$ a residual vector of the currently considered model, and $x \in \mathbb{R}^n$ a predictor variable. The set of all possible predictors will be denoted as $X = \{x_1, \ldots, x_p\}$. Subsets of X will be denoted as X_I, where $I \subseteq \{1, \ldots, p\}$ is the set of indices of variables. So, if $I = \{l_1, \ldots, l_k\}$, then $X_I = \{x_{l_1}, \ldots, x_{l_k}\}$.

In the paper we assume that each vector $x_i \in X$ as well as the response y are normalized i.e. they have zero mean ($\bar{x}_i = 0$) and l_2 norm equal to 1 ($\|x_i\| = 1$).

Let $I = \{l_1, \ldots, l_k\}$. The projection of a vector y onto the space spanned by $X_I = \{x_{l_1}, \ldots, x_{l_n}\}$ will be denoted as $Proj_{X_I} y$ and by $y \sim x_{l_1} + \ldots + x_{l_k}$ or $y \sim X_I$ we will denote a linear model with y as response and $x_{l_1}, \ldots, x_{l_k}$ as predictors.

For brevity, correlations of specific vectors x_i and x_j will be written as $c_{i,j} = \mathrm{cor}(x_i, x_j)$ and correlation of the variable x_i and current residual vector r as $c_{res,i} = \mathrm{cor}(r, x_i)$.

In Section 3 volumes of parallelotopes spanned by a set of vectors will be denoted as $\mu_k(\cdot)$. For example, the volume of a k-dimensional parallelotope spanned by $\{x_{l_1}, \ldots, x_{l_k}\}$ is $\mu_k(x_{l_1}, \ldots, x_{l_k})$.

2.2 Stepwise regression

The idea of stepwise regression was introduced in 1960 by Efroymson [14]. Here, by stepwise procedure we mean *forward stepwise selection* (see e.g. [13]). The algorithm works as follows. First we start with an empty model ($y \sim 1$) and find a variable (say x_{l_1}) which gives the lowest residual sum of squares (RSS) when added to the model. The variable is then included in the model which becomes: $y \sim x_{l_1}$. Then we check all two-variable models which include the variable x_{l_1}, that is $y \sim x_{l_1} + x_i$, for all $x_i \in X \setminus \{x_{l_1}\}$, select a variable x_{l_2} for which the RSS was lowest and add it to the model. We continue this procedure until the model no longer improves according to an appropriate criterion (such as AIC [15] or BIC [16]) or the maximum number of variables allowed in the model is reached. The algorithm is presented in Table 1.

Algorithm: Stepwise

1) $r := y$
   $I := \emptyset$
2) **For** $k = 1, \ldots, k_{max}$:
   1. **For** each $i \in \{1, \ldots, p\} \setminus I$:
      compute the residual of the model obtained
      by adding x_i to the current model: $r_i = y - Proj_{X_{I \cup \{i\}}} y$
   2. Find $l_k = \arg\min_{i \in \{1,\ldots,p\} \setminus I} {r_i}^T r_i$,
   3. **If** the model: $y \sim X_{I \cup \{l_k\}}$ is *better* than $y \sim X_I$:
      Add l_k to I: $I := I \cup \{l_k\}$ and **goto** 2)
      else break.

Fig. 1. The stepwise regression algorithm

The main problem with the stepwise algorithm is that in each iteration it requires building as many models as there are possible predictors (although some work can be shared between all models in some circumstances) and, as a result, becomes very inefficient for datasets with a large number of variables, such as the ones that may be obtained using Linked Open Data.

2.3 Multidimensional indices and correlations

To speed up the stepwise procedure described in Section 2.2 we proposed [10] an algorithm which limits the number of models built in each iteration, by using multidimensional indexing. We will now summarize the results of that paper.

A multidimensional index can be used to store a large number of points from an n-dimensional Euclidean space. Afterwards, we can use the index to quickly

answer two types of queries: (1) *nearest neighbor queries*, where given a query vector, find k nearest vectors in the index, and (2) *range queries*, where given a query vector and a radius, find all points within the given radius from the query. As a multidimensional index we used the FLANN library [11] which is very fast but gives approximate results and Ball Trees (see e.g. [17]) which are much slower, but give exact results.

A key observation is that, for appropriately normalized vectors, searching for a nearest neighbor corresponds to looking for the most correlated vector. Let x_i, x_j be vectors with zero mean and l_2 norm equal to 1 (i.e. $\bar{x}_i = \bar{x}_j = 0$ and $\|x_i\| = \|x_j\| = 1$), then

$$\|x_i - x_j\| = \sqrt{2 - 2\langle x_i, x_j\rangle} = \sqrt{2 - 2\mathrm{cor}(x_i, x_j)}.$$

Due to the above, in order to search for a vector most correlated with a given query vector x we need to normalize it

$$x' = \frac{x - \bar{x}}{\|x - \bar{x}\|}, \tag{1}$$

and perform a nearest neighbor search for both x' and $-x'$.

2.4 Fast stepwise regression

The main result of the paper [10] was to show how to quickly build a stepwise model on data with a large number of indexed variables. Here we restate this result briefly, starting with the following theorem.

Theorem 1. *Assume that the variables $x_{l_1}, \ldots, x_{l_{k-1}}$ currently in the model are orthogonal, let $r = y - Proj_{\{x_{l_1},\ldots,x_{l_{k-1}}\}}y$ denote the residual vector of the current model and take two variables x_{l_k}, $x_{l'_k}$. Then*

$$\|y - Proj_{\{x_{l_1},\ldots,x_{l_{k-1}},x_{l'_k}\}}y\| \leqslant \|y - Proj_{\{x_{l_1},\ldots,x_{l_{k-1}},x_{l_k}\}}y\| \tag{2}$$

implies

$$\max\{|c_{l_1,l'_k}|, \ldots, |c_{l_{k-1},l'_k}|, |c_{res,l'_k}|\} \geqslant \frac{|c_{res,l_k}|}{\sqrt{1 - \sum_{i=1}^{k-1} c^2_{l_i,l_k} + (k-1)c^2_{res,l_k}}}. \tag{3}$$

Suppose we considered x_{l_k} as a candidate for the model and computed the residual sum of squares for it. Theorem 1 states that if any variable is better than x_{l_k}, then it must be correlated to a sufficient degree either with the current residual vector or one of the predictors already in the model. This condition can easily be translated into a series of range queries to the index. The query points are $\pm r$, where r is the current residual and $\pm x_{l_i}$, where x_{l_i} are variables currently in the model. The query radius is given by the right hand side of (3).

The algorithm for fast stepwise regression is given in Figure 2. Lines 2 and 3.2 use a nearest neighbor query to the index, and lines 3.5 and 3.6 use range queries. The speed up comes from using the stepwise procedure only on the candidate set C which can be efficiently obtained using the multidimensional index.

Algorithm: Fast stepwise

1) $r := y$
   $I := \emptyset$
2) Find a variable $x_{l_1} \in X$ most correlated with r
   and add its index l_1 to the active index set $I := I \cup \{l_1\}$
3) **For** $k = 1, \ldots, k_{max}$:
   1. Compute the new residual vector $r = y - Proj_{X_I} y$
   2. Find a candidate variable index $u_k \in \{1, \ldots, p\} \setminus I$
      such that x_{u_k} is most correlated with r
   3. Initialize the candidate index set $C := \{u_k\}$
   4. $\eta := \frac{|\mathrm{cor}(r, x_{u_k})|}{\sqrt{1 - \sum_{l_i \in I} \mathrm{cor}(x_{l_i}, x_{u_k})^2 + |I| \mathrm{cor}(r, x_{u_k})^2}}$
   5. $C := C \cup \{i \in \{1, \ldots, p\} \setminus I : \|x_i - r\|^2 \leqslant 2 - 2\eta\}$
   6. **For** $j = 1, \ldots, k-1$:
      $C := C \cup \{i \in \{1, \ldots, p\} \setminus I : \|x_i - x_{l_j}\|^2 \leqslant 2 - 2\eta\}$
   7. Find the best variable x_{l_k} in X_C using an iteration
      of stepwise procedure
   8. Add l_k to the current active index set: $I := I \cup \{l_k\}$

Fig. 2. The fast stepwise regression algorithm based on a multidimensional index.

3 Geometric approach

In [10] a proof of Theorem 3 was given, which was based on linear algebra techniques. In this paper we would like to show a different approach concentrating on a geometric structure of variables and correlations between them. The geometric proof is based on the following lemma.

Lemma 1. *If adding the variable $x_{l'_k}$ to the model decreases the residual sum of squares more than adding x_{l_k}, i.e.*

$$\|y - Proj_{\{x_{l_1}, \ldots, x_{l'_k}\}} y\| \leqslant \|y - Proj_{\{x_{l_1}, \ldots, x_{l_k}\}} y\|, \tag{4}$$

then the following inequality is satisfied

$$\frac{c^2_{res, l_k}}{1 - \sum_{i=1}^{k-1} c^2_{l_i, l_k}} \leqslant \frac{c^2_{res, l'_k}}{1 - \sum_{i=1}^{k-1} c^2_{l_i, l'_k}}.$$

To prove Lemma 1 let us first state two facts:

FACT 31
For any $x_1, \ldots, x_n \in \mathbb{R}^n$ and matrix $X = [x_1|\ldots|x_n]$ we can find a rotation matrix R such that RX is upper triangular.

FACT 32
The volume of a parallelotope spanned by vectors $x_1, \ldots, x_n \in \mathbb{R}^n$ is equal to

$$\mu_n(x_1, \ldots, x_n) = \det([x_1|\ldots|x_n]).$$

Let us now back to the proof of lemma 1.

Proof (Proof of lemma 1). First let us notice that

$$\|y - Proj_{\{x_{l_1},\ldots,x_{l_k}\}}y\| = \frac{\mu_{k+1}(r, x_{l_1}, \ldots, x_{l_k})}{\mu_k(x_{l_1}, \ldots, x_{l_k})}. \tag{5}$$

Due to Fact 31 and the fact that without loss of generality we may assume the vectors $x_{l_1}, \ldots, x_{l_{k-1}}$ already added to the model to be orthogonal, we can rotate matrices $[r|x_{l_1}|\ldots|x_{l_k}]$ and $[x_{l_1}|\ldots|x_{l_k}]$ such that matrices M_1 and M_2 are obtained with respectively only $k+1$ and k nonzero rows.

$$M_1 = \begin{pmatrix} z & 0 & \cdots & 0 & \frac{c_{res,l_k}}{z} \\ 0 & 1 & \cdots & 0 & c_{l_1,l_k} \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & c_{l_{k-1},l_k} \\ 0 & 0 & \cdots & 0 & \sqrt{1 - \frac{c^2_{res,l_k}}{z^2} - \sum_{i=1}^{k-1} c^2_{l_i,l_k}} \\ 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \end{pmatrix}, M_2 = \begin{pmatrix} 1 & \cdots & 0 & c_{l_1,l_k} \\ \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & 1 & c_{l_{k-1},l_k} \\ 0 & \cdots & 0 & \sqrt{1 - \sum_{i=1}^{k-1} c^2_{l_i,l_k}} \\ 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \vdots \end{pmatrix},$$

where $z = \|r\|$. Due to Fact 32 the volumes in Equation 5 can be calculated as follows

$$\mu_{k+1}(z, x_{l_1}, \ldots, x_{l_k}) = \det M_1 = \sqrt{z^2(1 - \sum_{i=1}^{k-1} c^2_{l_i,l_k}) - c^2_{res,l_k}},$$

$$\mu_k(x_{l_1}, \ldots, x_{l_k}) = \det M_2 = \sqrt{1 - \sum_{i=1}^{k-1} c^2_{l_i,l_k}}.$$

And then equation (5) can be written as

$$\|y - Proj_{\{x_{l_1},\ldots,x_{l_k}\}}y\| = \frac{z^2(1 - \sum_{i=1}^{k-1} c^2_{l_i,l_k}) - c^2_{res,l_k}}{1 - \sum_{i=1}^{k-1} c^2_{l_i,l_k}},$$

which combined with (4) leads to

$$\frac{c^2_{res,l_k}}{1-\sum_{i=1}^{k-1} c^2_{l_i,l_k}} \leqslant \frac{c^2_{res,l'_k}}{1-\sum_{i=1}^{k-1} c^2_{l_i,l'_k}}.$$

As we can see the above proof is based on the geometric structure of the variables of a linear model. The rest of the proof of theorem 1 is the same as in [10]. We restate it below for the sake of completeness.

Proof (Proof of Theorem 1). If for any $i = 1, \ldots, k-1$:

$$|c_{l_i,l'_k}| \geqslant \frac{|c_{res,l_k}|}{\sqrt{1-\sum_{i=1}^{k-1} c^2_{l_i,l_k} + (k-1)c^2_{res,l_k}}}$$

then the inequality is true. Otherwise for all $i = 1, \ldots, k-1$:

$$|c_{l_i,l'_k}| < \frac{|c_{res,l_k}|}{\sqrt{1-\sum_{i=1}^{k-1} c^2_{l_i,l_k} + (k-1)c^2_{res,l_k}}} \tag{6}$$

and we need to show that this implies $|c_{res,l'_k}| \geqslant \frac{|c_{res,l_k}|}{\sqrt{1-\sum_{i=1}^{k-1} c^2_{l_i,l_k}+(k-1)c^2_{res,l_k}}}$. Notice first that the inequalities (6) imply

$$1-\sum_{i=1}^{k-1} c^2_{l_i,l'_k} > \frac{1-\sum_{i=1}^{k-1} c^2_{l_i,l_k}}{1-\sum_{i=1}^{k-1} c^2_{l_i,l_k} + (k-1)c^2_{res,l_k}}. \tag{7}$$

Using inequality (7) and Lemma 1 we get the desired result:

$$c^2_{res,l'_k} \geqslant c^2_{res,l_k}\frac{1-\sum_{i=1}^{k-1} c^2_{l_i,l'_k}}{1-\sum_{i=1}^{k-1} c^2_{l_i,l_k}} > \frac{c^2_{res,l_k}}{1-\sum_{i=1}^{k-1} c^2_{l_i,l_k} + (k-1)c^2_{res,l_k}}.$$

3.1 Optimality of the constraint

We will now show that the inequality (3) in Theorem 1 cannot be improved. This is illustrated graphically in Figures 3 and 4 and proved in Theorem 2.

Figures 3 and 4 present results on simulated data illustrating Theorem 1. Each point in each figure corresponds to a single simulation run, where random vectors were drawn, normalized and values of both sides of the bound calculated.

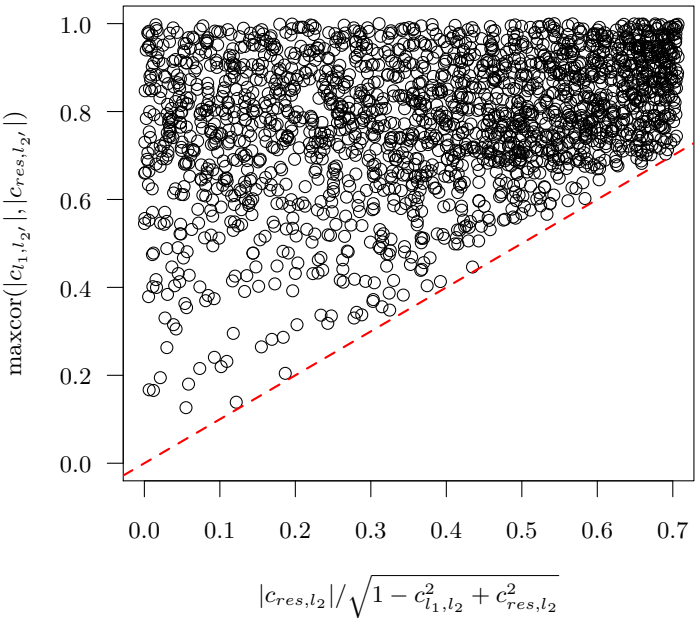

Fig. 3. Illustration of theorem 1 for adding 2nd variable for $n = 4$. Presented points correspond to vectors satisfying theorem assumptions and dashed red line is identity.

The first simulation (Figure 3) was performed as follows. The first predictor x_{l_1} was sampled as a normally distributed vector of a given length $n = 4$, then the response variable y was built as a sum of the vector x_{l_1} and some normally distributed noise. Two other vectors were then sampled similarly to x_{l_1}. The better of them (in the sense of lower RSS) was chosen as $x_{l_2'}$, the worse as x_{l_2} and the values $\max\{|c_{l_1,l_2'}|, |c_{res,l_2'}|\}$ and $|c_{res,l_2}|/\sqrt{1-c^2_{l_1,l_2}+c^2_{res,l_2}}$ were calculated. Then results were plotted as a scatterplot. The red dashed line corresponds to identity, so all vectors for which the inequality $\max\{|c_{l_1,l_2'}|, |c_{res,l_2'}|\} \geqslant |c_{res,l_2}|/\sqrt{1-c^2_{l_1,l_2}+c^2_{res,l_2}}$ is satisfied lie above that line. As we can see, vectors tend to get arbitrarily close to the line, suggesting that the inequality is tight.

The second simulation (Figure 4) is very similar, but instead of adding the second variable we add the third one. Moreover $n = 5$ was chosen. First, two variables x_{l_1} and x_{l_2} ware sampled and orthogonalized, then y was calculated as the sum of x_{l_1}, x_{l_2} and a normally distributed noise. Then two more variables were sampled, and the better one was used as $x_{l_3'}$ and the worse as x_{l_3}. Again,

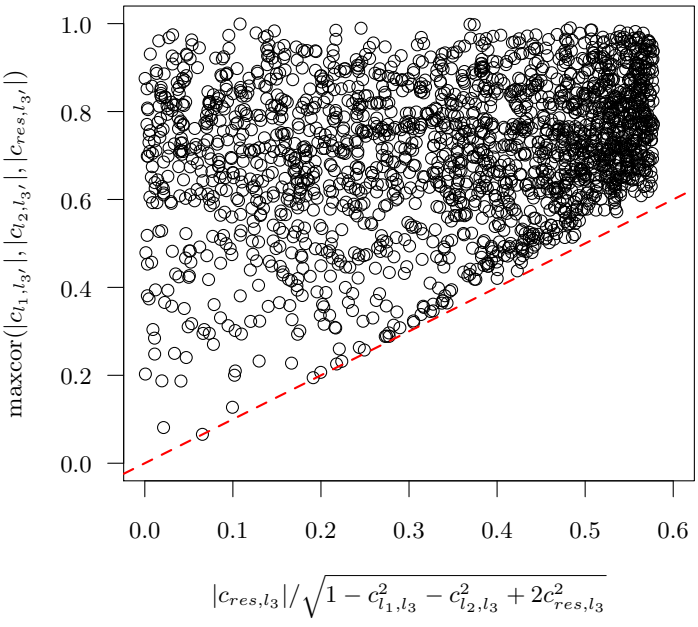

Fig. 4. Illustration of theorem 1 for adding 3rd variable for $n = 5$. Presented points correspond to vectors satisfying theorem assumptions and dashed red line is identity.

the values $\max\{|c_{l_1,l_3'}|, |c_{l_2,l_3'}|, |c_{res,l_3'}|\}$ and $|c_{res,l_3}|/\sqrt{1-c^2_{l_1,l_2}-c^2_{l_1,l_3}+c^2_{res,l_3}}$ were calculated and plotted in the figure. Again, points get arbitrarily close to the line, suggesting tightness of the bound. The theorem below proves that this is indeed the case.

Theorem 2. *The inequality*

$$\max\{|c_{l_1,l'_k}|,\ldots,|c_{l_{k-1},l'_k}|,|c_{res,l'_k}|\} \geqslant \frac{|c_{res,l_k}|}{\sqrt{1-\sum_{i=1}^{k-1} c^2_{l_i,l_k}+(k-1)c^2_{res,l_k}}}$$

form theorem 1 cannot be improved.

Proof. To prove the theorem it is enough to find vectors x_{l_k} and $x_{l'_k}$ such that

$$\max\{|c_{l_1,l'_k}|,\ldots,|c_{l_{k-1},l'_k}|,|c_{res,l'_k}|\} = \frac{|c_{res,l_k}|}{\sqrt{1-\sum_{i=1}^{k-1} c^2_{l_i,l_k}+(k-1)c^2_{res,l_k}}}$$

Let $x_{l_k} = \frac{1}{\sqrt{k}}\frac{r}{\|r\|} + \frac{1}{\sqrt{k}}x_{l_1} + \ldots + \frac{1}{\sqrt{k}}x_{l_{k-1}}$ and $x_{l'_k} = x_{l_k}$, then x_{l_k} is properly normalized ($\bar{x}_{l_k} = 0$, $\|x_{l_k}\| = 1$). Due to the fact that $r, x_{l_1}, \ldots, x_{l_k}$ are uncorrelated, the following correlations are equal to

$$c_{res,l_k} = c_{res,l'_k} = \frac{1}{\sqrt{k}},$$
$$c_{l_i,l_k} = c_{l_i,l'_k} = \frac{1}{\sqrt{k}},$$

thus

$$\max\{|c_{l_1,l'_k}|,\ldots,|c_{l_{k-1},l'_k}|,|c_{res,l'_k}|\} = \frac{1}{\sqrt{k}} = \frac{|c_{res,l_k}|}{\sqrt{1-\sum_{i=1}^{k-1} c^2_{l_i,l_k}+(k-1)c^2_{res,l_k}}}.$$

4 Conclusions

The paper presents an alternative, geometric proof of theorem enabling finding stepwise regression model faster on large data sets, presented in paper [10]. It also shows that the bound in this theorem cannot be improved. Paper discusses stepwise regression with no penalties, which is left for the future research.

5 Acknowledgements

The paper is co-founded by the European Union from resources of the European Social Fund. Project PO KL ,,Information technologies: Research and their interdisciplinary applications", Agreement UDA-POKL.04.01.01-00-051/10-00.

References

1. http://ec.europa.eu/eurostat
2. Schmachtenberg, M., Bizer, C., Jentzsch, A., Cyganiak, R.: Linking open data cloud diagram. http://lod-cloud.net/
3. http://stack.lod2.eu
4. http://linkeddata.org/
5. Heath, T., C., B.: Linked Data: Evolving the Web into a Global Data Space. 1 edn. Synthesis Lectures on the Semantic Web: Theory and Technology. Morgan & Claypool (2011)
6. Bizer, C., Heath, T., Berners-Lee, T.: Linked data - the story so far. International Journal on Semantic Web and Information Systems (IJSWIS) **5**(3) (2009) 1–22
7. Mohebbi, M., Vanderkam, D., Kodysh, J., Schonberger, R., Choi, H., Kumar, S.: Google correlate whitepaper. (2011)
8. http://www.google.com/trends/correlate
9. Vanderkam, D., Schonberger, R., Rowley, H., Kumar, S.: Technical report: Nearest neighbor search in google correlate. (2013)
10. Zogala-Siudem, B., Jaroszewicz, S.: Fast stepwise regression on linked data
11. Muja, M., Lowe, D.: FLANN - Fast Library for Approximate Nearest Neighbors. (2013)
12. Muja, M., Lowe, D.G.: Scalable nearest neighbor algorithms for high dimensional data. IEEE Trans. on Pattern Analysis and Machine Intelligence **36** (2014)
13. Hastie, T., Tibshirani, R., Friedman, J.: The Elements of Statistical Learning: Data Mining, Inference, and Prediction. Springer-Verlag (2009)
14. Efroymson, M.A.: Multiple Regression Analysis. Wiley (1960)
15. Akaike, H.: A new look at the statistical model identification. IEEE Transactions on Automatic Control **AC-19**(6) (1974) 716–723
16. Schwarz, G.: Estimating the dimension of a model. The Annals of Statistics **6**(2) (1978) 461–464
17. Kibriya, A.M., Frank, E.: An empirical comparison of exact nearest neighbour algorithms. In: PKDD, Springer (2007) 140–151