Why Geometric Progression in Selecting the LASSO Parameter: A Theoretical Explanation

William Kubin\textsuperscript{1}, Yi Xie\textsuperscript{1}, Laxman Bokati\textsuperscript{1}, Vladik Kreinovich\textsuperscript{1}, and Kittawit Autchariyapanitkul\textsuperscript{2}

\textsuperscript{1}Computational Science Program
University of Texas at El Paso
ElPaso, Texas 79968, USA
wkubin@miners.utep.edu, yxie3@miners.utep.edu, lbokati@miners.utep.edu, vladik@utep.edu

\textsuperscript{2}Maejo University, Thailand, kittawit_a@mju.ac.th
1. Need for Regression

• In many real-life situations:
  – we know that the quantity $y$ is uniquely determined by the quantities $x_1, \ldots, x_n$, but
  – we do not know the exact formula for this dependence.

• For example, in physics:
  – we know that the aerodynamic resistance increases with the body’s velocity, but
  – we often do not know how exactly.

• In economics:
  – we know that a change in tax rate influences the economic growth, but
  – we often do not know how exactly.
2. Need for Regression (cont-d)

- In all such cases, we need to find the dependence \( y = f(x_1, \ldots, x_n) \) between several quantities.

- This dependence must be determined based on the available data.

- We need to use previous observations \((x_{k1}, \ldots, x_{kn}, y_k)\) in each of which we know both:
  - the values \( x_{ki} \) of the input quantities \( x_i \) and
  - the value \( y_k \) of the output quantity \( y \).

- In statistics, determining the dependence from the data is known as *regression.*
3. Need for Linear Regression

- In most cases, the desired dependence is smooth – and usually, it can even be expanded in Taylor series.

- In many practical situations, the range of the input variables is small, i.e., we have $x_i \approx x_i^{(0)}$ for some $x_i^{(0)}$.

- In such situations, after we expand the desired dependence in Taylor series, we can:
  
  - safely ignore terms which are quadratic or of higher order with respect to the differences $x_i - x_i^{(0)}$ and
  
  - only keep terms which are linear in terms of these differences:

    $$y = f(x_1, \ldots, x_n) = c_0 + \sum_{i=1}^{n} a_i \cdot (x_i - x_i^{(0)}) .$$

- Here $c_0 \stackrel{\text{def}}{=} f(x_1^{(0)}, \ldots, x_n^{(0)})$ and $a_i \stackrel{\text{def}}{=} \frac{\partial f}{\partial x_i} |_{x_i = x_i^{(0)}}$. 
4. Need for Linear Regression (cont-d)

- This expression can be simplified into:

\[ y = a_0 + \sum_{i=1}^{n} a_i \cdot x_i, \text{ where } a_0 \overset{\text{def}}{=} c_0 - \sum_{i=1}^{n} a_i \cdot x_i^{(0)}. \]

- In practice, measurements are never absolutely precise.

- So, when we plug in the actually measured values \( x_{ki} \) and \( y_i \), we will only get an approximate equality:

\[ y_k \approx a_0 + \sum_{i=1}^{m} a_i \cdot x_{ki}. \]

- Thus, the problem of finding the desired dependence can be reformulated as follows:
  
  - given the values \( y_k \) and \( x_{ki} \),
  
  - find the coefficients \( a_i \) for which the approximate equality holds for all \( k \).
5. The Usual Least Squares Approach

- We want each left-and side $y_k$ of the approximate equality to be close to the corresponding right-hand side.
- In other words, we want the left-hand-side tuple $(y_1, \ldots, y_K)$ to be close to the right-hand-sides tuple
  $$\left( \sum_{i=1}^{m} a_i \cdot x_{1i}, \ldots, \sum_{i=1}^{m} a_i \cdot x_{Ki} \right).$$
- It is reasonable to select $a_i$ for which the distance between these two tuples is the smallest possible.
- Minimizing the distance is equivalent to minimizing the square of this distance, i.e., the expression
  $$\sum_{k=1}^{K} \left( y_k - \left( a_0 + \sum_{i=1}^{m} a_i \cdot x_{ki} \right) \right)^2.$$
- This minimization is know as the Least Squares method.
6. The Least Squares Approach (cont-d)

• This is the most widely used method for processing data.

• The corresponding values $a_i$ can be easily found if:
  – we differentiate the quadratic expression with respect to each of the unknowns $a_i$ and then
  – equate the corresponding linear expressions to 0.

• Then, we get an easy-to-solve systems of linear equations.
7. Discussion

- The above heuristic idea becomes well-justified:
  - when we consider the case when the measurement errors are normally distributed
  - with 0 mean and the same standard deviation $\sigma$
- This indeed happens:
  - when the measuring instrument’s bias has been carefully eliminated, and
  - most major sources of measurement errors have been removed.
- In such situations, the resulting measurement error is a joint effect of many similarly small error components.
- For such joint effects, the Central Limit Theorem states that the resulting distribution is close to Gaussian.
8. Discussion (cont-d)

• Once we know the probability distributions, a natural idea is to select the most probable values $a_i$.

• In other words, we select the values for which the probability to observe the values $y_k$ is the largest.

• For normal distributions, this idea leads exactly to the least squares method.
9. Need to Go Beyond Least Squares

• Sometimes, we know that all the inputs $x_i$ are essential to predict the value $y$ of the desired quantity.

• In such cases, the least squares method works reasonably well.

• The problem is that in practice, we often do not know which inputs $x_i$ are relevant and which are not.

• As a result, to be on the safe side, we include as many inputs as possible.

• Many of them will turn out to be irrelevant.

• If all the measurements were exact, this would not be a problem:
  – for irrelevant inputs $x_i$, we would get $a_i = 0$, and
  – the resulting formula would be the desired one.
10. Need to Go Beyond Least Squares (cont-d)

- However, because of the measurement errors, we do not get exactly 0s.

- Moreover, the more such irrelevant variables we add:
  - the more non-zero “noise” terms $a_i \cdot x_i$ we will have, and
  - the larger will be their sum.

- This will negatively affecting the accuracy of the formula,

- Thus, it will negative affect the accuracy of the resulting desired (non-zero) coefficients $a_i$. 
11. LASSO Method

• We know that many coefficients will be 0; so, a natural idea is:
  – instead of considering all possible tuples
    \[ a \overset{\text{def}}{=} (a_0, a_1, \ldots, a_n), \]
  – to only consider tuples for which a bounded number of coefficients is 0: \( \|a\|_0 \leq B \) for some constant \( B \).

• Here, \( \|a\|_0 \) (known as the \( \ell_0 \)-norm) denotes the number of non-zero coefficients in a tuple.

• The problem with this natural idea is that the resulting optimization problem becomes NP-hard.

• This means, crudely speaking, that:
  – no feasible algorithm is possible
  – that would always solve all the instances of this problem.
12. LASSO Method (cont-d)

- A usual way to solve such problem is:
  - by replacing the $\ell_0$-norm with an $\ell_1$-norm $\sum_{i=0}^{n} |a_i|$; 
  - this norm is convex, therefore, the optimization problem is easier to solve.
- So:
  - instead of solving the problem of unconditionally minimizing the quadratic expression,
  - we minimize this expression under the constraint $\sum_{i=0}^{n} |a_i| \leq B$ for some constant $B$.
- This minimum can be attained when we have strict inequality or when the constraint becomes an equality.
- If the constraint is a strict inequality, then we have a local minimum.
13. LASSO Method (cont-d)

- For quadratic functions, a local minimum is exactly the global minimum that we try to avoid.

- Thus, we must consider the case when the constraint becomes an equality $\sum_{i=0}^{n} |a_i| = B$.

- The Lagrange multiplier method leads to minimizing the expression:

$$
\sum_{k=1}^{K} \left( y_k - \left( a_0 + \sum_{i=1}^{m} a_i \cdot x_{ki} \right) \right)^2 + \lambda \cdot \sum_{i=0}^{n} |a_i|.
$$

- This minimization is known as the *Least Absolute Shrinkage and Selection Operator* method – *LASSO*, for short.
14. How $\lambda$ Is Selected: Main Idea

- The success of the LASSO method depends on what value $\lambda$ we select.
- When $\lambda$ is close to 0, we retain all the problems of the usual least squares method.
- When $\lambda$ is too large, the $\lambda$-term dominates.
- So we select all the values $a_i = 0$, which do not provide any good description of the desired dependence.
- In different situations, different values $\lambda$ will work best.
- The more irrelevant inputs we have:
  - the more important it is to deviate from the least squares, and
  - thus, the larger the parameter $\lambda$ – that describes this deviation – should be.
15. How \( \lambda \) Is Selected: Main Idea (cont-d)

- We rarely know beforehand which inputs are relevant – this is the whole problem.
- So we do now know beforehand what value \( \lambda \) we should use.
- The best value \( \lambda \) needs to be decided based on the data.
- A usual way of testing any dependence is by randomly dividing the data into:
  - a (larger) training set and
  - a (smaller) testing set.
- We use the training set to find the value of the desired parameters (in our case, the parameters \( a_i \)).
- Then we use the testing set to gauge how good is the model.
16. How $\lambda$ Is Selected: Main Idea (cont-d)

- To get more reliable results, we can repeat this procedure several times.

- In precise terms, we select several training subsets

  \[ S_1, \ldots, S_m \subseteq \{1, \ldots, K\}. \]

- For each of these subsets $S_j$, we find the values $a_{ij}(\lambda)$ that minimize the functional

  \[
  \sum_{k \in S_j} \left( y_k - \left( a_0 + \sum_{i=1}^{m} a_i \cdot x_{ki} \right) \right)^2 + \lambda \cdot \sum_{i=0}^{n} |a_i|.
  \]

- We can then compute the overall inaccuracy, as

  \[
  \Delta(\lambda) \overset{\text{def}}{=} \sum_{j=1}^{m} \left( \sum_{k \notin S_j} \left( y_k - \left( a_{j0}(\lambda) + \sum_{i=1}^{m} a_{ji}(\lambda) \cdot x_{ki} \right) \right)^2 \right).
  \]

- We then select $\lambda$ for which $\Delta(\lambda)$ is the smallest.
17. How \( \lambda \) Is Selected: Details

- In the ideal world, we should be able to try all possible real values \( \lambda \).

- However, there are infinitely many real numbers, and in practice, we can only test finitely many of them.

- Which set of values \( \lambda \) should we choose?

- Empirically, the best results are obtained if we use the values \( \lambda \) from a geometric progression \( \lambda_n = c_0 \cdot q^n \).

- Of course, a geometric progression also has infinitely many values, but we do not need to test all of them.

- Usually, as \( \lambda \) increases from 0, the value \( \Delta(\lambda) \) first decreases then increases again.

- So, it is enough to catch a moment when this value starts increasing.
18. How $\lambda$ Is Selected: Details (cont-d)

- A natural question is: why geometric progression works best?
- In this talk, we provide a theoretical explanation for this empirical fact.
19. What Do We Want?

• At first glance, the answer is straightforward: we want to select a discrete set of values, i.e., a set

\[ S = \{ \ldots < \lambda_n \lambda_{n+1} < \ldots \} \].

• However, a deeper analysis shows that the answer is not so simple.

• Indeed, what we are interested in is the dependence between the quantities \( y \) and \( x_i \).

• However, what we have to deal with is not the quantities themselves, but their numerical values.

• And the numerical values depend on what unit we choose for measuring these quantities; for example:
  - a person who is 1.7 m high is also 170 cm high,
  - an April 2020 price of 2 US dollars is the same as the price of \( 2 \cdot 23500 = 47000 \) Vietnam Dong, etc.
20. What Do We Want (cont-d)

• In most cases, the choice of the units is rather arbitrary.
• It is therefore reasonable to require that the results of data processing should not depend on the unit.
• And hereby lies a problem.
• Suppose that we keep the same units for \( x_i \) but change a measuring unit for \( y \) to a one which is \( \alpha \) times smaller.
• In this case, the new numerical values of \( y \) become \( \alpha \) times larger: \( y \rightarrow y' = \alpha \cdot y \).
• To properly capture these new values, we need to increase the original values \( a_i \) by the same factor:

\[
a_i \rightarrow a'_i = \alpha \cdot a_i.
\]
21. What Do We Want (cont-d)

- In terms of these new values, the minimized expression takes the form

\[ \sum_{k=1}^{K} \left( y'_k - \left( a'_0 + \sum_{i=1}^{m} a'_i \cdot x_{ki} \right) \right)^2 + \lambda \cdot \sum_{i=0}^{n} |a'_i|. \]

- Taking into account that \( y'_k = \alpha \cdot y_k \) and \( a'_i = \alpha \cdot a_i \), we get:

\[ \alpha^2 \cdot \sum_{k=1}^{K} \left( y_k - \left( a_0 + \sum_{i=1}^{m} a_i \cdot x_{ki} \right) \right)^2 + \alpha \cdot \lambda \cdot \sum_{i=0}^{n} |a_i|. \]

- Minimizing an expression is the same as minimizing \( \alpha^{-2} \) times this expression, i.e., the modified expression

\[ \sum_{k=1}^{K} \left( y_k - \left( a_0 + \sum_{i=1}^{m} a_i \cdot x_{ki} \right) \right)^2 + \alpha^{-1} \cdot \lambda \cdot \sum_{i=0}^{n} |a_i|. \]
• This new expression is similar to the original one, but with a new value of the LASSO parameter $\lambda' = \alpha^{-1} \cdot \lambda$.

• So, when we change the measuring units, the values of $\lambda$ are also re-scaled – i.e., multiplied by a constant.

• What was the set $\{\lambda_n\}$ in the old units becomes the re-scaled set $\{\alpha^{-1} \cdot \lambda_n\}$ in the new units.

• This is, in effect, the same set but corresponding to different measuring units.

• So, we cannot say that one of these sets is better than the other, they clearly have the same quality.

• Thus, we cannot choose a single set $S$, we must choose a family of sets $\{c \cdot S\}_c$, where

$$c \cdot S \overset{\text{def}}{=} \{c \cdot \lambda : \lambda \in S\}.$$
23. Natural Uniqueness Requirement

• Eventually, we need to select some set \( S \).

• We cannot select one set a priori, since with every set \( S \), a set \( c \cdot S \) also has the same quality.

• To fix a unique set, we can, e.g., fix one of the values \( \lambda \in S \).

• Let us require that with this fixture, we will be end up with a unique optimal set \( S \).

• This means, in particular, that:
  
  – if we select a real number \( \lambda \in S \),
  
  – then the only set \( c \cdot S \) that contains this number will be the same set \( S \).

• Let us describe this requirement in precise terms.
24. Definitions and the Main Result

• A set $S \subseteq \mathbb{R}^+$ is called discrete if:
  
  – for every $\lambda \in S$,
  – there exists a $\varepsilon > 0$ such that $|\lambda - \lambda'| > \varepsilon$ for all other $\lambda' \in S$.

• For such sets, for each element $\lambda$:
  
  – if there are larger elements,
  – then there is the “next” element – i.e., the smallest element which is larger than $\lambda$.

• Similarly:
  
  – if there are smaller elements,
  – then there exists the “previous” element – i.e., the largest element which is smaller than $\lambda$.

• Thus, such sets have the form
  
  \[ \{ \ldots < \lambda_{n-1} < \lambda_n < \lambda_{n+1} < \ldots \} \].
25. Definitions and the Main Result (cont-d)

- A discrete set $S$ is called uniquely determined if for every $\lambda \in S$ and $c > 0$, if $\lambda \in c \cdot S$, then $c \cdot S = S$.

- Proposition. A set $S$ is uniquely determined if and only if it is a geometric progression, i.e.:

$$S = \{c_0 \cdot q^n : n = \ldots, -2, -1, 0, 1, 2, \ldots\} \text{ for some } c_0 \text{ and } q.$$

- This results explains why geometric progression is used to select the LASSO parameter $\lambda$. 
26. Proof

- It is easy to prove that every geometric progression is uniquely determined.
- Indeed, if for \( \lambda = c_0 \cdot q^n \), we have \( \lambda \in c \cdot S \), this means that \( \lambda = c \cdot c_0 \cdot q^m \) for some \( m \), i.e., \( c_0 \cdot q^n = c \cdot c_0 \cdot q^m \).
- Dividing both sides by \( c_0 \cdot q^m \), we conclude that \( c = q^{n-m} \) for some integer \( n - m \).
- Let us show that in this case, \( c \cdot S = S \).
- Indeed, each element \( x \) of the set \( c \cdot S \) has the form \( x = c \cdot c_0 \cdot q^k \) for some integer \( k \).
- Substituting \( c = q^{n-m} \) into this formula, we conclude that \( x = c_0 \cdot q^{k+(n-m)} \), i.e., that \( x \in S \).
- Similarly, we can prove that if \( x \in S \), then \( x \in c \cdot S \).
27. Proof (cont-d)

- Vice versa, let us assume that the set $S$ is uniquely determined.
- Let us pick any element $\lambda \in S$ and denote it by $\lambda_0$.
- The next element we will denote by $\lambda_1$, the next to next by $\lambda_2$, etc.
- Similarly, the element previous to $\lambda_0$ will be denoted by $\lambda_{-1}$, previous to previous by $\lambda_{-2}$, etc.
- Thus, $S = \{\ldots, \lambda_{-2}, \lambda_{-1}, \lambda_0, \lambda_1, \lambda_2, \ldots\}$.
- Clearly, $\lambda_1 \in S$, and for $q \overset{\text{def}}{=} \lambda_1/\lambda_0$, we have $\lambda_1 \in q \cdot S$ – since $\lambda_1 = (\lambda_1/\lambda_0) \cdot \lambda_0 = q \cdot \lambda_0$ for $\lambda_0 \in S$.
- Since the set $S$ is uniquely determined, this implies that $q \cdot S = S$.
- Since $S = \{\ldots, \lambda_{-2}, \lambda_{-1}, \lambda_0, \lambda_1, \lambda_2, \ldots\}$, we have
  $$q \cdot S = \{\ldots, q \cdot \lambda_{-2}, q \cdot \lambda_{-1}, q \cdot \lambda_0, q \cdot \lambda_1, q \cdot \lambda_2, \ldots\}.$$
28. Proof (cont-d)

- The sets $S$ and $q \cdot S$ coincide.
- We know that $q \cdot \lambda_0 = \lambda_1$; thus:
  - the element next to $q \cdot \lambda_0$ in the set $q \cdot S$ – i.e., the element $c \cdot \lambda_1$,
  - must be equal to the element which is next to $\lambda_1$ in the set $S$, i.e., to the element $\lambda_2$:
    $$\lambda_2 = q \cdot \lambda_1.$$  
- For next to next elements, we get $\lambda_3 = q \cdot \lambda_2$ and, in general, we get $\lambda_{n+1} = q \cdot \lambda_n$ for all $n$.
- This is exactly the definition of a geometric progression.
- The proposition is proven.
29. Discussion

- Machine learning (e.g., deep learning) uses the gradient method \( x_{i+1} = x_i - \lambda_i \cdot \frac{\partial J}{\partial x_i} \) to minimize \( J \).

- Empirically the best strategy for selecting \( \lambda_i \) also follows approximately a geometric progression.

- For example, some algorithms use:
  - \( \lambda_i = 0.1 \) for the first ten iterations,
  - \( \lambda_i = 0.01 \) for the next ten iterations,
  - \( \lambda_i = 0.001 \) for the next ten iterations, etc.

- In this case, similarly, re-scaling of \( J \) is equivalent to re-scaling of \( \lambda \).

- Thus, we need to have a family of sequences \( \{c \cdot \lambda_i\} \) corresponding to different \( c > 0 \).

- A natural uniqueness requirement – as we have shown – leads to the geometric progression.
30. Acknowledgments

This work was supported in part by the National Science Foundation grants:

- 1623190 (A Model of Change for Preparing a New Generation for Professional Practice in Computer Science),
- HRD-1242122 (Cyber-ShARE Center of Excellence).