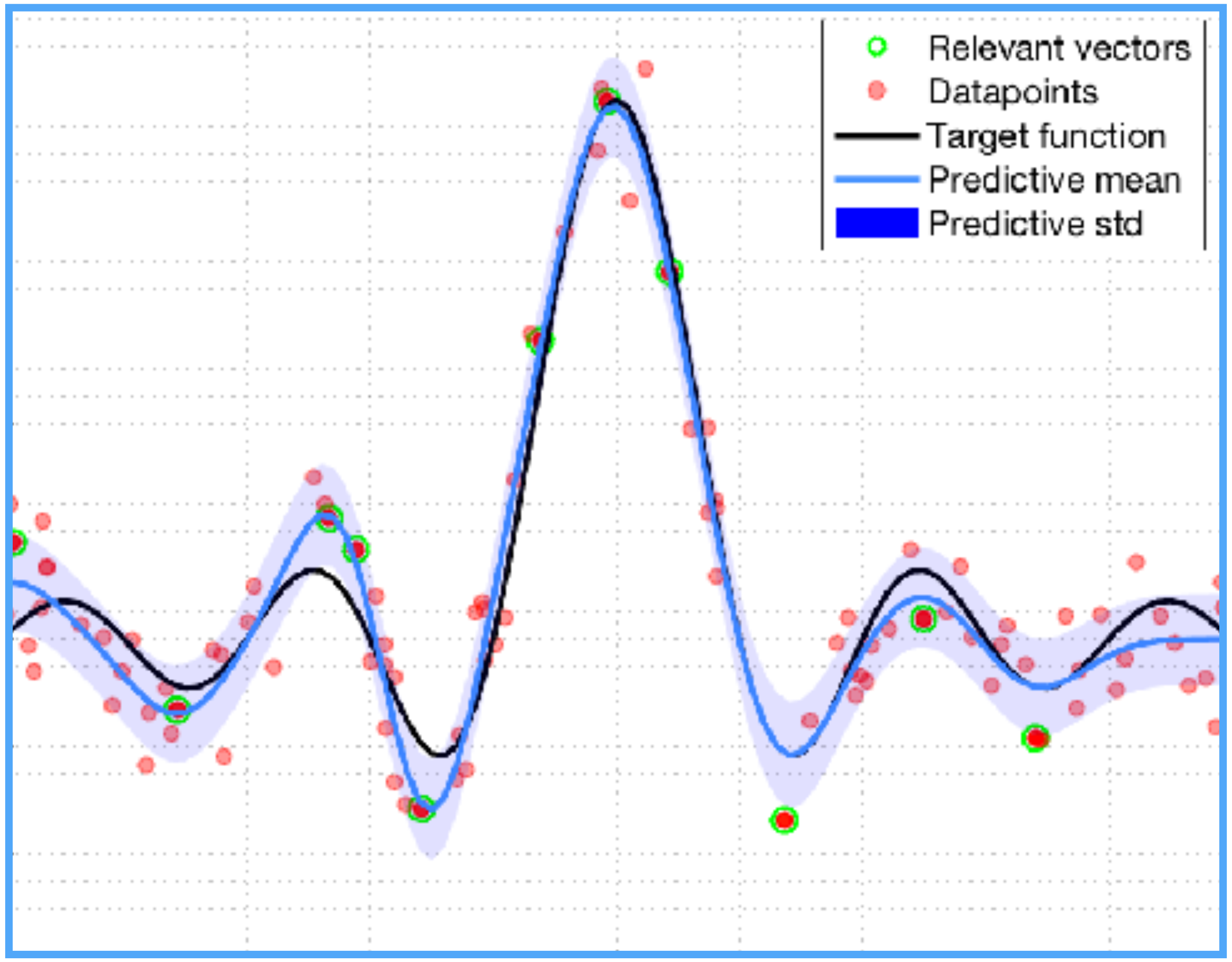
Support Vector Regression vs. Relevance Vector Regression

a sparsity / performance study



L.Faury G.Gallois-Montbrun H.Hendrikx

26/05/2017

Outline

- **Theoretical** reminders on both methods
- Introduction to a **sparse-regression** metric, experimental justification
- Sparse-regression metric based **cross-validation**
- Performance vs. sparsity discussion

Regression

Learn $f : \mathbb{R}^d \rightarrow \mathbb{R}$ thanks to a dataset $\{X, t\} \in (\mathbb{R}^d)^n \times \mathbb{R}^n$

Assuming a Gaussian **conditional p.d.f** around a linear transformation of features :

$$p(t \,|\, x, w) = \mathcal{N}(t \,| w^T \phi(x), \beta^{-1})$$

the maximum-likelihood estimator (MLE) writes :

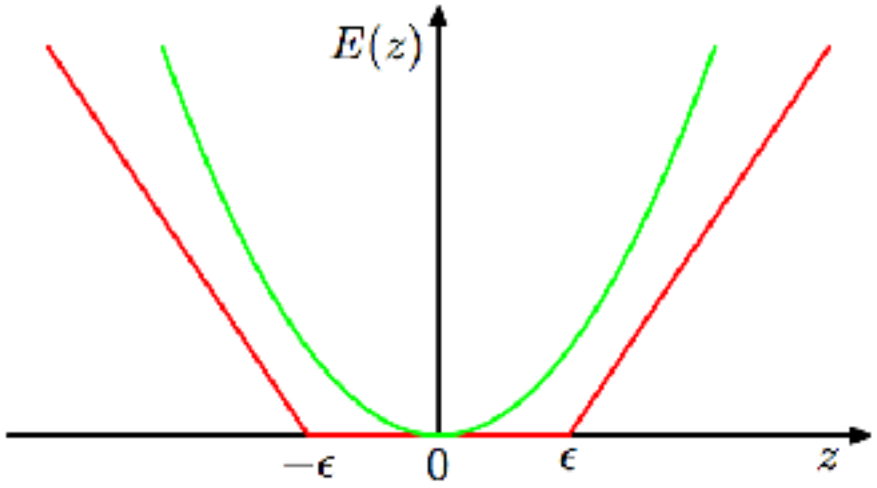
$$\begin{aligned} \hat{w} &= \text{argmax}_w \, p(t \,|\, X, w) \\ &= \text{argmin}_w \, \frac{1}{2} \sum_{i=1}^{n} \|w^t \phi(x) - t\|^2 \end{aligned}$$

- **Support Vector Regression**

[1] *Vladimir Vapnik*, The nature of statistical learning theory, *1995*

Support Vector Regression

- Introduce the ε-insensitive[1] loss-function.



Source : Bishop, Pattern Recognition and Machine Learning (2006)

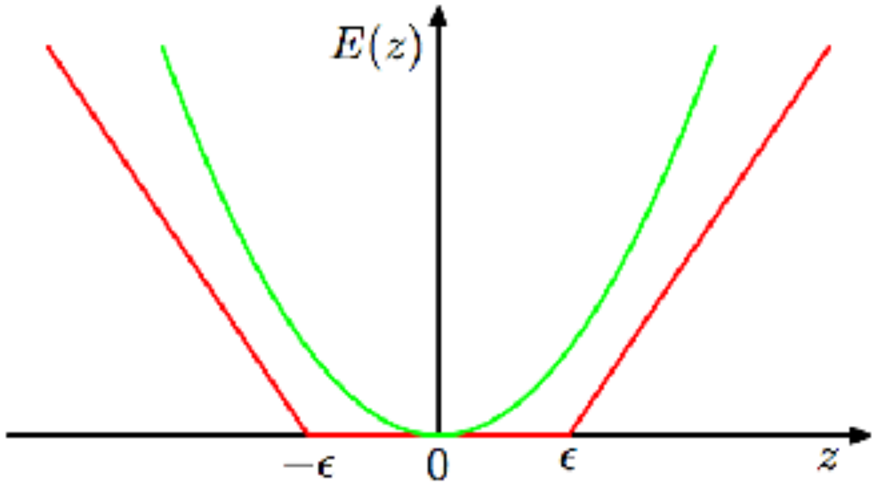
[1] *Vladimir Vapnik*, The nature of statistical learning theory, *1995*

Support Vector Regression

- Introduce the ε-insensitive[1] loss-function.

$$min_w \frac{C}{n}\sum_n(\xi_n + \hat{\xi}_n) + \frac{1}{2}\|w\|^2$$

$$\text{s.t} \quad \begin{cases} \xi, \hat{\xi} \geq 0 \\ w^T\phi(x_n) + \xi_n + \varepsilon \geq t_n \\ w^T\phi(x_n) - \hat{\xi}_n - \varepsilon \leq t_n \end{cases}$$



Source : Bishop, Pattern Recognition and Machine Learning (2006)

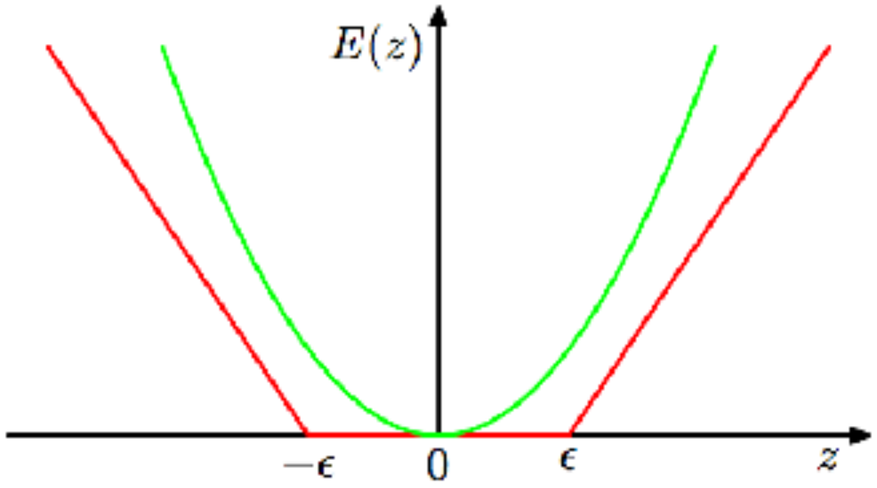
[1] *Vladimir Vapnik*, The nature of statistical learning theory, *1995*

Support Vector Regression

- Introduce the ε-insensitive[1] loss-function.

$$min_w \frac{C}{n}\sum_n(\xi_n + \hat{\xi}_n) + \frac{1}{2}\|w\|^2$$

$$\text{s.t} \quad \begin{cases} \xi, \hat{\xi} \geq 0 \\ w^T\phi(x_n) + \xi_n + \varepsilon \geq t_n \\ w^T\phi(x_n) - \hat{\xi}_n - \varepsilon \leq t_n \end{cases}$$



Source : Bishop, Pattern Recognition and Machine Learning (2006)

- Only points outside the ε-tube (**active constraints**) are used for predictions :

$$y(x) = \textstyle\sum_{n\in\mathcal{S}}(a_n - \hat{a}_n)k(x, x_n) \longrightarrow \text{Posterior } \textbf{decision}$$

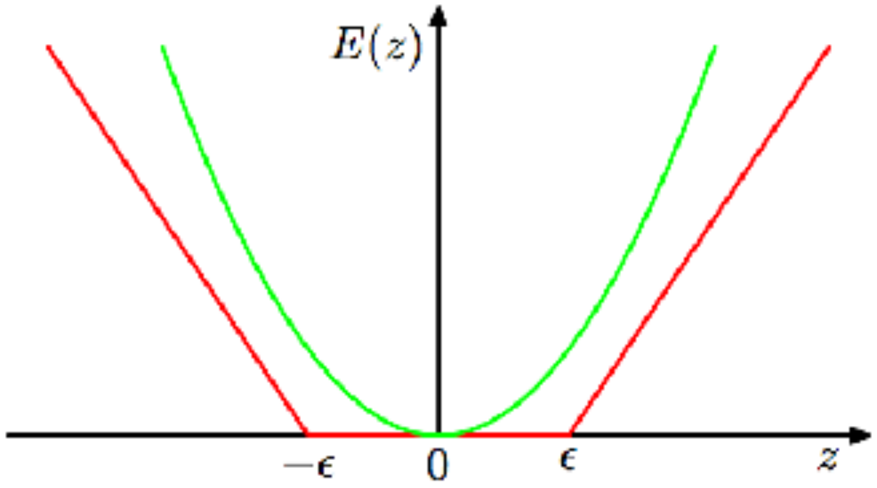
[1] *Vladimir Vapnik,* The nature of statistical learning theory, *1995*

Support Vector Regression

- Introduce the ε-insensitive[1] loss-function.

$$min_w \boxed{C} \sum_n (\xi_n + \hat{\xi}_n) + \frac{1}{2}\|w\|^2$$

$$\text{s.t} \quad \begin{cases} \xi, \hat{\xi} \geq 0 \\ w^T\phi(x_n) + \xi_n + \boxed{\varepsilon} \geq t_n \\ w^T\phi(x_n) - \hat{\xi}_n - \varepsilon \leq t_n \end{cases}$$



Source : Bishop, Pattern Recognition and Machine Learning (2006)

- Only points outside the ε-tube (**active constraints**) are used for predictions :

$$y(x) = \sum_{n \in \mathcal{S}} (a_n - \hat{a}_n) \boxed{k(x, x_n)} \longrightarrow \text{Posterior } \textbf{decision}$$

[1] *Vladimir Vapnik,* The nature of statistical learning theory, *1995*

Relevance Vector Regression[2]

[2] *Tipping Michael,* Sparse Bayesian learning and the relevance vector machine, *Journal of machine learning research, 2001*

Relevance Vector Regression[2]

- Provide the predictor with a Gaussian prior : $w \sim \prod_i \mathcal{N}(w_i \,|\, 0, \alpha_i^{-1})$

$$y(x) = \sum_n w_n k(x, x_n)$$

[2] *Tipping Michael,* Sparse Bayesian learning and the relevance vector machine, *Journal of machine learning research, 2001*

Relevance Vector Regression[2]

- Provide the predictor with a Gaussian prior : $w \sim \prod_i \mathcal{N}(w_i \,|\, 0, \alpha_i^{-1})$

$$y(x) = \sum_n w_n k(x, x_n)$$

- Use **type-2 likelihood** *(evidence approximation)* to determine :

$$(\alpha^*, \beta^*) = \text{argmax}_{\alpha,\beta} \left[p(t \,|\, \alpha, \beta) = \int_w p(t \,|\, w, \beta) p(w \,|\, \alpha) \right]$$

[2] *Tipping Michael,* Sparse Bayesian learning and the relevance vector machine, *Journal of machine learning research, 2001*

Relevance Vector Regression[2]

- Provide the predictor with a Gaussian prior : $w \sim \prod_i \mathcal{N}(w_i \,|\, 0, \alpha_i^{-1})$

$$y(x) = \sum_n w_n k(x, x_n)$$

- Use **type-2 likelihood** *(evidence approximation)* to determine :

$$(\alpha^*, \beta^*) = \text{argmax}_{\alpha,\beta} \left[p(t \,|\, \alpha, \beta) = \int_w p(t \,|\, w, \beta) p(w \,|\, \alpha) \right]$$

- Automatic Relevance Detection : drives some α_i to $+\infty$ (*sparse model*). Others are called **relevant** vectors.

[2] *Tipping Michael,* Sparse Bayesian learning and the relevance vector machine, *Journal of machine learning research, 2001*

Relevance Vector Regression[2]

- Provide the predictor with a Gaussian prior : $w \sim \prod_i \mathcal{N}(w_i \,|\, 0, \alpha_i^{-1})$

$$y(x) = \sum_n w_n k(x, x_n)$$

- Use **type-2 likelihood** *(evidence approximation)* to determine :

$$(\alpha^*, \beta^*) = \text{argmax}_{\alpha,\beta} \left[p(t \,|\, \alpha, \beta) = \int_w p(t \,|\, w, \beta) p(w \,|\, \alpha) \right]$$

- Automatic Relevance Detection : drives some α_i to $+\infty$ (*sparse model*). Others are called **relevant** vectors.

- Compute posterior and **predictive distribution**

[2] *Tipping Michael,* Sparse Bayesian learning and the relevance vector machine, *Journal of machine learning research, 2001*

Relevance Vector Regression(2)

- Provide the predictor with a Gaussian prior : $w \sim \prod_i \mathcal{N}(w_i \,|\, 0, \alpha_i^{-1})$

$$y(x) = \sum_n w_n \boxed{k(x, x_n)}$$

- Use **type-2 likelihood** *(evidence approximation)* to determine :

$$(\alpha^*, \beta^*) = \operatorname{argmax}_{\alpha,\beta} \left[p(t \,|\, \alpha, \beta) = \int_w p(t \,|\, w, \beta) p(w \,|\, \alpha) \right]$$

- Automatic Relevance Detection : drives some α_i to $+\infty$ (*sparse model*). Others are called **relevant** vectors.

- Compute posterior and **predictive distribution**

[2] *Tipping Michael,* Sparse Bayesian learning and the relevance vector machine, *Journal of machine learning research, 2001*

Comparison

SVR

RVR

[3] *John Platt,* Sequential Minimal Optimization : A fast algorithm for training support vector machines. *1998*

Comparison

SVR	RVR
▶ Decision choices	▶ Posterior probability

[3] *John Platt*, Sequential Minimal Optimization : A fast algorithm for training support vector machines. *1998*

Comparison

SVR

- Decision choices
- Held-out method for hyper-parameters (at least 3)

RVR

- Posterior probability
- Hyper-parameters are determined automatically (except kernel)

[3] *John Platt*, Sequential Minimal Optimization : A fast algorithm for training support vector machines. *1998*

Comparison

SVR	RVR
Decision choices	Posterior probability
Held-out method for hyper-parameters (at least 3)	Hyper-parameters are determined automatically (except kernel)
Mercer kernel	Arbitrary base functions

[3] *John Platt*, Sequential Minimal Optimization : A fast algorithm for training support vector machines. *1998*

Comparison

SVR	RVR
Decision choices	Posterior probability
Held-out method for hyper-parameters (at least 3)	Hyper-parameters are determined automatically (except kernel)
Mercer kernel	Arbitrary base functions
Training : SMO(3) (somewhere between linear and quadratic)	**Training** : cubic complexity

[3] *John Platt*, Sequential Minimal Optimization : A fast algorithm for training support vector machines. *1998*

Comparison

SVR

- Decision choices
- Held-out method for hyper-parameters (at least 3)
- Mercer kernel
- **Training** : SMO(3) (somewhere between linear and quadratic)
- **Testing** : linear in the SV

RVR

- Posterior probability
- Hyper-parameters are determined automatically (except kernel)
- Arbitrary base functions
- **Training** : cubic complexity
- **Testing** : linear in the RV

[3] *John Platt*, Sequential Minimal Optimization : A fast algorithm for training support vector machines. *1998*

Comparison

SVR	RVR
Decision choices	Posterior probability
Held-out method for hyper-parameters (at least 3)	Hyper-parameters are determined automatically (except kernel)
Mercer kernel	Arbitrary base functions
Training : SMO(3) (somewhere between linear and quadratic)	**Training** : cubic complexity
Testing : linear in the SV	**Testing** : linear in the RV

[3] *John Platt*, Sequential Minimal Optimization : A fast algorithm for training support vector machines. *1998*

- **Sparsity / Performance**

[4] *Christopher Bishop,* Pattern Recognition Machine Learning, *2006*

Sparsity / Performance

- **Question** : Compare the tradeoff found between performance (MSE minimization) and sparsity

[4] *Christopher Bishop,* Pattern Recognition Machine Learning, *2006*

Sparsity / Performance

- **Question** : Compare the tradeoff found between performance (MSE minimization) and sparsity

- **Literature**[4] : RVR reaches sparser models with equivalent generalization skills.

[4] *Christopher Bishop,* Pattern Recognition Machine Learning, *2006*

Sparsity / Performance

- **Question** : Compare the tradeoff found between performance (MSE minimization) and sparsity

- **Literature**[4] : RVR reaches sparser models with equivalent generalization skills.

- **Initial idea** : Test (**experimentally**) this assertion
  - what is performance ?
  - what do we want with sparsity ?
  - how to measure the tradeoff ?

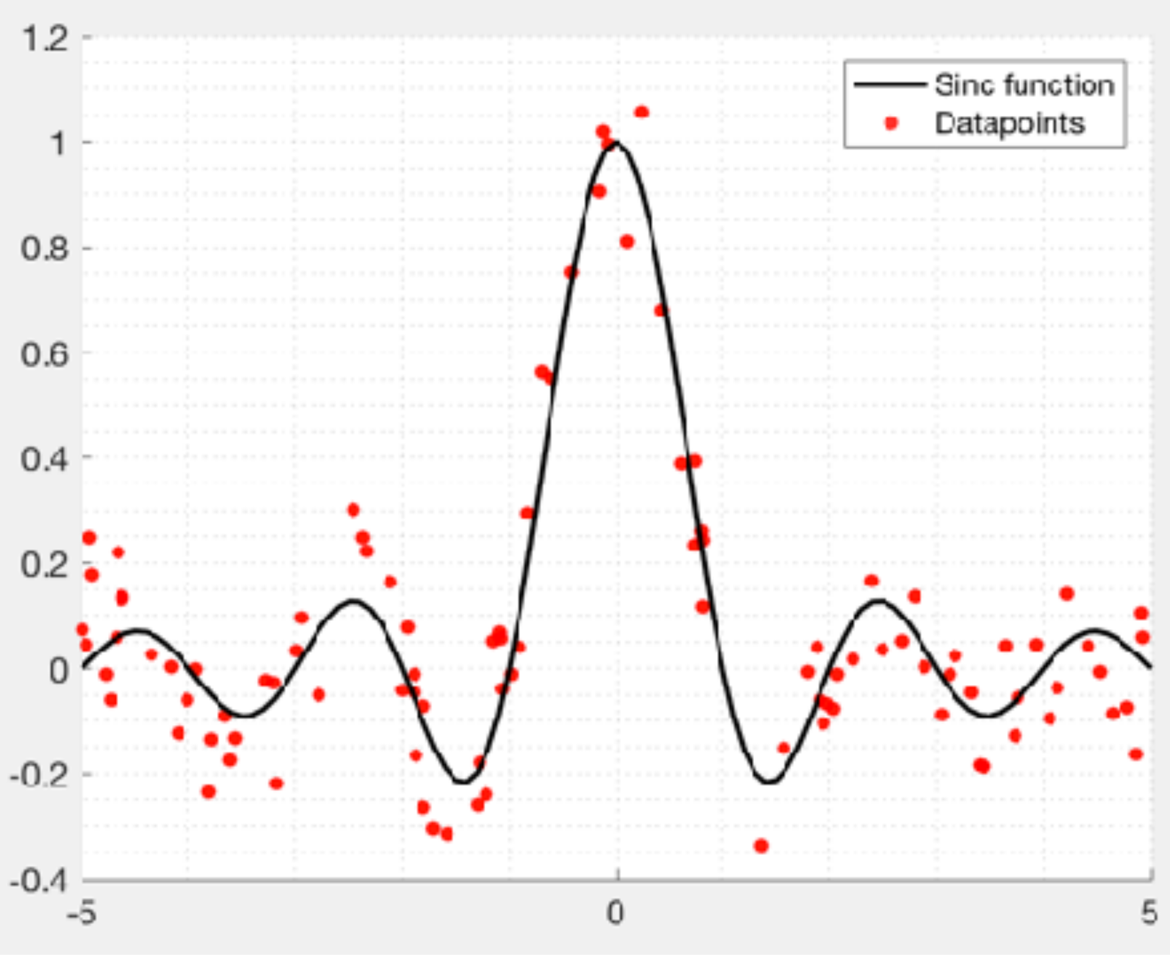
[4] *Christopher Bishop*, Pattern Recognition Machine Learning, *2006*

- **Datasets**

[5] *T.F. Brooks, D.S. Pope, and A.M. Marcolini.* Airfoil self-noise and prediction. *Technical report NASA. 1989.*

Datasets

Artificial (1d)

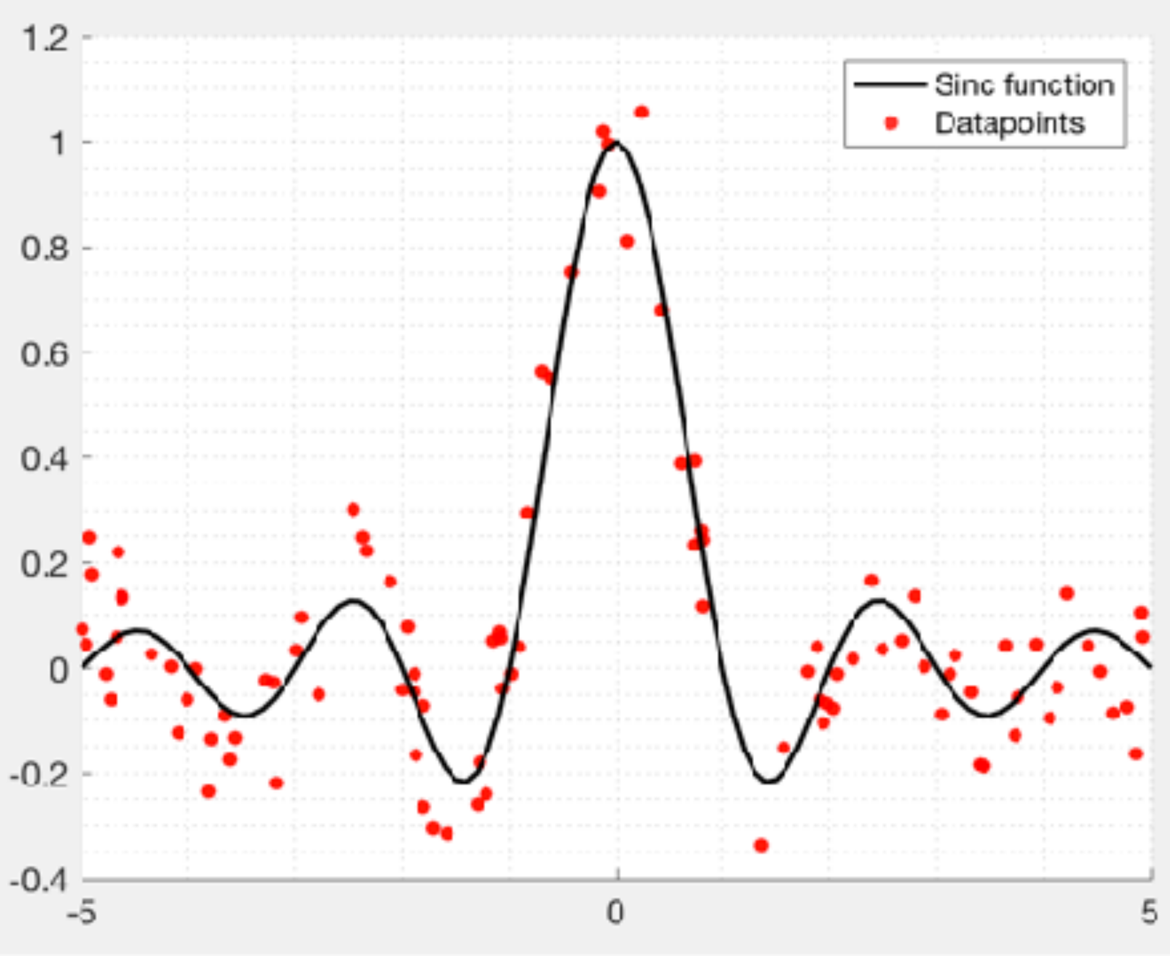


Dimension	Points	Support	Noise variance	Outlier
1	100	[-5,5]	0.01	No

[5] *T.F. Brooks, D.S. Pope, and A.M. Marcolini.* Airfoil self-noise and prediction. *Technical report NASA. 1989.*

Datasets

Artificial (1d)



Dimension	Points	Support	Noise variance	Outlier
1	100	[-5,5]	0.01	No

Real (5d)

- Airfoil Self-Noise Data Set (NASA)[5]

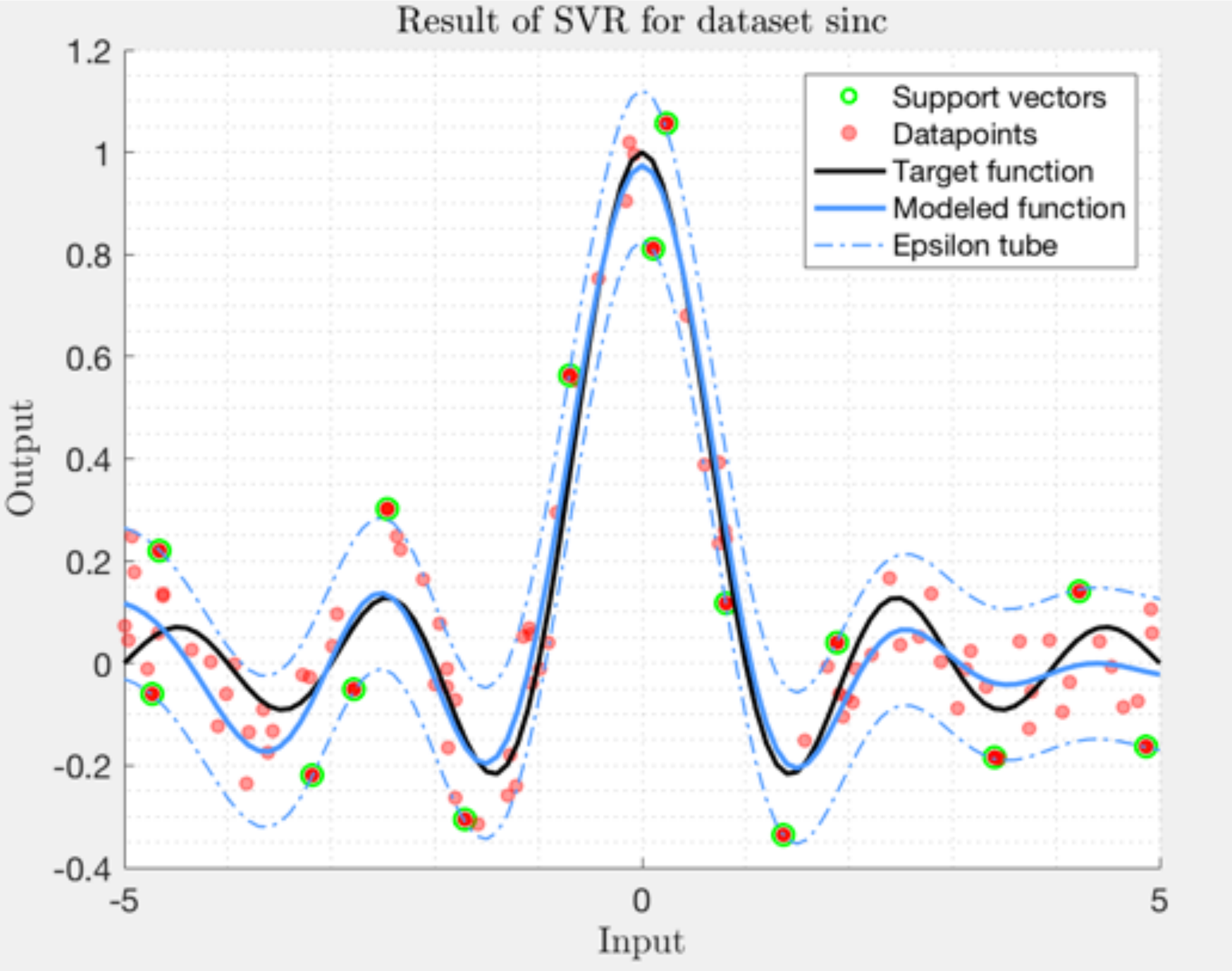
Dimension	Points
5	1503

- Predict sound pressure (dB) according to few features :
  - Eigen frequency
  - Angle of attack
  - Chord Length
  - Free stream
  - Suction side displacement thickness

[5] *T.F. Brooks, D.S. Pope, and A.M. Marcolini.* Airfoil self-noise and prediction. *Technical report NASA. 1989.*

- **Test Run**

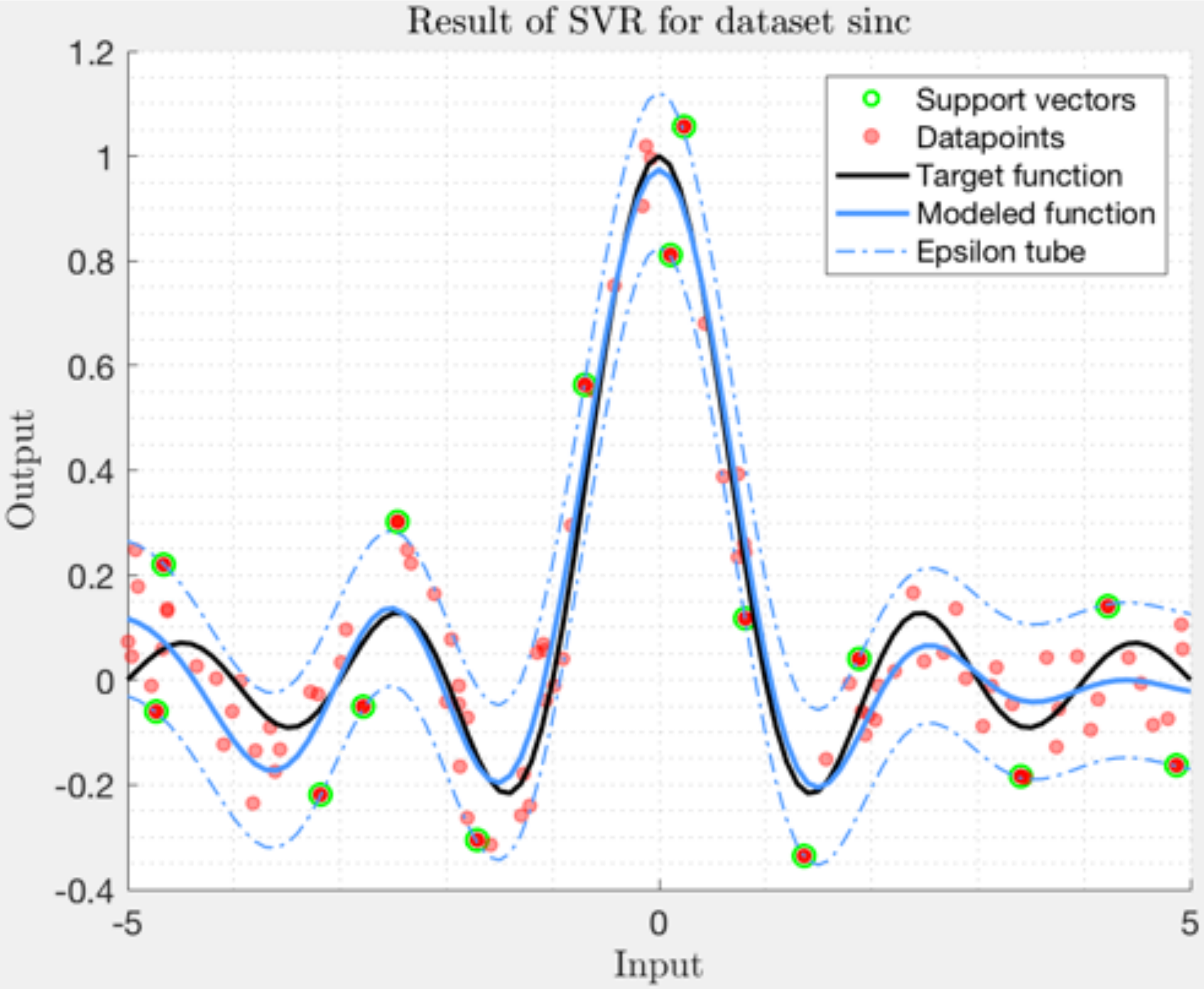
Test Run

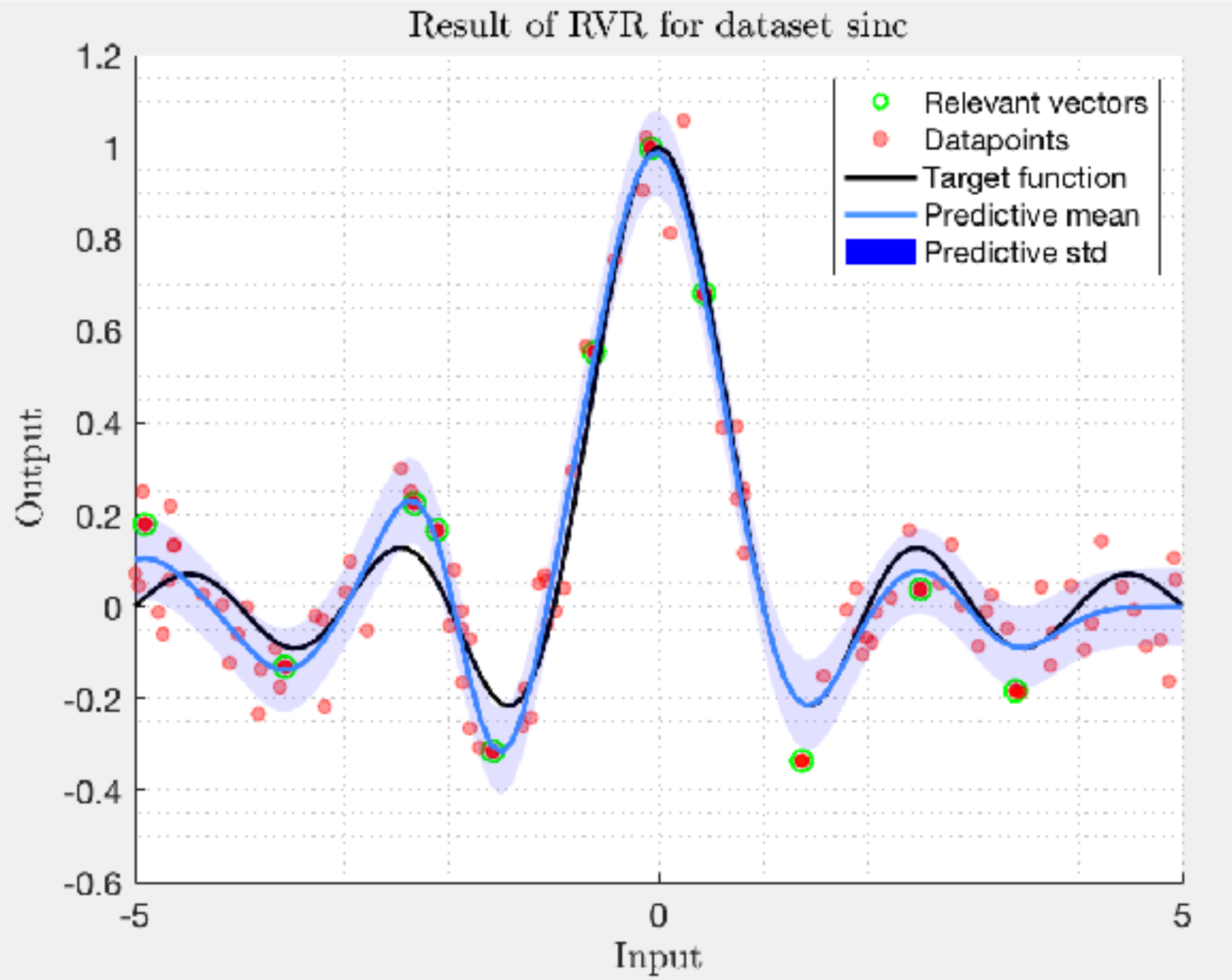


$\nu-$SVR, RBF kernel with:

$$\begin{cases} \nu & = 0.08 \\ C & = 8.5 \\ \sigma & = 1.4 \quad \text{(kernel width)} \end{cases}$$

Test Run





$\nu-$SVR, RBF kernel with:

$$\begin{cases} \nu & = 0.08 \\ C & = 8.5 \\ \sigma & = 1.4 \quad \text{(kernel width)} \end{cases}$$

RVR, RBF kernel with:

$$\sigma = 1$$

- **Intuition**

Intuition

- **Goal**: Maximize performance while penalizing complexity

Intuition

- **Goal**: Maximize performance while penalizing complexity
- BIC (Bayesian Information Criterion) for model selection:

$$BIC = -2 \underbrace{\log \mathcal{L}(x)}_{\text{likelihood}} + \underbrace{M \log N}_{\text{model complexity}}$$

Intuition

- **Goal**: Maximize performance while penalizing complexity

- BIC (Bayesian Information Criterion) for model selection:

$$BIC = -2 \underbrace{\log \mathcal{L}(x)}_{\text{likelihood}} + \underbrace{M \log N}_{\text{model complexity}}$$

- Adaptation to regression :

$$-\log \mathcal{L}(x) = \beta N \cdot MSE(x, t)$$ Gaussian likelihood

$$M \overset{\triangle}{\sim} |SV| = k$$ complexity = number of support vectors

Intuition

- **Goal**: Maximize performance while penalizing complexity

- BIC (Bayesian Information Criterion) for model selection:

$$BIC = -2 \underbrace{\log \mathcal{L}(x)}_{\text{likelihood}} + \underbrace{M \log N}_{\text{model complexity}}$$

- Adaptation to regression :

$$-\log \mathcal{L}(x) = \beta N \cdot MSE(x, t)$$

Gaussian likelihood

$$M \overset{\triangle}{\approx} |SV| = k$$

complexity = number of support vectors

- BICSR (BIC for Sparse Regression)

$$\boxed{BICSR = \beta N \cdot MSE + k \log N}$$

- **Experimental Evaluation**

Experimental Evaluation

- **Goal**: Evaluate the tradeoff found by the BICSR metric

Experimental Evaluation

- **Goal**: Evaluate the tradeoff found by the BICSR metric

  - For each method (SVR and RVR) :
    - Cross-validation to find the best hyper-parameters according to BICSR and MSE
    - Compare them with arbitrary models

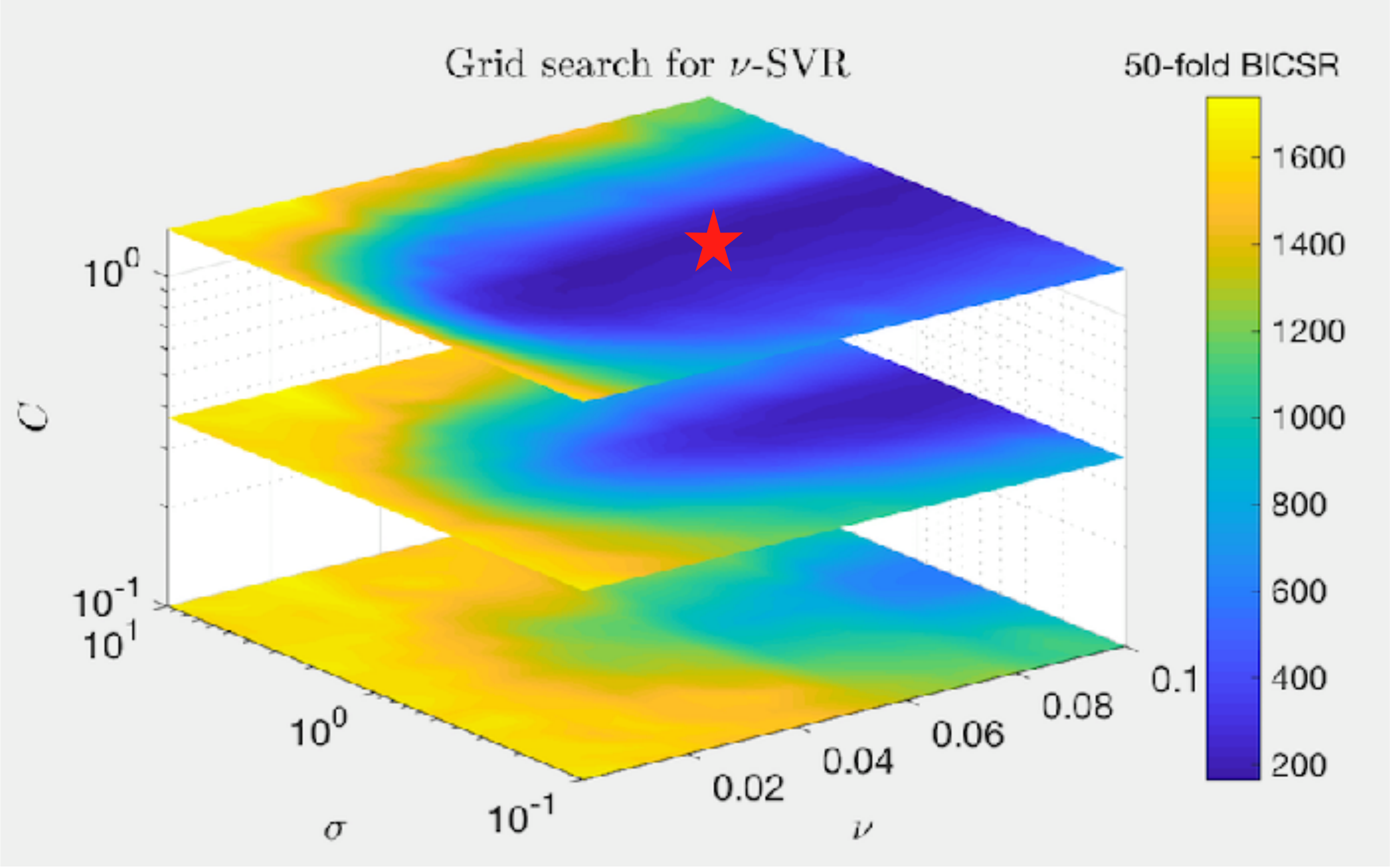
- **Best hyper-parameters selection**

Best hyper-parameters selection

- **Example for SVR with BICSR:**

Best hyper-parameters selection

- **Example for SVR with BICSR:**



Best hyper-parameters selection

- **Example for SVR with BICSR:**

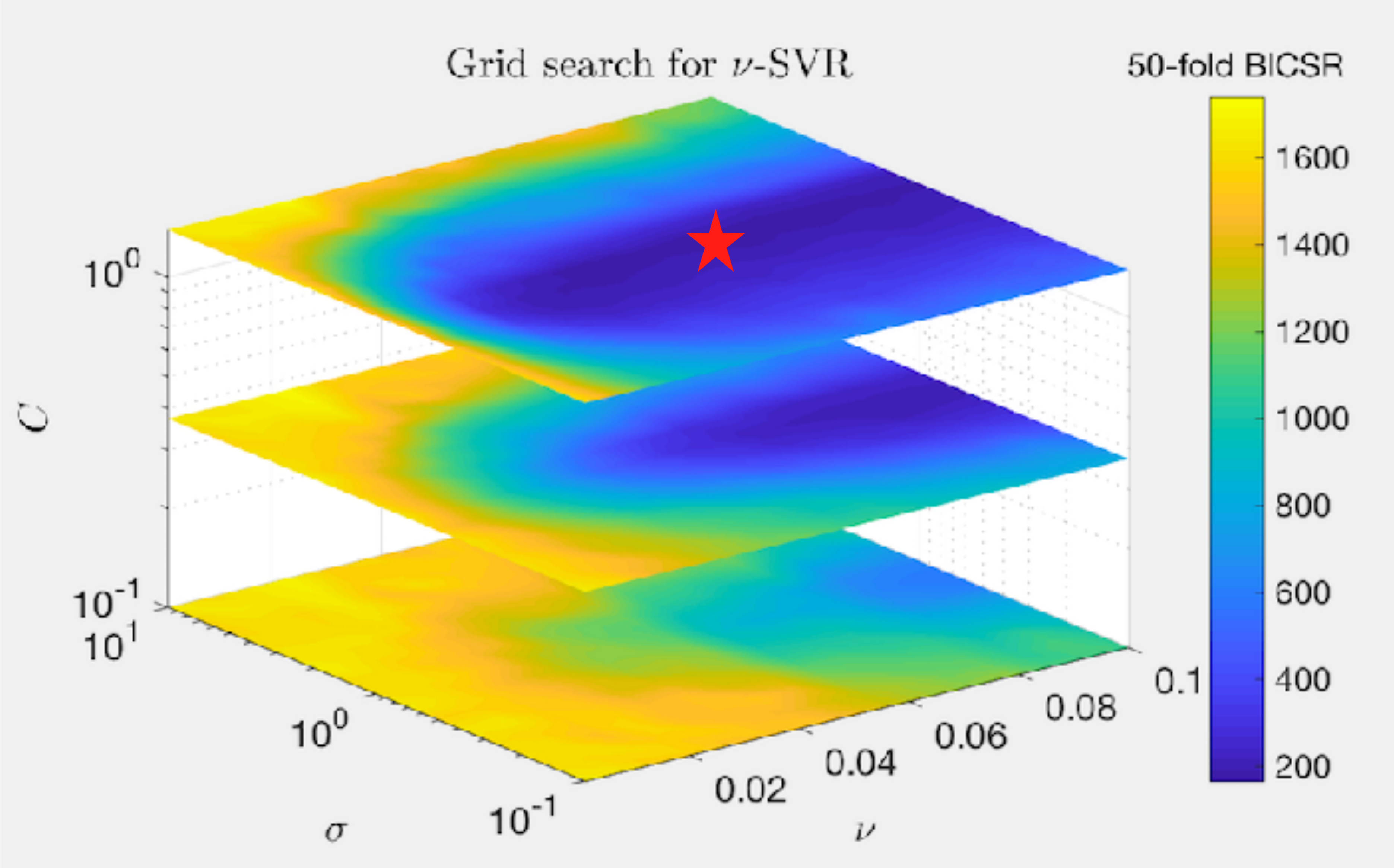
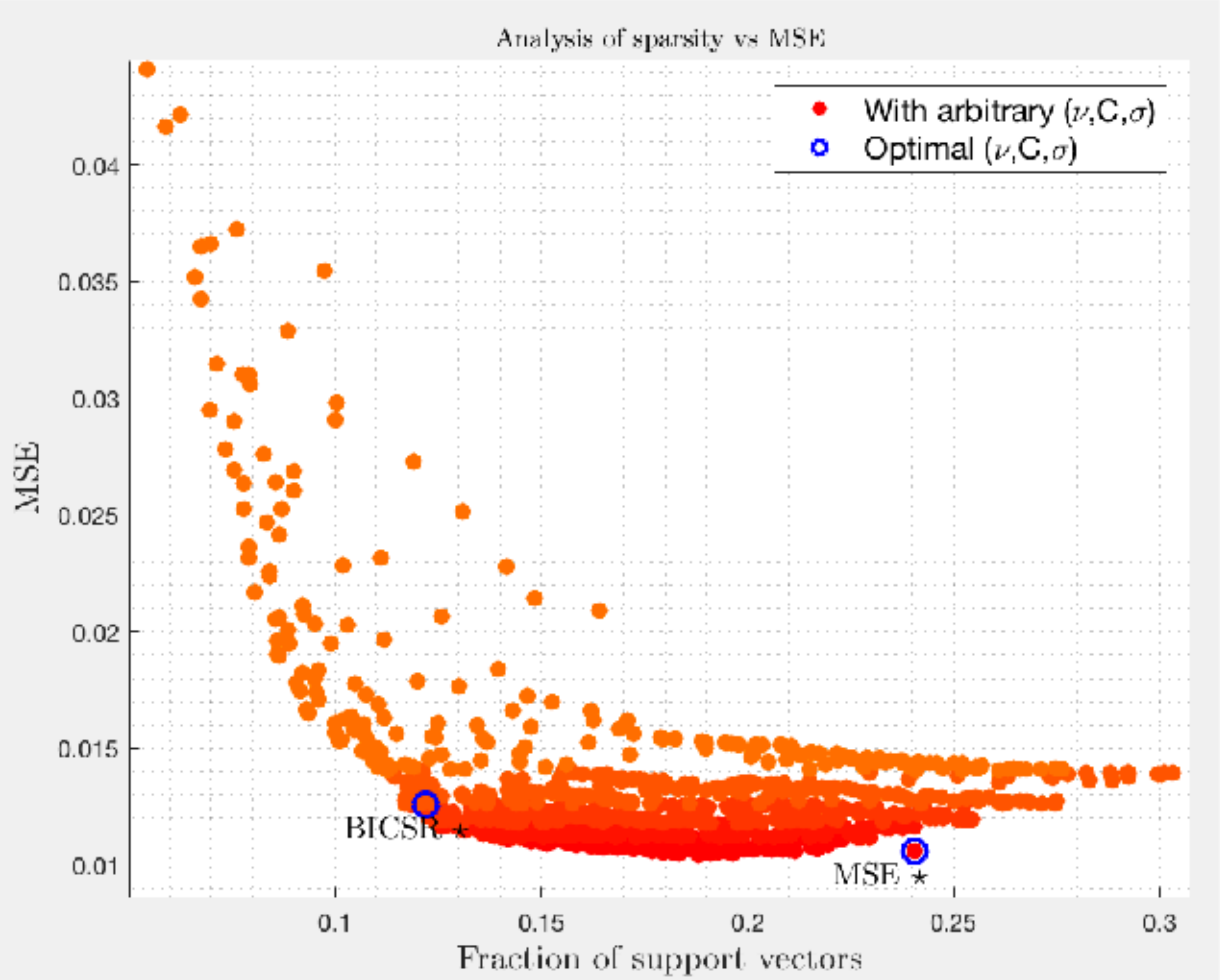


Figure : 50-fold cross-validation (0.75 training/test ratio)

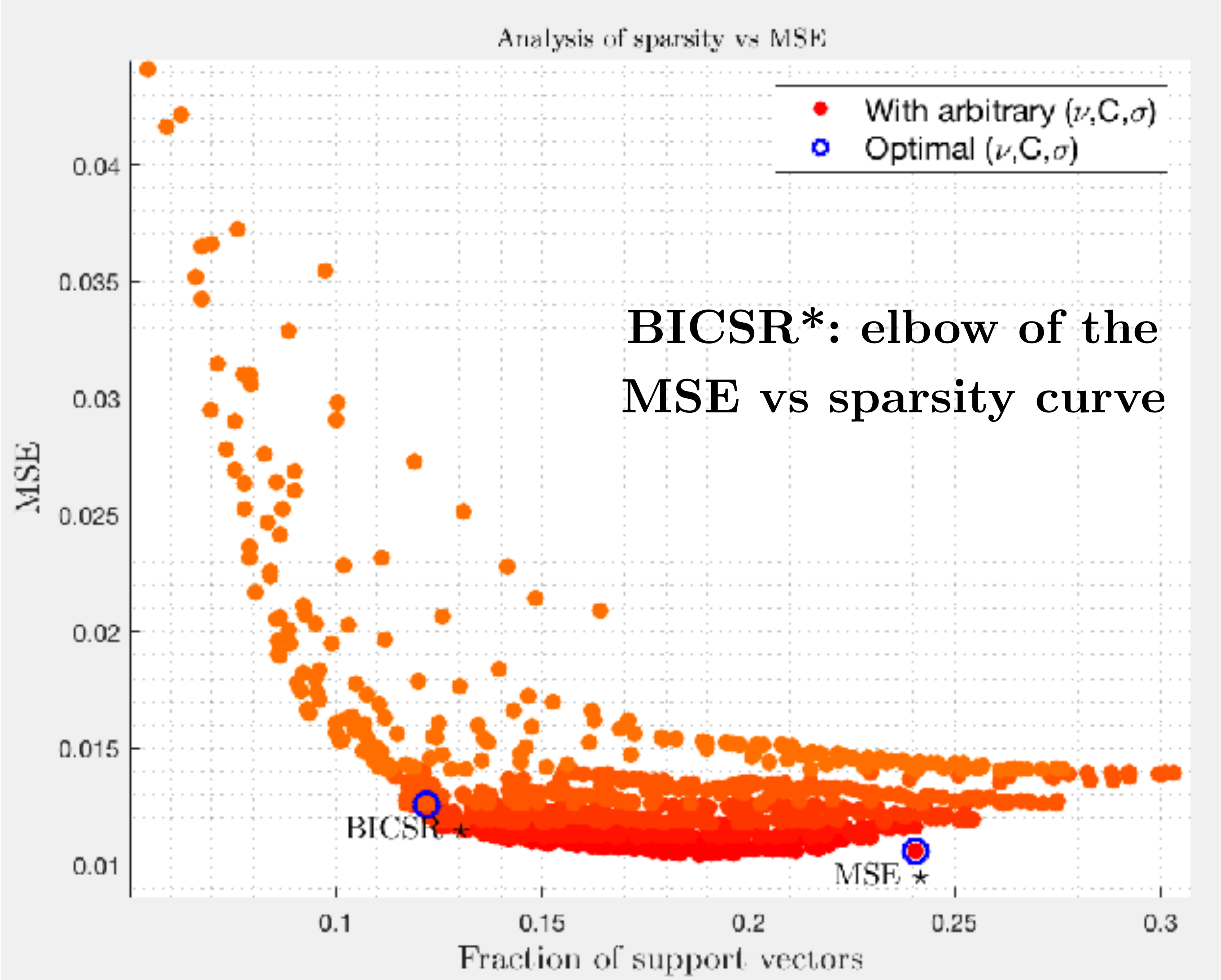
Tradeoff evaluation (artificial dataset)

- **Example for SVR** (50 fold, 75 training/test ratio)**:**



Tradeoff evaluation (artificial dataset)

- **Example for SVR** (50 fold, 75 training/test ratio)**:**



Tradeoff evaluation (artificial dataset)

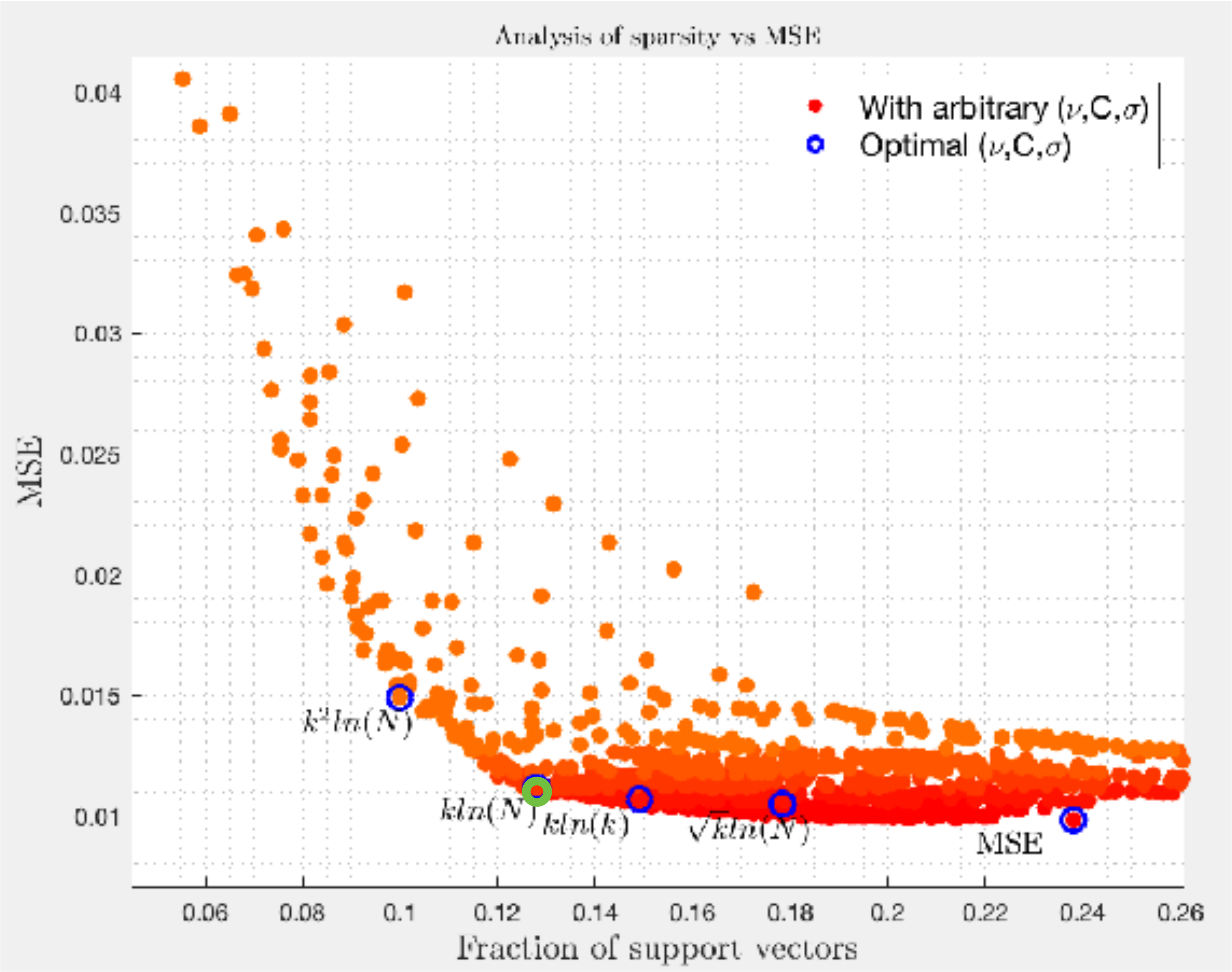
- **Can we do better (different penalization) ?**

$$BICSR = \beta^{-1} N \cdot MSE + \boldsymbol{k} \log \boldsymbol{N}$$

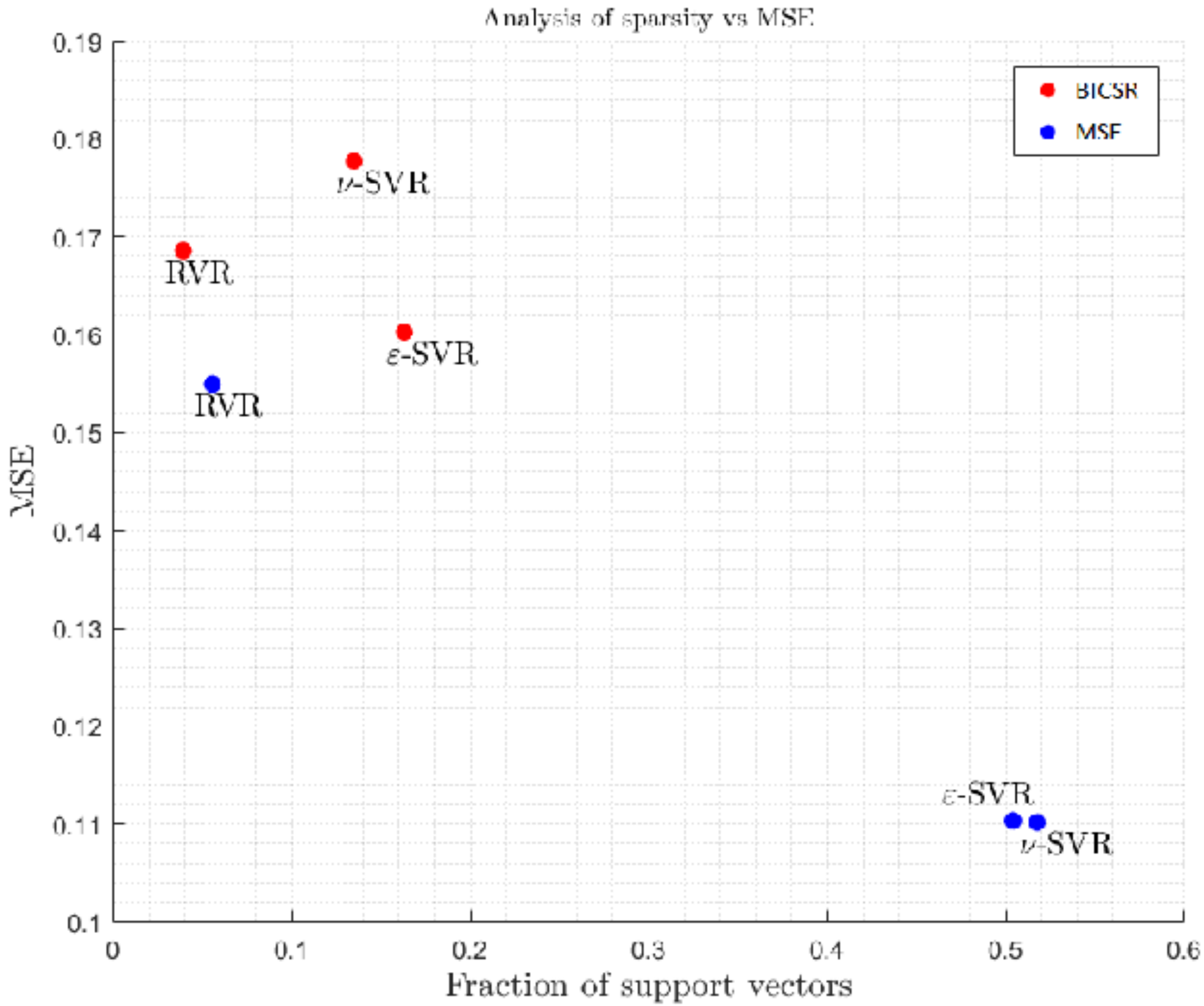
Tradeoff evaluation (artificial dataset)

- **Can we do better (different penalization) ?**

$$BICSR = \beta^{-1}N \cdot MSE + \boldsymbol{k} \log \boldsymbol{N}$$



Model Comparaison (real dataset)



Conclusions

- BICSR seems to be a well-behaved sparse-regression metric
(tradeoff between sparsity and performance)

- Even without sparsity penalization, RVR finds a fairly good compromise
  - most suited for fast predictions !

- SVR can be tuned to achieve either high sparsity or high regression performance

Other aspects :

- Behavior far from data
- Training cost
- Decision theory for predictions (predictive distribution)
- ..

Thank you for your attention !