A Survey Of Regression Algorithms And Connections With Deep Learning

Yunpeng Tai

Abstract—Regression has attracted immense interest lately due to its effectiveness in tasks like predicting values. And Regression is of widespread use in multiple fields such as Economics, Finance, Business, Biology and so on. While considerable studies have proposed some impressive models, few of them have provided a whole picture regarding how and to what extent Regression has developed. With the aim of aiding beginners in understanding the relationships among different Regression algorithms, this paper characterizes a broad and thoughtful selection of recent regression algorithms, providing an organized and comprehensive overview of existing work and models utilized frequently. In this paper, the relationship between Regression and Deep Learning is also discussed and a conclusion can be drawn that Deep Learning can be more powerful as an combination with Regression models in the future.

Index Terms—Regression, Survey, Comparison Between Algorithms, A Different Vision Of Ordinary Least Squares, Insight Of Regression Future.

1 INTRODUCTION

Regression is an approach of obtaining a relationship between input space and output space. The relationship is represented by a function \( f : X \mapsto Y \) and \( X \) is known as the independent variable, and \( Y \) as the dependent variable. It was originally put forward by Legendre [1] in 1805, who applied least squares in Regression. And then Gauss [2] published a further development of the theory of least squares featuring ordinary least squares in 1821.

Regression belongs to supervised learning and \( Y \) is continuous, i.e. \( Y \in \mathbb{R} \). Undoubtedly, Regression is powerful and has made tremendous impact in enormous fields. As such, an increasing number of research has made fundamental improvement to Regression models over the past few decades.

There has been such a surge of Regression models proposed recently, that researchers and beginners may find it challenging to figure out what exactly every model means and the relationship between them. Thus, a survey of the existing Regression models is beneficial both to beginners who just want to scratch the surface of Regression and researchers willing to have a systematic view of Regression models and gain insight from those smart models.

The key component of Regression is ordinary least squares. It is capable of producing an unbiased linear model of minimum variance as long as six necessary assumptions is satisfied according to Gauss-Markov Theorem [2]. However, if OLS is applied in specific areas, some of the assumptions are likely to be violated so that OLS fails to play its part in predicting values. Hence, it is essential to grasp those assumptions and figure out possible solutions when one of them is broken. And those well-known concerns with OLS contributes to extensive models designed to fix those violations such as Ridge [3] [4], Lasso [5], Elastic Net [6] and so on.

What distinguishes this paper from others is the earlier part of this paper is OLS-centered and alternative solutions provided by different models are discussed at length when some assumptions are broken (Figure 1). This paper views the relationship between models on the whole and discusses the details of distinct models specifically and explicitly. And this paper also provides a walk-through of some uncommon Regression models. What’s more, a possible direction to which Regression is going to develop is covered.

Fig. 1: Problems With OLS And Possible Solutions Provided By Distinct Models. N is the size of input space. P is the number of features of every sample in input space.

This paper is organized as follows. A brief introduction of the Regression task and convention in this paper is included in Section 2. In Section 3, OLS, its assumptions and possible solutions for violation are comprehensively explained. Section 4 is composed of a number of Regression models which enable OLS’s potential to be stimulated although the real data challenges its assumptions and some unexpected situations happen. Generalized Linear Models and an uncommon Regression named as Step-Wise are explored in Section 5. Section 6 sets out to provide a quick overview of the strong bond between Regression and Deep Learning.
Learning. In Section 7, conclusions about Regression are drawn and possible combination of Regression and Deep Learning in the future is discussed. In a nutshell, this well-established paper is an overview of Regression models and the relationship between Regression and Deep Learning, and hopefully this paper does make sense.

2 THE REGRESSION TASK

Our data looks like \( \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\} \), which \( y_i \in \mathbb{R} \). We intend to train a model from our data set and implement it in unknown test sets. A standard for machine performs well is that low residuals (distance from predicted values to labels). When it comes to regression task, it is common way to implement Linear Regression.

\[
y = \hat{y} + \epsilon = \theta x + b + \epsilon
\]  

\( \theta \) is called slope (gradient) or coefficient and \( b \) is called intercept. \( \theta \) explains when \( x \) changes to what extent \( \hat{y} \) is going to change. \( X = \{x_1, x_2, \ldots, x_n\} \) is known as input space and \( Y = \{y_1, y_2, \ldots, y_n\} \) as output space. \( x_i \) is called a sample, and \( x_{ij} \) means the \( j \)-th feature of the \( i \)-th sample. \( y \) is the label and \( \hat{y} \) is the prediction. And \( \epsilon \) is the error accompanied by every prediction and is also called the distance from \( \hat{y} \) to \( y \) (residual). In ordinary least squares, the model assumes that \( y \) is actually sampled from Gaussian Distribution and every sample is with noise. Thus, it can also called noise in statistics. But in this paper, I am going to use error for that.

3 ORDINARY LEAST SQUARES

In Machine Learning, we always figure out the best model by minimizing our objective function, which is also known as cost function. OLS (Ordinary Least Squares) serves as an effective loss function as long as the model satisfies six necessary assumptions. Then it can choose an unbiased model of minimum variance by minimizing the function below. And \( J(\theta) \) is convex. Thus, set partial derivative of \( \theta \) zero and then we can get the best parameter \( \theta^* \). Note that only when \( X^T X \) is full rank, equation(3) does make sense.

Some books may multiply \( J(\theta) \) by \( 1/n \), which is convenient for computation. Note that \( X \) and \( Y \) are matrices.

\[
J(\theta) = \sum_{i=1}^{n} (y_i - (\theta x_i + b))^2
\]

(2)

\[
\theta^* = (X^T X)^{-1} X^T Y
\]

(3)

3.1 Prior Assumptions

In this section, six necessary assumptions is studied. And possible solutions for unsatisfied situations are also covered.

- Linearity. In other words, only Straight Line models are permitted. If the relationship between \( X \) and \( y \) is a non-linear model, e.g. \( y = X^4 \), the whole regression model crashes. And the recipe for this situation is applying feature transformation. By doing so, the whole relationship between \( X \) and \( y \) is changed for good. Hence, we must also take the correlation between \( X \) and \( y \) into consideration. Correlation can be told by calculating \( R^2 \) (Coefficient Of Determination), which stands for the ability to predict \( y \) by observing \( X \). When \( R^2 = 1 \), it means the loss of this predictor is 0. If \( R^2 = 0 \), it is equivalent to that the predictor is constant, which indicates \( X \) has nothing to do with \( y \). As shown in Table 1, in Column \( R^2 \), the value in bracket stands for the original coefficient. And the same goes for Column Linearity. Note that I use 10000 random points from \( y = 2(X - 3)^2 + 5 + N \), which \( N \) stands for noise and it varies from 0 to 1. \( X \) follows random and even distribution.

\[
\bar{y} = \frac{1}{n} \sum_{i=1}^{n} y_i
\]

(4)

\[
R^2 = 1 - \frac{\sum_{i=1}^{n} (y_i - \bar{y})^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2}
\]

(5)

As shown in Table 1, \( R^2 \) changes when feature transformation is applied. And the purpose is to find linear model with the best \( R^2 \). Note that linear means \( y \) w.r.t transformed \( X \). And if so, scatter plot of
Fig. 3: In Figure(a), errors are not symmetrical and Q-Q plot doesn’t look like a line, which indicates errors don’t follow Normal Distribution. Figure(b) and Figure(c) show when errors follow Normal Distribution, the hist plot should look like bell curve and Q-Q plot should be a line.

**TABLE 1: Transformation Results About $R^2$ And Linearity**

| $X$  | $y$  | Transformation | $R^2(-1.602)$ | Linearity(Non) |
|------|------|----------------|----------------|----------------|
| 1.450| 9.994| $X_T = \log_{10}(x)$ | 0.764 | Non |
| 1.003| 13.475| $X_T = \sqrt{x}$ | 0.693 | Non |
| 3.179| 5.935| $X_T = \exp(x)$ | 0.281 | Non |
| 3.801| 6.726| $X_T = \frac{1}{x}$ | 0.870 | Non |
| 1.294| 11.704| $X_T = (x - 3)^2$ | 0.983 | Linear |
| 1.399| 10.915| $X_T = x^3$ | 0.344 | Non |
| 3.919| 5.951| $X_T = x^4$ | 0.253 | Non |
| 2.987| 6.742| $X_T = x^5$ | 0.188 | Non |
| 3.161| 5.906| $X_T = x + 1$ | 0.616 | Non |
| 2.065| 7.647| $X_T = \frac{1}{x}$ | 0.904 | Non |

original $X$ and predicted $y$ should fit the original distribution plot. This assumption is the most significant for Linear Regression and it may explain why Ridge or Lasso also performs badly when the relationship is nonlinear.

- **Constant Error Variance** [7]. It means errors are uniformly distributed, which in statistics is called no Heteroscedasticity. When we apply our model, we can get a bunch of predicted values via observing $X$. Then we can calculate errors between true values and predicted values. And we can also calculate the variance of errors. If errors follow normal distribution (equation 5), thus its variance is constant($\sigma^2$). Also, its distribution is symmetrical. In turn, the distribution of errors should also be uniform and symmetrical. So we can use error plot to detect it (Figure 3). And Q-Q Plot can detect whether the errors follow normal distribution. The data I use can be downloaded here. I choose X2 house age as $X$ and Y house price of unit area for $y$. Note that I remove points that $x$ equals 0. And I choose 200 points for study.

$$f(x) = \frac{1}{\sigma \sqrt{2\pi}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

equation(6) represents standard normal distribution, $\mu = 0$, $\sigma = 1$. As shown in Figure 3, if errors follow normal distribution, the distribution should be uniform just like Figure (b) and histogram should be like bell curve. We can also draw a safe conclusion that if errors follow normal distribution, their distribution should fit the line in Q-Q Plot.

After figuring out heteroscedasticity, we can come up with a question that how it influences our model and how to improve it. The most common way is to try feature transformation e.g. Log. As shown in Figure 4, it can, to some degree, make our errors’ distribution slightly more stable. It’s always an option to try, but not an effective method to handle the problem.

What’s more, we can apply Box Cox Transformation, which can make data move closer to normal distribution. In statistics, if data follows normal distribution and then the variance of noise(error) is a constant($\sigma^2$). Thus, normality of data is likely to relieve heteroscedasticity. And data can be downloaded here. I choose total sulfur dioxide for $X$ and quality for $y$. In Figure 5, it may relieve heteroscedasticity. Note you can’t always depend on it. Also, it can be worse(Figure 6).

- **Independent Errors(no autocorrelation, AC for short).** For instance, you want to predict the shares in stock market. But the errors are correlated while they should be i.i.d (independent identically distributed). When a financial crisis happens, the shares is going to be extremely unstable in the next few months, which means errors are going to increase sharply. It can be detected by Durbin Watson Test (Table 3) or drawing AC Plot. And if values of y axis in AC Plot vary from (0, 1), they mean Positive AC. If values equal 0, they mean Non AC. Otherwise, they mean Negative AC.

**TABLE 2: Durbin Watson Test**

| Value | Relationship |
|-------|--------------|
| 2.0   | no Ac        |
| 0.0-2.0 | positive AC  |
| 2.0 - 4.0 | negative AC  |

The autocorrelation can affect errors’ standard deviation while it’s unlikely to have an influence on model’s coefficient and intercept [8].

There’re two common ways to fix it. The first is to add omitted variables. For example, you want to predict stock performances by time. Undoubtedly, the model is of high autocorrelation. We can, however, add S&P 500. And hopefully, it may relieve autocorrelation. The second is to switch another function. You can transform your linear model into a squared model.
4.1 Ridge Regression

- Background: high variance. OLS enables the predictor to perform good on training sets, however, this is an invitation to poor performance on testing sets, which is also known as overfitting. Generally speaking, the more complicated a model is, the poorer performance of the model on unknown sets. According to Occam’s rule, a model is more likely to do a good job on unknown sets if the model is simple. And the model is of high generalization ability.

- Shrinkage. When OLS is applied in real life, the predictor’s coefficients can be too large in absolute. The coefficient of variable more correlated to y is large while the one of variable less correlated to y is also large, which is misleading to figure out the relationship between X and y [3]. This phenomenon can account for poor performance on unknown sets. Thus, Ridge is going to implement shrinkage on coefficients and the extent of shrinkage counts on the degree of correlation. Typically, if a variable holds much predicting power, its coefficient is more likely to be big [4].

- Nonorthogonal Solution [3]. Whether OLS can be directly computed just by derivative relies mainly on if $X^TX$ is orthogonal. If $X^TX$ is nonorthogonal, this means $X^TX$ is not reversible and the direct computation can’t work. I is a unit matrix. There’re only small positive quantity on the diagonal of $kI$, the diagonal looks like a ridge in comparison with zero’s distribution (equation 9).

\[
softmax(x) = \frac{e^{x_i}}{\sum_{i=1}^{n} e^{x_i}} \quad (8)
\]

Our goal of Regression model is to figure out the relationship between independent variable(X) and dependent variable(y) by finding a proper coefficient. But when there’s Multicollinearity, the coefficient is unable to interpret. We actually don’t know what exactly the relationship is. However, if we just want to make good predictions, it’s still effective [7]. And if the degree of Multicollinearity is moderate, you don’t have to care about it too much. We can remove highly correlated variables or increase sample size.

- Normality Of Data [7]. Box-Cox is the efficient transformation to make data more close to normal distribution. Normalization and some basic feature transformation may help. And also try increasing data size.

- No Exogeneity. If $X$ we choose itself is of little influence on $y$, which means the real prediction is not based on $X$, then there’s exogeneity [7]. And the best solution is to make a good analysis about what on earth affects our predicted values and choose a suitable $X$.

4 ALTERNATIVE MODELS

In this section, OLS’s weaknesses are going to appear in Background. And each Background stands for a specific problem.

4.1 Ridge Regression

- Background: high variance. OLS enables the predictor to perform good on training sets, however, this is an invitation to poor performance on testing sets, which is also known as overfitting. Generally speaking, the more complicated a model is, the poorer performance of the model on unknown sets. According to Occam’s rule, a model is more likely to do a good job on unknown sets if the model is simple. And the model is of high generalization ability.

\[
\theta^* = (X^TX + kI)^{-1}X^TY \quad (9)
\]

\[
J(\theta) = \frac{1}{n} \sum_{i} (y_i - \hat{y}_i)^2 + \frac{\lambda}{2} ||\theta||^2 \quad (10)
\]
4.2 Lasso Regression

- Background: Ridge’s poor performance on outliers. Compared to OLS, Ridge is quite powerful but shrinkage means it just cuts down on coefficients’ ability of affecting the result. As such, each coefficient still has influence on the result which further indicates Ridge still cares about every sample’s loss. However, when outliers appear in the data, Ridge fails to deal with them in that it is sensitive to outliers.
- Feature Selection [5]. Unlike Ridge, Lasso implements feature selection because only one coefficient is saved and others are set zero, which is known as sparse solution. Therefore, only one sample has effect on the prediction in the sense that Lasso is insensitive to outliers and robust to small changes. Feature selection results in oscillation in optimization, which means Ridge is more stable than Lasso in gradient descent(Figure 7). What’s more, although Lasso has looked at all the data, only one sample does make sense. Thus, Lasso is also able to avoid overfitting. If performance of Lasso is excellent, then we can say which sample does work. So Lasso is an interpretable model compared to OLS and Ridge.

\[ \frac{\partial J(\theta)}{\partial \theta} = \frac{\partial MSE}{\partial \theta} + \lambda \theta \]  
(11.1)
\[ = \left[ \begin{array}{c} \frac{\partial M(\theta)}{\partial \theta_1} \\ \frac{\partial M(\theta)}{\partial \theta_2} \\ \vdots \\ \frac{\partial M(\theta)}{\partial \theta_n} \end{array} \right] + \lambda \left[ \begin{array}{c} \theta_1 \\ \theta_2 \\ \vdots \\ \theta_n \end{array} \right] \]  
(11.2)

\[ J(\theta) = \frac{1}{n} \sum_{i} (y_i - \hat{y}_i)^2 + \lambda ||\theta|| \]  
(12)
\[ \frac{\partial J(\theta)}{\partial \theta} = \frac{\partial MSE}{\partial \theta} + \lambda sign(\theta) \]  
(13.1)
\[ = \left[ \begin{array}{c} \frac{\partial M(\theta)}{\partial \theta_1} \\ \frac{\partial M(\theta)}{\partial \theta_2} \\ \vdots \\ \frac{\partial M(\theta)}{\partial \theta_n} \end{array} \right] + \lambda \left[ \begin{array}{c} sign(\theta_1) \\ sign(\theta_2) \\ \vdots \\ sign(\theta_n) \end{array} \right] \]  
(13.2)

4.3 Support Vector Regression

- Margin Maximization. SVM(Support Vector Machine) is originally invented for classification problems [9]. Unlike other algorithms, SVM not only needs to classify all the data correctly but also requires the distance of data to the Hyper plane to be the biggest, which is known as widest street. Our linear model is \( y = \theta^T X + b \). Among all the data points, the distance of the closest positive point to the hyper plane pluses the same distance of closest negative point is Margin(\( \gamma \)). Then SVM is turned into maximizing Margin. And the same for SVR. We can turn the maximizing \( \frac{2}{||\theta||} \) into minimizing \( ||\theta||^2 \). Because \( ||\theta|| \) is bigger than 0, \( ||\theta||^2 \) is proportional to \( ||\theta|| \). Then the problem for SVR is like equation 15. In equation 15, C is a coefficient for regularization and \( L(x) \) is an undefined loss function.

\[ \min_{\theta,b} \frac{1}{2} ||\theta||^2 + C \sum_{i=1}^{n} L(y_i - \hat{y}_i) \]  
(14)

\[ \epsilon \] - insensitive loss. SVR can tolerate mistakes which are no more than \( \epsilon \), if those points predicted wrongly in dashed area \( [f(x) - \epsilon, f(x) + \epsilon] \), then the losses of those points equal zero, which is called \( \epsilon \) - insensitive loss(Figure 8). In equation 16,\( z \) stands for loss. Actually, it just makes a trade-off between errors and complexity of the model. Hence, SVR is unlikely to overfit the data. Thus, SVR can reduce the variance of OLS.

\[ l_\epsilon(z) = \begin{cases} 0 & , if \ |z| \leq \epsilon \\ |z| - \epsilon & , otherwise \end{cases} \]  
(15)
• Dual Problem. Slack variables can be introduced to optimization problem (equation 18). It means how many errors are allowed to make beyond ϵ.

\[
\min_{\theta, b} \frac{1}{2} ||\theta||^2 + C \sum_{i=1}^{n} L(\xi_i + \tilde{\xi}_i) \tag{16}
\]

\[
y_i - \tilde{y}_i \leq \epsilon + \xi_i \tag{17}
\]

\[
\tilde{y}_i - y_i \leq \epsilon + \tilde{\xi}_i \tag{18}
\]

\[
\xi_i \geq 0, \tilde{\xi}_i \geq 0 \tag{19}
\]

We can introduce Lagrange Multiplier \(\mu_i \geq 0, \tilde{\mu}_i \geq 0\).

\[
L(\theta, b, \alpha, \tilde{\alpha}, \xi, \tilde{\xi}, \mu, \tilde{\mu})
\]

\[
= \frac{1}{2} ||\theta||^2 + C \sum_{i=1}^{n} (\xi_i + \tilde{\xi}_i) - \sum_{i=1}^{n} \mu_i \xi_i - \sum_{i=1}^{n} \tilde{\mu}_i \tilde{\xi}_i
\]

\[
+ \sum_{i=1}^{n} \alpha_i (y_i - \tilde{y}_i - \epsilon - \xi_i) + \sum_{i=1}^{n} \tilde{\alpha}_i (\tilde{y}_i - y_i - \epsilon - \tilde{\xi}_i)
\]

And set partial derivative of \(w, b, \xi_i, \tilde{\xi}_i\) zero.

\[
w = \sum_{i=1}^{n} (\tilde{\alpha}_i - \alpha_i) x_i \tag{21}
\]

\[
0 = \sum_{i=1}^{n} (\tilde{\alpha}_i - \alpha_i) \tag{22}
\]

\[
C = \alpha_i + \mu_i \tag{23}
\]

\[
C = \tilde{\alpha}_i + \tilde{\mu}_i \tag{24}
\]

Thus, we can get Dual Problem for SVR.

\[
\max_{\alpha, \tilde{\alpha}} \sum_{i=1}^{n} \bar{y}_i (\tilde{\alpha}_i - \alpha_i) - \epsilon (\tilde{\alpha}_i + \alpha_i)
\]

\[
- \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} (\tilde{\alpha}_i - \alpha_i) (\tilde{\alpha}_j - \alpha_j) (x_i)^T x_j \tag{25}
\]

\[
\text{s.t. } \sum_{i=1}^{n} (\tilde{\alpha}_i - \alpha_i) = 0
\]

• Kernel Trick. Typically, the model is nonlinear in 1D space. Thus, we use \(\phi(x)\) to represent the transformed \(x\). Some common kernel functions are listed in table 3.

\[
w = \sum_{i=1}^{n} (\tilde{\alpha}_i - \alpha_i) \phi(x) \tag{26}
\]

\[
f(x) = \sum_{i=1}^{n} (\tilde{\alpha}_i - \alpha_i) k(x, x_i) + b \tag{27}
\]

4.4 Random Forest Regression

• Tree Structure [12]. A walk-through of tree structure is going to be given in this section. In regression tree, there’re root node, internal nodes and leaf node. For instance, if we’re going to predict a man’s height. And we get men’s and women’s heights. Thus, the root node is man or not. If a point which we need to predict is \((68, 171)\). It means the height of a man weighing 68kg is 171cm. If we use tree regression, then our internal node is > 60? And the second internal node is < 70? and so on. And at last, the leaf node is the result we predict. Hence, the leaf node is the predicted height.

• Classification And Regression Tree [13]. CART includes feature selection, generating trees and pruning. CART assumes that decision tree is a Binary tree. The decision tree is equivalent to dividing features into two groups recursively. It divides the input space into limited units and predict the distribution. Suppose our data \(D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_n, y_n)\}\). Decision tree splits input space into \(M\) units \(R_1, R_2, \ldots, R_M\) and at the end of every unit is the output value \(c_m\). And we can minimize squared errors of output values and true values. And it’s obvious that the best output value in every unit should be the average value. But the question is that how to split the input space. And we choose \(x_j\) randomly and its output value \(s\). And we split the input space into two space by their output values. \(R_1(j, s) = \{x|x_j \leq s\}\) and \(R_2(j, s) = \{x|x_j > s\}\) \(x_j\) is called splitting variable and \(s\) is splitting point. And we find \(c_1, c_2\) in \(R_1, R_2\) via squared least errors. We also want \(c_1 + c_2\) to be small enough. That’s how we finally get \(j, s\).

\[
\min_{i=1}^{n} (y_i - \tilde{y}_i)^2 \tag{28}
\]

\[
c_m = \text{average}(y_i|x_i \in R_m) \tag{29}
\]

\[
\min_{j, s} \min_{c_1} \sum_{x_i \in R_1} (y_i - c_1)^2 + \min_{c_2} \sum_{x_i \in R_2} (y_i - c_2)^2 \tag{30}
\]

• Pruning [13]. Decision Tree implements recursive binary splitting to make more accurate predictions. And if the size of data and the input space is quite large. The structure of the tree is complicated. Sadly, a complex model easily gives birth to Overfitting. So it’s necessary to make our model more simple. Therefore, from the bottom of the tree, we cut down some child trees. Then we can get a tree sequence \(\{T_0, T_1, \ldots, T_n\}\) \(T_0\) is the root node. Then we employ cross-validation to choose the best child tree from it. At the same time, we also expect our model to perform well. Hence, we apply a loss function to measure the differences of performances in the process of pruning. In equation (32), \(T\) is arbitrary child tree. \(C(T)\) means errors on the training data. \(|T|\) is the

| Name          | Expression                  | Parameters                                                                 |
|---------------|-----------------------------|----------------------------------------------------------------------------|
| Linear Kernel | \(k(x_i, x_j) = (x_i)^T x_j\) | \(d = 1\)                                                                  |
| Polynomial Kernel | \(k(x_i, x_j) = (x_i)^T x_j)^d\) | \(\geq 1\)                                                                |
| Gaussian Kernel | \(k(x_i, x_j) = \exp(-\frac{|x_i - x_j|^2}{2\sigma^2})\) | \(\sigma > 0\)                                                           |
| Sigmoid Kernel | \(k(x_i, x_j) = \tanh(\beta (x_i)^T x_j + \theta)\) | \(\beta > 0, \theta < 0\)                                               |

TABLE 3: Common Kernel Functions
number of leaf nodes in a child tree. \( \alpha \) is a parameter, which decides the regularization term. If \( \alpha \) is big, it means hard punishment and this results in a simple tree. If \( \alpha \) is small, it means soft punishment and this leads to a more complicated tree respectively.

\[
C_\alpha(T) = C(T) + \alpha |T|
\]

(31)

- Ensemble Learning [14]. We use 1/3 of our data to evaluate our model, which is called out of bag data. And 2/3 of our data to be a new data set. Then we select subsets randomly from the new data set, which is known as Bagging. Every time we select one subset of the complete data set and then the subset is placed back. The number of points in different subsets is the same. We train different tree models for every different data. Finally we make an average of all trees’ variables as our final model variable in order to cut down on the variance. Hence, it’s an accurate model. And it is able to maintain accuracy although most of data is missing in that the model only randomly select a subset to train. However, it may overfit data when there’re some outliers in data.

4.5 Boosted Regression Tree

- Biased Feature Selection [15]. It’s almost the same as Random Forest. They both select the random subset and then a new tree is generated in Random Forest, the probability of data is selected is the same, which is almost unbiased. But in Boosted Regression Tree, it is going to give weights to every data point. For instance, first time we select a subset and we build a tree model for it. Before this subset is placed back, prediction errors are calculated for every point. If the error is high, this point is likely to be given large weights, which indicates its probability of being selected is higher than others. To summarize, Boosted Regression Tree focuses on the errors and is going to mix it. Whereas, if there’re many outliers in data, it just sucks. But it is robust to missing values just like Random Forest because they both select subsets to fit.

4.6 Elastic Net Regression

- Background: Multicollinearity. One of OLS’s assumptions is that no multicollinearity. However, in multivariate regression, \( X \) can be sometimes dependent. If this happens, OLS, Ridge and Lasso fail to play their part.

- Encourage Group Effect [6]. In multivariate regression, if some samples is correlated to each other, OLS is likely to take one sample, not caring which one is selected while strongly correlated samples are on the same boat in Elastic Net Regression. And it view them as a whole, which is called group effect. It does automatic variable selection and continuous shrinkage, and it select groups of correlated samples [6], which is similar to clustering methods. However, this model doesn’t reduce the variance and extra bias increases.

4.7 Least Angle Regression

- Joint Least Squares Direction [16]. To begin with, the model selects the coefficient \( \beta_j \) and calculate errors. When some other sample \( x_k \) has more correlation with errors than \( x_j \) has. Hence, the model increases \((\beta_j, \beta_k)\) in their joint least squares direction until \( x_m \) has more correlation with errors. Thus, the model increases \((\beta_j, \beta_k, \beta_m)\) in their joint least squares direction. The model comes to its end until all samples in the model. Therefore, it can also settle samples’ autocorrelation in high dimension. OLS is the special case of LARS. When LARS doesn’t increase in joint least squares direction, the model becomes OLS. By the way, LARS is also powerful when \( N \gg P \) just like Elastic Net Regression.

- Sensitive To Outliers. LARS is a method which iteratively refit the errors. When there’re outliers in data, then LARS doesn’t make sense.

- Easily Modified. It’s simple for LARS to join with other models such as Lasso. LARS-Lasso employs the Lasso’s loss function and applies the LARS’s method of coefficient selection.

4.8 RANSAC Regression

- Background. Although Lasso is insensitive to a few outliers, Lasso doesn’t work when data is filled with a large number of outliers, let alone OLS.

- Random Sample Consensus Set [17]. RANSAC is an iterative method. And it has a error threshold \( \epsilon \). To begin with, it select a subset of the whole data and find a model to fit it. Then, use the model to test the rest of the data. If point’ loss on the model is no more than \( \epsilon \), then add it to the consensus set, which is full of inliers. And this process is iterative. When iterative times is reached, the process comes to its end. RANSAC is going to make the number of points in consensus set as large as possible.

- Voting Scheme. RANSAC is kind of like voting process. A subset of data claims its idea and then the rest of the data votes for the idea. In every independent process, there are two kinds of data. One agrees with the idea while the other is against it. And RANSAC is going to select a process where the number of supporters is max. Thus, RANSAC is biased. If the
model is going to be robust to outliers, then there must be enough good features which vote for correct model and outliers can’t vote consistently.

- Disadvantages. When there’re few outliers in the data, RANSAC can’t make sense in that the difference between every process is little. Only when the data is heavily contaminated, RANSAC can play its part. Besides, the threshold must be set by hand, which requires users to decide specific threshold on different data. Last but not least, the cost of computation is high because it is an iterative method and the number of times required in the model is unknown.

4.9 Theil-Sen Regression

- Median Method [18]. There’re many data pairs used to calculate coefficient. \( \theta = y_a - y_b / x_a - x_b \) And \( \theta \) is the median of all \( \theta \)s. \( b = y - \theta x \) \( b \) is also the median of \( bs \). Hence, it’s a nonparametric technique. However, complete computation leads to low speed.

- Breaking Point. In particular, Theil-Sen only can tolerate 29.3% of data is outliers. And when the model is applied in high-dimensional regression, the rate is going to decrease.

4.10 Huber Regression

- Huber Loss [19]. It transforms its loss function when faced with different values. When values are large, which is of high possibility of being outliers, Huber turns their loss function into Linear Loss in order to minimize their influence on the model. \( \delta \) serves as a threshold, deciding how large data is to need a linear loss. And it is fastest in three robust regression.

\[
HuberLoss = \begin{cases} 
\frac{1}{2}a^2 & \text{if } |a| \leq \delta \\
\delta(|a| - \frac{1}{2}\delta) & \text{otherwise}
\end{cases}
\]  

(33)

4.11 Multivariate Adaptive Regression Splines

- Partitioning [20]. MARS begins with partitioning data and then runs linear regression on each different partition. And it makes no assumptions about the relationship between the labels and samples. MARS originally has a large collection of basis functions. Each meeting point of two linear models is called a knot. And each knot has a pair of basis functions. And these functions are used to describe the relationship between \( x \) and \( y \). The first basis function is \( \max(0, x - y) \). The second is \( \max(0, y - x) \).

- Remove Basis Functions [20]. After MARS partitions data and builds models, it applies least-squares model to fit data. And each knot has two basis functions. The results of them can be viewed as input variables. Least-Squares model estimate the loss of each basis function’s output value. If a basis function has little influence on model fitting, then it is going to be removed.

- Advantages And Disadvantages. It can fit a large number of predictor variables. And it is an effective and fast algorithm. Also, it is robust to outliers. However, it begins with a large set of models and this easily leads to overfitting. And it is vulnerable to missing data problems.

4.12 Polynomial Regression

- Background: When the relationship between \( X \) and \( y \) is nonlinear, OLS sucks.

- Polynomial Transformation [21]. Polynomial Regression replaces original \( X \) with Polynomial in order to attain a more linear relationship than before or change features for some reason. Hence, it is not interpretable. Interestingly, it is somewhat like Taylor Extend. When your model breaks the assumption of linearity, then you can try all polynomial regression to find a best one, which is a good recipe. If feature’s dimension is 2,

\[
\begin{align*}
\text{order} = 1 & \Rightarrow [1, X_1, X_2] \\
\text{order} = 2 & \Rightarrow [1, X_1, X_2, X_1^2, X_2^2] \\
\text{order} = 3 & \Rightarrow [1, X_1, X_2, X_1^2, X_1X_2, X_2^2, X_1^3, X_1^2X_2, X_1X_2^2, X_2^3]
\end{align*}
\]

4.13 Weighted Least Squares

- Background. One of the OLS’s assumptions is constant error variance. In section 3, I put forward log method. However, it is ineffective.

- Transformed Weights [22]. When \( w_i \) all equals 1, it should be the OLS. In OLS, the model gives every point the same attention. But it’s under homoskedasticity while we come across more heteroskedastic scenarios. The idea is that we gives more attention to those points of which error is small. Thus, the model gives those points bigger weights.

- Stable Intercept and Sensitive Coefficient [23]. Note I choose different values for the first 20 weights and others are always 1. Different weights are equivalent for errors’ abnormal distribution. As shown in Table 4, the intercept is right regardless. However, the coefficient changes sharply. Hence, we can’t use it to draw inferences and test our hypotheses with regard to coefficient.

\[
WLS = \sum_{i=1}^{n} w_i (y_i - \hat{y}_i)^2
\]  

(34)
Fig. 9: The middle line shows OLS and MLS fits the data. And others show the range of predicted values of different algorithms.

### TABLE 4: Slopes And Intercepts

| Weights | Slopes | Intercepts |
|---------|--------|------------|
| 1.0     | 1.444  | 0.059      |
| 10.0    | 1.4887 | 0.059      |
| 20.0    | 1.5146 | 0.058      |
| 40.0    | 1.5407 | 0.058      |
| 80.0    | 1.5615 | 0.057      |
| 160.0   | 1.5755 | 0.057      |

As shown in Figure 9, Figure(a) shows that OLS is the special case of WLS when weights = 1. And Figure(b) and Figure(c) show when weights change, the intercept almost stays the same.

#### 4.14 Generalized Least Squares

- **Background.** OLS assumes that the error must be independent in the sense that one error can’t be correlated to others. But when error autocorrelation happens, OLS has no way but to fail.

- **Better OLS** [24]. GLS is similar to OLS on a linearly transformed version of the data. And GLS is unbiased, consistent and effective. WLS is the special case of GLS, which means GLS can also solve heteroskedasticity [25].

#### 4.15 Feasible Generalized Least Squares

- **Implementable GLS.** While GLS sounds powerful, but it can’t be applied in specific regression tasks. FGLS is an implementable version of GLS. And FGLS needs some crucial assumptions to ensure a consistent estimator for errors covariance matrix.

- **Inefficiency On Little Data.** Whereas GLS is more powerful than OLS under heteroscedasticity or autocorrelation, this is not the case for FGLS. When the size of data is quite small, FGLS is ineffective than OLS. Thus, some people prefer FGLS over OLS under small data. But when the size of data becomes large, FGLS is a better choice.

#### 4.16 Bayesian Regression

- **Background.** As is known to all, OLS exactly makes an estimation of the mean of the values, which fails to provide a whole picture of the relationship between independent variables and dependent variables. And in some cases, we want to obtain a possible distribution of labels instead of a mean value.

- **Bayesian Theorem** [27]. For instance, we’re going to employ a model to distinguish whether an email is normal or spam. So what our model faces is that it has to make predictions about the unknown email. Our data has 100 emails and 10% of them is spam. Hence, the percentage of spam is 10%. But that’s absolutely not the whole story. In Bayesian, it’s called Prior Probability, which means the Basic Assumption of the distribution and that’s where Bayesian Begins. At the beginning of the algorithm, Bayesian is biased in return the model is easily affected by the original distribution. For example, if we only have all 10 normal emails, it’s impossible that we wouldn’t get any spam emails in future. In other words, if the size of our data is quite small, it’s not incentive for us to implement Bayesian. However, when training times keep increasing, we should get ideal results ultimately. In the equation below, P(B) is a Normalization term and P(A) is Prior Probability. P(A|B) is called Posterior Probability(Conditional Probability).

To conclude, when we have much data, Bayesian may be a good choice to try while it exactly performs like other algorithms.

\[
P(A|B) = \frac{P(B|A)P(A)}{P(B)}
\]  

(35)

- **Maximum Likelihood Estimation** [28]. Generally speaking, our goal is to figure out the real data distribution, which is almost impossible. Therefore, we want a data distribution which is close to our data distribution from a problem domain. MLE(Maximum Likelihood Estimation) indicates that we want to maximize the probability that real data is sampled from the Hypothesis Distribution.

\[
\beta^* = \arg \max_{\beta} P(\beta|D)
\]  

(36)

- **Maximum Posterior Estimation** [29]. Typically, we can use MAP(Maximum A Posterior Estimation) to replace MLE. It’s based on Bayesian Theorem. And MAP is fundamental to Bayesian Regression(equation 37). Rather than other standard algorithms, Bayesian Regression doesn’t produce a single value but a range of possible distribution. And in most cases, MLE and MAP are likely to get the same results. However, when the hypothesis of MAP is different from MLE, they fail to reach the same destination. When Prior Probability is uniformly distributed, they can make it. From another point of view, if we have some precise understanding of data, Bayesian Regression is a excellent choice in that it serves as Prior Probability or we can weigh every different choice just like Weighted Least Errors. Interestingly, prior can be kind of Regularization or bias of the model, as such prior can be interpreted as L2 norm, which is also called Bayesian Ridge Regression. Equation(38) means given a model m, the probability of output y. And \(\beta\) (Coefficients) and \(\sigma\) (Standard Deviation) are arbitrary values.
\[ P(\beta|D) = \frac{P(D|\beta)P(\beta)}{P(D)} \]  
\[ P(y|m) = \frac{P(\beta, \sigma|m)P(y|X, \beta, \sigma, m)}{P(\beta, \sigma|y, X, m)} \]
\[ y \sim (\beta^T X, \sigma^2) \]

4.17 Quantile Regression

- Transformed Loss Function [30]. QR has a parameter \( q \) which decides the proportion to split the data. One is \( q \% \) of the data and the other is \((1-q)\%\) of the data. For instance, if \( q = 0.5 \), then data is split in two. We minimize the squared loss in OLS while we now minimize the absolute loss in QR.

\[ J(\theta) = \sum_{i=1}^{n} q|y_i - \hat{y}_i| + \sum_{i=1}^{n} (1-q)|y_i - \hat{y}_i| \]

- Advantages. It goes without saying that QR can provide a more complete view of the relationship than OLS. What’s more, it is also robust to outliers and situations where the variance of errors is not a constant.

4.18 Ordinal Regression

- Background. In some cases, the values for labels are ranking numbers. For instance, 0-5 can represent his ability of communicating with others in social science. And OLS can’t make accurate prediction about them.

- Ranking Learning [31]. In OR, the model has a set of thresholds \( \theta_1, \theta_2, \ldots, \theta_n \), which is used to split predictions into independent intervals and every interval corresponds to a \( y \). The model can be represented by sigmoid function of which output values stand for possibility.

\[ P(y \leq i|x) = \sigma(\theta_i - \hat{y}_i) \]

5 EXTRA MODELS

5.1 Generalized Linear Models

- Generalized Functions [32]. Just as the name implies, it is generalization of different functions. And it consists of two significant parts. The first part is the probability distribution of \( y \) such as normal distribution(OLS). The second is linear predictor, which decides how the coefficients combine with independent variables. And GLM includes several regression models such as Binominal Regression, Bernoulli Regression, Poisson Regression and so on. Their application is not so wide and they just swift the two parts compared to OLS. Hence, it’s left out in this paper.

- Advantages And Disadvantages. It can absolutely deal with situations where \( y \) doesn’t follow normal distribution. However, it needs large data sets and it is sensitive to outliers.

5.2 Step-Wise Regression

- Forward Selection [33]. This method begins with no variables. And it involves testing the addition of the variable in an iterative method which is of great use to the improvement of the accuracy. The model repeats until no improvement.

- Backward Elimination. This method starts with many candidate variables. It involves testing the loss of the model with the deletion of variables. If the loss is small, then the variable is going to be deleted. The model repeats until no variable can be deleted.

- Bidirectional Elimination. This is an combination of the above two methods. Whether adding or deleting a variable is decided on every step. To summarize, Step-Wise Regression contains a big space of possible models, which can lead to overfitting.

- Reasons For Stopping [34]. First, the tests such as F-tests and t-tests are biased, thus it may not be accurate. Second, widespread incorrect usage and availability of alternative models such as ensemble learning have led to calls to stop the use of this algorithm.

6 RELATIONSHIP WITH DEEP LEARNING

6.1 General Regression Neural Network

![GRNN Structure](https://www.mdpi.com/1424-8220/20/9/2625)

- Network Structure [35]. GRNN includes input, pattern, summation and output layers. The input and output layers are independent vector and dependent vector. The pattern layer can be seen as a vector full of coefficients. For instance, if we want to apply \( y = \theta x \). Then one pattern neuron stands for \( \theta_i x_i \). And output layer can be formulated as the equation below. And this model can maintain its accuracy with small data and it’s robust to outliers. However, the structure of the network is complicated so that it is computationally expensive.
Y(x) = \frac{\sum_{i=1}^{n} y_i K(x, x_i)}{\sum_{i=1}^{n} K(x, x_i)} \quad (42)
K(x, x_i) = e^{-d_i^2/2\sigma^2} \quad (43)
d_i = (x - x_i)^T (x - x_i) \quad (44)

- Widespread Application [36]. Many regression models like Poisson Regression and Ordinal Regression have succeeded in using GRNN. And we can draw a safe conclusion that a complicated network structure can represent any kind of regression. But only a few of them are proved successful. Neural Network is powerful and classic regression algorithms are well-structured. Maybe regression can be applied in neural network without missing its original function. Humans have made fundamental progress in Regression. If we can combine Regression with neural network perfectly, then it’s another picture.

7 Conclusions

In this paper, I set out necessary assumptions with OLS and little tricks to fix the problems when assumptions are violated. Amazingly, it seems that OLS is the beginning of almost every regression model. And a large number of Regression models are designed to be a better OLS. They can play their part in situations where OLS fails to work. I hold the belief that not every algorithm needs to be introduced in details. Hence, the widespread algorithms are given enough attention and others are quickly illustrated. Finally, I give a quick overview of GRNN. From this paper, I can draw three conclusions.

- Know Your Model. Note that regression algorithms aren’t plug-and-play. You must know every model’s range of application and are able to deal with situations where the model’s assumptions are unsatisfied.

- Regression In the Future. Regression is older compared to Deep Learning and great ideas behind every classic algorithm is never out of date. And people always want to predict unknown values and regression task is really fascinating. Deep Learning is quite powerful. If regression can learn from Deep Learning and keeps its excellent part, I do believe regression can be more powerful in the near future.

- It seems that regression algorithms are out of date. However, as far as I am concerned, beautiful ideas behind every algorithm are shared. In other words, dipping into these old algorithms can enable us to gain insight and intuition about algorithms and put forward exciting algorithms which share the same ideas with regression algorithms and are just different implementations of awesome ideas to handle new problems.

References

[1] A.M. Legendre. Nouvelles méthodes pour la détermination des orbites des comètes, Firmin Didot, Paris, 1805. “Sur la Méthode des moindres carrés” appears as an appendix.

[2] C.F. Gauss. Theoria combinationis observationum erroribus minimis obnoxiae

[3] Arthur E. Hoerl & Robert W. Kennard (1970) Ridge Regression: Biased Estimation for Nonorthogonal Problems, Technometrics, 12:1, 55-67, DOI:10.1080/00401706.1970.10488634

[4] Arthur E. Hoerl & Robert W. Kennard (1970) Ridge Regression: Applications to Nonorthogonal Problems, Technometrics, 12:1, 69-82, DOI: 10.1080/00401706.1970.10488635

[5] Tibshirani, R. (1996), Regression Shrinkage and Selection Via the Lasso. Journal of the Royal Statistical Society: Series B (Methodological), 58: 267-288. https://doi.org/10.1111/j.2517-6161.1996.tb02080.x

[6] Zou, H. and Hastie, T. (2005), Regularization and variable selection via the elastic net. Journal of the Royal Statistical Society: Series B (Statistical Methodology), 67: 301-320. https://doi.org/10.1111/j.1467-9868.2005.00503.x

[7] Hayashi, Fumio (2000). Econometrics. Princeton University Press. p. 6161.1996.tb02080.x

[8] Gubner, John A. (2006). Probability and Random Processes for Electrical and Computer Engineers. Cambridge University Press. ISBN 978-0-521-86470-1.

[9] V. Vapnik, “The support vector method of function estimation, “ in J.A.K. Suykens and J. Vandewalle (Eds) Nonlinear Modeling: Advanced Black-Box Techniques, Kluwer Academic Publishers, Boston, pp. 55-85, 1998.

[10] Drucker, Harris; Burges, Christ. C.; Kaufman, Linda; Smola, Alexander J.; and Vapnik, Vladimir N. (1997): “Support Vector Regression Machines”, in Advances in Neural Information Processing Systems 9, NIPS 1996, 155–161, MIT Press.

[11] Smola, A.J., Sch"olkopf, B. A tutorial on support vector regression. Statistics and Computing 14, 199–222 (2004). https://doi.org/10.1023/B:STCO.0000035301.49549.88

[12] S. R. Safavian and D. Landgrebe, “A survey of decision tree classifier methodology,” in IEEE Transactions on Systems, Man, and Cybernetics, vol. 21, no. 3, pp. 660-674, May-June 1991, doi: 10.1109/21.97458.

[13] Loh, W.-Y. (2011), Classification and regression trees. WIREs Data Mining and Knowledge Discovery, 1: 1-23. https://doi.org/10.1002/widm.8

[14] Liaw A, Wiener M. Classification and regression by randomForest(). R news, 2002, (23): 18-22.

[15] Elith J, Leathwick J R, Hastie T. A working guide to boosted regression trees[J]. Journal of Animal Ecology, 2008, 77(4): 802-813.

[16] Efron, Bradley; Hastie, Trevor; Johnstone, Iain; Tibshirani, Robert (2004). “Least Angle Regression” (PDF). Annals of Statistics. 32 (2): pp. 407–499. arXiv:math/0406456. doi:10.1214/009053604000000867. MR 2061016.

[17] Fischler, M. & Bolles, R. ( 1981). Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. Communications of the ACM, 24, 381-395.

[18] Theil, H. (1950), “A rank-invariant method of linear and polynomial regression analysis. I. I. II.”, Nederl. Akad. Wetensch., Proc., 53: 386–392, 521–525, 1397–1412, MR 0036489

[19] Huber, Peter J. (1964). “Robust Estimation of a Location Parameter”. Annals of Statistics. 2 (1): 73–101. doi:10.1214/aoms/1177703732. JSTOR 2238020.

[20] Friedman J H. Multivariate adaptive regression splines[J]. The annals of statistics, 1991: 1-67.

[21] Stigler, Stephen M. (November 1974). “Gergonne’s 1815 paper on the design and analysis of polynomial regression experiments”. Historia Mathematica. 1 (4): 431–439. doi:10.1016/0315-0860(74)90033-0.

[22] Ruppert D, Wand M P. Multivariate locally weighted least squares regression[J]. The annals of statistics, 1994: 1346-1370.

[23] Suykens J A K, De Brabanter J, Lukas L, et al. Weighted least squares support vector machines: robustness and sparse approximations[J]. Neurocomputing, 2002, 48(1-4): 85-105.

[24] Amemiya, Takeshi (1985). “Generalized Least Squares Theory”. Advanced Econometrics. Harvard University Press. ISBN 0-674-00560-0.

[25] Kmenta, Jan (1986). “Generalized Linear Regression Model and Its Applications”. Elements of Econometrics (Second ed.). New York: Macmillan. pp. 607–650. ISBN 0-07-02886-7

[26] Kariya T, Kurata H. Generalized least squares[M]. John Wiley & Sons, 2004.

[27] Bernardo J M, Smith A F M. Bayesian theory[M]. John Wiley & Sons, 2009.
