Stable Regression & Support Vector Regression
AMS 518

John Shutkind Forrest McMann Aman Arya

Stony Brook University

December 13, 2024

Outline

1. Overview
2. Traditional Randomization Approach
3. Stable Regression
4. Support Vector Regression
5. Optimization Formulas and CVaR
6. Mapping CVaR to SVR via Risk Quadrangle Framework
7. Establishing Equivalence Between SR and SVR
8. Numerical Analysis

Overview

First, we will introduce regression, and then the traditional randomized approach. Three commonly used randomized approaches will be given that attempt to improve out-of sample accuracy with 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, νSVR and ϵSVR. 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 νSVR as such, we can reformulate νSVR 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 ν SVR concluding the derivation part of this paper. The final aspect of the paper will employ an algorithm PSG, that can test SVR and SR using different datasets, hence, empirically justifying their equivalency.

Overview

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.

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.

Traditional Optimization Approach

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 Lasso Regression, Ridge Regression, Elastic Net Regression.

Traditional Randomization Approach

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. 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.

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.

Stable Regression

SR 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 SR model 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.

Support Vector Regression

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.

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.

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

RQ Framework

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.

RQ Framework

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

$R(X)$ provides a numerical surrogate for the overall hazard in X,

$D(X)$ measures the "nonconstancy" in X as its uncertainty,

$E(X)$ measures the "nonzeroness" in X,

$V(X)$ measures the "regret" in facing the mix of outcomes of X,

$S(X)$ is the "statistic" associated with X through E and V.

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

RQ Framework: Unified Methodology 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.

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. 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

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

$=$

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

If $q_\alpha^-(X) = q_\alpha^+(X)$, then $q_\alpha(X)$ is a singleton set containing the VaR at level α. At a 95 percent confidence level over one day, VaR represents the maximum loss not exceeded with 95 percent probability. Conversely, there's a 5 percent chance that the loss will exceed this value.

Conditional Value-at-Risk

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.

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

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

While VaR tells you the maximum expected loss at a certain confidence level, CVaR calculates the average loss assuming that the loss has exceeded the VaR threshold, giving a clearer picture of the severity of extreme losses.

RQ Framework: Regular Risk Measure

CVaR plays a pivotal role in uniting the methodologies due to it being a regular risk measure defined as follows:
A functional $R : L^2(\Omega) \to \mathbb{R} \cup \{+\infty\}$ is called a **regular measure of risk** if it satisfies the following properties:

- **Constant Neutrality (R1):** $R(C) = C$ for all constant C.
- **Convexity (R2):** For all $X, Y \in L^2(\Omega)$ and $\lambda \in [0, 1]$,

$$R(\lambda X + (1 - \lambda)Y) \leq \lambda R(X) + (1 - \lambda)R(Y).$$

- **Closedness (R3):** The set $\{X \in L^2(\Omega) \mid R(X) \leq c\}$ is closed for every $c < \infty$.
- **Aversity (R4):** $R(X) > \mathbb{E}[X]$ for all X that are not constant.

Optimization Formulas for CVaR

Furthermore, it is CVaR's characteristics that allow it to be integrated into the SVR formula. Now we will introduce a few of these aspects.

- **CVaR Optimization**
  For a random variable X and $\alpha \in (0,1)$,
  $$\mathrm{CVaR}_\alpha(X) = \min_{C \in \mathbb{R}} \left\{ C + \frac{1}{1-\alpha}\mathbb{E}[(X-C)_+] \right\},$$
  where $(X-C)_+ = \max\{X-C, 0\}$. The minimizer C^* is the VaR at level α, i.e., $C^* = \mathrm{VaR}_\alpha(X)$.
- C Represents the threshold loss. A higher C implies a stricter threshold, potentially reducing the expected excess loss.
- $$\frac{1}{1-\alpha}\mathbb{E}\left[(X-C)_+\right]:$$
  Represents the average loss beyond the threshold C, scaled by $\frac{1}{1-\alpha}$ to account for the proportion of losses that exceed C.

The goal is to minimize the sum of the threshold C and the scaled expected excess loss. By adjusting C, we seek the balance between setting a low threshold and a high threshold.

Optimization Formulas for CVaR

- **Dual CVaR Optimization Formula**
  For a random variable X and $x \in \mathbb{R}$,

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

  The set of maximizers for this problem is

  $$\alpha \in \big[P(X < x), P(X \leq x)\big].$$

- The formula establishes that the expected excess loss beyond x is equal to the maximum of the scaled CVaR minus x over all confidence levels α.
- This dual representation is valuable because it links a linear expectation with an optimization problem involving CVaR, a coherent risk measure.

Integrating CVaR into SVR via Risk Quadrangle Framework

Now that we have introduced the relevant context, we can reformulate SVR inside the RQ framework as a problem of CVaR:

Risk Quadrangle (RQ) Framework:

- Five components: **Error**, **Regret**, **Risk**, **Deviation**, and **Statistic**. It Provides a holistic approach to uncertainty and performance management in SVR.

Unified Methodological Approach:

- Embeds CVaR into SVR, addressing tail risks and extreme deviations. Aligns machine learning objectives with robust risk management practices.

Reformulating ν-SVR with CVaR:

$$\min_{w,b} \; \lambda\|w\|^2 + \nu\,\bar{q}\left(|z(w,b)|\right), \quad \bar{q} = \langle\!\langle\, |z(w,b)|\, \rangle\!\rangle_{1-\nu}$$

where $\bar{q}(|z(w,b)|)$: CVaR of $|z(w,b)|$ at confidence level $1-\nu$.

- Reformulates as:

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

  with $\alpha = 1 - \nu$.

  $\overline{q}(|z(w,b)|)$ (the risk measure) is directly replaced with $\mathrm{CVaR}_\alpha(|y - Xw|)$,

  emphasizing the role of CVaR in measuring the risk associated with residuals.

SR as a Min-Max Representation

Now we begin to establish the equivalence between ν-SVR and SR. 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:

$$\min_{\mathbf{w},b,\epsilon} \left(\mathbb{E}\left[|Z(\mathbf{w},b)| - \epsilon \right]_+ + \nu\epsilon + \lambda \|\mathbf{w}\|^2 \right), \tag{1}$$

SR as a Min-Max Representation

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

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

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

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

By analyzing when $\alpha = 0$ and when $\alpha \to 1$, we can form the min-max:

$$\min_{\mathbf{w},b} \left(\max_i |y_i - \mathbf{w}^\top \mathbf{x}_i - b| + \lambda \|\mathbf{w}\|^2 \right).$$

Mapping SR to CVaR

- **Recapping the Important Theorems:**
  - **CVaR Optimization Formula:**

$$\bar{q}_\alpha(X) = \min_C \left\{ C + \frac{1}{1-\alpha}\,\mathbb{E}[X - C]_+ \right\}$$

    The set of minimizers is $q_\alpha(X)$.

  - **Dual CVaR Optimization Formula:**

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

    The set of maximizers is $[P(X < x),\, P(X \leq x)]$.

- **Understanding Constraints:**

SR Constraints:

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

z_i selects k data points with the largest errors.

SVR Constraints:

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

q_i distributes risk across all data points, bounded by $\frac{1}{n(1-\alpha)}$.

Establishing Equivalence: Aligning Parameters

Set

$$k = n(1-\alpha)$$

aligning the number of selected data points in SR with the confidence level α in SVR. This targets the same tail of the loss distribution in both methods.

Mapping z_i to q_i:

- Define $q_i = \frac{z_i}{k}$, where $z_i \in \{0,1\}$ and $\sum_{i=1}^{n} z_i = k$.
- Then:

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

- This ensures q_i satisfies SVR constraints:

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

Equivalence in Objective Functions

- **SVR Objective Function:**

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

- **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 $z_i = kq_i$ and choosing $k = n(1-\alpha)$ and $q_i = \frac{z_i}{k}$, the SR objective scales to:

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

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

This matches the SVR objective function, with adjusted regularization parameter $\frac{\lambda}{k}$. Now we will introduce the numerical analysis that helps empirically test this equivalency.

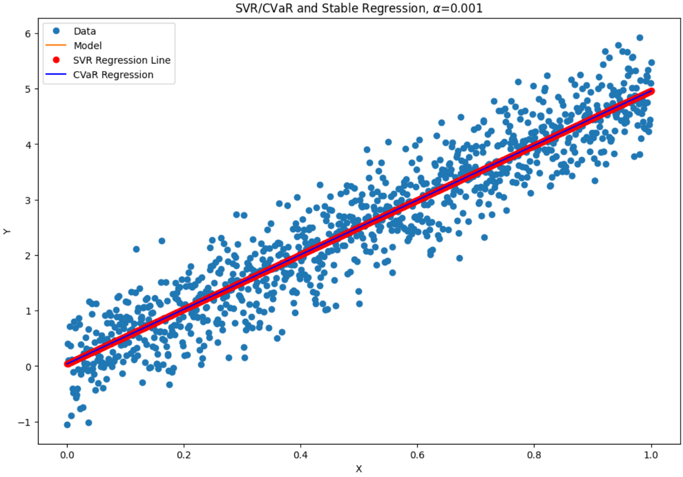
Numerical Analysis: General Algorithm

- General Algorithm for both regressions. SkLearn NuSVR method is used for the stable regression and PSG's CVaR_Risk is used for the SVR regression.

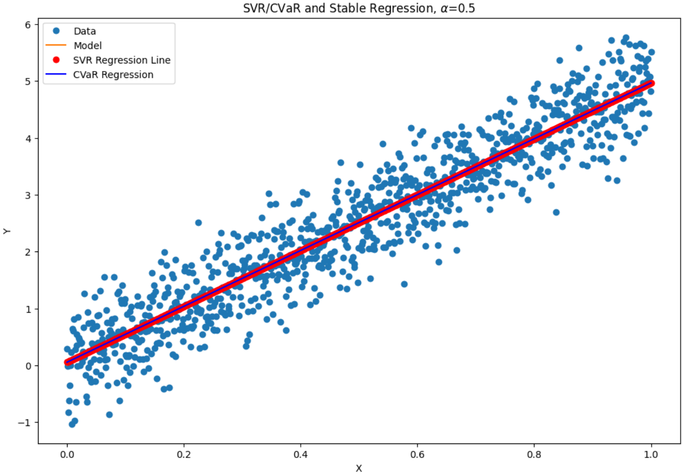
Algorithm 1 Workflow for Equivalent Regressions

```
Require: Dataset $\mathcal{D}$ with features $X$, target $y$, regularization $C$, and parameters $\alpha$.
Ensure: Coefficients $\beta$ and intercept $b$ for SVR and PSG.
Step 1: Preprocess Data
    • Normalize numerical features (StandardScaler).
    • Encode categorical features (OneHotEncoder).
Step 2: Train Models
    • SVR: Define SVR model with kernel $\mathcal{K}$, $C$, and $\nu$; train on $(X, y)$; extract $\beta_{\text{SVR}}$, $b_{\text{SVR}}$.
    • 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 $\beta_{\text{PSG}}$, $b_{\text{PSG}}$.
Step 3: Compare
    • Compare $\beta_{\text{SVR}}$ vs. $\beta_{\text{PSG}}$ and $b_{\text{SVR}}$ vs. $b_{\text{PSG}}$.
Step 4: Visualize (Optional)
    • Plot data with both regression lines.
```

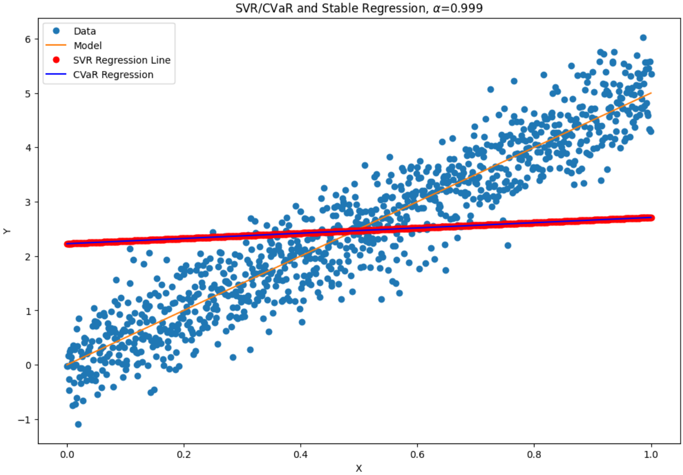
- SVR/CVaR and Stable Regression for $\alpha = 0.001$



- SVR/CVaR and Stable Regression for $\alpha = 0.5$



- SVR/CVaR and Stable Regression for $\alpha = 0.999$



Numerical Analysis: Multivariate Model

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

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

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

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

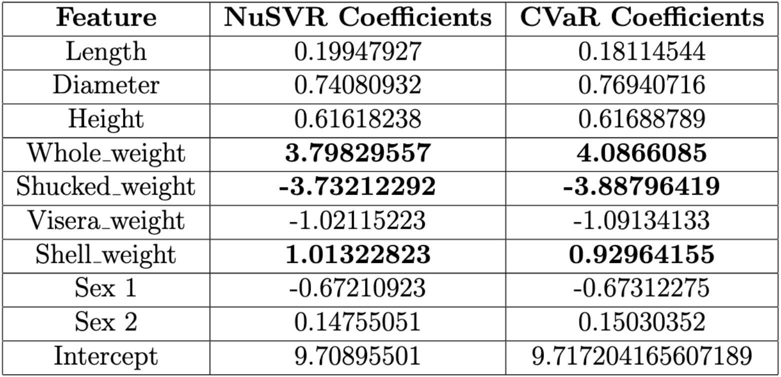
Table 4: Comparison of Regression Coefficients

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

Numerical Analysis: Abalone Model

- The UCI Abalone data is an 8 feature dataset, containing physical measurements of an Abalone. The target variable is the number of rings.

Table 5: Comparison of Regression Coefficients



Feature	**NuSVR Coefficients**	**CVaR Coefficients**
Length	0.19947927	0.18114544
Diameter	0.74080932	0.76940716
Height	0.61618238	0.61688789
Whole_weight	**3.79829557**	**4.0866085**
Shucked_weight	**-3.73212292**	**-3.88796419**
Visera_weight	-1.02115223	-1.09134133
Shell_weight	**1.01322823**	**0.92964155**
Sex 1	-0.67210923	-0.67312275
Sex 2	0.14755051	0.15030352
Intercept	9.70895501	9.717204165607189

Thank you!
Questions?