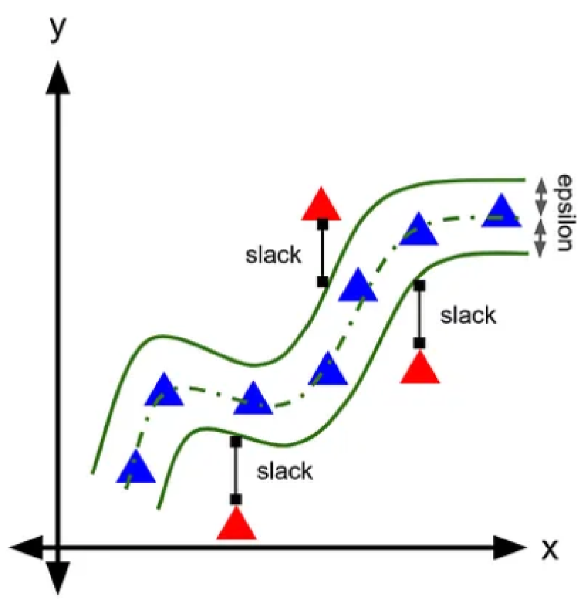
Effects of Different Kernels in Support Vector Machine Regression

By: Rohith Jasthi (ID - 23007975)

Let's start with the Introduction,

What does SVM Regression mean?

Support Vector Machine regression belongs to the supervised-learning algorithms domain and is critical in applying to regression problems. Regression does not require such a condition where the technique searches for the hyperplane (or a curve in the higher dimensions), which will fit well with the data points but, at the same time helps in maintaining tolerance for errors. It aims to minimize prediction errors subject to the condition that most of the data is bounded within a certain distance from the hyperplane.



SVM Regression

Important aspects of SVM regression are:

- It works by identifying support vectors that define the boundaries of the model.
- SVM Regression does not minimize error directly; rather, it minimizes error within specific boundaries bounded by the margin epsilon (ε).
- It is strong for outliers since the algorithm does not consider all data points but considers only support vectors.

- SVM regression is best applicable in cases in which the relationship among the variables is non-linear or complex.

Now move to the theory of SVM regression,

Theory

Overview of SVM Regression

SVM regression works by finding a function f(x), which predicts the target variable given input x, keeping the predictions within a margin of tolerance (epsilon).

1. Epsilon-Insensitive Loss:
    a. Errors are penalized only if they exceed some level, specifically ϵ.
    b. The margin is then defined to capture all predictions falling within this margin as being correct.

2. Objective: To minimize the following cost function,

$$1/2||w||^2 + C\Sigma(Loss\ beyond\ margin)$$

   Where w represents the model weight and C is the regularization parameter which controls the tradeoff between margin width and tolerance for the errors.

3. Support Vectors: Only data points outside the boundary can affect the model of the SVM.

Mathematics Intuition Which Lies Behind Kernel Functions:

Kernel functions are mathematical instruments intended for measuring the similarity between two given data points in machine learning. Algorithms such as Support Vector Machines (SVMs) furnish their operation in a higher-dimensional space without explicitly transforming the data. This method is referred to as the kernel trick.

What Actually Kernel Does?

Imagine two data points in a simple 2D space, each represented by a vector (a list of numbers). You may want to transform them to a higher dimension (3D or beyond) to make classification easier. Transformations of this sort usually cost far too much computational effort.

Kernel would not deal with transformation; instead, it would compute the dot product in this higher-dimensional space without the transformation. It is like indirectly accessing the properties of this new space while staying in the original space.

How does the kernel compute the similarity?

Let's consider an example - the RBF (Radial Basis Function) kernel which is formulated as:

$$K(x,x') = \exp(-\gamma \|x-x'\|^2)$$

Now the declarations in that equation:

1. x and x′: Two feature vectors (data points).
   Example: Let x=[1,2] and x′=[3,4]. These two points can describe any point in 2-dimensional space.

2. $\| x-x' \|$: The distance between the two points x and x′, that is the distance between the other.
   - If the two points lie near each other, this distance will be small.
   - And when there is a large distance between these points, then this distance will be large.

3. $\exp(-\gamma\|x-x'\|^2)$: The exponential function exp(x) will decay toward zero quite rapidly when x is regarded as becoming very negative.
   In other words, this means:
   - If they've been close to each other, the value: distance $\| x-x' \|$ would be small thus the value of the exponent would be high (close to 1).
   - In case x and x′ are quite remote from one another, the value of the exponent turns small (close to 0).

4. γ: It is a scaling factor inappropriate for the kernel for distances. The bigger the γ, the more sensitive the kernel in the distances.
   - A Large value of γ leads to narrow regions of interest inside the kernel.
   - Not-so-large γ would be such that values of dissimilarity become more strongly yielding positive values, rather than indicating similarity.

Kernel Types:

1. **Linear Kernel:**
   - Models data which is linearly separable.
   - Equations: $K(x,x') = x \cdot x'$, (Dot product of two feature vectors).
   - A simple and computationally efficient but not very suitable for non-linear data.

2. Polynomial Kernel:

   - It describes non-linear relationships by polynomial transformations.
   - Equations: $K(x,x') = (x \cdot x'+c)d$, where d is the degree of the polynomial and c is a constant.
   - It suits the interactions between features, which can be written in polynomial terms.

3. Radial Basis Function (RBF) Kernel:

   - One of the most popular applications is non-linearity.
   - Equations: $K(x,x') = \exp(-\gamma \|x-x'\|2)$, where γ is a parameter that controls the impact of individual data points.
   - Infinite dimensional mapping in most extraordinary cases makes an equation powerful about complex relationships.

4. Sigmoid Kernel:

   - Approximates the behavior shown by neural networks.
   - $K(x,x') = \tanh(\gamma x \cdot x' + c)$; γ and c are parameters.
   - Used less but very successful in specific problems.

Now let's see the suitable conditions for each kernel function:

When to Use What Kernel: Trade-off and Use Cases

Linear Kernel:

- Use linear separation if the data can be completely or almost linearly separated.
- Most efficient for high-dimensional sparse data (like text data).
- Example: Basic financial predictions or sentiment analyses.

Polynomial Kernel:

- Effective for datasets having little or no relationship within the features or their degree have the interaction.
- High degree polynomial tends to overfit, thus careful tuning of parameters is required.
- Example: Problems with moderate non-linear dependence, such as stock price prediction with polynomial trends.

RBF Kernel:

- Powerfully complex and typically approaches non-linear problems.
- Most flexible in understanding relations in the infinite-dimensional space.
- Example: Image recognition, forecasting the weather.

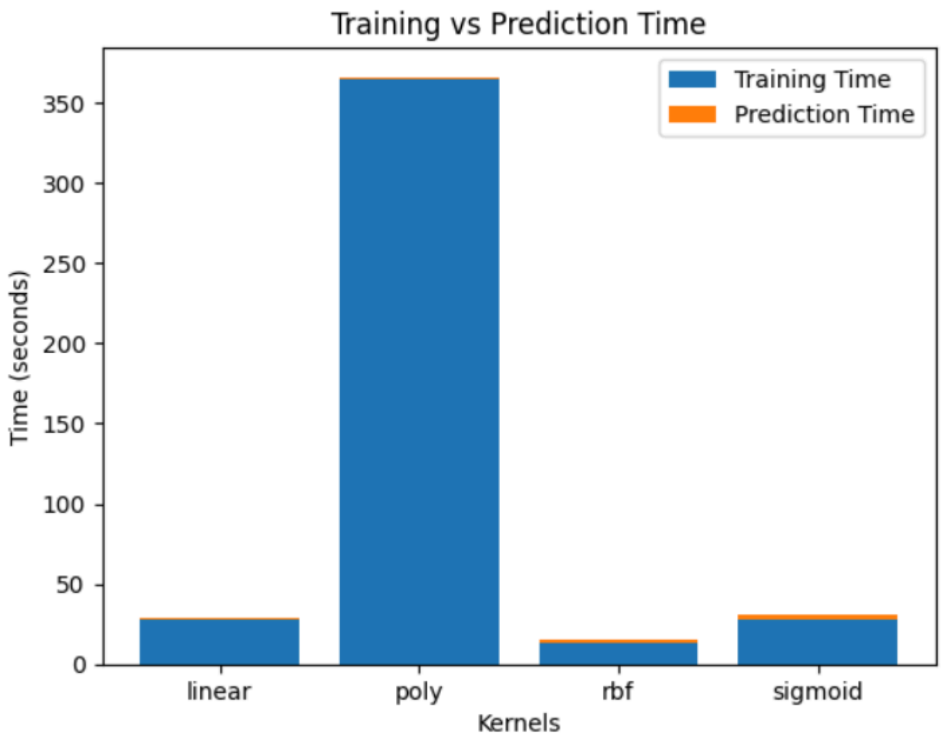
Sigmoid Kernel:

- Chooses a more appropriate application where the data settings closely resemble that of behavior in neural networks.
- Requires rigorous tuning of parameters γ and c.
- Example: Problems involving probabilities or thresholds.

By understanding SVM regression and its kernel functions, practitioners can choose the appropriate methods for their datasets so that they get the most out of their predictive modeling effort.

Comparing Kernel Performances in SVM Regression:

On comparing the kernels over the California Housing Dataset over the parameters Train and Test duration, we got the following result,



- Long training time indicates polynomial kernel complexity, which is beneficial for very highly complex data sources. However, sometimes this kernel would be overkill or inefficient in the case of less complex data.

- In addition, the training times for linear and RBF kernels are lower, which suggests they are far better suited to datasets in which linearity or moderate nonlinearity would suffice for modeling.

- The measures should be dependent on the desired balance between accuracy and efficiency, for instance: When the performance of the polynomial kernel is only marginally better than an RBF or a sigmoid, the last two would be preferable for the fact that they train faster.

- When time is really of the essence (in real-time applications), then the Rbf kernel would probably be the best option. But does the speed have any effect on the accuracy?

The following table given below will explain the accuracy by using the parameters such as Mean Squared Error and R Squared Error,

Kernel	MSE	R^2 Error
Linear	0.5792	0.5580
Polynomial	1.0048	0.2332
RBF	0.3570	0.7276
Sigmoid	15252.4545	-11638.4612

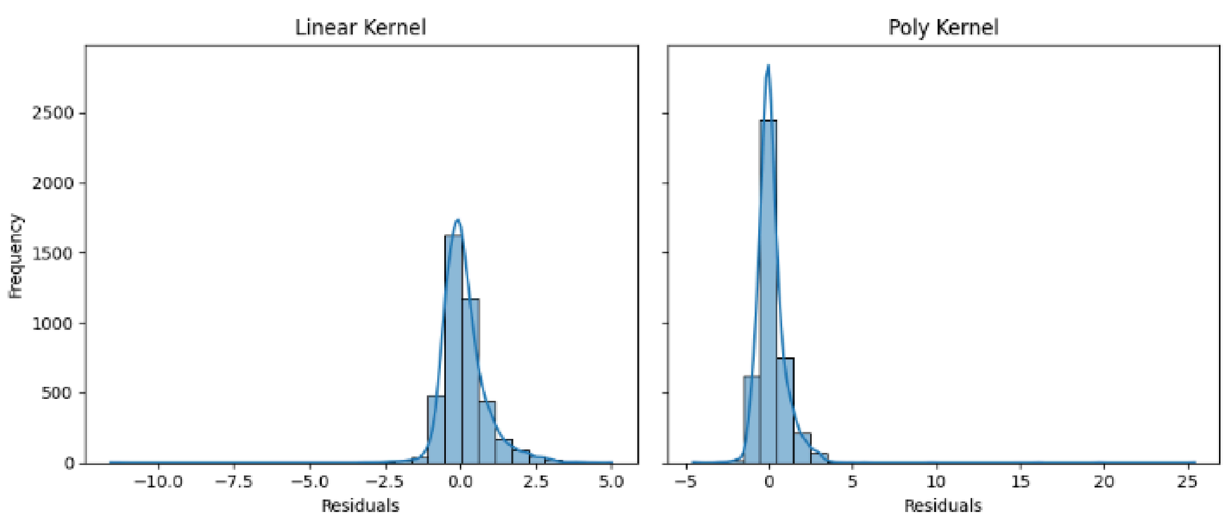
Best Performer: sure enough, top performer. The RBF kernel stands out as performing the best, scoring the lowest MSE, the highest R^2, and then making it to the fast training time.

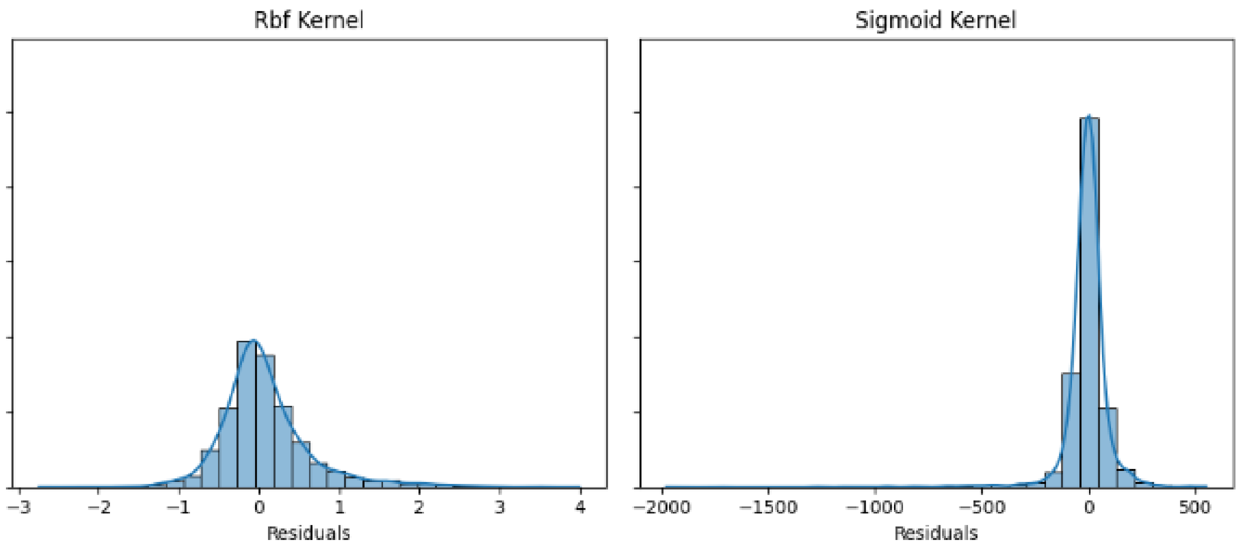
Underperformer-sigmoid kernel: Among these, this kernel was put to be an inappropriate kernel for the dataset, as it exhibits a very high error and negative value for R^2.

Efficiency Issues: The polynomial kernel takes a long for train and performs pretty poorly, thus not making it of much use.

Linear Kernel: A good approach for very simple, linear connections, but not so good for those that are nonlinear by nature.

The model behaves differently over the residuals distribution over and these can be visualized in the below diagram,

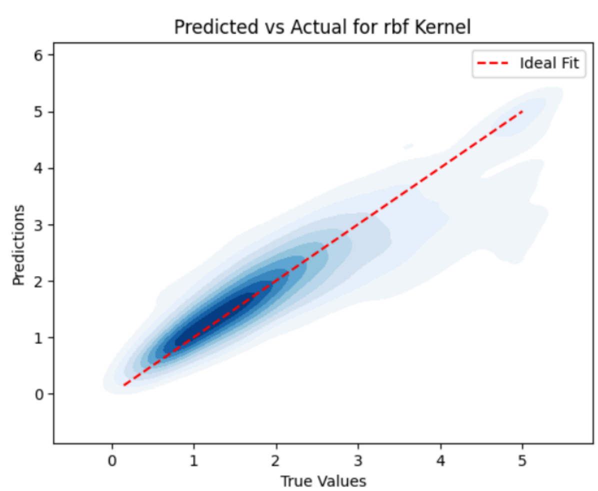


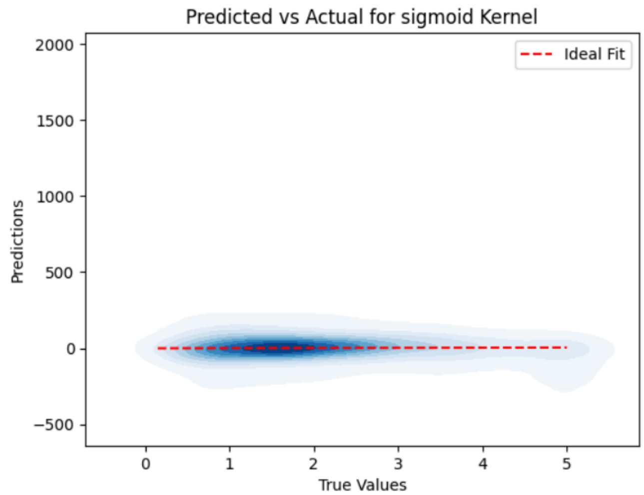


From these graphs, we can conclude:

When it comes to the distribution of residuals, the RBF kernel have the best profile, with a very narrow spread and very well-centered residuals. The Polynomial kernel does fairly well, but as in some cases, it suffers from either overfitting or high influence values. The Linear kernel is not too bad, but depending on the degree to which the dataset is linear, it may or may not be applicable. The Sigmoid kernel has demonstrated very poor performance, with its large residuals being indicative of its incapacity to capture the actual patterns of the data.

Here it is evident that for this data the kernel which is best is RBF and the worst is sigmoid, this can be shown over the prediction accuracy while comparing the actual and predicted values.





In terms of Sigmoid Kernel, prediction mostly portrayed values hugging the line of zero with some values straying away at both poles of the positive-negative direction. This would imply that the kernel fails to reflect the relationship between the features and the target variable.
The ideal case where predictions equal true values is shown with the red dashed line, which indicates the level of deviation as given by the predictions. This is a characteristic of poor model performance with a lack of correlation.

But, there is a clear case where rbf kernel scatter density of predictions follows as closely as possible the red dashed line, suggested as the ideal fit, which through this indicates brilliant effectiveness of the RBF kernel in capturing the relationship between true values and predicted values.Most points cluster closely to the diagonal line; they show very high correlation and very low residual errors and thus suggest that the model is a highly accurate one.

To conclude, kernel functions in SVM regression are very much determined by the nature of data and the problem it is solving. For instance, Sigmoid kernel was not good enough under these cases; however, that does not necessarily mean that it is bad for other tasks-it might perform well under other scenarios. Similar is the case with all other kernels like RBF or Linear, which also vary with data complexity and feature distribution as well as more complicated overheads. This is what this tutorial tries to bring out- familiarity with your data and experimentation with diverse kernels, so you can select the right one appropriate to your own task.

Check out the References used in this tutorial,

[1] Smola, Alex J., and Bernhard Schölkopf. "A tutorial on support vector regression." *Statistics and computing* 14 (2004): 199-222.

[2] Cristianini, Nello. *An introduction to support vector machines and other kernel-based learning methods*. Cambridge University Press, 2000.

[3] Scholkopf, Bernhard, and Alexander J. Smola. *Learning with kernels: support vector machines, regularization, optimization, and beyond*. MIT press, 2018.

For Code Check out the Github Link: https://github.com/Rohith23007975/SVR_Tutorial.git