Non-Linear Regression: comparison between P-Splines and SVM Regression Using Sine-Exponential Simulated Data

First Author and Second Author

Abstract
DO NOT FORGET TO INCLUDE SECTION HERE!!!

KEY WORDS: non-linear regression; GAM; univariate analysis; P-Splines; SVM Regression; simulation study

1 INTRODUCTION

There has been a significant development in non-linear regression techniques due to the advent of modern high-speed computing. This new family of techniques, often originating from machine learning and computer science environments directly compete with more traditional statistics techniques whose theory predates the contemporary machine learning algorithm.

This simulation study aims to compare two such techniques: Splines which have origins in 20th century, and the Support Vector Machines that became a technique-of-choice in the early 21st century before neural networks became more viable option in recent years.

1.1 Splines

Splines were first proposed in 1946 by Schoenberg who provided mathematical theory for piecewise polynomial functions that are connected via knots [1]. This was later improved by the introduction of smoothing splines in 1967 by Reinsch for balancing smoothness with data fitting [2].

However, the previous approaches relied on subjective parameter selection. The key development came with the introduction of generalised cross-validation in 1978 [3]. Smoothing spline theory as well as cubic splines were then formalused by Wahba in 1990 [4].

At the same time, there was a development in an alternative approach using basis splines (B-splines) that, due to them using local rather than global support, are computationally more efficient [5]. However, B-splines require manual knot placement and selection which makes their use more subjective and prone to errors in practice.

The key development that brought splines to prominance was Eilers and Marx's introduction of penalised B-splines (P-splines) in 1996 [6]. The technique combines computational efficiency of B-splines with the adaptive smoothing penalty techniques. Due to its use of large number of equally-spced knots. The penalty is then applied between adjacing B-spline coefficients. This approach balances smoothness and fit through a single tuning parameter and avoids a subjective choice of knot locations. However, the number of appropriate knots still remains subjective with larger number of knots may still lead to overfitting, albeit much less pronounced compared to B-splines.

The contemporary developments include the introduction of both frequentist [7] and Bayesian [8] uncertainty quantification which allowed for scientific applcation of splines. There has also been significant development in other fields, such as multivariate extensions and functional data analysis [8]

The univariate P-spline model uses the following two equations:

$$f(x) = \sum_{j=1}^{n} \beta_j B_j(x) \tag{1}$$

where $B_j(x)$ are B-spline basis functions; β_j are coefficients; and n is the number of basis functions.

The above is subject to the following minimisation contraint for coefficient estimation:

$$\sum_{i=1}^{m}(y_i - f(x_i))^2 + \lambda \sum_{j=d+1}^{n} (\Delta^d \beta_j)^2 \tag{2}$$

where the first term measures fit to the data; the second term is a penalty on differences between adjacent coefficients; λ is the smoothing parameter; and Δ^d represents the d order difference operator.

1.2 Support Vector Reggresion

Vapnik [9] proposed the Support Vector Regression (SVR) framework by extending Support Vector Machine (SVM) to regression models. SVR constructs a regression function based on a subset of the data, known as support vectors. In its original form, Vapnik introduced the following representation:

$$F_2(x, \mathbf{w}) = \sum_{i=1}^{N} (\alpha_i^* - \alpha_i)(v_i^\top x + 1)^p + b \tag{3}$$

Here, α_i, α_i^* are Lagrange multipliers obtained from the dual optimization problem, v_i represents the data samples, p is the degree of the polynomial kernel, and the term $(v_i^\top x + 1)^p$ corresponds to the kernel function. This formula use a specific polinomial kernel.

The SVR model can be more generalized replacing the polynomial kernel with any positive-definite kernel function $K(x_i, x)$, leading to the more flexible representation:

$$f(x) = \sum_{i=1}^{N} (\alpha_i^* - \alpha_i) K(x_i, x) + b \tag{4}$$

This form allows SVR to capture complex, non-linear relationships in data. Commonly used kernels include the radial basis function, linear and sigmoid kernels, making SVR a powerful and adaptable tool for regression tasks involving non-linear data structures.

1.2.1 Loss Function and Hyperparameters

The loss function utilized in this model is the ε-insensitive loss function, first introduced by Vapnik [9]. This loss function defines a tube of tolerance around the regression function, within which errors are not penalized. The idea is to focus on fitting the data while ignoring small deviations that are likely due to noise (just like the svm do to separed classes). The loss function L_ε is defined as:

$$L_\varepsilon(y, f(x)) = \begin{cases} 0, & \text{if } |y - f(x)| \le \varepsilon \\ |y - f(x)| - \varepsilon, & \text{otherwise} \end{cases} \tag{5}$$

Only data points that lie outside the ε-tube contribute to the cost function, making the solution sparse in terms of support vectors [10].

The Hyperparameters that are involve in this model are three and they are use to control the flexibility and generalization of the model:

- **Regularization parameter** C: This controls the bias-variance trade-off of the model. A high C encourages the model to minimize errors at the risk of overfitting, while a small C yields a smoother function that may tolerate larger errors.
- **Epsilon** ε: Determines the width of the tube around the regression function where no penalty is applied. Larger values of ε simplify the model by ignoring small deviations, while smaller values increase sensitivity to minor variations.
- **Kernel function** $K(x_i, x_j)$: Enables SVR to perform non-linear regression by implicitly mapping the input data into a higher-dimensional feature space. Common kernel choices include:
  - *Linear*: $K(x_i, x_j) = x_i^\top x_j$
  - *Polynomial*: $K(x_i, x_j) = (\gamma x_i^\top x_j + r)^d$
  - *Radial Basis Function (RBF)*: $K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2)$
  - *Sigmoid*: $K(x_i, x_j) = \tanh(\gamma x_i^\top x_j + r)$

As noted by Hastie *et al.* [11], the choice and tuning of these hyperparameters are crucial for model performance and are selected through cross-validation techniques.

2 SIMULATION STUDY

This section discusses the specific apporach that was taken for the simulation study, decisions taken regarding each implementation, and presentation of final comparison results:

2.1 Methodology

2.1.1 Data Simulation

To evaluate the performance, sine-exponential function was used to simulate data. This nonlinear function combines periodic oscillations with exponential growth, creating a challenging test case for nonparametric regression techniques.

The underlying deterministic function is defined as:

$$f(x) = \sin(x) \cdot e^{\alpha x}$$

where $\alpha = 0.2$ represents the exponential growth rate parameter. The complete data generating process is formalised as:

$$y_i = \sin(x_i) \cdot e^{0.2x_i} + \varepsilon_i, \quad i = 1, \ldots, n$$

with the following specifications:$x_i \sim \mathcal{U}(0, 6\pi)$ (*random sampling specification*; $\varepsilon_i \sim \mathcal{N}(0, \sigma^2)$, where $\sigma = 10$ represents the standard deviation of Gaussian noise; and $n = 1000$ represents the total number of observations.

2.1.2 P-Spline Implementation

P-spline model was implemented using **mgcv** package that is from Wood [12] and is still activelly developed. The package includes automatic GCV which was used for smoothing parameter tuning. According to Woods [8] second-order difference penalty is best choice for fitting a data with oscillating pattern and therefore was used in our model specification.

However, the choice of number of knots is still subjective and relative to data. Based on graphical analysis, the appropriate number of knots was determined to be 40. Despite the overlapping with its double, 40 was chosen due to its reduced complexity. However, the effective degrees of freedom (EDF) is 16.159 meaning the number of knots was effectively reduced to prevent overfitting which suggest the penalisation function works as intended.

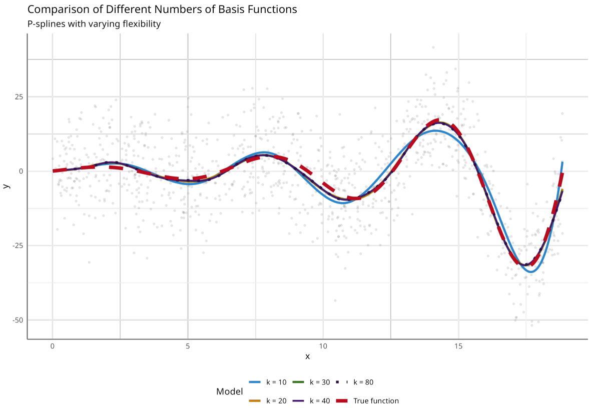

Figure 1: Comparison of different numbers of basis functions

Finally, residual plot was analysed to determine as well as the residual distribution. From these plots it was determined that residuals are univorm, homoscedastic with approximately normal distribution with Anderson-Darling Normality tests p-value being 0.114 which does not reject the normality hypothesis. The model explained 53.4% of variance.

Overall, the model succesfully captured the non-linear relationwhip with the smoot term being highly significant (p-value <0.001; F = 60.11).

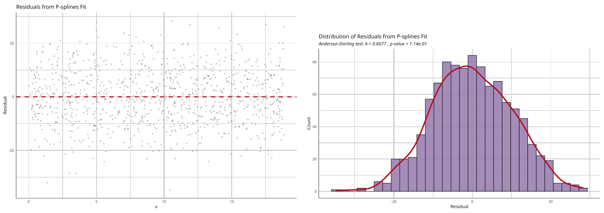

Figure 2: Left image shows residual plot and right their distribution

2.1.3 SVR Implementation

Support Vector Regression was implemented using the **e1071** package, which interfaces with *libsvm*. Two versions of the SVR model were explored: a base model using default parameters and a tuned model optimized using grid search.

This models were trained on scaled input data to avoid issues related to kernel sensitivity to variable magnitude.The base SVR model was trained using the radial basis function (RBF) kernel. A second, tuned SVR model was fitted by optimizing the ϵ-insensitive margin and the regularization parameter C, with a search grid defined as $\epsilon \in [0, 1]$ and $C \in \{2^{-1}, 2^0, \ldots, 2^7\}$. The best model was selected based on RMSE.

Prediction accuracy was evaluated by comparing the fitted values to true values using a visual overlay (Figure 4) and RMSE scores across the linear model, base SVR, and tuned SVR (Table 1). The SVR tuned model clearly outperformed both the base SVR and linear regression models.

Model	**RMSE**
Linear Regression	14.350
SVR Base	11.720
SVR Tuned	10.220

Table 1: Root Mean Squared Error (RMSE) comparison across models.

Residual analysis was performed to assess model performance beyond accuracy. The residuals from the tuned SVR model were plotted against fitted values and examined through histogram and density analysis (Figure 4). The residuals appeared randomly scattered around zero

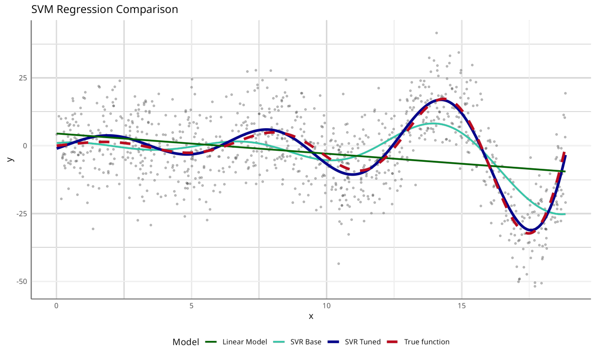

Figure 3: Overlay of predicted vs actual values for Linear Regression, SVR Base, SVR Tuned models, and Actual Model

and approximately normally distributed, supported by the Anderson-Darling test with a p-value of `0.076`, indicating no strong evidence against normality.

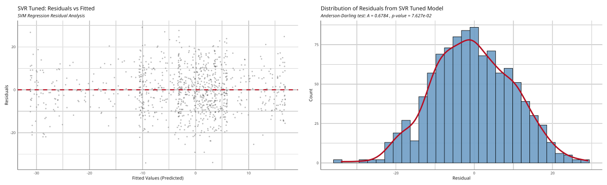

Figure 4: Left: Residuals vs Fitted plot. Right: Residual distribution with Anderson-Darling test result.

2.1.4 Prediction Testing Method

Firstly, a sample fo 200 new data was generated as a testing data set, meaning this data was not used for training of any models. We then made predictions using our models. Prediction interval(95%) was calculated for both P-splines and SVMwhich was later used for calculating Prediction Interval Coverage Probability(PICP).

For P-splines, the Wood's package was used to calculate fit and standard error mmetrics. For SVM, a simulation method had to be used to achieve prediction interval. While using simulation on top of simulation reduces credibility of this aproach, this is the best solution given the lack of interval theory in SVM implementation. The number of bootstrap iteration was set to 1000 to calculate confidence intervals and later prediction intervals.

2.2 Results

The P-spline results suggest moderate prediction accuracy(RMSE = 10.178). The lower MAE(8.241) suggest there are not many extreme outliers with 54% of explained variance.

Table 2: Goodness-of-fit and Prediction

Model	RMSE	MAE	R-Sq
P-spline	10.177	8.241	0.534
SVM Tuned	10.220	8.277	0.530
Model	**Pred-RMSE**	**PICP**	**Av. Interval Width**
P-spline	10.024	0.97	40.553
Svm Tuned	10.034	0.97	40.267

3 CONCLUSION

REFERENCES

[1] I. J. Schoenberg, "Contributions to the problem of approximation of equidistant data by analytic functions. part b. on the problem of osculatory interpolation. a second class of analytic approximation formulae," *Quarterly of Applied Mathematics*, vol. 4, pp. 112–141, 1946. [Online]. Available: https://api.semanticscholar.org/CorpusID:125667988

[2] C. H. Reinsch, "Smoothing by spline functions," *Numerische Mathematik*, vol. 10, pp. 177–183, 1967. [Online]. Available: https://api.semanticscholar.org/CorpusID:26555189

[3] C. P. Wahba, G., "Smoothing noisy data with spline functions. estimating the correct degree of smoothing by the method of generalized cross-validation." *Numerische Mathematik*, vol. 31, pp. 377–404, 1978/79. [Online]. Available: http://eudml.org/doc/132586

[4] G. Wahba, *Spline Models for Observational Data*. Society for Industrial and Applied Mathematics, 1990. [Online]. Available: https://epubs.siam.org/doi/abs/10.1137/1.9781611970128

[5] C. de Boor, *A Practical Guide to Splines*, 1st ed., ser. Applied Mathematical Sciences. New York, NY: Springer-Verlag, 1978, vol. 27, published: 29 November 2001, Series ISSN: 0066-5452, Series E-ISSN: 2196-968X.

[6] P. H. C. Eilers and B. D. Marx, "Flexible smoothing with *b*-splines and penalties," *Statistical Science*, vol. 11, no. 2, pp. 89–102, 1996. [Online]. Available: http://www.jstor.org/stable/2246049

[7] T. Krivobokova, T. Kneib, and G. Claeskens, "Simultaneous confidence bands for penalized spline estimators," *Journal of the American Statistical Association*, vol. 105, pp. 852 – 863, 2009. [Online]. Available: https://api.semanticscholar.org/CorpusID:59487175

[8] S. N. Wood, *Generalized Additive Models: An Introduction with R*, 2nd ed. New York: Chapman and Hall/CRC, 2017, eBook Published: 18 May 2017.

[9] V. N. Vapnik, *The Nature of Statistical Learning Theory*. New York: Springer, 1995.

[10] A. J. Smola and B. Schölkopf, "A tutorial on support vector regression," NeuroCOLT2 Technical Report Series, Tech. Rep. NC2-TR-1998-030, 1998. [Online]. Available: https://alex.smola.org/papers/1998/SmoSch98.pdf

[11] T. Hastie, R. Tibshirani, and J. Friedman, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed. New York: Springer, 2009.

[12] S. Wood, *mgcv: Mixed GAM Computation Vehicle with Automatic Smoothness Estimation*, R Foundation for Statistical Computing, 2025, r package version 1.9-3. [Online]. Available: https://CRAN.R-project.org/package=mgcv

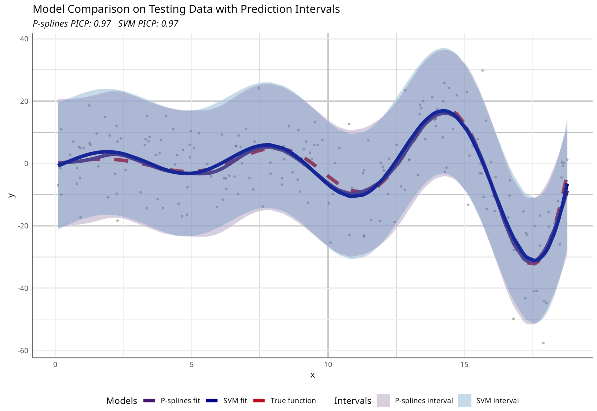

Figure 5: Comparison predictive power between P-splines SVM Tuned models