AMS 518 Project: Stable Regression and Support Vector Regression

John Shutkind, john.shutkind@stonybrook.edu

Forrest McMann, forrest.mcmann@stonybrook.edu

Aman Arya, aman.arya@stonybrook.edu

December 13, 2024

Abstract

First, we will introduce regression, and then the traditional randomized approach. Three commonly used randomized approaches will be given that attempt to accurately train the regression model and improve out-of sample accuracy by utilizing penalization and regularization. Subsequently, we will introduce Stable Regression (SR) as an alternative approach in an optimization form, that seeks to optimally train a regression mode. Then, we will discuss Support Vector Regression and its two popular formulations, $\nu - SVR$ and $\epsilon - SVR$. By establishing the Risk Quadrangle framework, we will be able to demonstrate Conditional Value at Risk (CVaR) as a regular risk measure and show some of CVaR's characteristics inside the RQ framework. After formulating $\nu - SVR$ as such, we can reformulate $\nu - SVR$ in terms of CVaR, and then equivalently, as a min-max problem. Upon this final reformulation, we can mathematically demonstrate the equivalency between the two min-max formulations of SR and $\nu - SVR$ concluding the derivation part of this paper.

The final aspect of the paper will employ an algorithm in Portfolio Safe-Guard (PSG), that can test SVR and SR using different datasets, hence, empirically justifying their equivalency.

Contents

1 Introduction

Regression approximates a random variable Y by a function $\hat{f}$ of an observed random vector $X = (X_1, \ldots, X_n)^\top$. By minimizing an error function applied to a regression residual $Z_f = Y - f(X)$, we can find the function $\hat{f}(X)$ from a given class $\mathcal{F}$. We will first discuss stabilizing regression and the approach of robust optimization. Then we will introduce a direct extension from traditional regression of Support Vector Regression, which focuses on fitting the data within a certain error margin, ϵ, and aims to generalize well on unseen data by controlling model complexity. Support Vector Regression (SVR) has two major frameworks: VC theory and RQ theory. The former is applicable to machine learning, and the latter is applicable to risk management. SVR has been long used with the VC framework, but by introducing SVR into the risk quadrangle framework, we are able to establish a connection between SVR as a machine learning tool and risk management, classical statistics and distributionally robust optimization.

2 Traditional Randomization Approach

There are multiple popular techniques that aim to stabilize regressions, however, a widely used method is that of randomization towards training. The approach goes as follows; you randomly pick a subset from the data to put aside as a testing set, and then the remaining data is randomly split into training and validation sets. Then the model is trained accordingly on the training data, tested on it's accuracy through numerous iterations on the validation set, and finally assessed on the validation set. This is seen as an optimization problem:

$$\min_{\beta} \sum_{i=1}^{n} \left| y_i - x_i^\top \beta \right|$$

Which finds a $\beta \in \mathbb{R}^p$ through minimization, given labeled points (x_i, y_i), $i = 1, \ldots, n$, where $x_i \in \mathbb{R}^p$ and $y_i \in \mathbb{R}$. Typically the joined set of the training data and validation subset include the whole data, and their intersection set is empty.

Then we use the data points in the training sets to find an optimal β^*:

$$\beta^* = \arg\min_{\beta} \sum_{i \in A_{\text{train}}} \left| y_i - x_i^\top \beta \right|,$$

and then using β^* to evaluate performance on the validation set, i.e., the subset of points $(x_i, y_i),\ i \in A_{\text{val}}$, typically via:

$$\frac{1}{|A_{\text{val}}|} \sum_{i \in A_{\text{val}}} d\left(y_i, x_i^\top \beta^*\right),$$

where $d\left(y_i, x_i^\top \beta^*\right) = \left| y_i - x_i^\top \beta^* \right|$ or $d\left(y_i, x_i^\top \beta^*\right) = \left(y_i - x_i^\top \beta^*\right)^2$.

However, because this often can lead to over-fitting the coefficients of the linear regression model are regularized or penalized. Three of the popular regularized regression models are given as follows:

1. **Lasso Regression** adds an L1 penalty to the loss function, promoting sparsity in the coefficients:

$$\min_{\beta} \sum_{i=1}^{n} |y_i - \mathbf{x}_i^T \beta| + \lambda \sum_{j=1}^{p} |\beta_j|$$

2. **Ridge Regression** incorporates an L2 penalty, which shrinks the coefficients but does not enforce sparsity:

$$\min_{\beta} \sum_{i=1}^{n} |y_i - \mathbf{x}_i^T \beta| + \lambda \sum_{j=1}^{p} \beta_j^2$$

3. **Elastic Net Regression** combines both L1 and L2 penalties, leveraging the benefits of both Lasso and Ridge:

$$\min_{\beta} \sum_{i=1}^{n} |y_i - \mathbf{x}_i^T \beta| + \lambda_1 \sum_{j=1}^{p} |\beta_j| + \lambda_2 \sum_{j=1}^{p} \beta_j^2$$

A crucial aspect of training these regularized models is tuning the regularization parameter(s) λ. This tuning is typically performed using cross-validation over a range of λ values. The optimal λ and corresponding coefficients β that yield the smallest error on the validation set are selected.

Once the best λ is identified, the model is evaluated on an independent test set to assess its performance, commonly using the mean squared error (MSE) as the evaluation metric. Typically, the dataset is split by reserving 10% of the data as the test set, while the remaining 90% is used for training and validation purposes. This ensures that the model's performance is assessed on unseen data, providing an unbiased estimate of its predictive capability.

3 Stable Regression

However, a problem arises when training the linear regression models. Since the cross-validation procedures involves randomization, optimally and efficiently choosing a partition of training and validation sets that gives the best out-of-sample performance can prove rather difficult and complex. Therefore, we introduce a different approach to train a regularized regression model is that of the following form:

$$\min_{\beta} \max_{z \in Z} \sum_{i=1}^{n} z_i \left| y_i - x_i^\top \beta \right| + \lambda \sum_{i=1}^{p} \Gamma(\beta_i), \quad \text{with} \quad Z = \left\{ z \in \{0,1\}^n \;\middle|\; \sum_{i=1}^{n} z_i = k \right\}.$$

Where, instead of randomly assigning the data to training and validation sets, this approach simultaneously determines the optimal regression coefficients β and selects the best subset of data points for training during the optimization process. Hence, it is able to solve two problems as one linear optimization problem. By choosing the hardest set of data to train on, the model is in a sense training on the worst situation. The indictor variable z_i is valued at 0 or 1, determining whether the data point is included in the training set or validation set, while $\Gamma(\cdot)$ is the chosen regularization function. To dictate the relative proportion of data being assigned to training sets as opposed to validation sets, the parameter k can be set (at k=0.7, the ratio of data in the training set versus validation set would be 70/30). However, we can imagine a situation where stable regression and its performance can be limited due to the nature of the data. Suppose we had a relatively small set of observations with a small set of outliers. If we condition the stable regression where the percent of data it is to be trained upon is roughly equivalent to the percent of outliers then we could create a model that may perform poorly on the out-of sample data.

4 Formulating SVR

Now that we have introduced SR as a robust approach to training a regression model, we will now introduce Support Vector Regression (SVR). There two popular formulations for SVR, ϵ-SVR and ν-SVR.

4.1 ϵ-SVR

ϵ-SVR seeks to find a hyperplane that not only fits the data but also allows for a margin of tolerance, $\epsilon > 0$, within which no penalty is incurred for deviations. The optimization problem for ϵ-SVR is formulated as:

$$\min_{\mathbf{w},b} \left(\lambda \|\mathbf{w}\|^2 + \sum_{i=1}^{l} \left[|y_i - \mathbf{w}^\top \mathbf{x}_i - b| - \epsilon \right]_+ \right),$$

where:

- $\|\mathbf{w}\|$ denotes the Euclidean norm (L2-norm) of the weight vector $\mathbf{w}$.
- $\lambda > 0$ is the regularization parameter that controls the trade-off between the flatness of the hyperplane and the penalty for deviations exceeding ϵ.
- $[a]_+ = \max\{0, a\}$ represents the hinge loss, which penalizes only those predictions where the absolute error exceeds ϵ.

Regularization Term ($\lambda\|\mathbf{w}\|^2$): The term $\lambda\|\mathbf{w}\|^2$ serves as a regularizer that penalizes large weights, promoting a flatter and simpler hyperplane. A smaller $\|\mathbf{w}\|$ implies a model with lower complexity, which helps in preventing overfitting and enhancing the model's generalization capabilities.

Hinge Loss ($\sum_{i=1}^{l} \left[|y_i - \mathbf{w}^\top \mathbf{x}_i - b| - \epsilon \right]_+$): The hinge loss component measures the extent to which predictions deviate from the actual target values beyond the margin ϵ. Specifically, it calculates the absolute error $|y_i - \mathbf{w}^\top \mathbf{x}_i - b|$ for each data point and subtracts ϵ. Only the errors exceeding ϵ contribute to the loss, allowing the model to tolerate small deviations and focus on correcting larger discrepancies.

Probabilistic Interpretation: In a probabilistic framework, let $z = z(\mathbf{w}, b)$ be a random variable representing the deviations:

$$z = \begin{bmatrix} y_1 - \mathbf{w}^\top \mathbf{x}_1 - b \\ y_2 - \mathbf{w}^\top \mathbf{x}_2 - b \\ \vdots \\ y_l - \mathbf{w}^\top \mathbf{x}_l - b \end{bmatrix}.$$

Assuming that each component of z is equally likely, the expected loss can be expressed as:

$$\mathbb{E}\left[(|z(\mathbf{w}, b)| - \epsilon)_+ \right] = \frac{1}{l} \sum_{i=1}^{l} \left(|y_i - \mathbf{w}^\top \mathbf{x}_i - b| - \epsilon \right)_+.$$

Therefore, the ϵ-SVR optimization problem can be reformulated as:

$$\min_{\mathbf{w},b} \ \mathbb{E}\left[(|z(\mathbf{w}, b)| - \epsilon)_+ \right] + \lambda \|\mathbf{w}\|^2.$$

This formulation emphasizes the balance between minimizing the expected loss for deviations beyond ϵ and maintaining a simple (flat) hyperplane through regularization.

4.2 ν-SVR

The second commonly seen form of SVR is ν-SVR, which takes the form of solving the optimization problem:

$$\min_{w,b,\varepsilon} \ \lambda\frac{1}{2}\|w\|^2 + \varepsilon\nu + \mathbb{E}\left[\,|z(w,b)| - \varepsilon\,\right]_+$$

subject to:

$$z(w,b) = y - w^\top x - b$$

where:

- λ is the regularization parameter,
- ε is the error margin,
- $\nu \in (0, 1]$ controls the number of support vectors,
- $[\,\cdot\,]_+$ denotes the positive part function,
- $\mathbb{E}$ represents the expectation operator.

Support Vectors are the data points that lie closest to the regression hyperplane (or decision boundary in classification tasks). These points are critical because they directly influence the position and orientation of the hyperplane. In ν-SVR, $\nu\ x \in A\ (0,\ 1]$ controls the number of support vectors:

Bounds on Support Vectors: The parameter ν sets an upper and lower limit on the fraction of training errors and support vectors. Specifically, it ensures that no more or no less than a fraction ν of the training data points become support vectors. When ν is set closer to 1, the model is encouraged to have more support vectors. This typically results in a more flexible model that can fit the training data more closely, potentially capturing more complex patterns but also increasing the risk of overfitting. Conversely, a smaller ν value restricts the number of support vectors, promoting a simpler model that may generalize better to new data but might underfit if the data is complex. Therefore, ν acts as a trade-off parameter between the model's flatness (controlled by $\|w\|$) and the number of support vectors (which affects the model's ability to capture variability in the data). By adjusting ν, practitioners can fine-tune the model to achieve the desired balance between bias and variance

5 Optimization Formulas and CVaR

When dealing with random variables in optimization contexts, particularly those representing levels of risk such as financial loss, pollution, or contamination, it is crucial to rank or order these variables to facilitate comparison. This ranking is achieved by assigning numerical values through risk measures. In this section, we discuss the risk quadrangle framework and focus on Conditional Value-at-Risk (CVaR), a prominent risk measure, and explore its optimization formulations.

The Risk Quadrangle (RQ) Framework plays a crucial role in incorporating Conditional Value-at-Risk (CVaR) into Support Vector Regression (SVR). This integration enhances SVR by providing a comprehensive approach to managing and quantifying various aspects of risk., and it will allow us to show the equivalence between SR and SVR.

5.1 Comprehensive Risk Management

The RQ framework encompasses four key components, alongside the statistic component:

- **Error,** E: The difference between predicted and actual values.
- **Regret,** V: The gap between the chosen prediction and the best possible prediction.
- **Risk,** R: The likelihood of adverse outcomes beyond expected performance.
- **Deviation,** D: The variability in prediction errors.

By addressing these dimensions, the RQ framework offers a holistic view of uncertainty and performance in regression models, enabling effective risk management within SVR.

5.1.1 Regular Risk Measures

A *risk measure* quantifies the risk associated with random variables, enabling the comparison and ranking of different risk levels. We concentrate on risk measures that satisfy specific properties, collectively known as regular risk measures. Under this framework, a regular risk measure adheres to four fundamental axioms, described in the following definition:

Definition 5.1 (Regular Risk Measure)**.** *A functional* $R : L^2(\Omega) \to \mathbb{R} \cup \{+\infty\}$ *is called a* ***regular measure of risk*** *if it satisfies the following properties:*

1. ***Constant Neutrality (R1):*** $R(C) = C$ *for all constant* C*.*
2. ***Convexity (R2):*** *For all* $X, Y \in L^2(\Omega)$ *and* $\lambda \in [0, 1]$*,*
$$R(\lambda X + (1 - \lambda)Y) \leq \lambda R(X) + (1 - \lambda)R(Y).$$
3. ***Closedness (R3):*** *The set* $\{X \in L^2(\Omega) \mid R(X) \leq c\}$ *is closed for every* $c < \infty$*.*
4. ***Aversity (R4):*** $R(X) > \mathbb{E}[X]$ *for all* X *that are not constant.*

5.1.2 Unified Methodological Approach

The RQ framework seamlessly integrates stochastic optimization, risk management, and statistical estimation. This unification is essential for embedding risk measures like CVaR into regression analysis, ensuring that all aspects of risk are systematically addressed within a single model.

Traditional SVR, based on Vapnik-Chervonenkis (VC) Theory, focuses on minimizing prediction errors using specific loss functions, such as the ϵ-insensitive loss. However, VC Theory does not inherently account for risk measures that evaluate extreme deviations or tail behavior.

In contrast, the RQ framework, prominent in risk management, introduces CVaR as a measure capturing the expected loss beyond a certain quantile. By situating SVR within the RQ framework, researchers align machine learning objectives with robust risk assessment practices, enhancing SVR's capability to manage extreme risks.

Incorporating CVaR through the RQ framework enables SVR to minimize not only average prediction errors but also the risk associated with significant deviations. This dual focus ensures that SVR models are both accurate on average and resilient against outliers or high-impact errors.

5.2 Quantiles and Value-at-Risk (VaR)

To address CVaR and its importance, it is crucial to understand the method it was created to improve, Value-at-Risk. Quantiles provide a fundamental tool for risk assessment by identifying specific points in the distribution of a random variable. The **Value-at-Risk (VaR)** at a confidence level α is a widely used quantile-based risk measure.

Definition 5.2 (Quantile (Value-at-Risk, VaR))**.** *The quantile, also known as Value-at-Risk (VaR), of a random variable X at confidence level $\alpha \in [0,1]$ is defined as the set:*

$$q_\alpha(X) = [q_\alpha^-(X), q_\alpha^+(X)],$$

where

$$q_\alpha^-(X) = \begin{cases} \textit{ess inf}(X) & \textit{if } \alpha = 0, \\ \inf\{x \in \mathbb{R} \mid F_X(x) > \alpha\} & \textit{if } \alpha \in (0,1), \end{cases}$$

$$q_\alpha^+(X) = \begin{cases} \textit{ess sup}(X) & \textit{if } \alpha = 1, \\ \sup\{x \in \mathbb{R} \mid F_X(x) < \alpha\} & \textit{if } \alpha \in [0,1). \end{cases}$$

If $q_\alpha^-(X) = q_\alpha^+(X)$, then $q_\alpha(X)$ is a singleton set containing the VaR at level α.

Remark 5.1 (Sum and Scaling of Quantiles)**.** *Since a quantile is defined as a set, the sum of two quantiles is given by the Minkowski sum of convex sets. Specifically, for $\alpha_1, \alpha_2 \in [0,1]$,*

$$q_{\alpha_1}(X) + q_{\alpha_2}(X) = \{v + w \mid v \in q_{\alpha_1}(X), w \in q_{\alpha_2}(X)\}.$$

Similarly, scaling a quantile by a constant $\lambda \in \mathbb{R}$ is defined as:

$$\lambda q_\alpha(X) = \{\lambda w \mid w \in q_\alpha(X)\}.$$

5.3 Conditional Value-at-Risk (CVaR)

CVaR extends the concept of VaR by considering the expected loss exceeding the VaR threshold. It is also known as Tail Value-at-Risk, Average Value-at-Risk, or Expected Shortfall. CVaR is favored for its coherent risk measure properties and suitability for optimization. Now I will introduce numerous definitions, cited in the bibliography, that help give CVaR is characteristics and importance inside the RQ framework.

Definition 5.3 (Conditional Value-at-Risk (CVaR))**.** *The Conditional Value-at-Risk (CVaR) of a random variable X at confidence level $\alpha \in [0,1]$ is defined as:*

$$\mathit{CVaR}_\alpha(X) = \begin{cases} \frac{1}{1-\alpha} \int_\alpha^1 \mathit{VaR}_\beta(X)\, d\beta & \textit{if } \alpha \in (0,1), \\ \mathbb{E}[X] & \textit{if } \alpha = 0, \\ \textit{ess sup}(X) & \textit{if } \alpha = 1. \end{cases}$$

Definition 5.4 (Scaled CVaR Norm)**.** *Let $X \in L^1(\Omega)$ be a real-valued random variable. The* scaled Conditional Value-at-Risk (CVaR) norm *of X with parameter $\alpha \in [0,1]$ is defined as*

$$\langle\langle X \rangle\rangle_{S_\alpha} = \overline{q}_\alpha(|X|).$$

This definition shows how the scaled CVaR norm normalizes the risk measure by parameter α allowing for consistent comparison and integration within the SVR framework.

Definition 5.5 (Non-scaled CVaR Norm)**.** *Let $X \in L^1(\Omega)$ be a real-valued random variable. The* non-scaled CVaR norm *of X with parameter $\alpha \in [0, 1)$ is defined as*

$$\langle\langle X \rangle\rangle_\alpha = (1 - \alpha)\overline{q}_\alpha(|X|).$$

The non-scaled CVaR norm aligns with definitions used in prior research (e.g., Pavlikov and Uryasev, 2014), facilitating easier comparison and integration of new findings. By incorporating the factor (1-α), this norm directly scales the CVaR measure, providing a straightforward mechanism to adjust the weight given to tail risks in the SVR optimization process. The non-scaled version often simplifies the mathematical formulation of optimization problems within SVR, making it computationally more efficient to incorporate CVaR into the regression model.

Definition 5.6 (CVaR Norm Quadrangle)**.** *For $\alpha \in [0, 1)$, the error measure $E(X) = \langle\langle X \rangle\rangle_\alpha$ generates the following regular quadrangle:*

$$\begin{aligned}
S(X) &= \frac{1}{2} q\left(\frac{1-\alpha}{2}\right)(X) + q\left(\frac{1+\alpha}{2}\right)(X), \\
R(X) &= R_1(X) - (1-\alpha)x, \quad \forall \alpha \in A_x(X), \\
D(X) &= D_1(X) - (1-\alpha)x, \quad \forall \alpha \in A_x(X), \\
V(X) &= \mathbb{E}[|X| - x]_+ + \mathbb{E}[X], \\
E(X) &= \langle\langle X \rangle\rangle_\alpha.
\end{aligned}$$

The quadrangle encapsulates four key risk-related measures—Error (E), Regret (R), Risk (V), and Deviation (D)—providing a structured framework to analyze and manage different aspects of risk within SVR. By defining relationships between these measures, the proposition ensures that changes in one aspect of risk (e.g., Deviation) are coherently reflected in others (e.g., Risk and Regret), promoting balanced optimization. The quadrangle framework offers a mathematical structure that can be leveraged to develop and apply optimization algorithms tailored to minimize CVaR within the SVR context.

Definition 5.7 (Quantile Symmetric Average Quadrangle)**.** *Let $X \in L^2(\Omega)$, $0 \leq x < \frac{1}{2}\left(\text{ess sup}(X) - \text{ess inf}(X)\right)$, and let (R_1, D_1, V_1, E_1) be the CVaR Norm Quadrangle quartet with statistic S_1. Then the set*

$$A_x(X) = \left\{\alpha \in [0, 1) \mid q^-(X) - q^-(X) \leq x \leq q^+(X) - q^+(X)\right\}$$

is nonempty, and the Vapnik error generates the following quadrangle:

$$\begin{aligned}
S(X) &= S_1(X), \quad \forall \alpha \in A_x(X), \\
R(X) &= R_1(X) - (1-\alpha)x, \quad \forall \alpha \in A_x(X), \\
D(X) &= D_1(X) - (1-\alpha)x, \quad \forall \alpha \in A_x(X), \\
V(X) &= \mathbb{E}[|X| - x]_+ + \mathbb{E}[X], \\
E(X) &= \mathbb{E}[|X| - x]_+ = \textit{Vapnik error}.
\end{aligned}$$

This proposition establishes a direct relationship between the Vapnik error (a measure used in SVR) and the CVaR norm within the quadrangle framework, enabling the

incorporation of CVaR into the SVR optimization objective. By utilizing quantiles in defining the quadrangle, the proposition ensures that SVR models can effectively capture and minimize risks associated with specific parts of the error distribution, particularly the tails. Therefore, this provides a precise mathematical relationship ensures that the integration of CVaR into SVR is not only intuitive but also grounded in rigorous theoretical principles, promoting consistency and reliability in model performance.

Remark 5.2. *The regression residual $Z(w,b) = Y - (w^\top X + b)$ can be generalized to $Z(w,b) = Y - f(w,X)$, where f belongs to a broader class of functions. This proposition holds in a more general nonlinear setting. However, in the case of ℓ_2 regularization, the class of affine functions suffices due to the applicability of the "kernel trick."*

5.4 Optimization Formulations of CVaR

CVaR can be characterized through optimization problems, providing a foundation for efficient computational algorithms.

Theorem 5.1 (CVaR Optimization Formula)**.** *For a random variable X and $\alpha \in (0,1)$,*

$$CVaR_\alpha(X) = \min_{C\in\mathbb{R}} \left\{ C + \frac{1}{1-\alpha}\mathbb{E}[(X-C)_+] \right\},$$

where $(X-C)_+ = \max\{X-C, 0\}$. The minimizer C^* is the VaR at level α, i.e., $C^* = VaR_\alpha(X)$.

Theorem 5.2 (Dual CVaR Optimization Formula)**.** *For a random variable X and $x \in \mathbb{R}$,*

$$\mathbb{E}[(X-x)_+] = \max_{\alpha\in[0,1]} (1-\alpha)(CVaR_\alpha(X) - x).$$

The set of maximizers for this problem is $\alpha \in [P(X<x), P(X\le x)]$.

Remark 5.3. *Note that $(1-\alpha)CVaR_\alpha(X)$ is a concave function of α. Consequently, the dual optimization problem presented in Theorem 3.2 is a concave maximization problem.*

5.5 Relationship Between CVaR and Mean Excess Function

There exists a profound relationship between the mean excess function and CVaR. Specifically, the mean excess function can be expressed in terms of CVaR, further highlighting the integral role of CVaR in risk assessment and optimization.

$$\mathbb{E}[(X-x)_+] + x = \mathbb{E}[X \mid X > x], \tag{1}$$

which connects the mean excess function to the expectation of X conditional on exceeding x.

Theorem 5.3 (Relationship Between Mean Excess Function and CVaR)**.** *For a random variable X and $x \in \mathbb{R}$,*

$$\mathbb{E}[(X-x)_+] = \max_{\alpha\in[0,1]} (1-\alpha)(CVaR_\alpha(X) - x).$$

This theorem establishes that the mean excess function is maximized by the CVaR at appropriate confidence levels.

5.6 Quadrangle Representation

Within the RQ framework, SVR variants like ϵ-SVR and ν-SVR correspond to specific quadrangles defined by their error, risk, deviation, and regret measures. This mathematical representation facilitates easier manipulation and optimization of SVR models by leveraging established properties of CVaR and related risk measures.

The RQ framework assists in deriving alternative dual formulations of SVR that incorporate CVaR. These dual forms often result in optimization problems with lower complexity or improved computational characteristics, allowing the use of efficient solvers and enhancing the scalability of SVR models in practical applications.

Following the methodology of Takeda and Sugiyama (2008), we can reformulate the ν-SVR problem. Based on the work of Pavlikov and Uryasev (2014) and Bertsimas et al. (2011), the objective function can be rewritten using the Conditional Value-at-Risk (CVaR):

$$\min_{w,b} \ \lambda\|w\|^2 + \nu\, \bar{q}\left(|z(w,b)|\right)$$

where:

$$\bar{q} = \frac{1-\nu}{\varepsilon} + 1 - \nu = \langle\!\langle\, |z(w,b)|\, \rangle\!\rangle_{1-\nu}$$

In this context:

- $\bar{q}\left(|z(w,b)|\right)$ represents the CVaR of $|z(w,b)|$ at confidence level $1-\nu$ (refer to Definition 3.3),
- $\langle\!\langle\, |z(w,b)|\, \rangle\!\rangle_{1-\nu}$ denotes the CVaR norm.

Therefore, the ν-SVR optimization problem can be reformulated as:

$$\min_{w,b} \ \lambda\|w\|^2 + \langle\!\langle\, z(w,b)\, \rangle\!\rangle_{1-\nu}$$

This reformulation highlights the connection between ν-SVR and risk measures like CVaR, demonstrating its applicability in both machine learning and risk management (one could also establish the equivalency between ν-SVR and ϵ-SVR, although for the purpose of this paper, we need not establish this). We can also map this minimization into another form; by setting α= 1 - v, and interpreting the residuals consistently, we have:

$$\min_{w} \ \mathrm{CVaR}_{\alpha}\left(|y - Xw|\right) + \lambda\|w\|_2^2$$

6 Establishing Equivalence Between SR and SVR

6.1 ν-SVR as a Min-Max Problem

In this section, we establish the equivalence between ν-SVR and (CVaR) minimization framework. By leveraging the dual representation of CVaR, we demonstrate how ν-SVR can be reformulated as a saddle point problem, highlighting its connection to distributionally robust optimization.

The ν-SVR optimization problem can be framed as a regularized CVaR norm minimization problem. Specifically, consider the ν-SVR formulation:

$$\min_{\mathbf{w},b,\epsilon} \left(\mathbb{E}\left[|Z(\mathbf{w},b)| - \epsilon \right]_+ + \nu\epsilon + \lambda \|\mathbf{w}\|^2 \right), \tag{2}$$

where:

- $Z(\mathbf{w},b) = Y - (\mathbf{w}^\top \mathbf{X} + b)$ is the regression residual.
- $\nu \in (0,1]$ controls the number of support vectors.
- $\lambda > 0$ is the regularization parameter.

Using the dual representation of CVaR, the ν-SVR problem admits an equivalent saddle point formulation:

$$\min_{\mathbf{w},b} \max_{Q \in \mathcal{Q}_\alpha} \left(\mathbb{E}_Q \left[|Z(\mathbf{w},b)| \right] + \lambda \|\mathbf{w}\|^2 \right). \tag{3}$$

Here, $\mathcal{Q}_\alpha$ is the uncertainty set defined as:

$$\mathcal{Q}_\alpha = \left\{ Q \ll P \;\middle|\; 0 \leq \frac{dQ}{dP} \leq \frac{1}{1-\alpha} \right\}.$$

The saddle point formulation (3) reveals that ν-SVR is equivalent to a Regularized Distributionally Robust Optimization (DRO) problem. Specifically, it can be interpreted as a distributionally robust L_1-regression where the model seeks to minimize the worst-case expected absolute loss within the uncertainty set $\mathcal{Q}_\alpha$.

The equivalence between ν-SVR and DRO formulations allows us to identify specific instances based on the parameter α:

- **When $\alpha = 0$:**
$$\mathcal{Q}_0 = \left\{ Q \ll P \;\middle|\; \frac{dQ}{dP} \leq 1 \right\} = \{P\},$$
which implies that formulation (3) reduces to the standard regularized L_1-regression:
$$\min_{\mathbf{w},b} \left(\mathbb{E}\left[|Z(\mathbf{w},b)| \right] + \lambda \|\mathbf{w}\|^2 \right).$$

- **As $\alpha \to 1$:**
$$\mathcal{Q}_\alpha = \left\{ Q \ll P \;\middle|\; \frac{dQ}{dP} \leq \frac{1}{1-\alpha} \right\},$$
where $\frac{1}{1-\alpha} \to \infty$. In this limit, the uncertainty set $\mathcal{Q}_\alpha$ encompasses all possible probability measures, and formulation (3) approaches the regularized L_∞-regression:
$$\min_{\mathbf{w},b} \left(\max_i |y_i - \mathbf{w}^\top \mathbf{x}_i - b| + \lambda \|\mathbf{w}\|^2 \right).$$

By reformulating ν-SVR within the CVaR framework and utilizing the dual representation of CVaR, we have demonstrated that ν-SVR is equivalent to a distributionally robust L_1-regression problem. This equivalence not only bridges Support Vector Regression with robust risk management practices but also ensures that the regression model effectively minimizes both average prediction errors and the risk associated with extreme deviations.

6.2 Mapping SR to CVaR Framework

To demonstrate the equivalence between SR and SVR, we leverage the CVaR optimization formulas and interpret the SR's constraints in the context of CVaR. Using **Theorem 5.1**, CVaR can be expressed as a minimization problem involving the expectation of excess losses beyond a certain threshold. Similarly, **Theorem 5.2**, provides a dual formulation connecting CVaR with maximization over confidence levels. These theorems underpin the minimax formulations of both SR and SVR, reinforcing their equivalence through the lens of risk measures.

6.2.1 Understanding the Constraints

- **SR Constraints:**

$$z_i \in \{0, 1\}, \quad \sum_{i=1}^{n} z_i = k$$

  Here, z_i acts as an indicator selecting exactly k data points with the largest errors.

- **SVR Constraints:**

$$q_i \geq 0, \quad \sum_{i=1}^{n} q_i = 1, \quad q_i \leq \frac{1}{n(1-\alpha)}$$

  The weights q_i in SVR distribute the risk across all data points, bounded by $\frac{1}{n(1-\alpha)}$.

6.2.2 Aligning Parameters

Set $k = n(1-\alpha)$. This aligns the number of selected data points in SR with the confidence level α in SVR, effectively targeting the same tail of the loss distribution.

6.2.3 Mapping z_i to q_i

Define $q_i = \frac{z_i}{k}$. Given that $z_i \in \{0, 1\}$ and $\sum_{i=1}^{n} z_i = k$, it follows that:

$$q_i = \begin{cases} \frac{1}{k} & \text{if } z_i = 1 \\ 0 & \text{otherwise} \end{cases}$$

This mapping ensures that q_i satisfies the constraints of SVR:

$$\sum_{i=1}^{n} q_i = \frac{k}{k} = 1, \quad q_i \leq \frac{1}{k} = \frac{1}{n(1-\alpha)}$$

6.3 Equivalence in Objective Functions

1. **SR Objective Function:**

$$\min_{w} \max_{z \in Z} \sum_{i=1}^{n} z_i |y_i - w^\top x_i| + \lambda \|w\|_2^2$$

   Substituting $z_i = k q_i$:

$$\sum_{i=1}^{n} z_i |y_i - w^\top x_i| = k \sum_{i=1}^{n} q_i |y_i - w^\top x_i|$$

2. **SVR Objective Function:**

$$\min_{w} \max_{q \in Q} \sum_{i=1}^{n} q_i |y_i - w^\top x_i| + \lambda \|w\|_2^2$$

3. **Aligning Both Objectives:** By choosing $k = n(1-\alpha)$ and $q_i = \frac{z_i}{k}$, the SR objective scales to:

$$\min_{w} \max_{q \in Q} k \sum_{i=1}^{n} q_i |y_i - w^\top x_i| + \lambda \|w\|_2^2$$

Dividing the entire objective by k (which does not affect the optimization since k is a constant):

$$\min_{w} \max_{q \in Q} \sum_{i=1}^{n} q_i |y_i - w^\top x_i| + \frac{\lambda}{k} \|w\|_2^2$$

This is structurally identical to the SVR objective function, with adjusted regularization parameter $\frac{\lambda}{k}$.

7 Numerical Tests

Numerical simulation is conducted to verify the equivalence of stable regression and SVR regression. PSG is used to calculate the regression coefficients using the *cvar_risk* function. We calculate the SVR regression model coefficients using SK Learn's SVR model. The results show consistency except for extreme α values; at α=.999 we see a breakdown.

We examine two simulated data sets, and one real data set from UCI's Machine Learning Repository. The first simulated data set (Dataset 1) is a simple single variate linear dataset with $\mathcal{N}(0, .5)$ noise added. The second simulated data set is a 4 dimensional linear model with standard $\mathcal{N}(0, 1)$ noise added. The real data set from the UCI Machine Learning Repository is the Abalone data set [1]; this contains physical measurements of Abalones (type of mollusk) and the target variable is age.

Algorithm 1 describes the general data processing and regression process. For the UCI data sets, additional preprocessing is needed to convert categorical fields into binary fields.

Algorithm 1 Workflow for Equivalent Regressions

Require: Dataset $\mathcal{D}$ with features X, target y, regularization C, and parameters α.
Ensure: Coefficients β and intercept b for SVR and PSG.

1: **Step 1: Preprocess Data**

- Normalize numerical features (`StandardScaler`).
- Encode categorical features (`OneHotEncoder`).

2: **Step 2: Train Models**

- **SVR:** Define SVR model with kernel $\mathcal{K}$, C, and ν; train on (X, y); extract β_{SVR}, b_{SVR}.
- **PSG:** Build scenarios matrix $\mathcal{M}$, define regularization matrix $\mathcal{Q}$, and solve:

$$\min\left((1-\alpha)\cdot \text{CVaR}_{\alpha}(\mathcal{M}) + \text{Quadratic}(\mathcal{Q})\right).$$

  Extract β_{PSG}, b_{PSG}.

3: **Step 3: Compare**

- Compare β_{SVR} vs. β_{PSG} and b_{SVR} vs. b_{PSG}.

4: **Step 4: Visualize (Optional)**

- Plot data with both regression lines.

7.1 Simulated Datasets

Dataset 1: Simple Linear Regression

Both the CVaR and NuSVR regression perform the regression on the simple model very well. For all α values, the models are identical down to 10^{-6} for the slope coefficient and 10^{-3} for the intercept coefficient. This simple case is adapted from the *Support Vector Regression: Risk Quadrangle Framework* Case Study[2]. The following chart summarizes the results of data set 1 testing:

Table 1: Comparison of Regression Coefficients (Slope and Intercept), α=.001, c=1

Parameter	**Actual Coefficients**	**SVR Coefficients**	**PSG Coefficients**
Slope	5.0	4.921186994366259	4.921186279364816
Intercept	0	0.038615244869928476	0.03861553919046168

Table 2: Comparison of Regression Coefficients (Slope and Intercept), α=.5, c=1

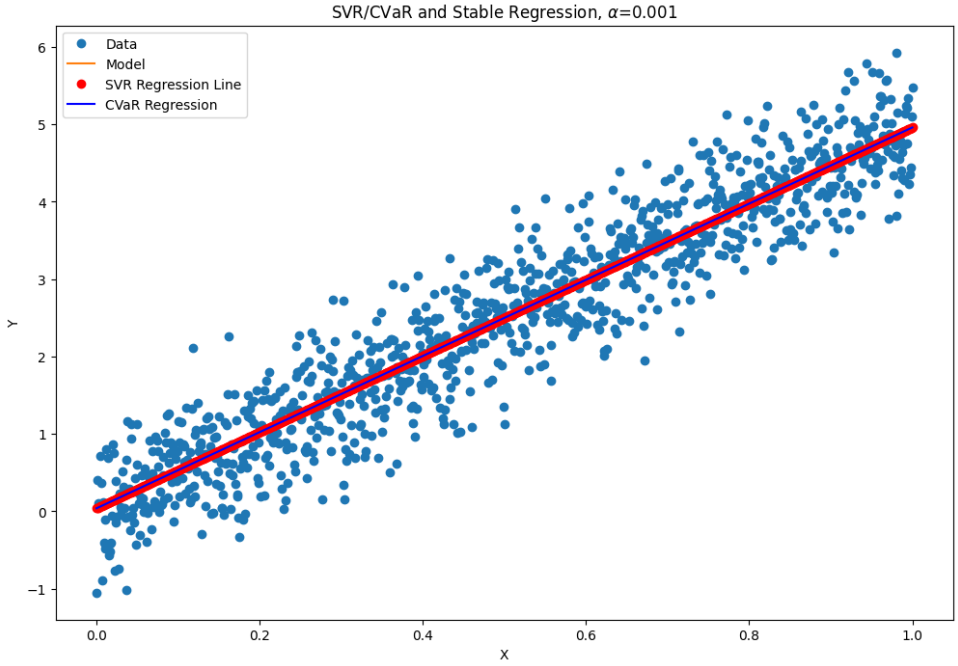
Parameter	**Actual Coefficients**	**SVR Coefficients**	**PSG Coefficients**
Slope	5.0	4.913970003708995	4.913970206058843
Intercept	0	0.05286945214212818	0.05286988905666538

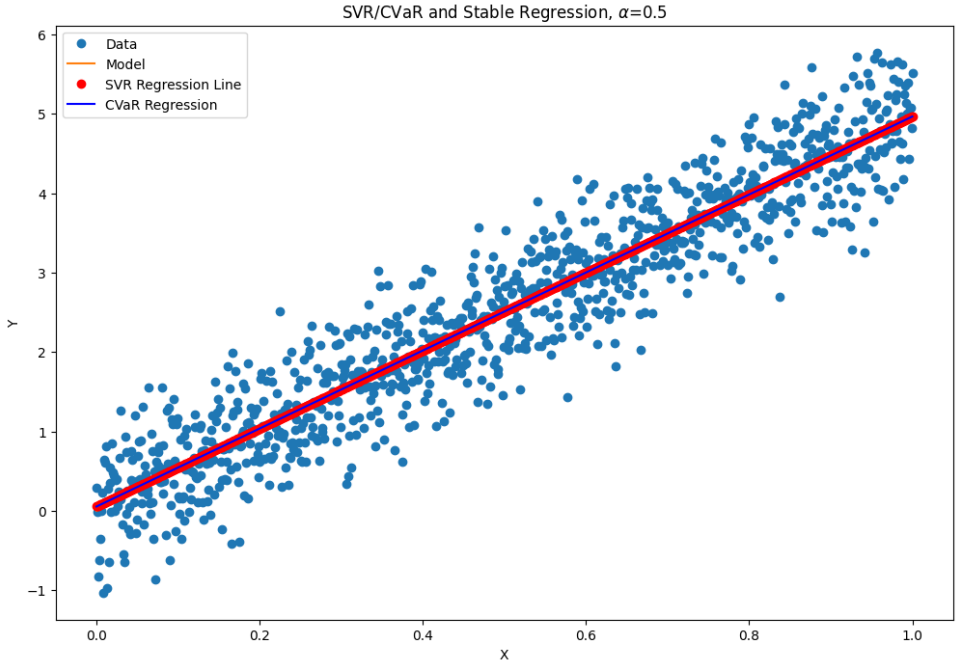
Table 3: Comparison of Regression Coefficients (Slope and Intercept), α=.999, c=1

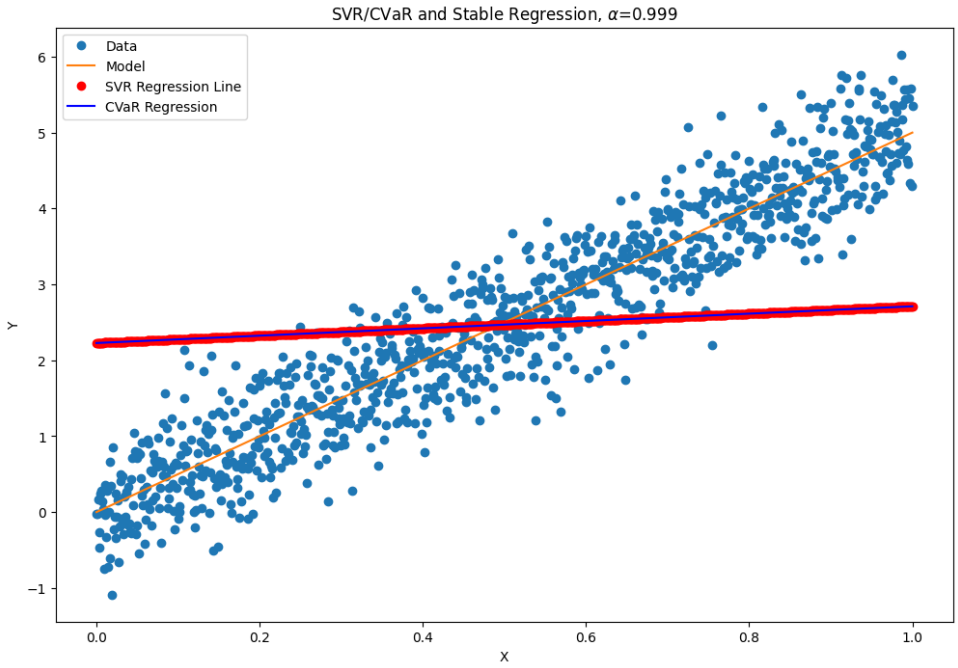
Parameter	**Actual Coefficients**	**SVR Coefficients**	**PSG Coefficients**
Slope	5.0	0.4834834834834839	0.48348375847170644
Intercept	0	2.227831084193122	2.227830951362656

Both the NuSVR and CVaR regression models break down with high alpha values; however the coefficients breakdown in similar ways.

The following figures show the original data, the original exact model, the CVaR Regression line and the NuSVR regression lines.


SVR/CVaR and Stable Regression, α=0.001
Data
Model
SVR Regression Line
CVaR Regression
X
Y


SVR/CVaR and Stable Regression, α=0.5
Data
Model
SVR Regression Line
CVaR Regression
X
Y



Dataset 2: Multivariate Linear Regression

Similar to dataset 1, dataset 2 is simulated and has an exact model we can compare the coefficients against. The model chosen for the project is defined as the following:

$$\mathbf{X} \sim \text{Uniform}(0, 10) \quad \text{(Randomly generated input matrix of size } N \times D\text{)} \tag{4}$$

$$\boldsymbol{\beta}_{\text{true}} = \begin{bmatrix} 2.0 \\ -3.5 \\ 1.0 \\ 4.0 \end{bmatrix} \quad \text{(True coefficients for the model)} \tag{5}$$

$$\boldsymbol{\epsilon} \sim \mathcal{N}(0, 1) \quad \text{(Gaussian noise with mean 0 and variance 1)} \tag{6}$$

$$\mathbf{y} = \mathbf{X} \cdot \boldsymbol{\beta}_{\text{true}} + \boldsymbol{\epsilon} \quad \text{(Target variable as a linear combination of } \mathbf{X} \text{ and noise).} \tag{7}$$

Table 4: Comparison of Regression Coefficients

Feature	**True Coefficients**	**NuSVR Coefficients**	**CVaR Coefficients**
Feature 1	2.0	1.99379647	1.99292758
Feature 2	-3.5	-3.52810274	-3.52867739
Feature 3	1.0	0.98760498	0.98788926
Feature 4	4.0	4.01559553	4.01838419
Intercept	0.0	0.1502509449730346	0.13962107610377297

With the increase in dimensions we see that the models are still close but we see deviations starting at 10^{-3} for some of the features, and deviations at 10^{-2} for the intercept terms.

7.2 UCI Datasets

Dataset 3: Abalone Dataset

The Abalone Dataset contains 8 features: Sex, Length, Diameter, Height, Whole_weight, Shucked_weight, Visera_weight, Shell_weight. The target for this regression is the number of Rings. This dataset requires an additional step, we need to transform the Sex features into a continuous feature. SkLearn's OneHotEncoder is used in the data preprocessing step to encode the categorical data in the Sex column into binary columns. We also use drop='first' in the OneHotEncoder to change the value of the binary columns to 0,0 if I is the value. The regression now has 9 features. The table below summarizes the regressions. This regression uses c=10.

Table 5: Comparison of Regression Coefficients

Feature	**NuSVR Coefficients**	**CVaR Coefficients**
Length	0.01180033	0.01001566
Diameter	0.92385042	0.9235053
Height	0.62961541	0.62587867
Whole_weight	**3.8785749**	**3.9435627**
Shucked_weight	-3.84235663	-3.87465006
Visera_weight	**-1.08063668**	**-1.10397277**
Shell_weight	1.04415583	1.04142231
Sex 1	-0.63380008	-0.63172853
Sex 2	0.06901395	0.07084149
Intercept	9.96343375	9.965103523524213

We now start to see the models diverging from each other when using real data. Since we don't have the true coefficients of the regression, we can only examine the difference in the CVaR and NuSVR regression coefficients. The bold entries in the table show coefficients with greater deviations (on the order 10^-1). The most likely cause of this is the presence of the categorical variable in the dataset. Not having truly continuous data seems to cause the divergence in the models.

Abalone Dataset Further investigation

With the slight divergence of the models with the real data set, we test an averaging method to see if we can get more consistent results. To average over many runs, there needs to be variations in each run; to accomplish this we randomly split the data in half 1000 times and average over the coefficients.The results are summarized below:

Table 6: Comparison of Regression Coefficients

Feature	**NuSVR Coefficients**	**CVaR Coefficients**	**NuSVR Coefficients Ave**	**CVaR Coefficients Ave**
Length	0.01180033	0.01001566	0.06663225645783867	0.06725650581450035
1Diameter	0.92385042	0.9235053	0.8687068186372268	0.8714386720709176
Height	0.62961541	0.62587867	0.5926547226134126	0.5905901438444047
Whole_weight	**3.8785749**	**3.9435627**	**3.735787346579565**	**3.8502966832228145**
Shucked_weight	-3.84235663	-3.87465006	**-3.7932459034503108**	**-3.849507755031685**
Visera_weight	**-1.08063668**	**-1.10397277**	-1.030707604870167	-1.0591340455234934
Shell_weight	1.04415583	1.04142231	**1.105807893445083**	**1.0717762642893147**
Sex 1	-0.63380008	-0.63172853	-0.6508626363564335	-0.6495582806691174
Sex 2	0.06901395	0.07084149	0.06483302365944914	0.06489035500052491
Intercept	9.96343375	9.965103523524213	9.96402642088119	9.965986342499919

After averaging over 1000 runs each with a different set of 50% of the data, we get about the same results in terms of consistency. Although this test does not improve the regression mdodel, we

8 Conclusion

We have shown an equivalency between Stable Regression and Support Vector Regression; we started with establishing SR and its role in regression. The we introduced SVR, and its two formulations, under the RQ framework. Under the RQ framework, CVaR is proven to be a regular risk measure with certain characteristics. Utilizing these characteristics, we reformulated SVR in the RQ framework as a CVaR norm problem, and then as a min-max problem with a saddle point. Utilizing this min-max formulation of SVR and the min-max formulation of stable regression, we can establish their equivalence by analyzing their parameters. However, to demonstrate this equivalency beyond just the theoretical, we will now discuss the numerical tests conducted.

Numerical studies using SkLearn NuSVR and PSG CVaR_Risk show the equivalency for simulated and real data sets. The regression on the simulated data sets show that both methods yield the same coefficients for both the univariate and multivariate sets. When using the real data we see a diverge in the regression models. This is most likely due to the categorical data in the Abalone dataset; using a fully continuous data set would probably yield better results.

References

[1] Nash, W., Sellers, T., Talbot, S., Cawthorn, A., & Ford, W. (1994). *Abalone [Dataset]*. UCI Machine Learning Repository. `https://doi.org/10.24432/C55C7W`.

[2] Anton Malandii and Stan Uryasev. *Support Vector Regression: Risk Quadrangle Framework*. Department of Applied Mathematics and Statistics, State University of New York, Stony Brook, NY, 2024.

[3] Dimitris Bertsimas and Ivan Paskov. Stable regression: On the power of optimization over randomization. *Journal of Machine Learning Research*, 21(230):1–25, 2020. Available at: `https://www.jmlr.org/papers/volume21/19-408/19-408.pdf`

[4] K. Pavlikov and S. Uryasev. CVaR Norm and Applications in Optimization. *Optimization Letters*, 8(7):1999–2020, 2014.