Support Vector Regression (SVR)

Descriptions of SVR in this discussion follow that in Refs. (2, 6, 7, 8, 9). The literature suggests the design variables should be normalized to a range of [-1,1] or [0,1]. Simply, SVR constructs a hyperplane that passes near each design point such that they fall within a specified distance of the hyperplane. In two dimensions, this hyperplane is simply a line. The hyperplane is then used to predict other responses. SVR estimates the real function as

$$y = r(x) + \delta \tag{2.28}$$

where δ is an independent random noise, *x* is the multivariable input, *y* is the scalar output, and *r* is the mean of the conditional probability (regression function). See Cherkassky and Ma[6] and Gunn[7] for more information. SVR technique selects the "best" approximate model from a group of selection models that minimize the prediction risk. Linear or nonlinear regression can be performed. When a linear regression is used, the pool of approximation models is given by

$$\hat{f}(x) = \langle \omega \cdot x \rangle + \mathrm{b} \tag{2.29}$$

where b is the bias term and $\langle \omega \cdot x \rangle$ is the dot product of ω and x. Minimizing empirical risk using the ε-insensitive loss function allows regression estimates. It is desirable to have a "flat" approximation function and this is achieved by minimizing $|\omega|^2$. Non-negative slack variables are introduced to account for training points that fall outside of the ε-insensitive zone. That is:

$$\begin{gathered} minimize\ \frac{1}{2}|\omega|^2 + C\sum_{i=1}^{N}(\xi_i + \xi_i^{*}) \\ s.t. \begin{cases} y_i - \langle \omega \cdot x_i \rangle - b \le \varepsilon + \xi_i^{*} \\ \langle \omega \cdot x_i \rangle + b - y_i \le \varepsilon + \xi_i \\ \xi_i^{*}, \xi_i \ge 0 \end{cases} \end{gathered} \tag{2.30}$$

where *C* is a positive constant, and *ε* is the insensitive zone, both are chosen by the user. *C* is also referred to as the regression parameter or penalty parameter. Cherkassky and Ma[6] propose *C* be chosen as

$$C = \max(|\mu_Y + 3\sigma_Y|, |\mu_Y - 3\sigma_Y|) \tag{2.31}$$

where μ_Y and σ_Y are the mean and standard deviation of the training point responses. Hsu et al.[8] suggest a cross-validation approach to find *C*.

The parameter ε determines the width of the ε-insensitive zone and affects the complexity/flatness of the model. The values of ε should be tuned to the input data, but a reasonable starting value is found using

$$\varepsilon = \frac{c}{100}|\tilde{y}_{max} - \tilde{y}_{min}| \tag{2.32}$$

with c = 1 where $|\tilde{y}_{max} - \tilde{y}_{min}|$ is the range of the responses at the training points. The value of *c* can be tuned for the function. Cherkassky and Ma[6] propose ε be chosen as

$$\varepsilon = 3\sigma\sqrt{\frac{\ln N}{N}} \tag{2.33}$$

where σ is the standard deviation of the noise associated with the training point response values and *N* is the number of training points. This assumes the noise is known or can be determined. Cherkassky and Ma[6] suggest the following to estimate the unknown variance of noise using a *k*-nearest neighbor technique

$$\sigma^2 = \frac{N^{1/5}k}{N^{1/5}-1} \cdot \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2 \tag{2.34}$$

where *k* is in the range [2,6] and $\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$ is the squared sum of the residuals.

The optimization problem in Eq. (2.30) written as a Lagrangian function is

$$\begin{aligned} L = \frac{1}{2}|\omega|^2 + C\sum_{i=1}^{N}(\xi_i + \xi_i^*) - \sum_{i=1}^{N}\alpha_i(\varepsilon + \xi_i - y_i + \langle\omega \cdot x_i\rangle + b)\,) \\ - \sum_{i=1}^{N}\alpha_i^*(\varepsilon + \xi_i^* + y_i - \langle\omega \cdot x_i\rangle - b) - \sum_{i=1}^{N}(\eta_i\,\xi_i + \eta_i^*\xi_i^* \end{aligned} \tag{2.35}$$

where η_i and η_i^* are additional slack variables. From Lagrangian theory, necessary conditions for α to be a solution are listed below

$$\partial_b L = \sum_{i=1}^{N}(\alpha_i^* - \alpha_i) = 0 \tag{2.36}$$

$$\partial_{\omega} L = \omega - \sum_{i=1}^{N} (\alpha_i{}^* - \alpha_i)\, x_i = 0 \tag{2.37}$$

$$\partial_{\xi_i} L = C - \alpha_i - \eta_i = 0 \tag{2.38}$$

$$\partial_{\xi_i{}^*} L = C - \alpha_i{}^* - \eta_i{}^* = 0 \tag{2.39}$$

Substituting Eqs. (2.36) - (2.39) into Eq. (2.30) gives the dual form optimization problem

$$Maximize \begin{cases} -\frac{1}{2}\sum_{i,j=1}^{N} (\alpha_i - \alpha_i{}^*)(\alpha_j - \alpha_j{}^*)\, \langle x_i \cdot x_j \rangle \\ -\varepsilon \sum_{i=1}^{N} (\alpha_i + \alpha_i{}^*) + \sum_{i=1}^{N} y_i (\alpha_i - \alpha_i{}^*) \end{cases} \\ s.t. \begin{cases} \sum_{i=1}^{N} (\alpha_i - \alpha_i{}^*) = 0 \\ (\alpha_i - \alpha_i{}^*)\, \epsilon\, [0, C] \end{cases} \tag{2.40}$$

Eq. (2.37) is rewritten as

$$\omega = \sum_{i=1}^{N} (\alpha_i{}^* - \alpha_i)\, x_i \tag{2.41}$$

The linear regression first expressed in Eq. (2.29) is written as

$$f(x) = \sum_{i=1}^{N} (\alpha_i - \alpha_i{}^*)\, \langle x_i \cdot x_i \rangle + b \tag{2.42}$$

Clarke et al.[9] summarize the process of transforming the problem into dual form by stating:

> "*Transforming the optimization problem into dual form yields two advantages. First, the optimization problem is now a quadratic programming problem with linear constraints and a positive definite Hessian matrix, ensuring a unique global optimum. For such problems, highly efficient and thoroughly tested quadratic solvers exist. Second, as can be seen in Eq. [(2.40)], the input vectors only appear inside the dot product. The dot product of each pair of input vectors is a scalar and can be preprocessed and stored in the quadratic matrix* $M_{ij} = (\langle x_i x_j \rangle)_{ij}$*. In this way, the dimensionality of the input space is hidden from the remaining computations, providing means for addressing the curse of dimensionality*."

A nonlinear regression model can be developed by replacing the dot product $\langle \boldsymbol{x_i} \cdot \boldsymbol{x_i} \rangle$ with a kernel function, *k*, rewriting the optimization problem in Eq. (2.40) as

$$Maximize \begin{cases} -\frac{1}{2}\sum_{i,j=1}^{N} (\alpha_i - \alpha_i{}^*)(\alpha_j - \alpha_j{}^*)\, k(x_i, x_j) \\ -\varepsilon \sum_{i=1}^{N} (\alpha_i + \alpha_i{}^*) + \sum_{i=1}^{N} y_i (\alpha_i - \alpha_i{}^*) \end{cases} \\ s.t. \begin{cases} \sum_{i=1}^{N} (\alpha_i - \alpha_i{}^*) = 0 \\ (\alpha_i - \alpha_i{}^*)\, \epsilon\, [0, C] \end{cases} \tag{2.43}$$

Replacing the dot product with a kernel function in the approximation function Eq. (2.42) gives the nonlinear SVR approximation as

$$f(x) = \sum_{i=1}^{N} (\alpha_i - \alpha_i^*)\, k(x_i, x_j) + b \qquad (2.44)$$

Common Kernel functions include the following:

- Linear: $k(x_i, x_j) = x_i^T, x_j$
- Polynomial: $k(x_i, x_j) = (\gamma \langle x_i \cdot x_j \rangle + r)^d \;\; \gamma > 0$
- Gaussian: $k(x_i, x_j) = \exp(-\frac{\|x_i - x_j\|^2}{2p^2})$
- Radial Basis Function: $k(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2) \qquad \gamma > 0$
- Sigmoid: $k(x_i, x_j) = \tanh(\gamma \langle x_i \cdot x_j \rangle + r)$

where γ,*d*, *p*, and *r* are kernel parameters and should be adjusted by the user for each data set. Listed below are some insights about setting kernel parameters:

1) The polynomial degree *d* is typically chosen to be 2. Gunn[7] uses *r* = 1 to "avoid problems with the hessian becoming zero".
2) Hsu et al.[8] use a cross-validation approach to determine γ for RBF. This procedure could be applied to γ for other kernels as well as other kernel parameters.
3) Cherkassky and Ma[6] suggest *p* for the Gaussian kernel (they call it RBF, the formulation is the same as "Gaussian" here) as $p \sim (0.1 - 0.5) * range(x)$ for single variable problems and $p^L \sim (0.1 - 0.5)$ for multi variable problems where *L* is the number of variables and all input variables are normalized to [0,1].

It should be noted that the "Gaussian" kernel here is sometimes referred to as "Radial Basis Function" and "Gaussian Radial Basis Function" in the literature. Experience shows that

SVR metamodels are highly sensitive to tuning parameters. The suggestions expressed in this discussion may not result in an acceptable accuracy level. The reader is encouraged to perform SVR tuning to ensure an accurate model for the selected response.

The SVR MATLAB toolbox developed by Gunn[7] was used in this study and the Linear kernel is used. The tuning parameters explored within SVR are penalty parameter, C and ε-insensitive zone parameter, c.

References

[2] Acar, E. and Rais-Rohani, M. "Ensemble of Metamodels with Optimized Weight Factors, *"Structural and Multidisciplinary Optimization,* Vol. 37, No. 3, 2008, pp. 279-294.

[6] Cherkassky, V., Ma, Y. "Practical selection of SVM parameters and noise estimation for SVM regression," *Neural Networks,* Vol. 17, 2004, pp. 113-126.

[7] Gunn, SR. "Support vector machines for classification and regression," Technical Report, University of Southampton, 1997.

[8] Hsu, CW., Chang CC., Lin CJ. "A Practical Guide to Support Vector Classification," National Taiwan University, last updated 2010.

[9] Clarke, S., Griebsch, J., Simpson, T. "Analysis of Support Vector Regression for Approximation of Complex Engineering Analyses," *Journal of Mechanical Design*, Vol. 127, November, 2005, pp. 1077-1087.