Support Vector Machines for Regression and Novelty Detection

Keith H. Bova

Abstract

This paper evaluates three regression models. We first construct a linear ridge regression model that makes a prediction y on some matrix X. We then compare this model to an ϵ SVR and a ν SVR.

Index Terms

Ridge regression, SVM, SVR, ϵ SVR, ν SVR.

I. Introduction

REGRESSION refers to the estimation of random variables using a set of observations. If we exist inside a space $\mathbb{R}^n$, we can approximate a distribution of points–provided our approximation follows the same distribution as the original. For example, if we have a distribution we know is gaussian, and the distribution follows what appears to be a linear path in $\mathbb{R}^2$, we can approximate that path with a linear function. If we consider a distribution with an infinitely large margin, we can apply the SVM classifier where the points act practically as the slack variables ξ_i. Vapnik defines this process using an alternate form of the representer theorem, sometimes referred to as the epsilon-insensitive loss function [1].

II. The Data for Our Experiments

Our data is represented as a matrix in CSV format. The first column of the file represents our testing regressors without noise; likewise, the second column represents our training regressors–also without noise. Our matrices $\underline{\underline{\mathbf{X}}}_{test}$ and $\underline{\underline{\mathbf{X}}}_{train}$ come from columns 3-21 and 22-40 respectively. Our vectors $\underline{\mathbf{y}}_{test}$ and $\underline{\mathbf{y}}_{train}$ come from columns 41 and 42, respectively. For each experiment, we split the data 80:20 to validate that our predictions follow approximately the same distribution at the original.

III. Comparing Regression Algorithms

The following three experiments were conducted in Python. For each experiment, the program reads from a CSV using the Pandas library, performs regression, and examines the error by comparing the results of the predictions with the actual results from the CSV.

A. Linear Ridge Regression

This first model uses the methods of *Tikhonov Regularization* to apply a regression using MMSE. First, we define a vector we call γ that contains 1000 logarithmically spaced elements, such that:

$$\gamma_i \in [-1, 2] \tag{1}$$

For each value of γ, we perform ridge regression and calculate the MMSE. A plot of the validation square error as a function of γ is shown below.

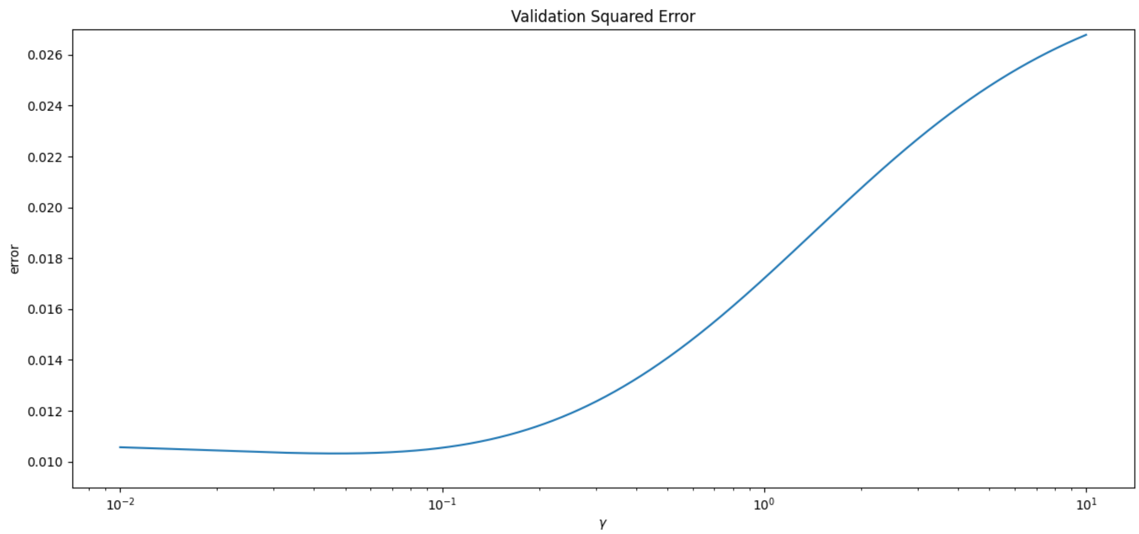

Fig. 1. Validation square error for linear ridge regression as a function of γ

After choosing the optimal value for γ (in our case, $\gamma \approx 0.047$), we apply ridge regression once more, and compare our predictions in this model with the actual value stored in the CSV.

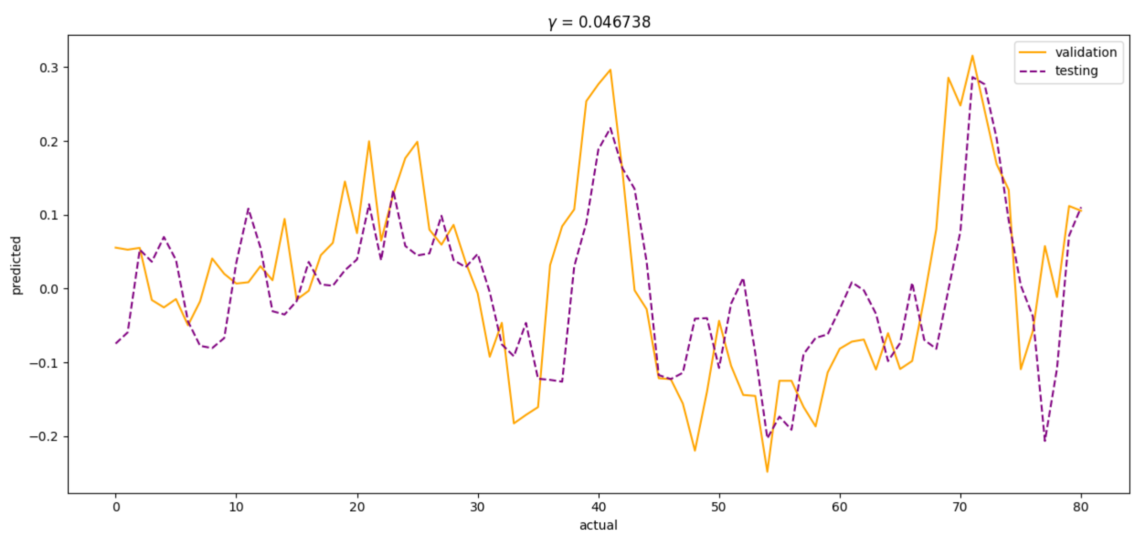

Fig. 2. Linear ridge regression distribution using γ = 0.46738

B. The ϵ SVR

The ϵ SVR is similar to linear ridge regression; however, this method uses cross-validation to determine an optimal value for ϵ–a free parameter in the SVR that represents a positive error tolerance. For this model, we represent ϵ and C as vectors of 100 logarithmically spaced elements such that:

$$\epsilon_i \in [-4, 1] \tag{2}$$

$$C_i \in [-2, 2] \tag{3}$$

After calculating the mean square error for all the possible combinations of C and ϵ, we pass the results into a contour plot to better visualize the gradient.

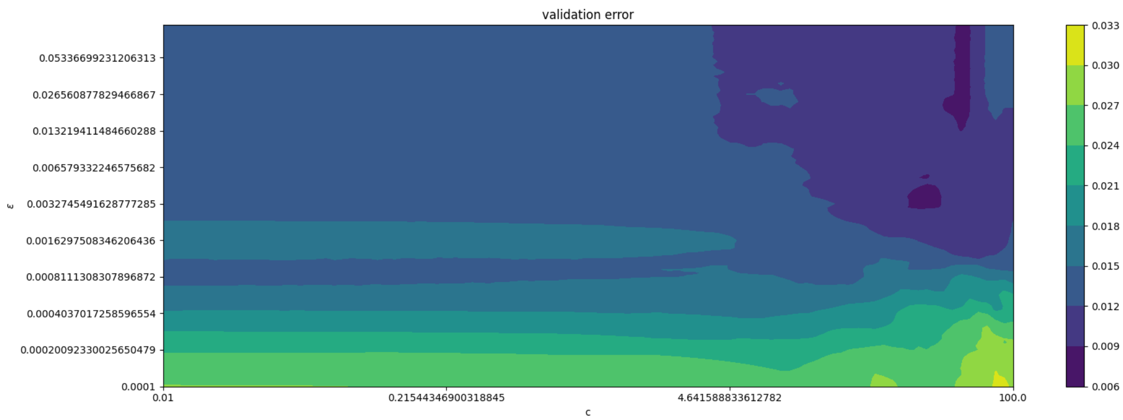

Fig. 3. MMSE for the ϵ SVR

We find that our minimum value occurs when $\epsilon \approx 0.498$ and $C \approx 1.050$. After setting these values as static and plotting the training data and the testing data, we see the following results.

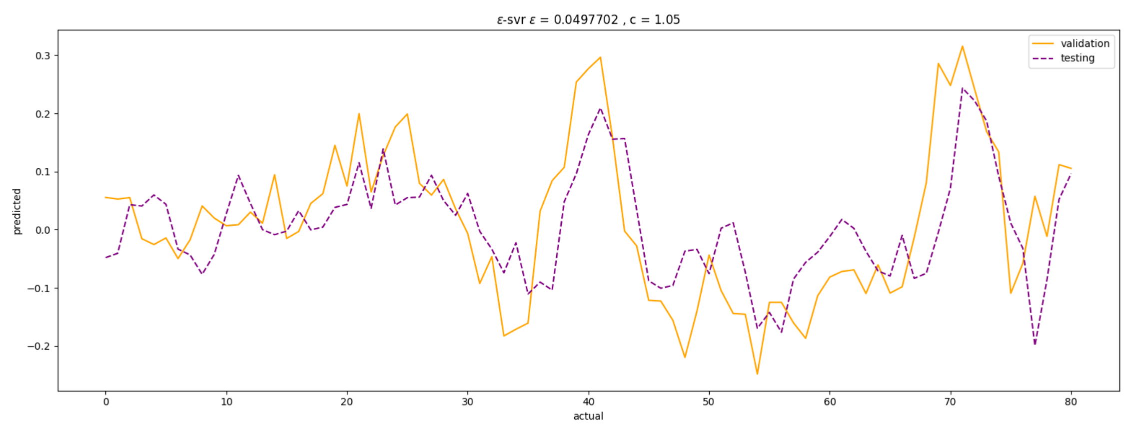

Fig. 4. ϵ SVR distribution using ϵ = 0.498 and C = 1.05

C. The ν SVR

The ν-SVR is similar to the ϵ-SVR; however, the ν-SVR tunes ϵ by applying the method described by Smola and Scholkopf [1]. For the ν-SVR, we represent C and ν as 100 logarithmically spaced vectors, such that:

$$\nu_i \in [-3, 0] \tag{4}$$

$$C_i \in [-2, 2] \tag{5}$$

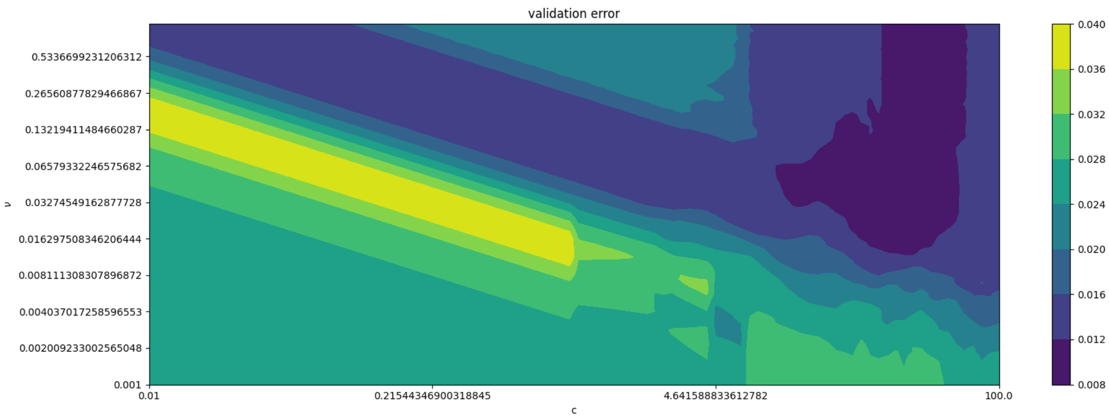

Fig. 5. MMSE for the ν SVR

Similar to the method in the ϵ-SVR, we use a contour plot of the values for ν and C so that we can find their optimal values using inspection. For our training data, we find an optimal value where $\nu \approx 0.572$ and $C \approx 1.05$. We plot the regressors again below.

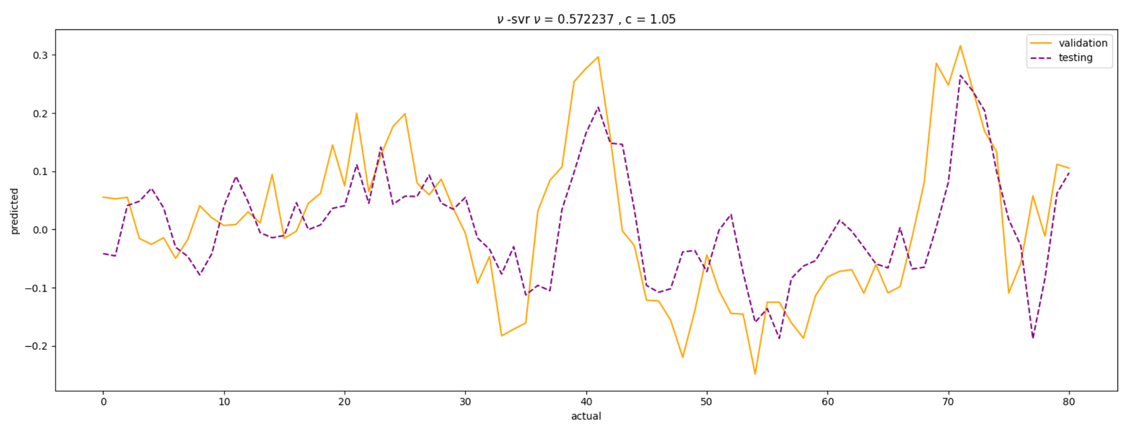

Fig. 6. ν SVR distribution using ν = 0.572 and C = 1.05

IV. Conclusion

From the following experiments we see that for this particular set of data, the three SVR algorithms produce similar regressors; however, the ν-SVR and the ϵ-SVR differ in the number of support vectors and complexity of the learning task.

V. Source Code

https://github.com/keithhbova/support_vector_machines/

References

[1] *A Tutorial on Support Vector Regression*. Alex J. Smola and Bernhard Scholkopf [Online]. Available: https://alex.smola.org/papers/2003/SmoSch03b.pdf