ADVANCED MACHINE LEARNING

Nonlinear Regression – Part I

Interactive lecture

SVR with polynomial kernel
Relevance Vector Regression
Ridge Regression

ε-Support Vector Regression

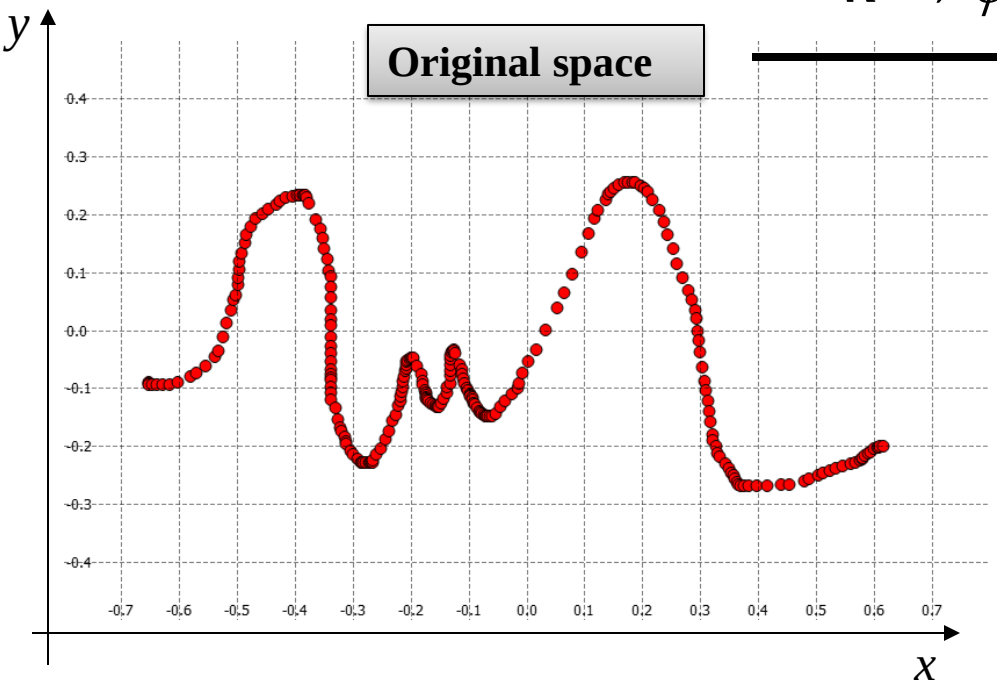
ε-Support Vector Regression: Recap

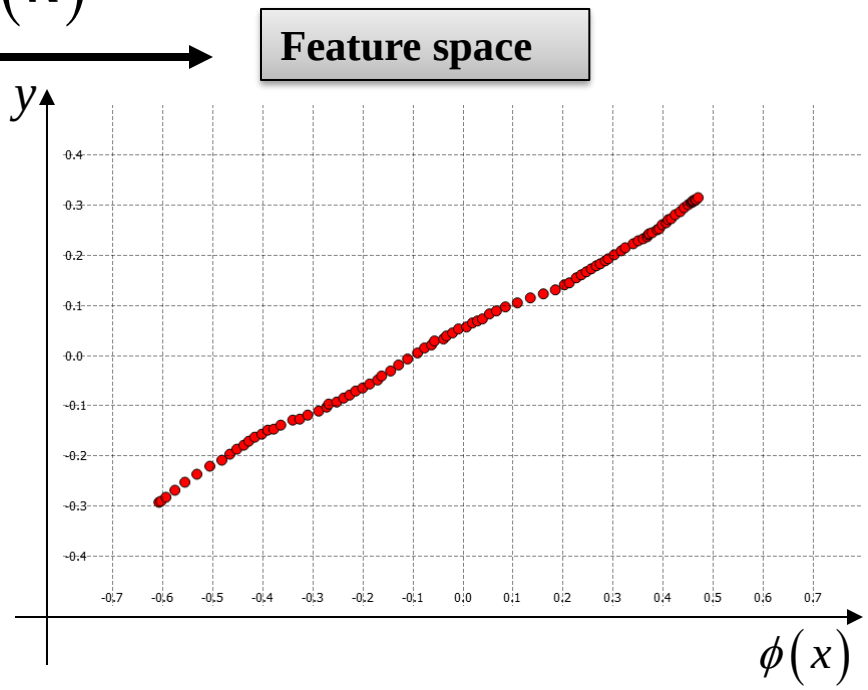
Transform the problem into non-linear regression using the kernel trick:

Inner product in feature space

$$y = f(x) = \sum_{i=1}^{M} \alpha_i \left\langle \phi(x^i), \phi(x) \right\rangle + b$$

$$x \rightarrow \phi(x)$$



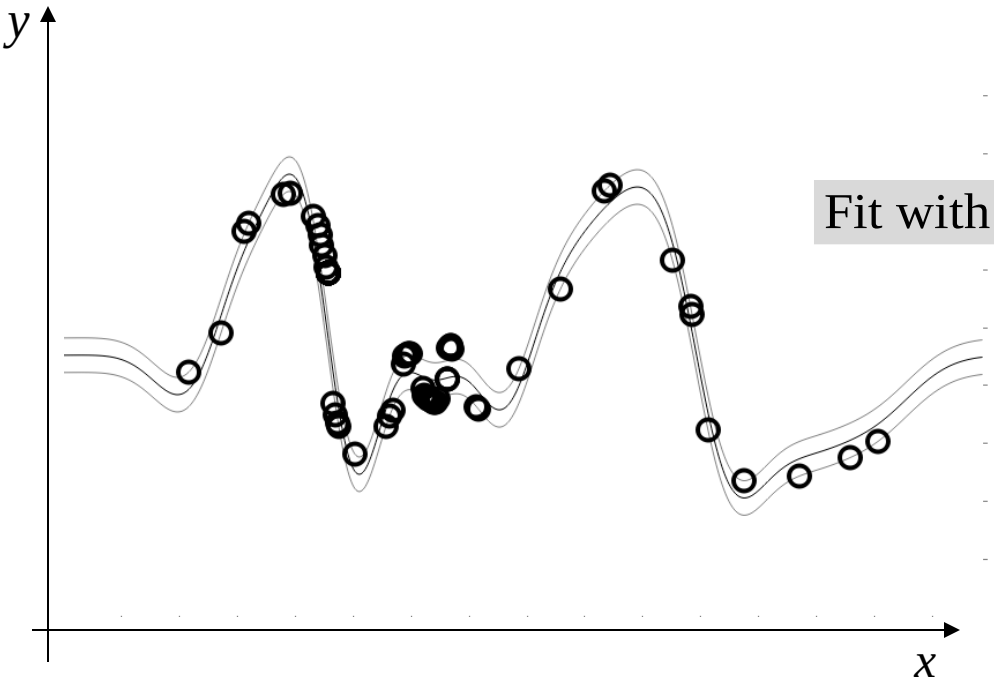


ε-Support Vector Regression: Recap

Transform the problem into non-linear regression using the kernel trick:

$$y = f(x) = \sum_{i=1}^{M} \alpha_i k(x^i, x) + b$$

Kernel computes inner product in feature space



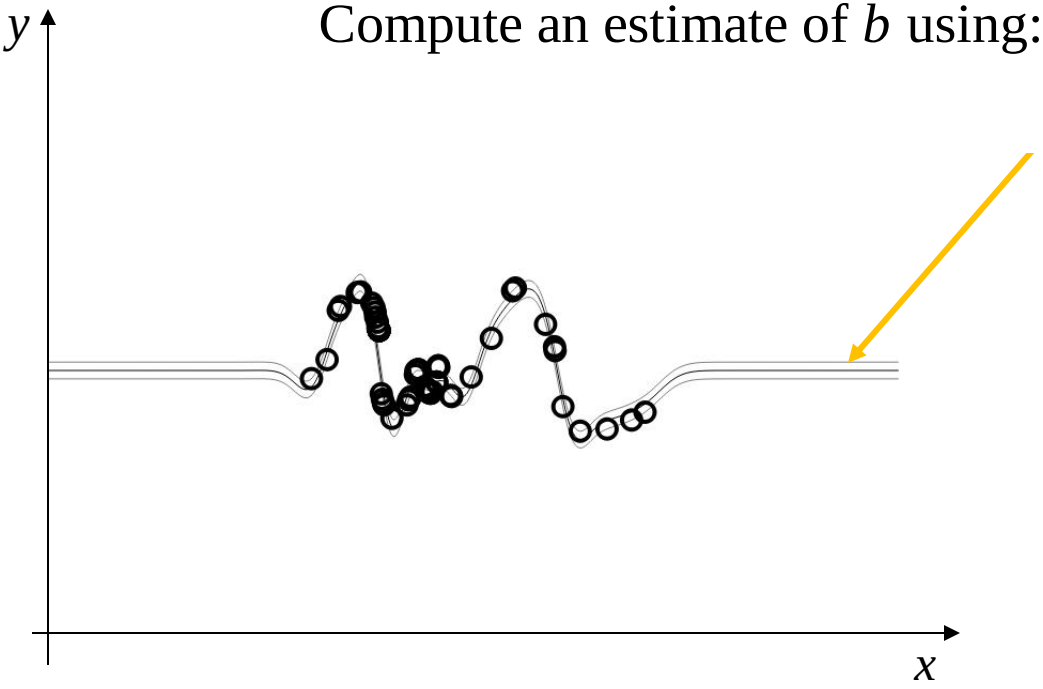
Fit with a RBF kernel

ε-Support Vector Regression: Recap

Transform the problem into non-linear regression using the kernel trick:

$$y = f(x) = \sum_{i=1}^{M} \alpha_i \, k\left(x^i, x\right) + b$$

Compute an estimate of b using: $b = \frac{1}{M} \sum_{\substack{j=1, \\ \alpha_j \neq 0}}^{M} \left(y^j - \sum_{i=1}^{M} \alpha_i k\left(x^i, x^j\right) \right)$



Support Vector Regression: polynomial kernel

$$y = f(x) = \sum_{i=1}^{M} \alpha_i \, k\left(x^i, x\right) + b$$

What type of function f can you model with the homogeneous polynomial kernel?

$$y = \sum_{i=1}^{M} \alpha_i \left(\left(x^i\right)^T x \right)^p + b$$

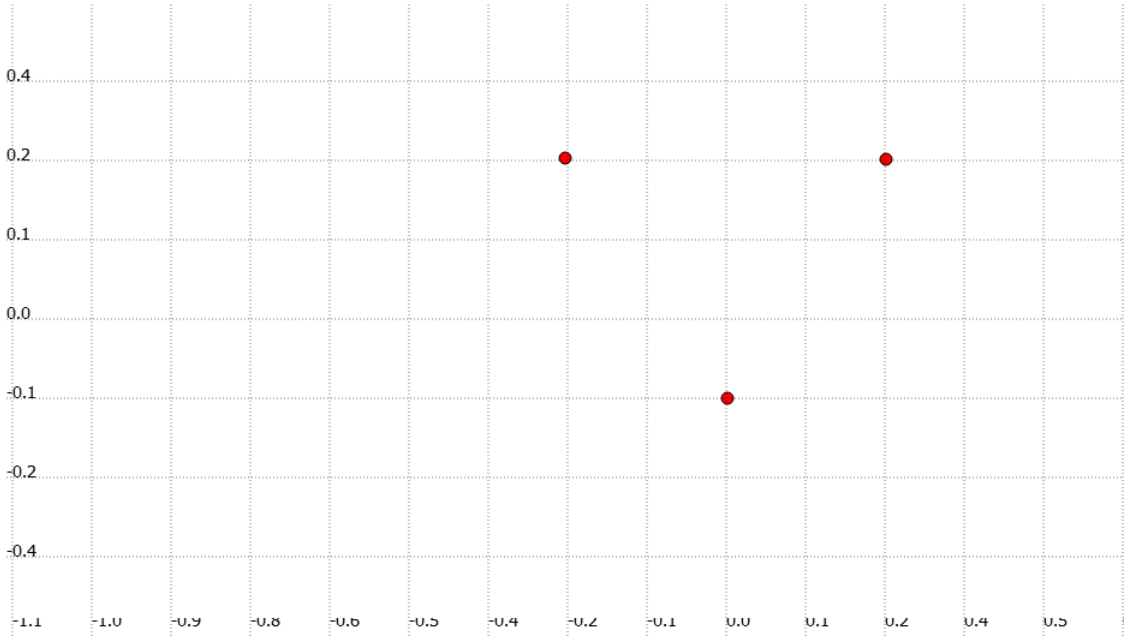
$$y = \beta x^P + b, \quad \beta = \sum_{i=1}^{M} \alpha_i \left(x_1^i\right)^p$$

Unidimensional x

SVR: polynomial kernel – choice of order

What is the minimum *p* (order of the polynomial) you need to achieve a good fit for the group of points below?

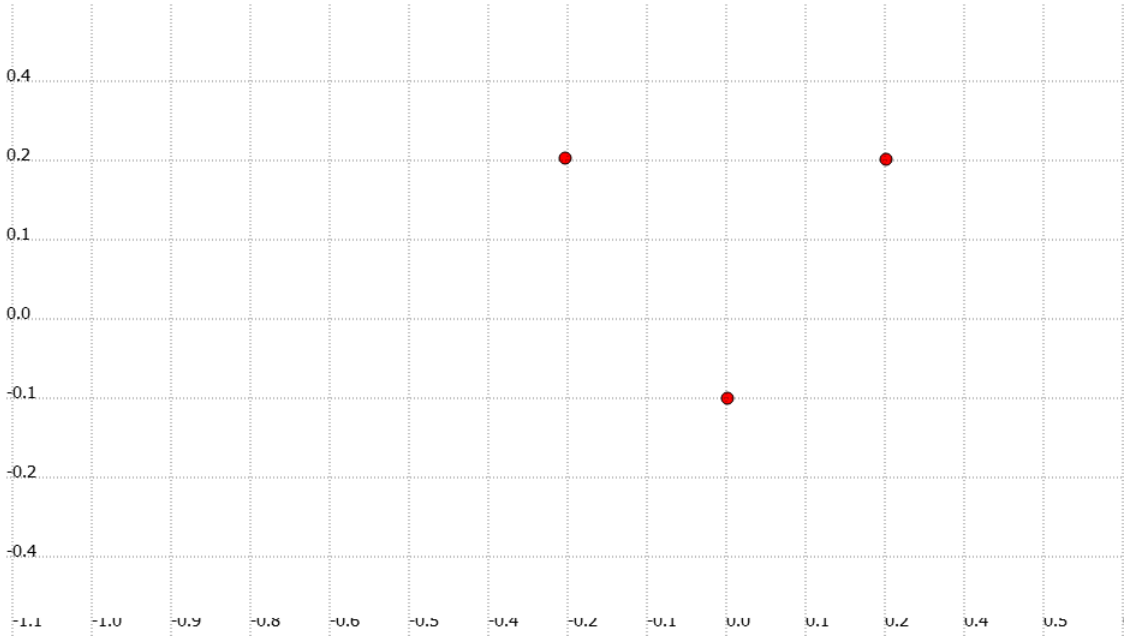
A. 1
B. 2
C. 3
D. >3



SVR: polynomial kernel – # of SV-s

What is the minimum *number of support vectors* ?

A. 1
B. 2
C. 3
D. I do not know



SVR: polynomial kernel – answers

Which is the minimum order of a homogeneous polynomial kernel you would need to achieve good regression on the set of 3 points below ?

To fit the set of points below,

we need at minimum p=2,

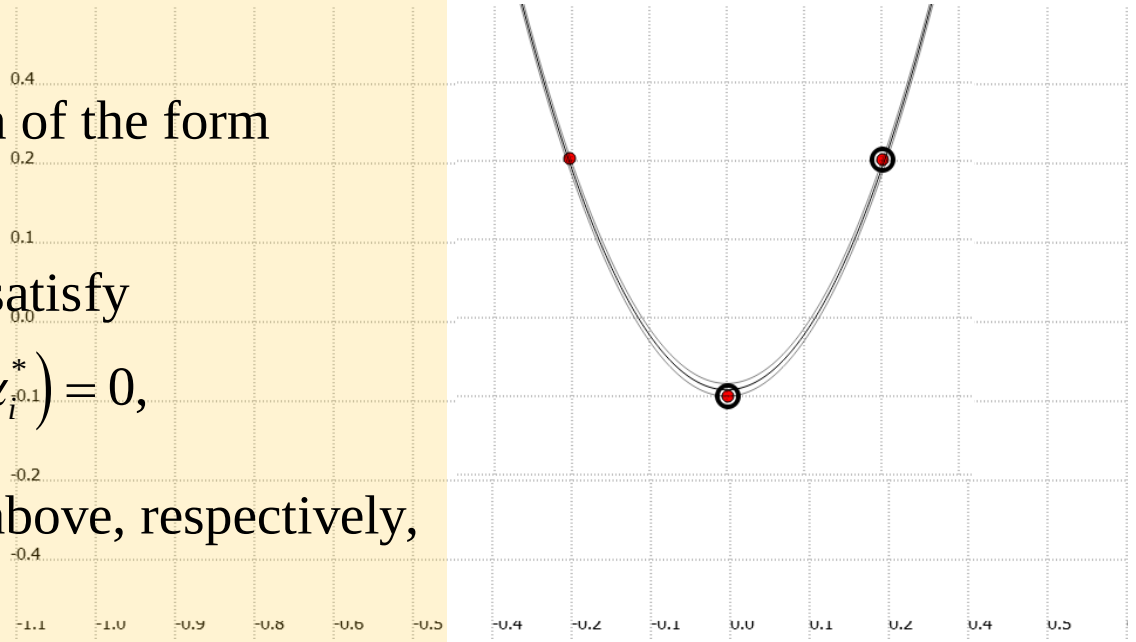
which leads to an equation of the form

$y = ax^2 + b$

and 2 Support Vectors to satisfy

KKT condition: $\sum\left(\alpha_i - \alpha_i^*\right) = 0,$

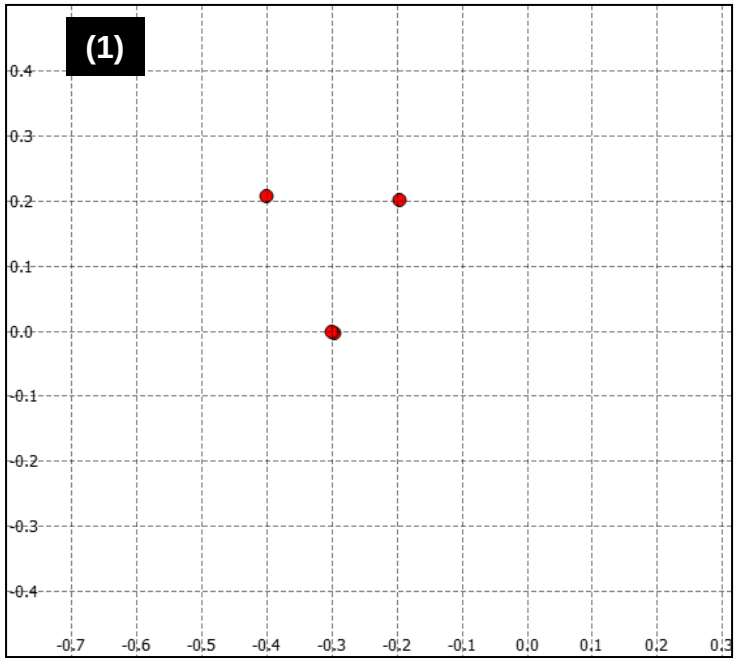
α_i, α_i^* denote SVs sitting above, respectively, below the regressive line

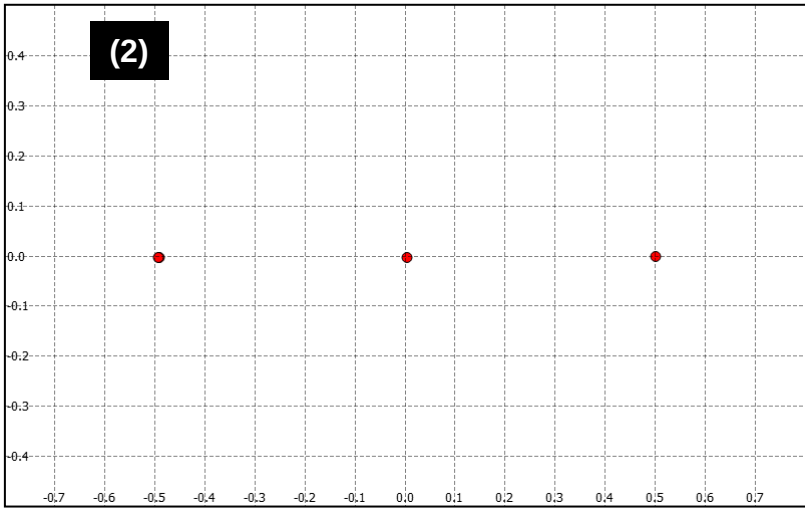


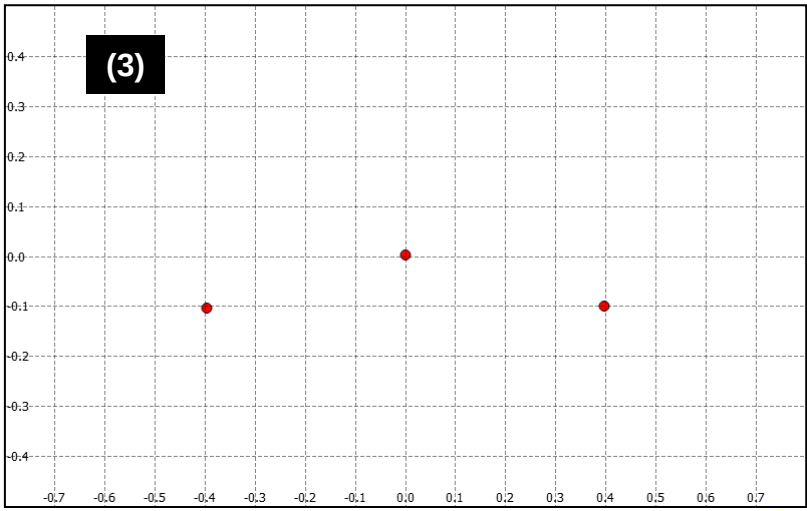
SVR: polynomial kernel – type of curve

Which of these datasets can you fit with homogeneous polynomial of order 2?

A. All
B. None
C. Dataset 1
D. Dataset 2
E. Dataset 3

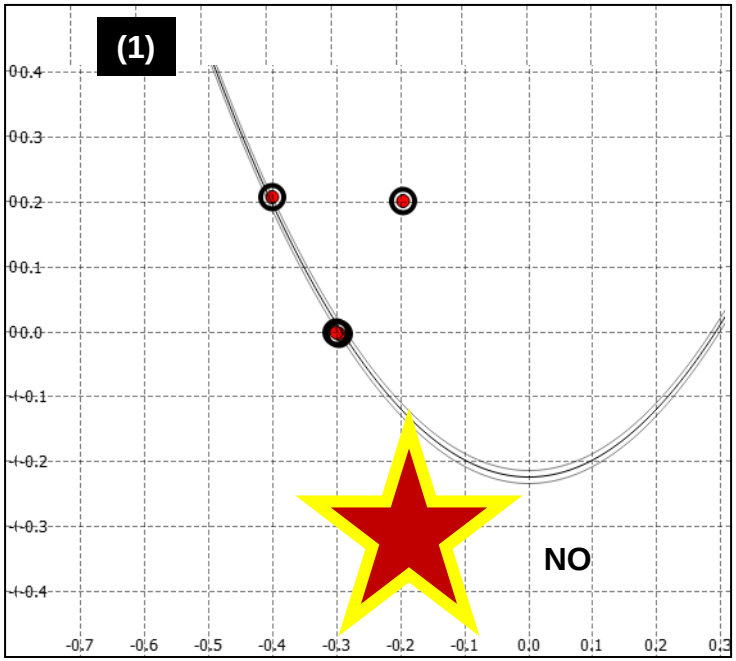


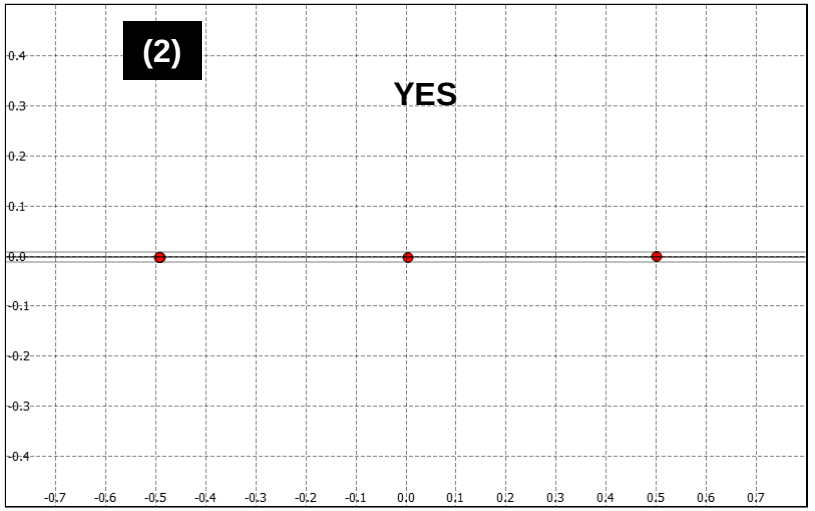


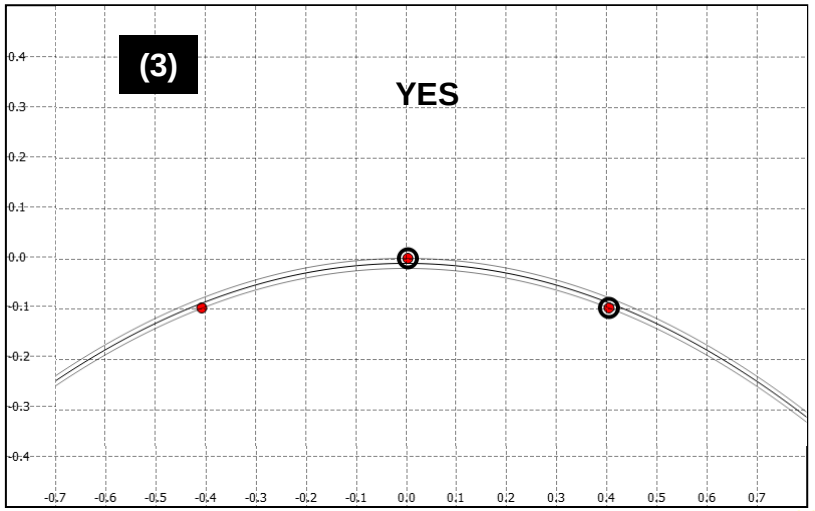


SVR: polynomial kernel – type of curve

Which of these datasets can you fit with homogeneous polynomial of order 2?







SVR: Inhomoneous polynomial kernel

The solution to SVR is:

$$y = f(x) = \sum_{i=1}^{M} \alpha_i k\left(x, x^i\right) + b$$

What type of function f can you model with the inhomogeneous polynomial?

$$y = f(x) = \sum_{i=1}^{M} \alpha_i \left(\left(x^i\right)^T x + c\right)^P + b$$

Combination of M polynomials of order p.

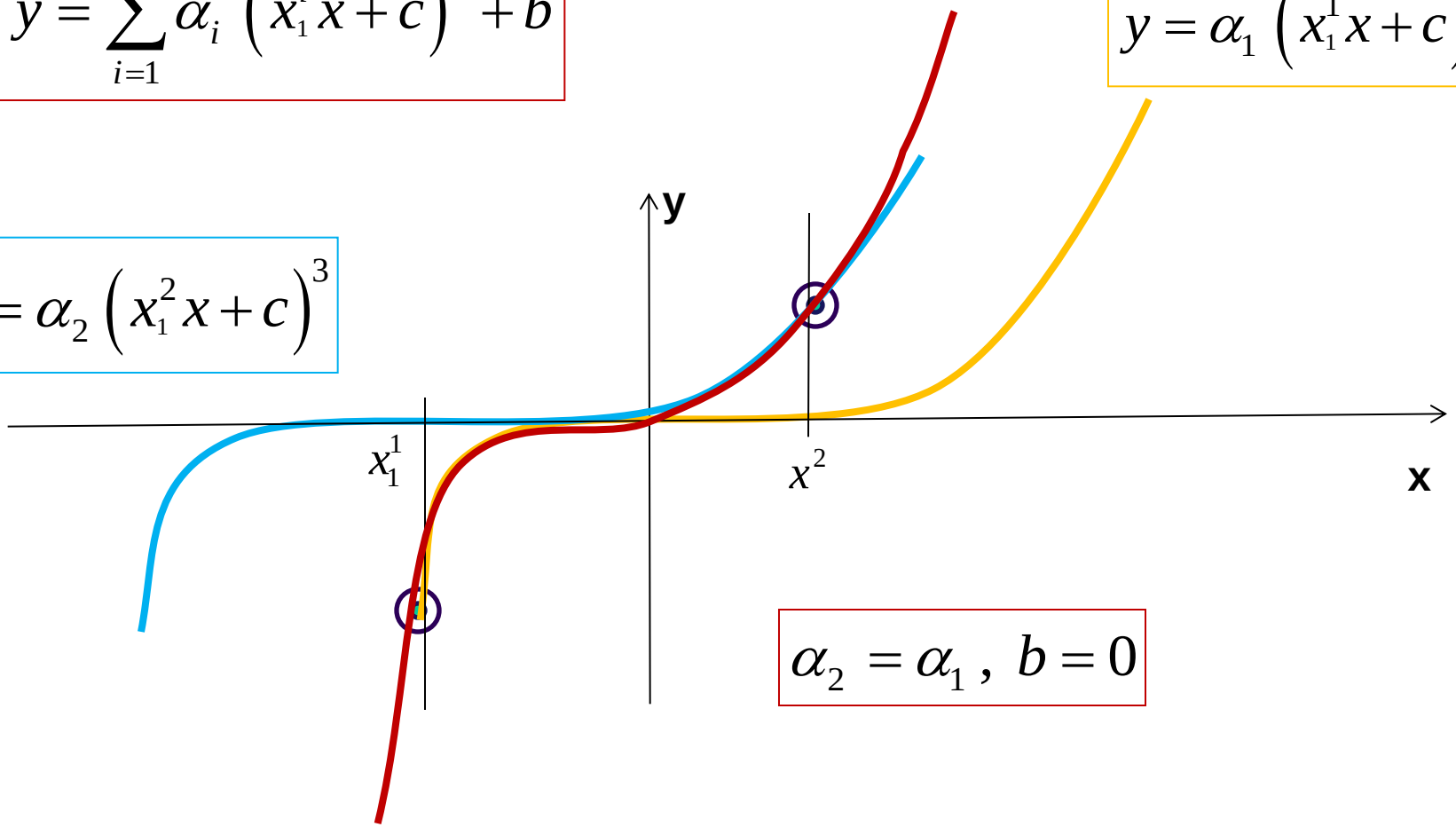
SVR: Inhomoneous polynomial kernel

Each polynomial passes through its own point (if there are no slack variables on the constraints and the ε-tube is extremely small).

$$y = \sum_{i=1}^{2} \alpha_i \left(x_1^i x + c \right)^3 + b$$

$$y = \alpha_1 \left(x_1^1 x + c \right)^3$$

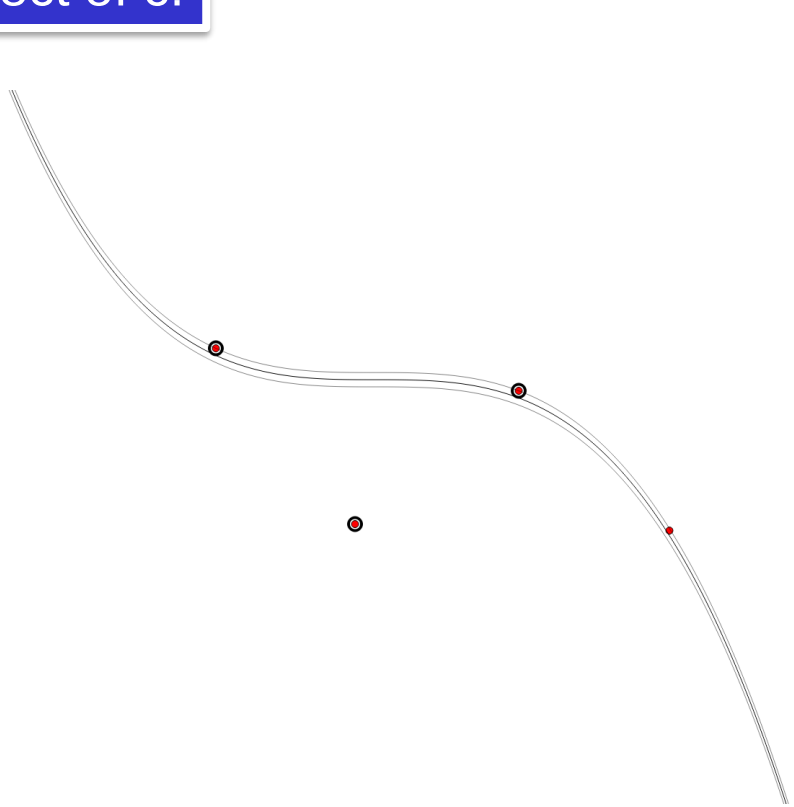
$$y = \alpha_2 \left(x_1^2 x + c \right)^3$$



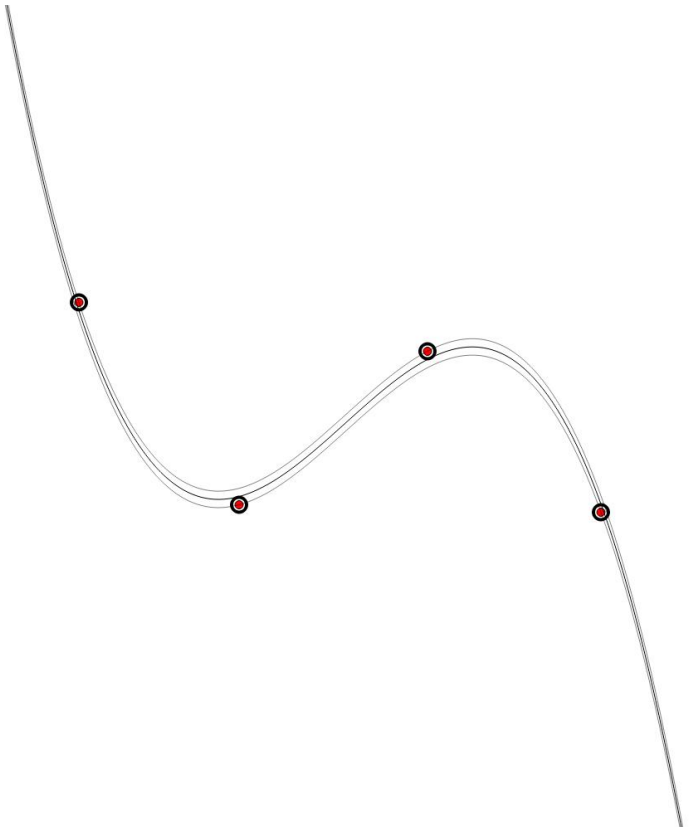
$$\alpha_2 = \alpha_1 , \; b = 0$$

SVR: inhom. polynomial kernel - hyperparameters

Effect of c:

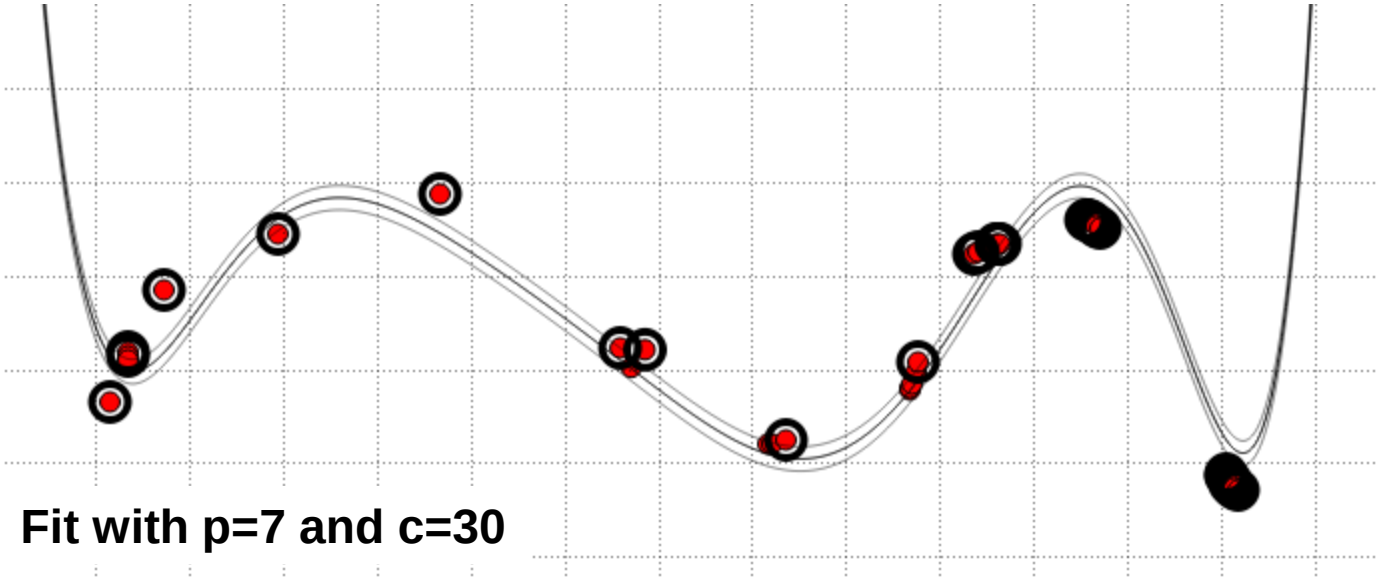


Fit with p=3 and c=0



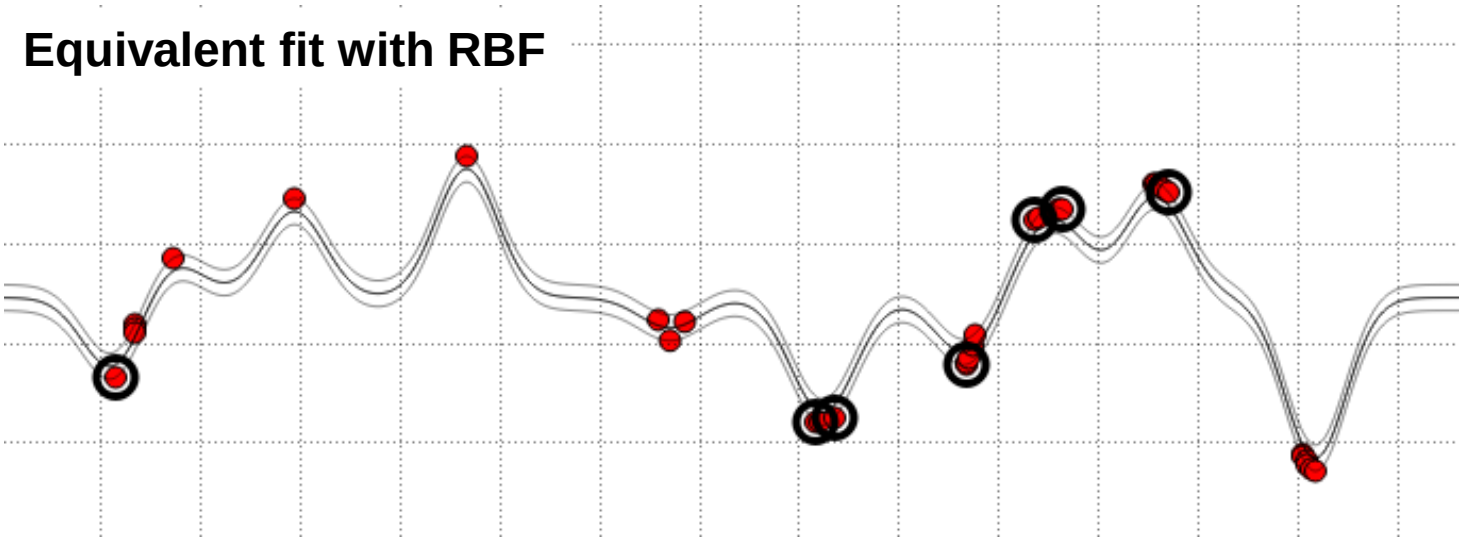
Fit with p=3 and c=0.1

ε-SVR – RBF versus Polynomial Kernels



Fit with p=7 and c=30

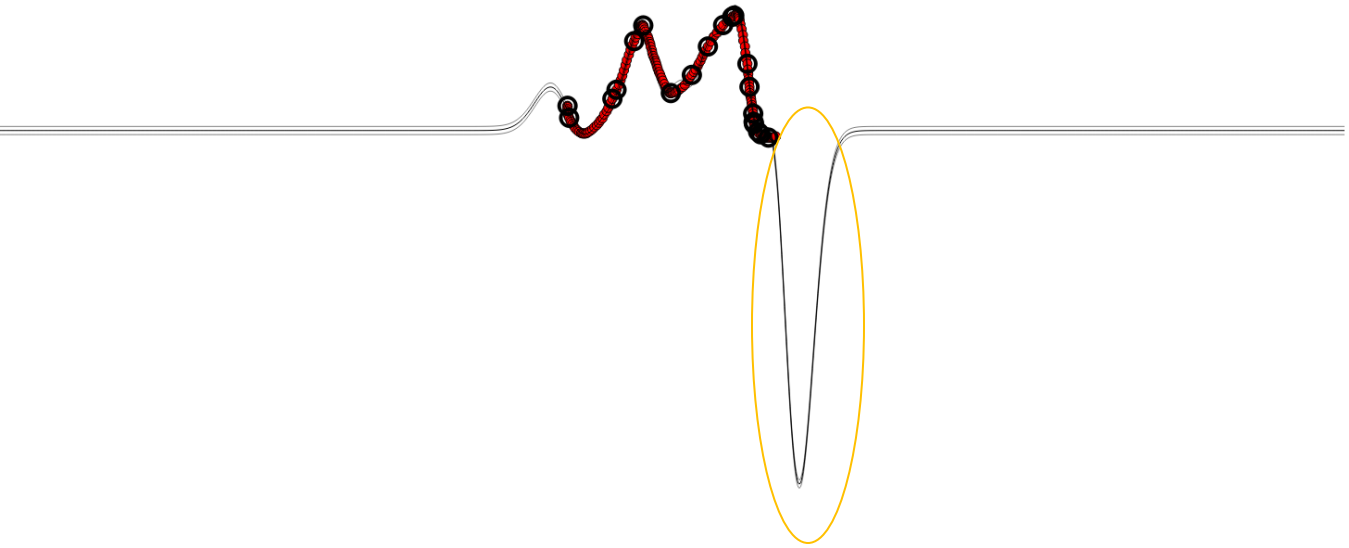
Equivalent fit with RBF



Effect of C in ε-SVR for RBF kernel

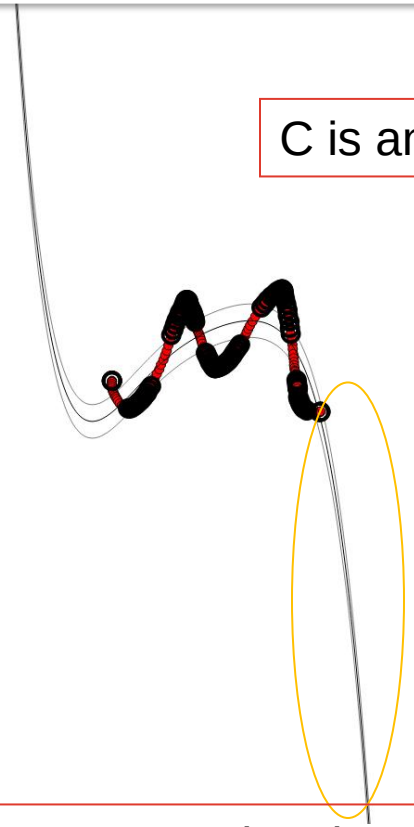
C is an upper bound on the absolute value of the α!

$$\alpha_i \in \left[0, \frac{C}{M}\right]$$



Does the effect remain with Polynomial kernel?

Effect of C in ε-SVR for RBF kernel



C is an upper bound on the absolute value of the α!

$$\alpha_i \in \left[0, \frac{C}{M}\right]$$

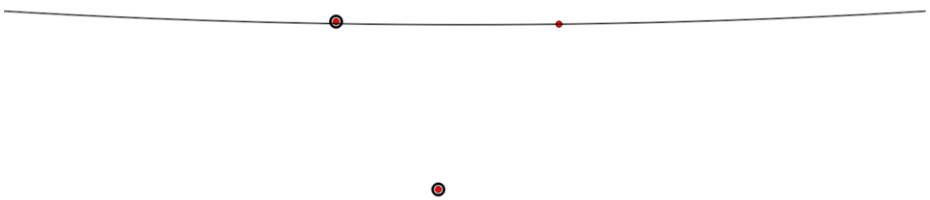
The larger α the steeper the slope.

$$y = f(x) = \sum_{i=1}^{M} \alpha_i \left(\left(x^i \right)^T x + c \right)^P + b$$

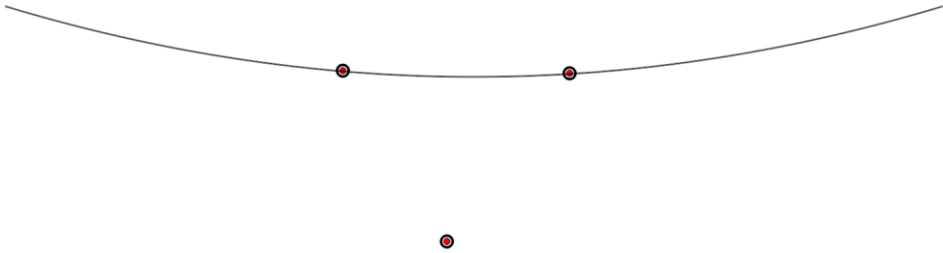
Effect of C in ε-SVR for RBF kernel

The larger C, the larger α and hence, the steeper the slope.

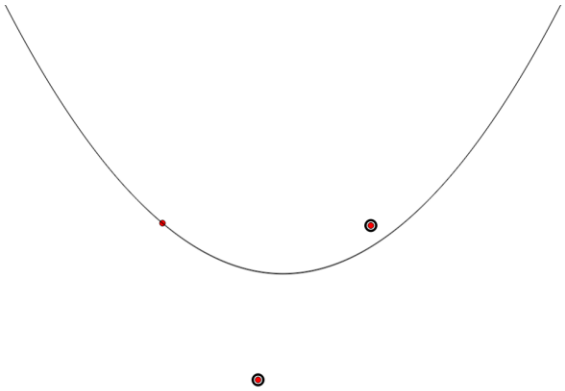
C=1



C=10



C=100



C=1000

For the polynomial kernel, picking a large C can be necessary to obtain a good fit.

Quick recap of SVR

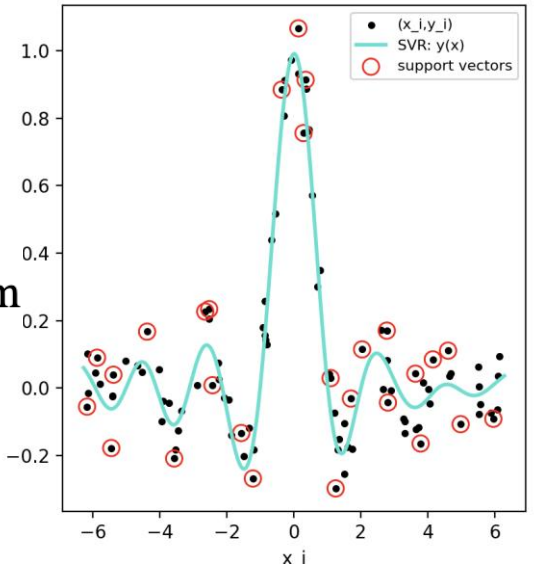
- Objective is to determine the function $y(x)$ at a test point x^*, given $\{x_i\}_{i=1}^M$ and $\{y_i\}_{i=1}^M$.

- The function is assumed to be linear $y(x) = w^\top x + b$ and weights w are to be determined.

- A convex optimization problem is solved to obtain $w = \sum_{i=1}^M \alpha_i x_i$, where α_i are dual variables corresponding to constraint: $\left| y_i - (w^\top x_i + b) \right| \leq \epsilon + \xi_i$.

- SVR: $y(x) = \sum_{i=1}^M \alpha_i K(x, x_i) + \alpha_0$.
- The intercept b is obtained in terms of dual variables.

- $\alpha_i \in (0, \frac{C}{M})$, where C is the regularization term in the dual problem



Relevance Vector Regression

Why RVR?

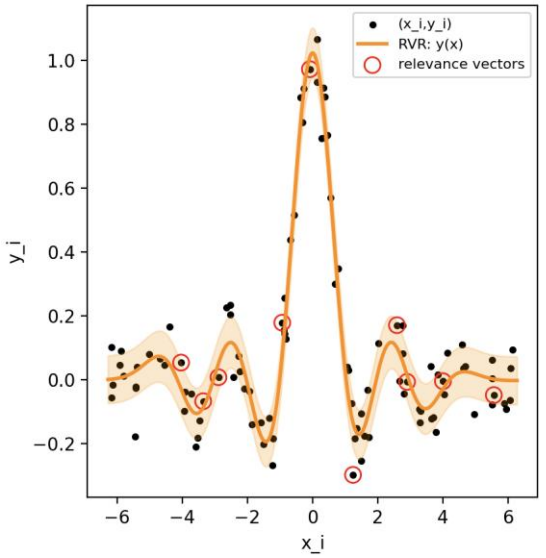
- Number of basis functions (kernels) in SVR is equal to number of support vectors (grows with complexity).
- Predictions are *not* probabilistic.
- C is hard to choose (no range or prior known).

What does RVR do?

- Considers each $y_i := y(x_i)$ is generated from a Gaussian distribution with prior $y(x_i)$ and variance σ^2.
- The function y is estimated as in SVR: $y(x_i) := \phi_i(x_i)\boldsymbol{\alpha}$, $\phi_i(x_i) = [1, K(x_i, x_1), \ldots, K(x_i, x_n)] \in \mathbb{R}^M$, $\boldsymbol{\alpha} = [\alpha_0, \ldots, \alpha_M]^\top$
- Mathematically, $p(y_i|\boldsymbol{\alpha}, \boldsymbol{x}, \sigma) = \mathcal{N}(\phi_i(x_i)\boldsymbol{\alpha}, \sigma^2)$, $\sigma \in \mathbb{R}$.
- All y_is are independent, so the likelihood of the complete data set is

$$p(\boldsymbol{y}|\boldsymbol{\alpha}, \boldsymbol{x}, \sigma) = \mathcal{N}(\Phi(\boldsymbol{x})\boldsymbol{\alpha}, \sigma^2)$$

$\boldsymbol{y} = [y_1, \ldots, y_M]^\top$, $\Phi = [\phi_1, \ldots, \phi_M] \in \mathbb{R}^{M \times (M+1)}$



Few Assumptions

- MLE of $\boldsymbol{\alpha}$ leads to over fitting, hence it is constrained by introducing a prior (similar to regularization parameter of SVM)
- Consider α_i is normally distributed with mean 0 and variance $\frac{1}{s_i}$.
- In this hierarchical model, a hyper-prior is assumed over each hyper parameter, contrary to a single variance distribution.
- The hyper-prior distribution is such that "any value is equally likely": Uniform or Gamma distribution.
- The model automatically rules out unlikely α_i values in the estimation procedure.

Hyperparamters in RVR are: $\{s_i\}_{i=1}^{M+1}$ and σ^2.

Making Predictions with RVR

- Prediction step (Bayesian inference):

$$p(y|\boldsymbol{y}, \boldsymbol{\alpha}, \boldsymbol{s}, \boldsymbol{\sigma}) = \int \underbrace{p(y|\boldsymbol{\alpha}, \boldsymbol{s}, \boldsymbol{\sigma})}_{\text{Assumed Gaussian}} p(\boldsymbol{\alpha}, \boldsymbol{s}, \boldsymbol{\sigma}|\boldsymbol{y}) \mathrm{d}\boldsymbol{\alpha} \mathrm{d}\boldsymbol{s} \mathrm{d}\boldsymbol{\sigma}$$

- The second term in the integrand is posterior of the hyperparameters and $\boldsymbol{\alpha}$:

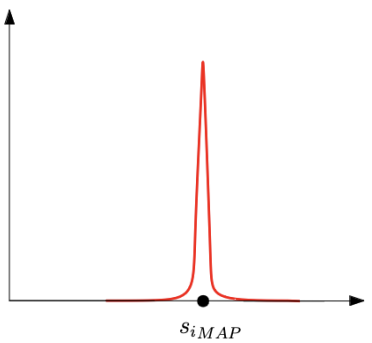
$$p(\boldsymbol{\alpha}, \boldsymbol{s}, \boldsymbol{\sigma}|\boldsymbol{y}) = p(\boldsymbol{\alpha}|\boldsymbol{y}, \boldsymbol{s}, \boldsymbol{\sigma}) p(\boldsymbol{s}, \boldsymbol{\sigma}|\boldsymbol{y})$$

- The MAP estimate of $\boldsymbol{s}$ and $\boldsymbol{\sigma}$ is

$$\begin{aligned} \boldsymbol{s}_{MAP}, \boldsymbol{\sigma}_{MAP} &= \arg\max_{\boldsymbol{s}, \boldsymbol{\sigma}} p(\boldsymbol{s}, \boldsymbol{\sigma}|\boldsymbol{y}) = \arg\max_{\boldsymbol{s}, \boldsymbol{\sigma}} p(\boldsymbol{y}|\boldsymbol{s}, \boldsymbol{\sigma}) \frac{p(\boldsymbol{s}, \boldsymbol{\sigma})}{p(\boldsymbol{y})} \\ &= \arg\max_{\boldsymbol{s}, \boldsymbol{\sigma}} p(\boldsymbol{y}|\boldsymbol{s}, \boldsymbol{\sigma}) p(\boldsymbol{s}, \boldsymbol{\sigma}) \end{aligned}$$

- MAP estimate assumption: $p(\boldsymbol{s}, \boldsymbol{\sigma}|\boldsymbol{y}) \approx \delta(\boldsymbol{s}_{MAP}, \boldsymbol{\sigma}_{MAP})$
- Delta function integrates to 1, hence,

$$\begin{aligned} p(y|\boldsymbol{y}) &= \int p(y|\boldsymbol{\alpha}, \boldsymbol{s}, \boldsymbol{\sigma}) p(\boldsymbol{\alpha}, \boldsymbol{s}, \boldsymbol{\sigma}|\boldsymbol{y}) \mathrm{d}\boldsymbol{\alpha} \mathrm{d}\boldsymbol{s} \mathrm{d}\boldsymbol{\sigma} \\ &= \int p(y|\boldsymbol{\alpha}, \boldsymbol{s}_{MAP}, \boldsymbol{\sigma}_{MAP}) p(\boldsymbol{\alpha}|\boldsymbol{y}, \boldsymbol{s}_{MAP}, \boldsymbol{\sigma}_{MAP}) \mathrm{d}\boldsymbol{\alpha} \end{aligned}$$



Dirac Delta function

Assumes that the parameters are at their most probable value when one samples.

Prediction step: Simplification

- We still have an integral to solve whose second integrand term is $p(\boldsymbol{\alpha}|\boldsymbol{y}, \boldsymbol{s}_{MAP}, \boldsymbol{\sigma}_{MAP})$
- Thanks to Bayes rule: $p(\boldsymbol{\alpha}|\boldsymbol{y}, \boldsymbol{s}, \boldsymbol{\sigma}) = \underbrace{\int p(\boldsymbol{y}|\boldsymbol{\alpha}, \boldsymbol{s}, \boldsymbol{\sigma}) p(\boldsymbol{\alpha}|\boldsymbol{s}, \boldsymbol{\sigma}) \mathrm{d}\boldsymbol{\alpha}}_{\text{Convolution Integral}}$
- Convolution of two Gaussian distributions is also a Gaussian distribution.
- Finally $p(y|\boldsymbol{y})$ is also a convolution of two Gaussian distributions.
- Mean of $p(y|\boldsymbol{y})$ is the predicted value of y

MAP Estimate: Simplification

- MAP Estimates are obtained by solving a maximization problem:

$$\boldsymbol{s}_{MAP}, \boldsymbol{\sigma}_{MAP} = \arg\max_{\boldsymbol{s},\boldsymbol{\sigma}} p(\boldsymbol{y}|\boldsymbol{s},\boldsymbol{\sigma}) p(\boldsymbol{s},\boldsymbol{\sigma})$$

Equivalently, minimize the negative log likelihood:

$$\boldsymbol{s}_{MAP}, \boldsymbol{\sigma}_{MAP} = \arg\min_{\boldsymbol{s},\boldsymbol{\sigma}} \underbrace{[-\ln p(\boldsymbol{y}|\boldsymbol{s},\boldsymbol{\sigma}) - \ln \boldsymbol{s} - \ln \sigma^2]}_{:=\mathcal{L}(\boldsymbol{s},\sigma)}$$

$$\mathcal{L}(\boldsymbol{s},\sigma) = \tfrac{1}{2}[\ln|C| + \boldsymbol{y}^\top C^{-1}\boldsymbol{y}], C = \sigma^2 I + \boldsymbol{\phi} S \boldsymbol{\phi}^{-1},$$
$$S = Diag(s_1, \ldots, s_{M+1})$$

- Isolate the effect of individual s_i and take derivative w.r.t s_i:

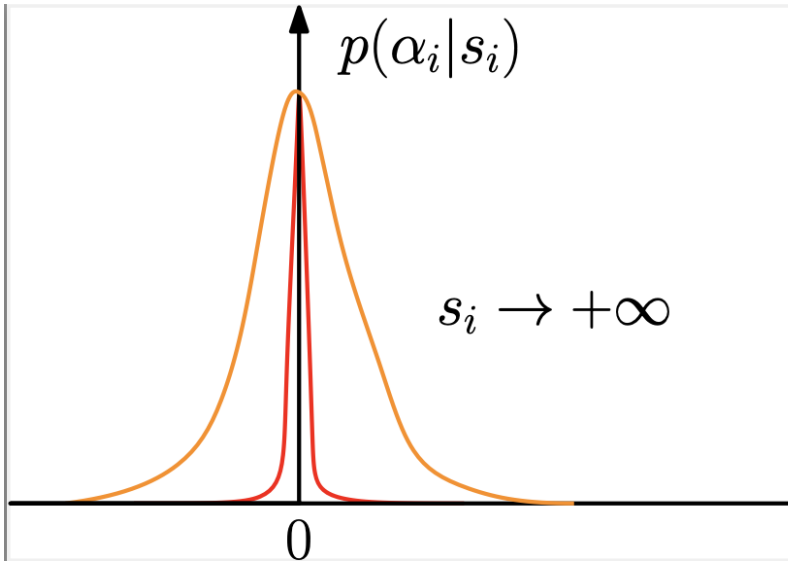
$$\frac{\partial \mathcal{L}(\boldsymbol{s},\sigma)}{\partial s_i} = \frac{s_i^{-1} A_i^2 - B_i}{2(s_i + A_i)^2}, \quad A_i, B_i \text{ are independent of } s_i$$

- $s_i \to +\infty$ is always an extremum, the other is $s_i = \frac{A_i^2}{B_i}$

- If $B_i < 0$, then $s_i \to +\infty$ is the unique minimum vice-versa.

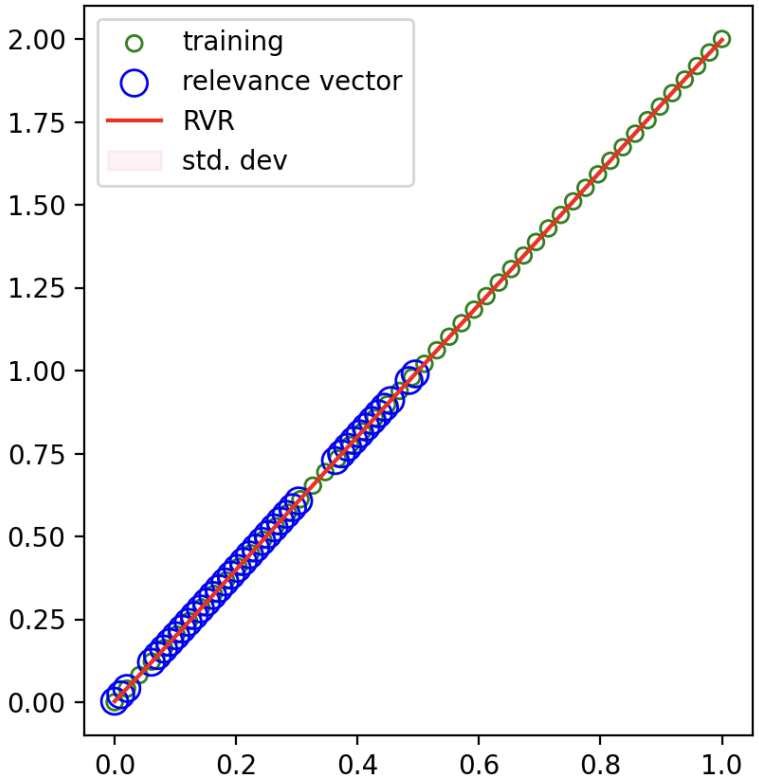
Discussion on RVR

- Hyper parameter choice for the kernel is simplified: grid search or score function or cross validation is not necessary.
- No closed form expression for hyper parameters.

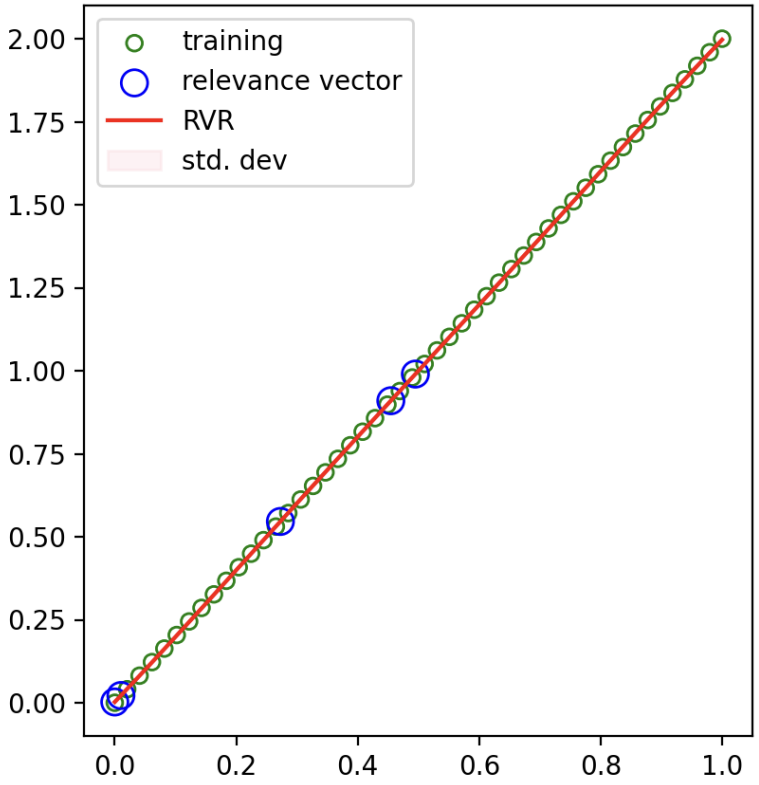


- Most $s_i \to \infty$ during MAP estimation: Corresponding $\alpha_i = 0$, basis function ϕ_i can be dropped
- Update rule of s_i, σ is obtained by setting the partial derivative of $\mathcal{L}(\boldsymbol{s}, \sigma)$ to 0 w.r.t s_i, σ respectively.
- Higher computation time than SVR as full set of basis functions is initially considered.

Points on line y=2x are trained with two RVR models

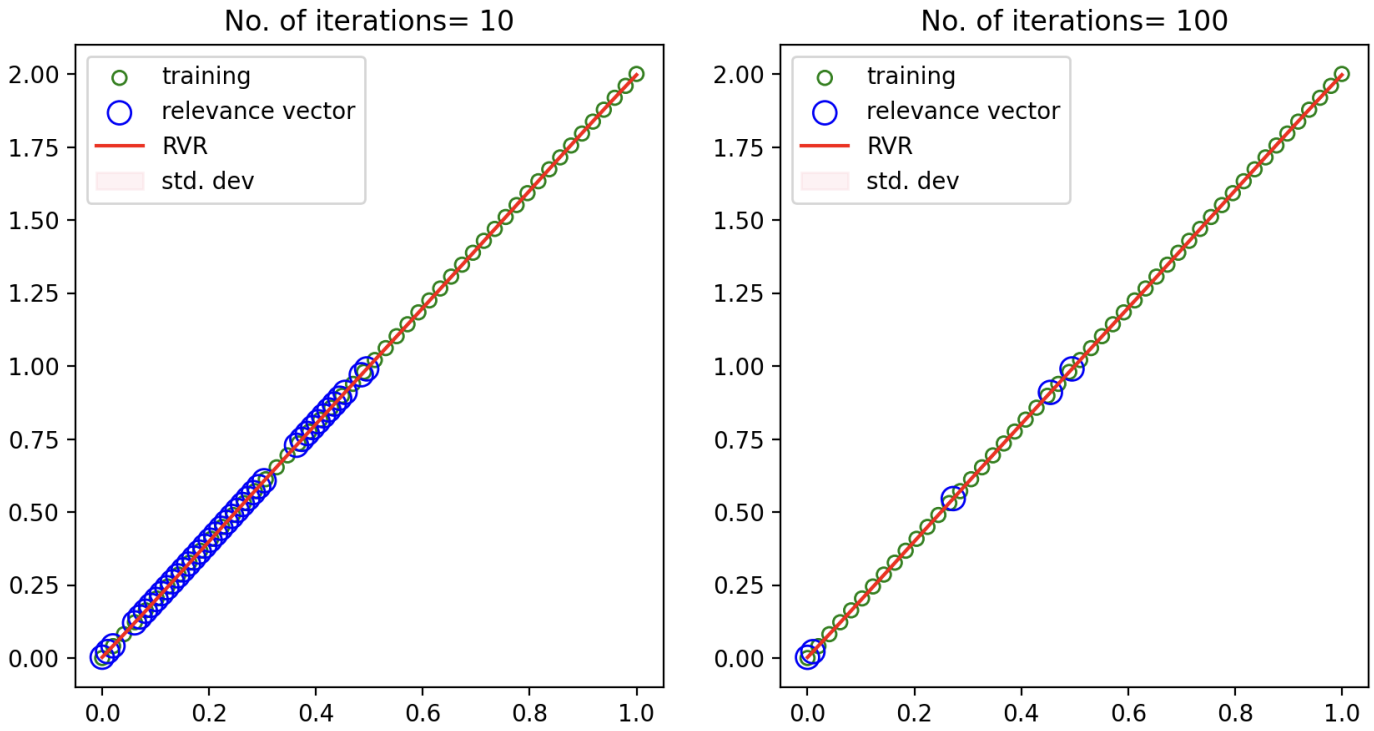


(a)



(b)

What is the issue with solution a and why do we get this result?



(a) (b)

The no. of relevance vectors in (a) and (b) are 72 and 5 respectively. The estimation of s is poor in (a), it is likely that the gradient descent algorithm has not converged due to smaller training time (or fewer number of iterations)

Which of the following statements are correct for RVR?

1. Hyper parameters are obtained in closed form.

2. Gradient Descent is used to solve for hyper parameters when using MAP estimation.

3. Estimates can be stuck at local minima when using MAP estimation.

4. It takes longer to train a RVR model than to test it.

- Hyper paramater updates are available in closed form as

$$\frac{\partial \mathcal{L}(\boldsymbol{s}, \sigma)}{\partial s_i} = \frac{s_i^{-1} A_i^2 - B_i}{2(s_i + A_i)^2} = 0 \implies \boxed{s_{i+1} = \frac{A_i^2}{B_i}} \quad \text{or} \quad \boxed{s_{i+1} = +\infty}$$

A_i, B_i depend on other $s_j, j \neq i$.

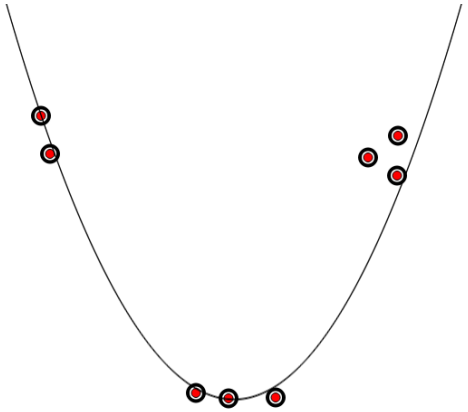
- There is no guarantee that $\mathcal{L}(\boldsymbol{s}, \sigma)$ has a global minimum.
- Computation of C^{-1} in $\mathcal{L}(\boldsymbol{s}, \sigma)$ is expensive as $C \in R^{M \times M}$, hence the time to train a RVR model is much larger than to test it.

Comparison ε-SVR, ν-SVR, RVR: Polynomial kernel

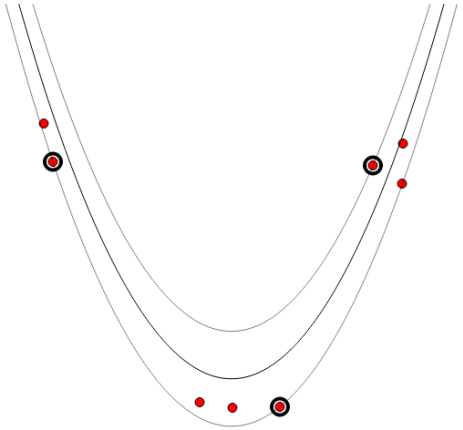
Would the use of ν-SVR or RVR help decrease
the number of support vectors with polynomial kernel?

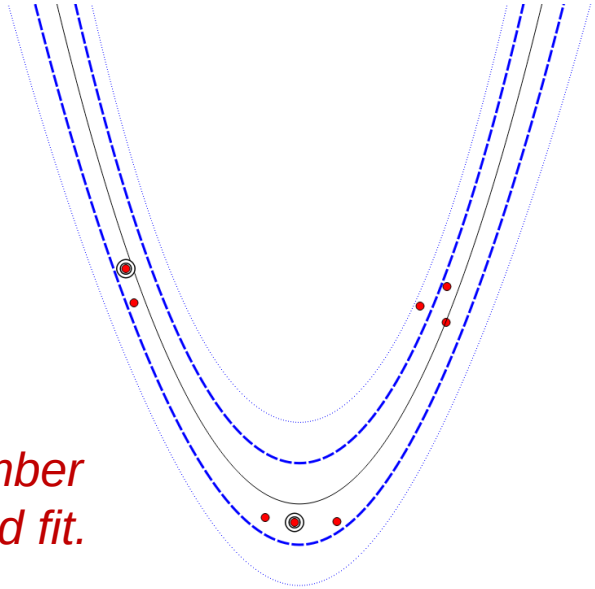
Comparison ε-SVR, ν-SVR, RVR: Polynomial kernel

Solution with **ε-SVR**,
All points become SV-s as we need a large C to obtain the right slope.



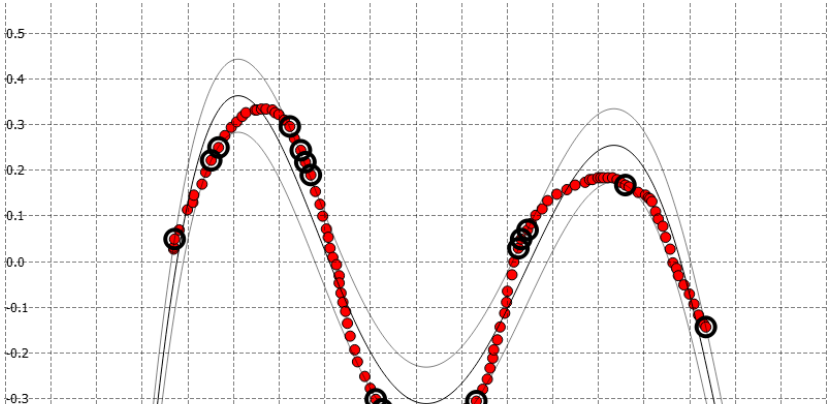
Solution with **ν-SVR**,
Automatically find the right slope while retaining few SV-s.



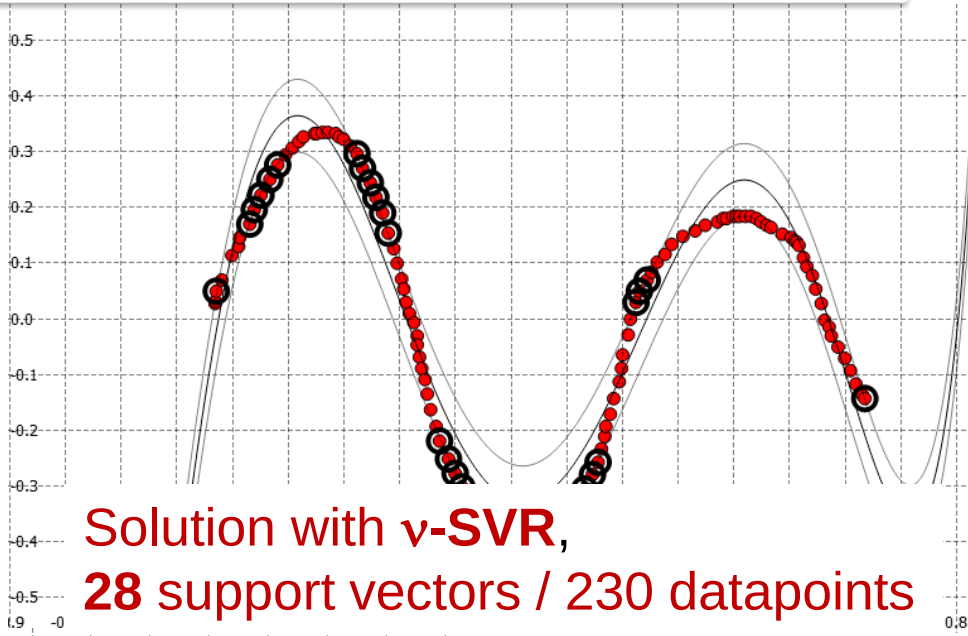


Solution with **RVR**,
Reduces even further number of SV-s required for a good fit.

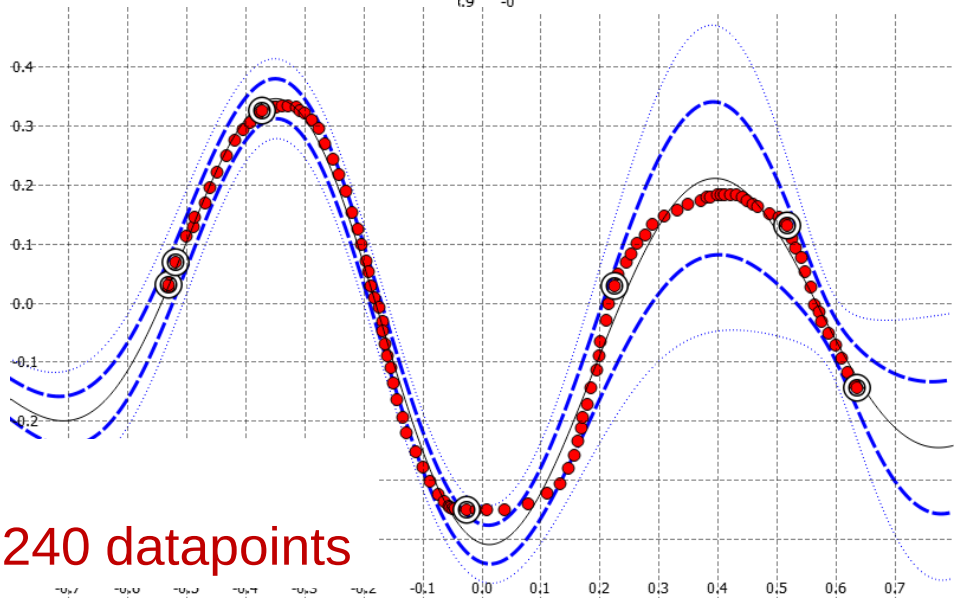
Comparison ε-SVR, ν-SVR, RVR: Polynomial kernel



Solution with **ε-SVR**,
17 support vectors / 240 datapoints



Solution with **ν-SVR**,
28 support vectors / 230 datapoints

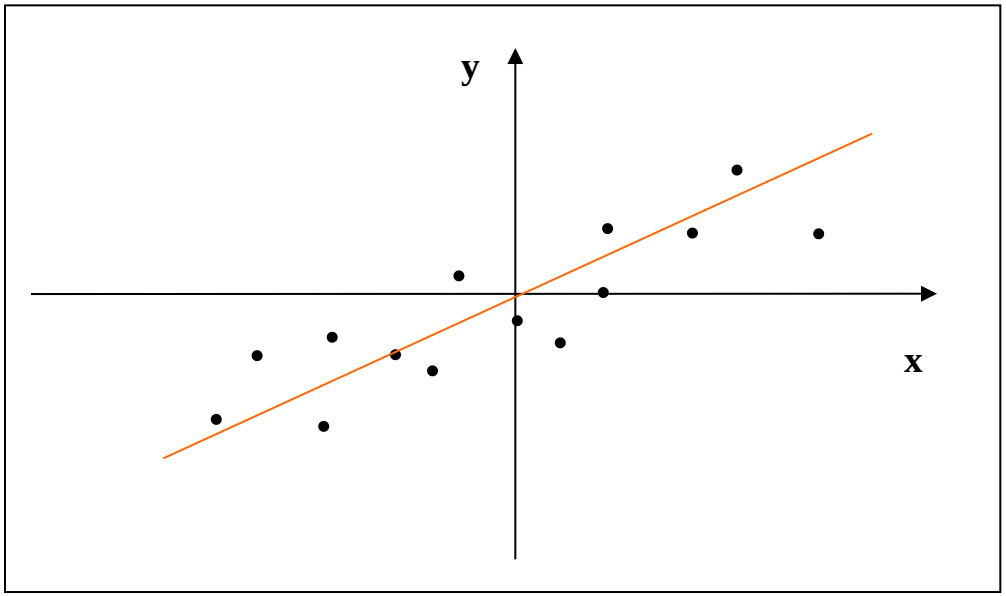


Solution with **RVR**,
7 support vectors / 240 datapoints

Ridge regression is the starting point to Gaussian Process Regression, see next week's lecture

Linear Regression

$$y = f(x; w) = w^T x$$



It has an exact solution $w^* = \left(XX^T\right)^{-1} X\mathbf{y}$ if:

a) XX^T is not singular (it is singular with not enough datapoints)

b) Data is not noisy (otherwise no single match to $y^i = \langle w, x^i \rangle$)

Ridge Regression: optimality

Is ridge regression always giving a unique optimal solution?

A. Yes
B. No
C. I do not know

Solution in linear case:

$$w^* = \left(XX^T + \lambda I\right)^{-1} X\mathbf{y}$$

always invertible for $\lambda > 0$.

Solution in nonlinear case:

$$y = k(X,x)\left(\underbrace{K(X,X)}_{\text{Gram Matrix in feature space}} + \lambda I\right)^{-1}\mathbf{y}, \quad k(X,x) = \begin{bmatrix} k(x^1,x) \\ \cdot \\ \cdot \\ k(x^M,x) \end{bmatrix}^T$$

It is unique if λ is fixed but optimality depends on λ.

Ridge Regression: computational costs

What affects most computational growth?

A. Number of datapoints
B. Dimension of the datapoints
C. I do not know

Solution in linear case:

$$w^* = \left(XX^T + \lambda I\right)^{-1} X\mathbf{y}$$

always invertible for $\lambda > 0$.

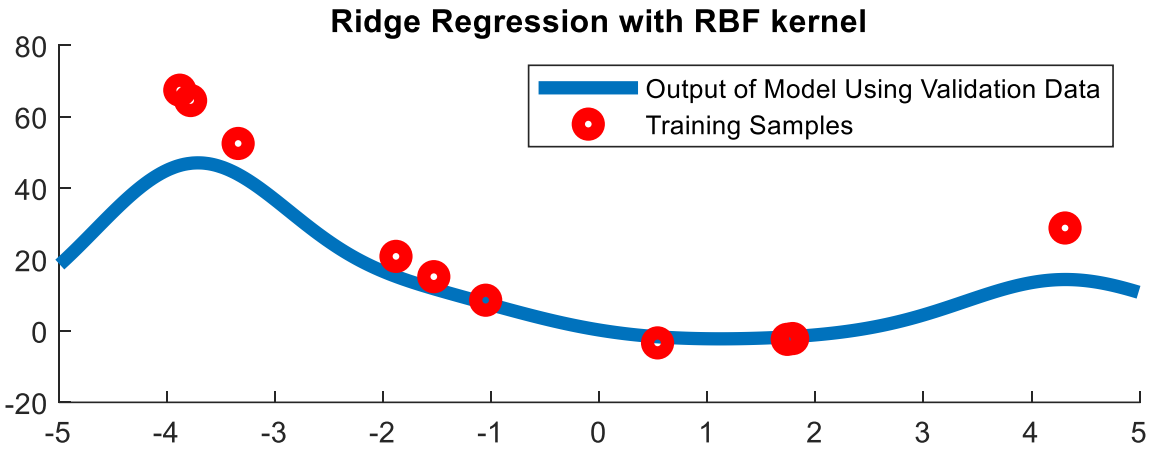
Solution in nonlinear case:

$$y = k(X,x)\left(\underbrace{K(X,X)}_{\text{Gram Matrix in feature space}} + \lambda I\right)^{-1}\mathbf{y}, \quad k(X,x) = \begin{bmatrix} k(x^1,x) \\ \cdot \\ \cdot \\ k(x^M,x) \end{bmatrix}^T$$

In linear ridge regression, the complexity is $O(N^3)$, N: dimension of datapoint

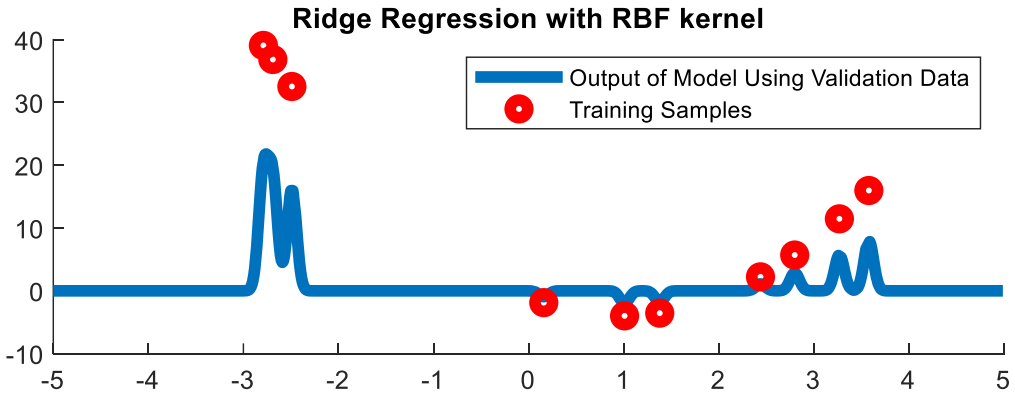
In nonlinear ridge regression, the complexity is: $O(M^3)$, M: number of datapoints

Ridge Regression: Kernel



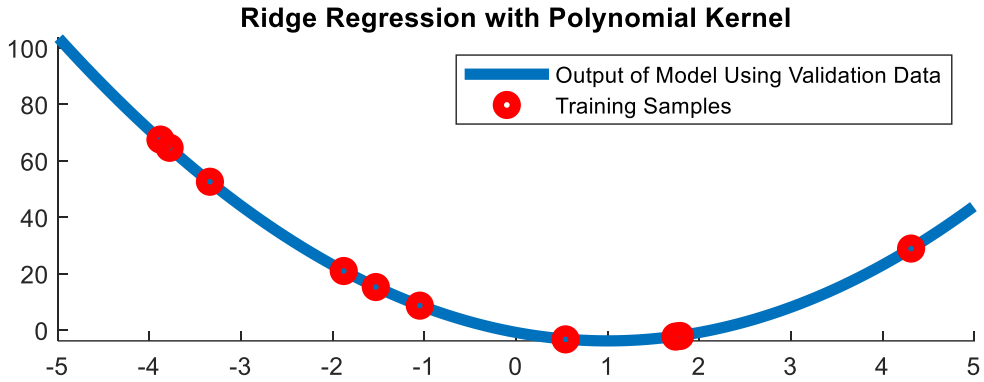
Large kernel width

Which kernel?



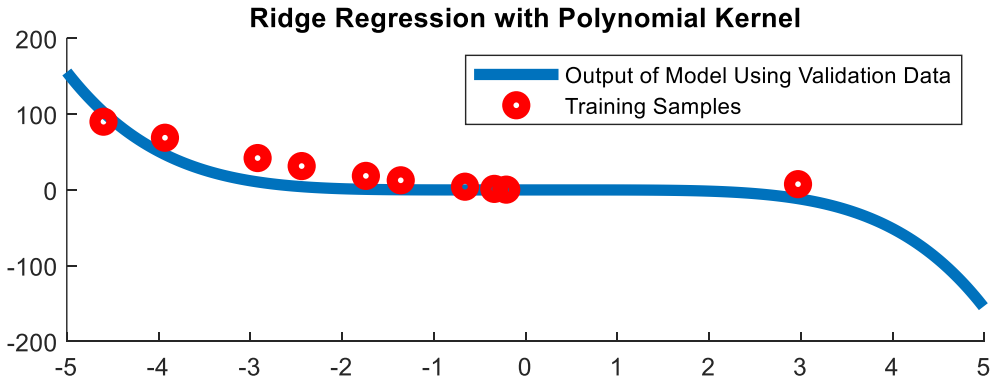
Small kernel width

Ridge Regression: kernel



P=2

P=5



Which kernel?