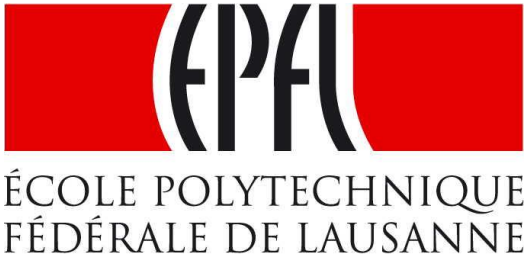
ADVANCED MACHINE LEARNING

Non-linear regression techniques

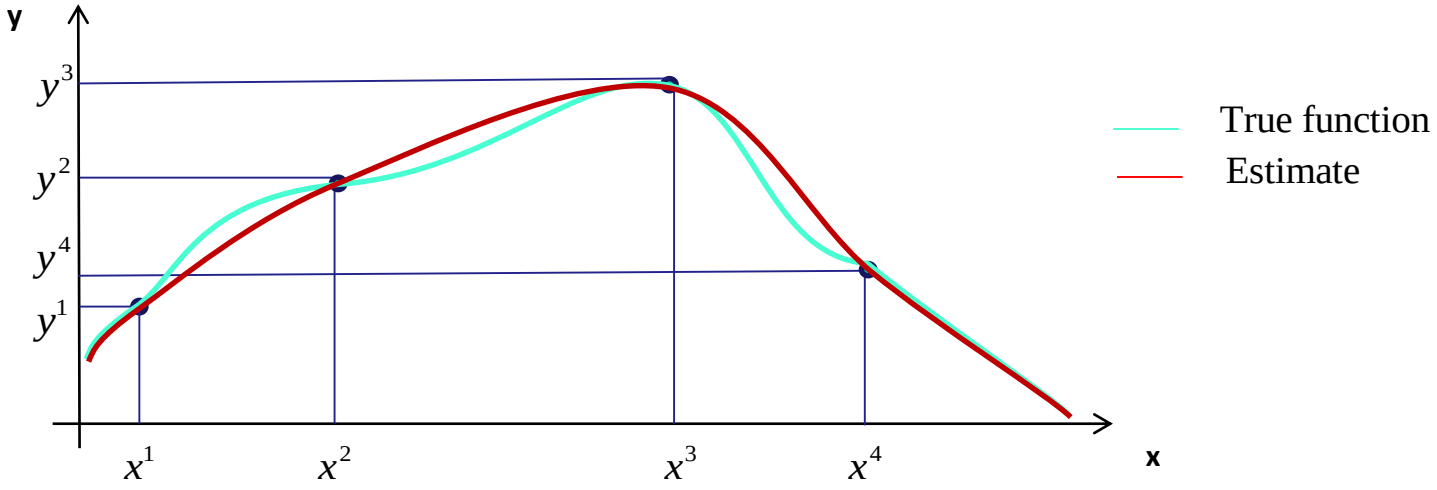


Regression: Principle

Map N-dim. input $x \in \mathbb{R}^N$ to a continuous output $y \in \mathbb{R}$.

Learn a function of the type:

$f : \mathbb{R}^N \rightarrow \mathbb{R}$ and $y = f(x)$.



Estimate f that best predict set of training points $\{x^i, y^i\}_{i=1,...M}$?

Regression: Issues

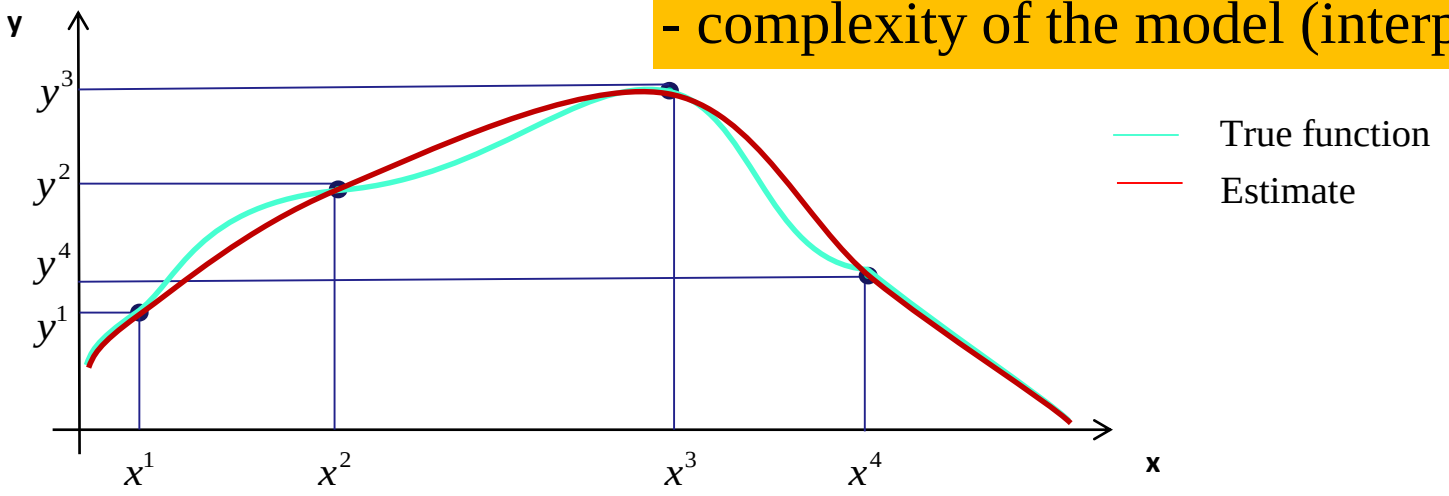
Map N-dim. input $x \in \mathbb{R}^N$ to a continuous output $y \in \mathbb{R}$.

Learn a function of the type:

$f : \mathbb{R}^N \rightarrow \mathbb{R}$ and $y = f(x)$.

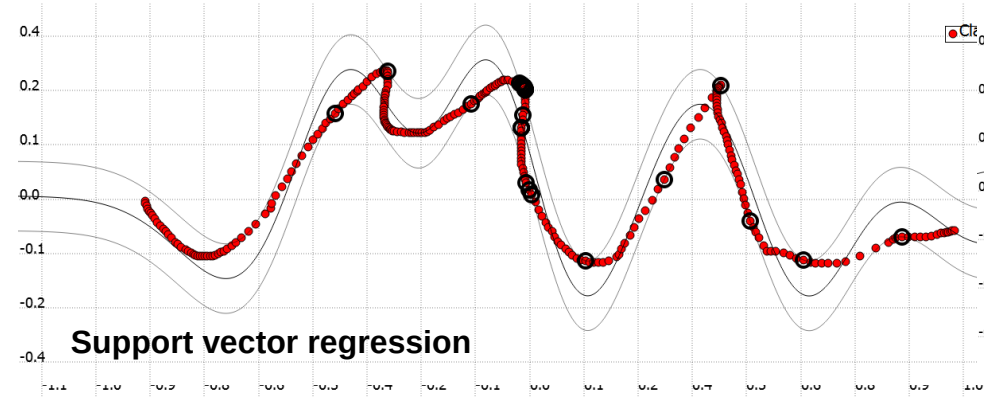
Fit strongly influenced by choice of:

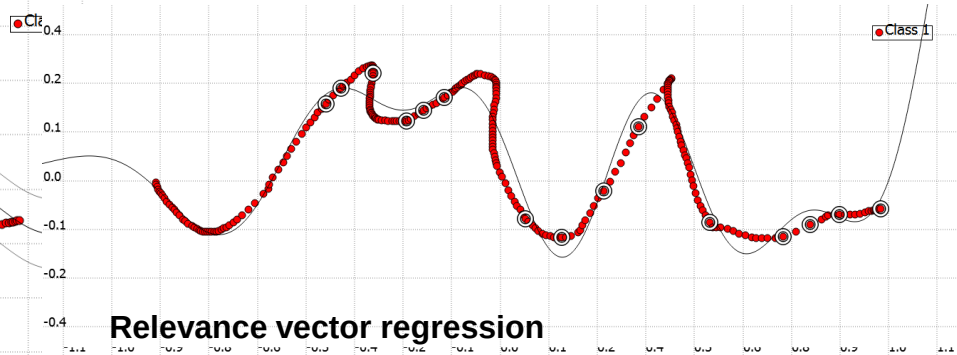
- datapoints for training
- complexity of the model (interpolation)

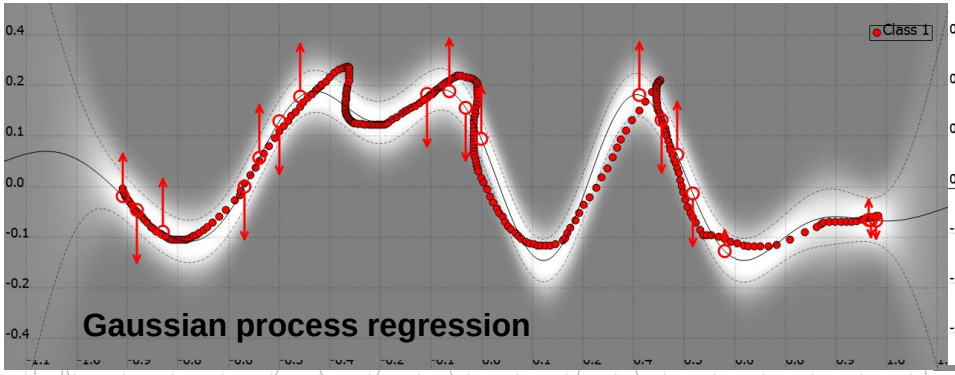


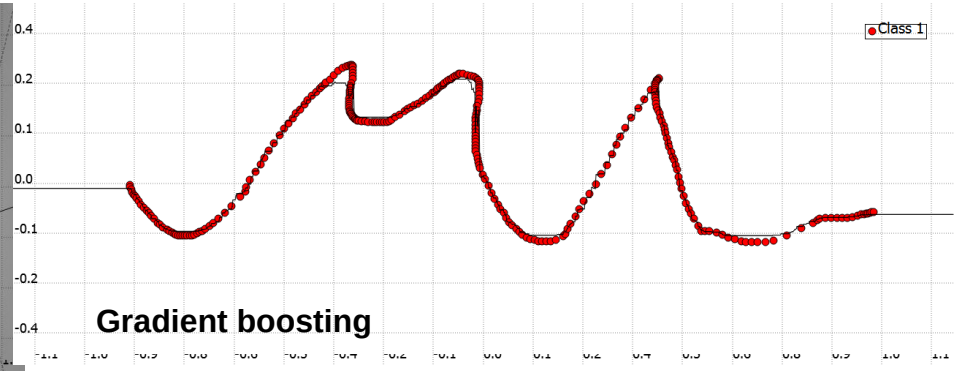
Estimate f that best predict set of training points $\{x^i, y^i\}_{i=1,...M}$?

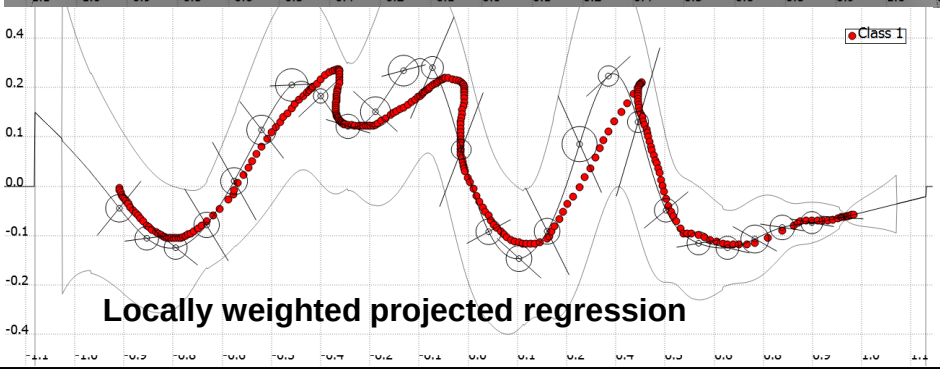
Regression Algorithms in this Course



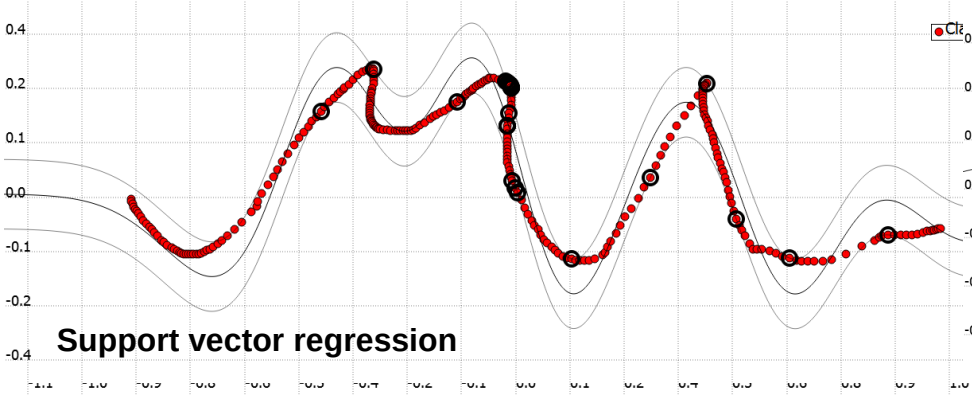


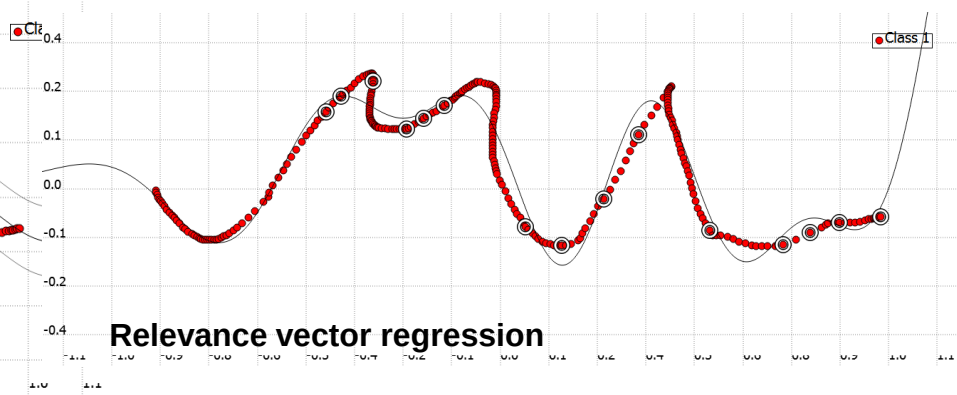






Today, we will see:





Support Vector Regression

Support Vector Regression

Assume a nonlinear mapping f , s.t. $y = f(x)$.

How to estimate f to best predict the pair of training points $\{x^i, y^i\}_{i=1,...M}$?

How to generalize the support vector machine framework for classification to estimate continuous functions?

1. **Assume a non-linear mapping through feature space and then perform *linear regression* in feature space**

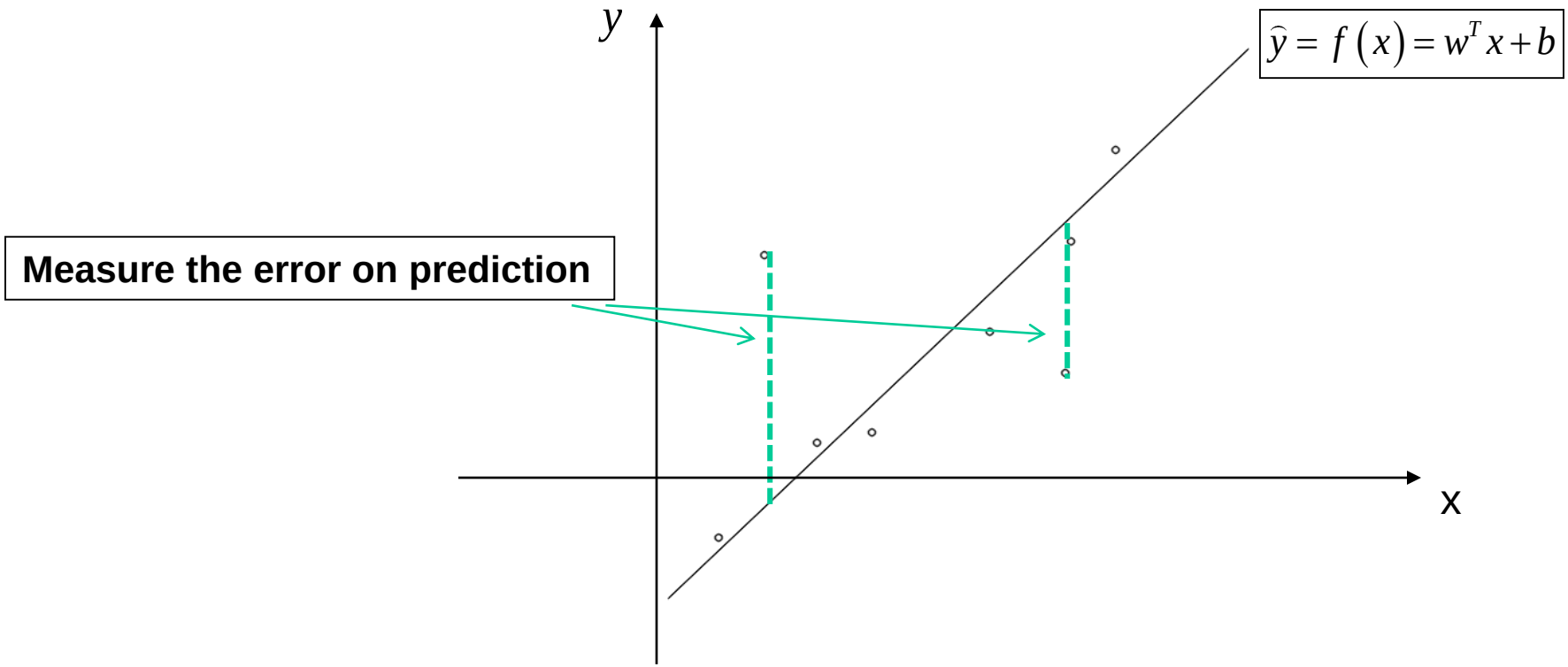
2. **Supervised learning – minimizes an error function.**

→ **First determine a way to measure error on testing set in the linear case!**

Support Vector Regression

Assume a linear mapping f , s.t. $y = f(x) = w^T x + b$.

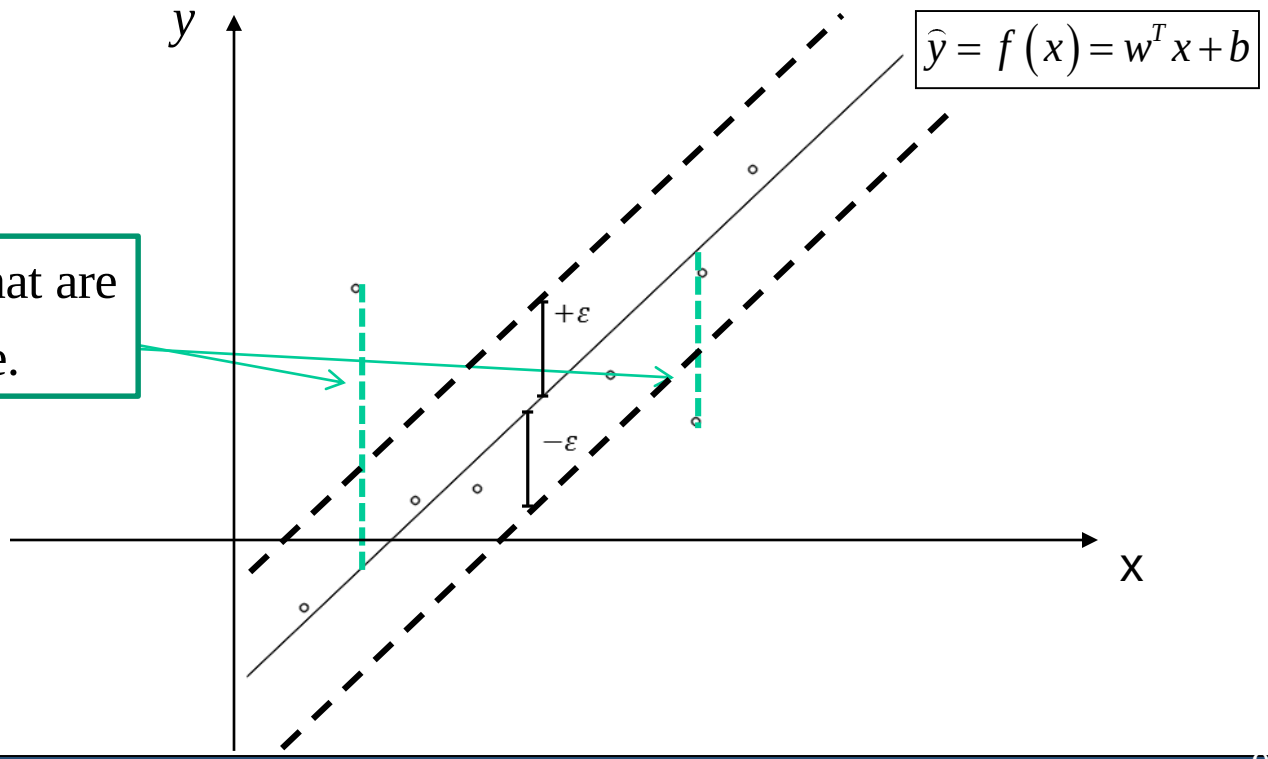
How to estimate w and b to best predict the pair of training points $\{x^i, y^i\}_{i=1,...M}$?



Support Vector Regression

Set an upper bound on the error ε and consider as correctly classified all points such that $|f(x) - y| \leq \varepsilon$.

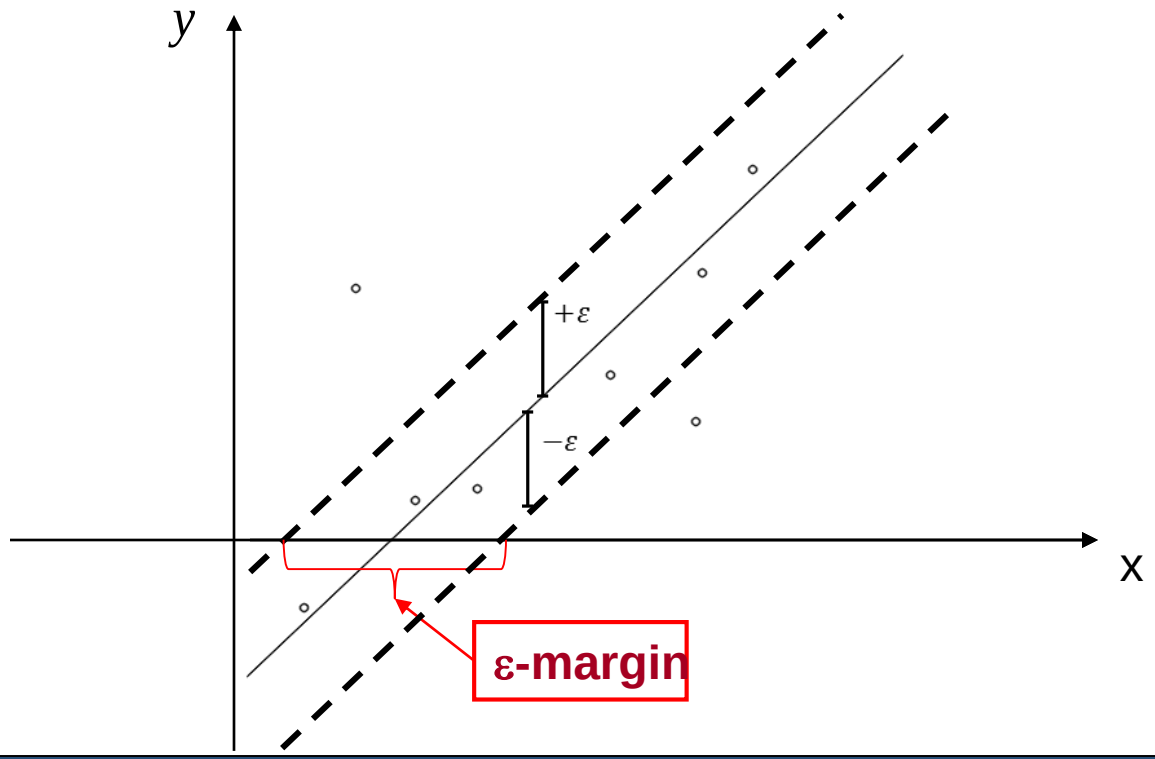
Penalize only datapoints that are not contained in the ε-tube.



Support Vector Regression

The ε-margin is a measure of the width of the ε-insensitive tube.
It is a measure of the precision of the regression.

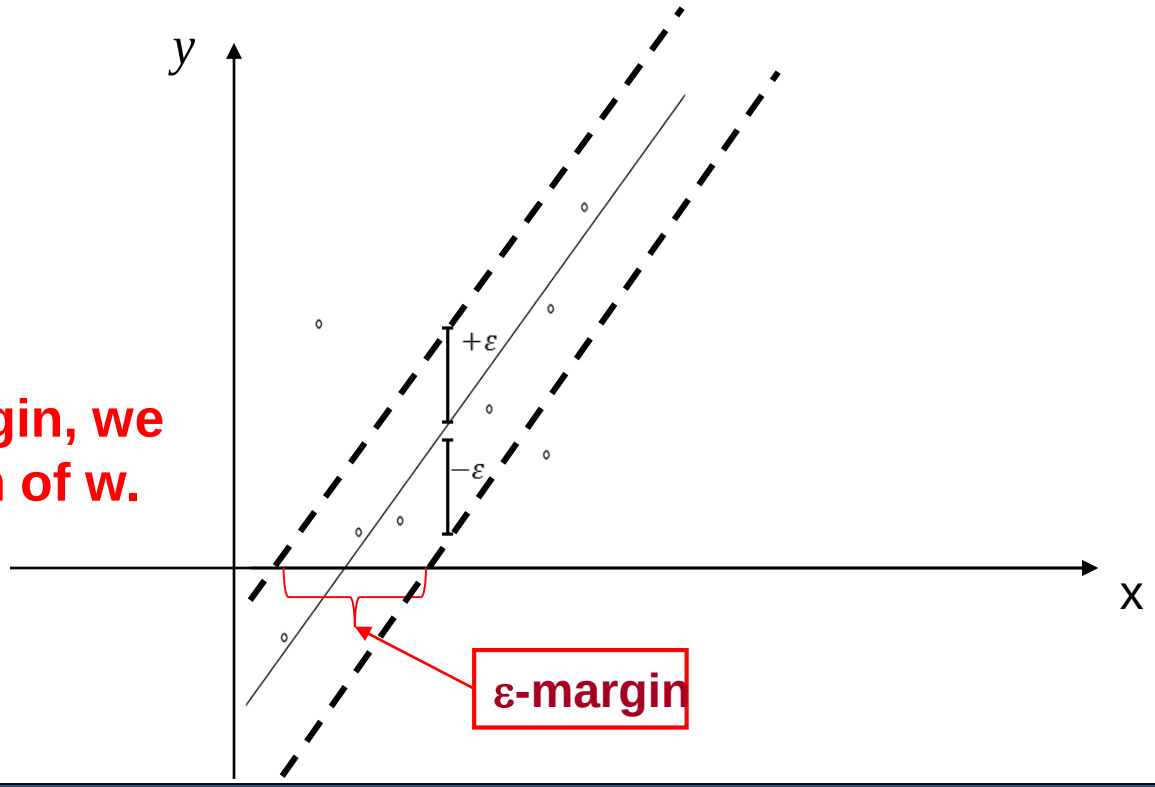
A small ||*w*|| corresponds to a small slope for *f*.
In the linear case, *f* is more horizontal.



Support Vector Regression

A large ||*w*|| corresponds to a large slope for *f*.
In the linear case, *f* is more vertical.

The flatter the slope of the function *f*, the larger the ε–margin.



→ To maximize the margin, we must minimize the norm of w.

Support Vector Regression

This can be rephrased as a constraint-based optimization problem of the form:

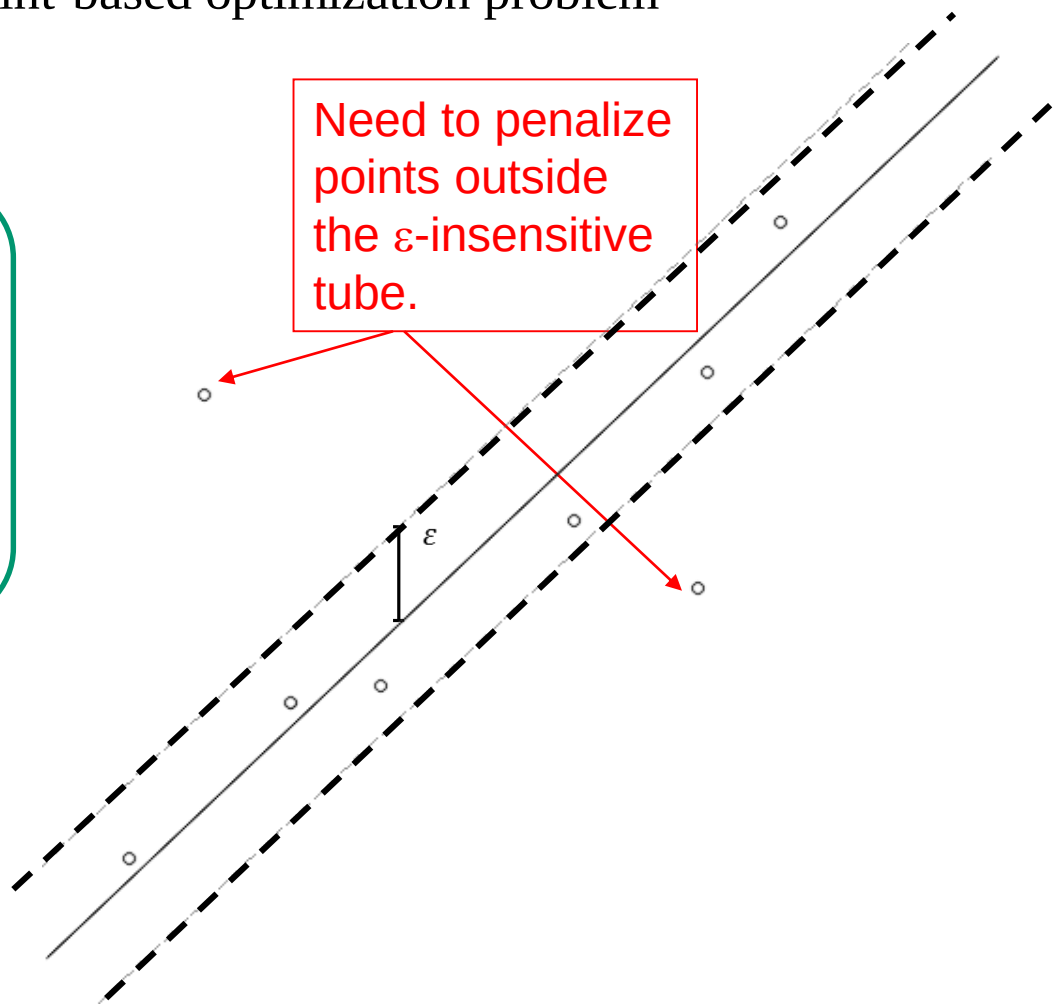
$$\text{minimize } \frac{1}{2}\|w\|^2$$

$$\text{subject to } \begin{cases} \langle w, x^i \rangle + b - y^i \leq \varepsilon \\ y^i - \langle w, x^i \rangle - b \leq \varepsilon \end{cases}$$

$$\forall i = 1, \ldots M$$

Consider as correctly classified all points such that $|f(x) - y| \leq \varepsilon$.

Need to penalize points outside the ε-insensitive tube.

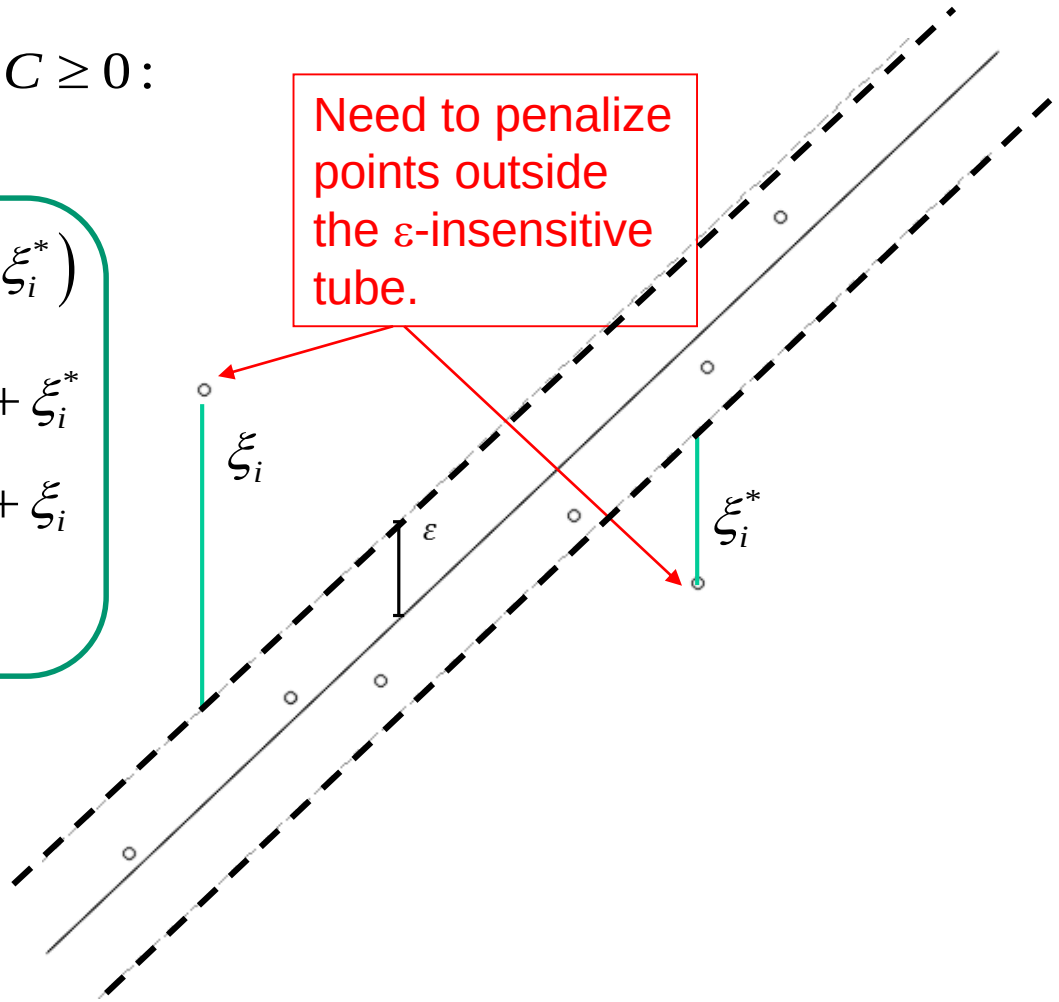


Support Vector Regression

Introduce slack variables $\xi_i, \xi_i^*, C \geq 0$:

$$\text{minimize} \quad \frac{1}{2}\|w\|^2 + \frac{C}{M}\sum_{i=1}^{M}\left(\xi_i + \xi_i^*\right)$$

$$\text{subject to} \quad \begin{cases} \left\langle w, x^i \right\rangle + b - y^i \leq \varepsilon + \xi_i^* \\ y^i - \left\langle w, x^i \right\rangle - b \leq \varepsilon + \xi_i \\ \xi_i \geq 0, \quad \xi_i^* \geq 0 \end{cases}$$



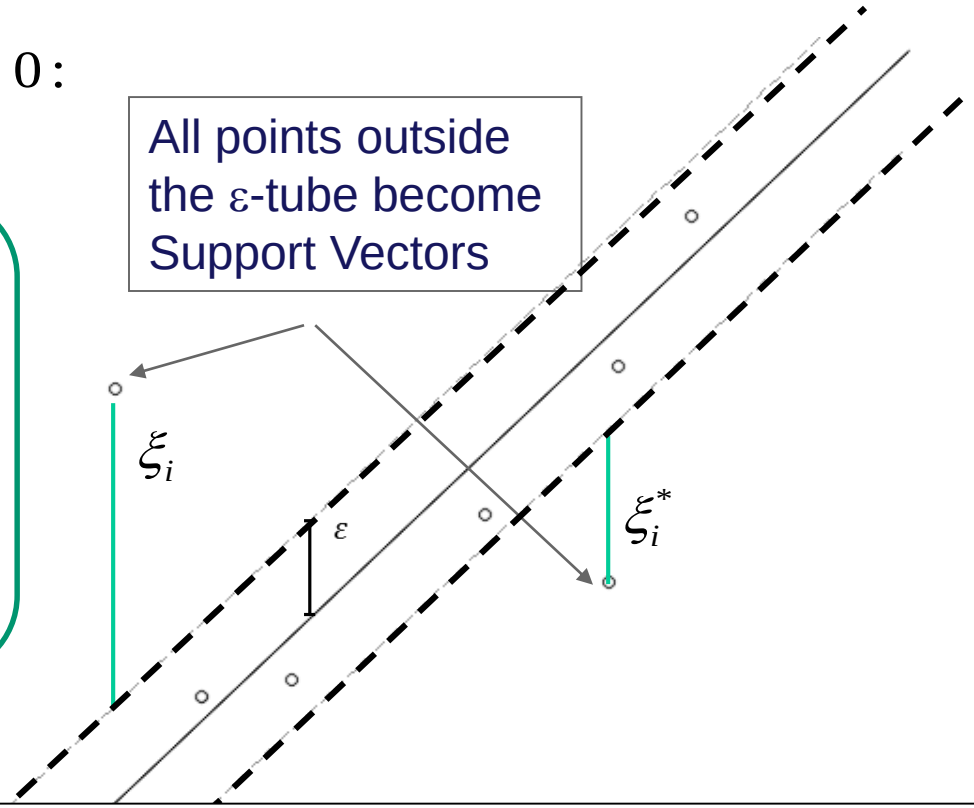
Support Vector Regression

Introduce slack variables $\xi_i, \xi_i^*, C \geq 0$:

$$\text{minimize} \quad \frac{1}{2}\|w\|^2 + \frac{C}{M}\sum_{i=1}^{M}\left(\xi_i + \xi_i^*\right)$$

$$\text{subject to} \quad \begin{cases} \left\langle w, x^i \right\rangle + b - y^i \leq \varepsilon + \xi_i^* \\ y^i - \left\langle w, x^i \right\rangle - b \leq \varepsilon + \xi_i \\ \xi_i \geq 0, \quad \xi_i^* \geq 0 \end{cases}$$

All points outside the ε-tube become Support Vectors



We now have the solution to the linear regression problem.

How to generalize this to the nonlinear case?

Support Vector Regression

We can solve this quadratic problem by introducing sets of $\alpha,\ \eta \in \mathbb{R}$ Lagrange multipliers and writing the Lagrangian :

Lagrangian = Objective function + multipliers * constraints

$$\mathrm{L}\left(w,\xi,\xi^{*},b\right)=\frac{1}{2}\|w\|^{2}+\frac{\mathrm{C}}{M}\sum_{i=1}^{M}\left(\xi_{i}+\xi_{i}^{*}\right)-\frac{\mathrm{C}}{M}\sum_{i=1}^{M}\left(\eta_{i}\xi_{i}+\eta_{i}^{*}\xi_{i}^{*}\right)$$

$$-\sum_{i=1}^{M}\alpha_{i}\left(\varepsilon+\xi_{i}-y^{i}+\left\langle w,x^{i}\right\rangle+b\right)$$

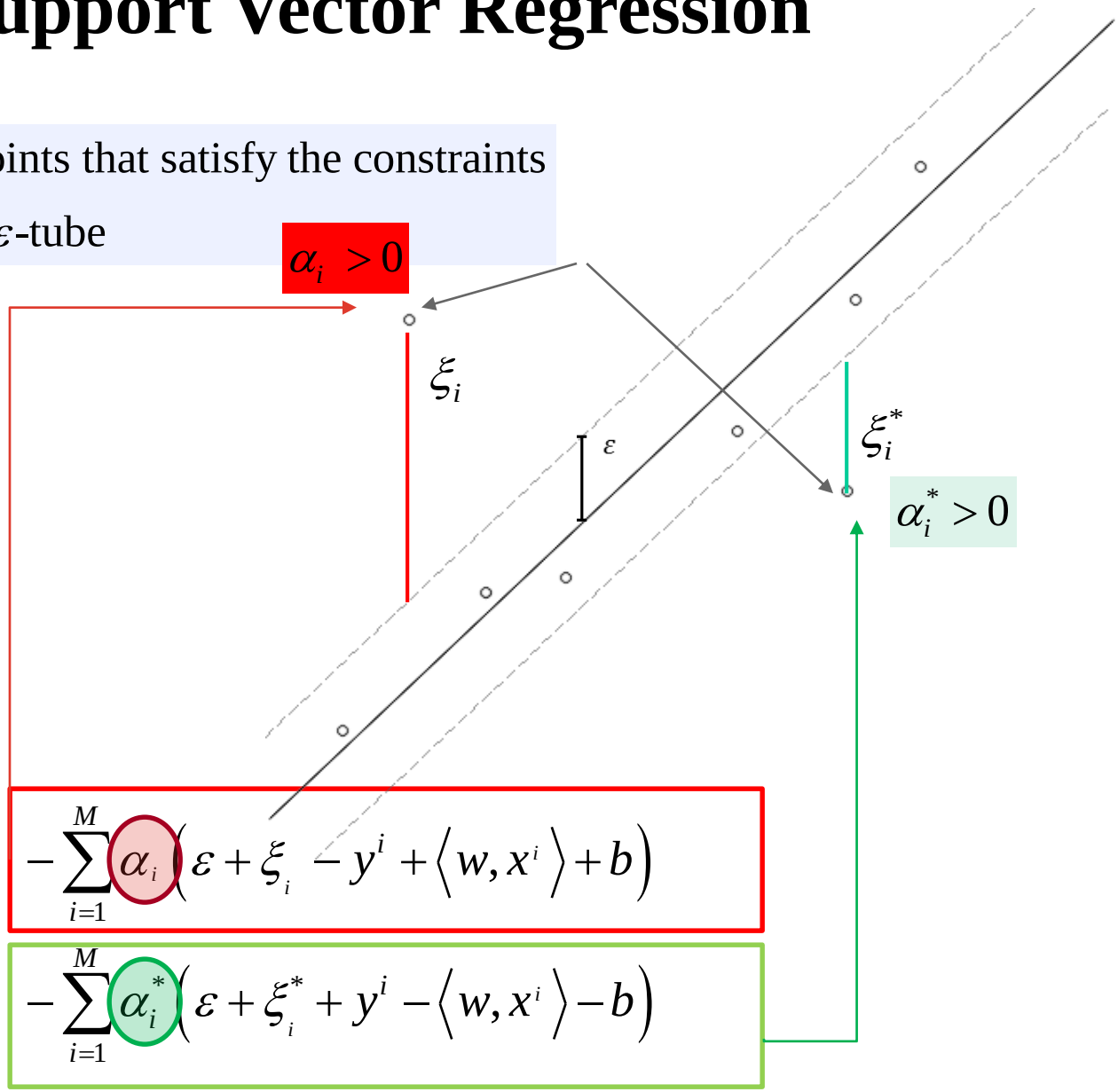
$$-\sum_{i=1}^{M}\alpha_{i}^{*}\left(\varepsilon+\xi_{i}^{*}+y^{i}-\left\langle w,x^{i}\right\rangle-b\right)$$

Support Vector Regression

$\alpha_i^* = \alpha_i = 0$ for all points that satisfy the constraints

$\rightarrow$ points inside the ε-tube

Constraints on points lying on either side of the ε-tube.



Support Vector Regression

Requiring that the partial derivatives are all zero:

$$\frac{\partial \mathrm{L}}{\partial b} = \sum_{i=1}^{M} \left(\alpha_i - \alpha_i^* \right) = 0. \qquad \rightarrow \sum_{i=1}^{M} \alpha_i = \sum_{i=1}^{M} \alpha_i^*$$

Rebalancing the effect of the support vectors on both sides of the ε-tube

$$\frac{\partial \mathrm{L}}{\partial w} = w - \sum_{i=1}^{M} \left(\alpha_i - \alpha_i^* \right) x^i = 0. \qquad \Rightarrow w = \sum_{i=1}^{M} \left(\alpha_i - \alpha_i^* \right) x^i.$$

Linear combination of support vectors

The solution is given by:

$$\begin{aligned} y &= f(x) \\ &= \langle w, x \rangle + b \\ &= \sum_{i=1}^{M} \left(\alpha_i - \alpha_i^* \right) \langle x^i, x \rangle + b \end{aligned}$$

Support Vector Regression

Lift x **into** feature space and then perform linear regression in feature space.

Linear Case:

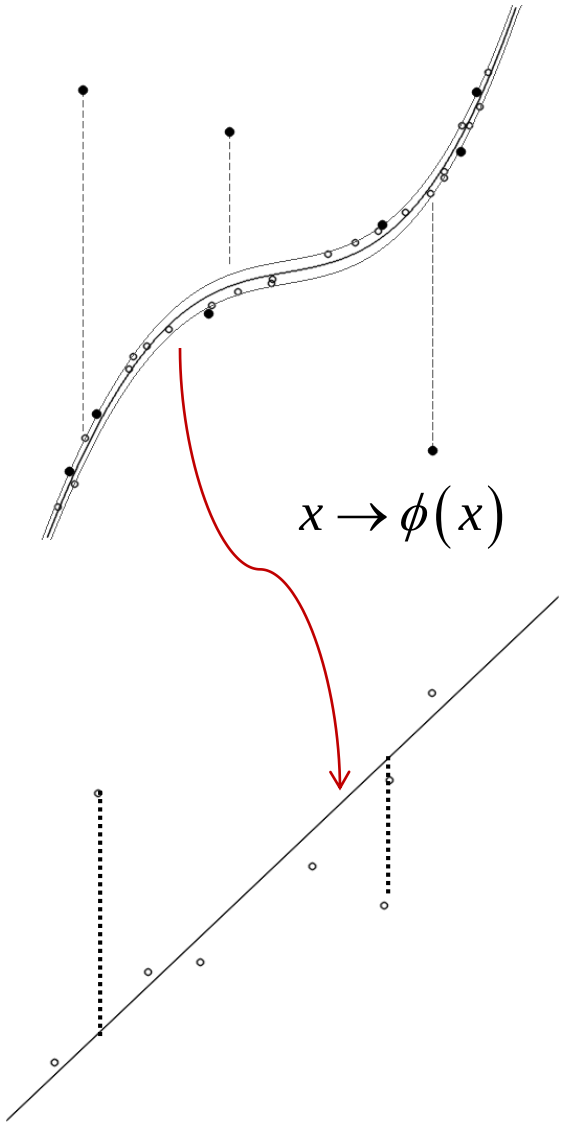
$$y = f(x) = \langle w, x \rangle + b$$

Non-Linear Case:

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

$$y = f(\phi(x)) = \langle w, \phi(x) \rangle + b$$

w lives in feature space!



Support Vector Regression

In feature space, we obtain the same constrained optimization problem:

$$\text{minimize} \quad \frac{1}{2}\|w\|^2 + \frac{C}{M}\sum_{i=1}^{M}\left(\xi_i + \xi_i^*\right)$$

$$\text{subject to} \quad \begin{cases} \left\langle w, \phi\left(x^i\right)\right\rangle + b - y^i \le \varepsilon + \xi_i^* \\ y^i - \left\langle w, \phi\left(x^i\right)\right\rangle - b \le \varepsilon + \xi_i \\ \xi_i \ge 0, \quad \xi_i^* \ge 0 \end{cases}$$

Support Vector Regression

We can solve this quadratic problem by introducing sets of $\alpha,\ \eta \in \mathbb{R}$ Lagrange multipliers and writing the Lagrangian :

Lagrangian = Objective function + multipliers * constraints

$$L\left(w,\xi,\xi^{*},b\right)=\frac{1}{2}\|w\|^{2}+\frac{C}{M}\sum_{i=1}^{M}\left(\xi_i+\xi_i^{*}\right)-\frac{C}{M}\sum_{i=1}^{M}\left(\eta_i\xi_i+\eta_i^{*}\xi_i^{*}\right)$$

$$-\sum_{i=1}^{M}\alpha_i\left(\varepsilon+\xi_i^{*}+y^{i}-\left\langle w,\phi\left(x^{i}\right)\right\rangle-b\right)$$

$$-\sum_{i=1}^{M}\alpha_i^{*}\left(\varepsilon+\xi_i-y^{i}+\left\langle w,\phi\left(x^{i}\right)\right\rangle+b\right)$$

Support Vector Regression

And replacing in the primal Lagrangian, we get the Dual optimization problem:

$$\max_{\alpha,\alpha^*} \begin{cases} -\frac{1}{2}\sum_{i,j=1}^{M}\left(\alpha_i^* - \alpha_i\right)\left(\alpha_j^* - \alpha_j\right)\cdot k\left(x^i, x^j\right) \\ -\varepsilon\sum_{i=1}^{M}\left(\alpha_i^* + \alpha_i\right) + \sum_{i=1}^{M} y^i\left(\alpha_i^* + \alpha_i\right) \end{cases}$$

$$\text{subject to } \sum_{i=1}^{M}\left(\alpha_i^* - \alpha_i\right) = 0 \text{ and } \alpha_i^*, \alpha_i{}^i \in \left[0, \frac{C}{M}\right]$$

Kernel Trick

$$k\left(x^i, x^j\right) = \left\langle \phi\left(x^i\right), \phi\left(x^j\right)\right\rangle$$

Support Vector Regression

The solution is given by:

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

Linear Coefficients **(Lagrange multipliers for each constraint).**

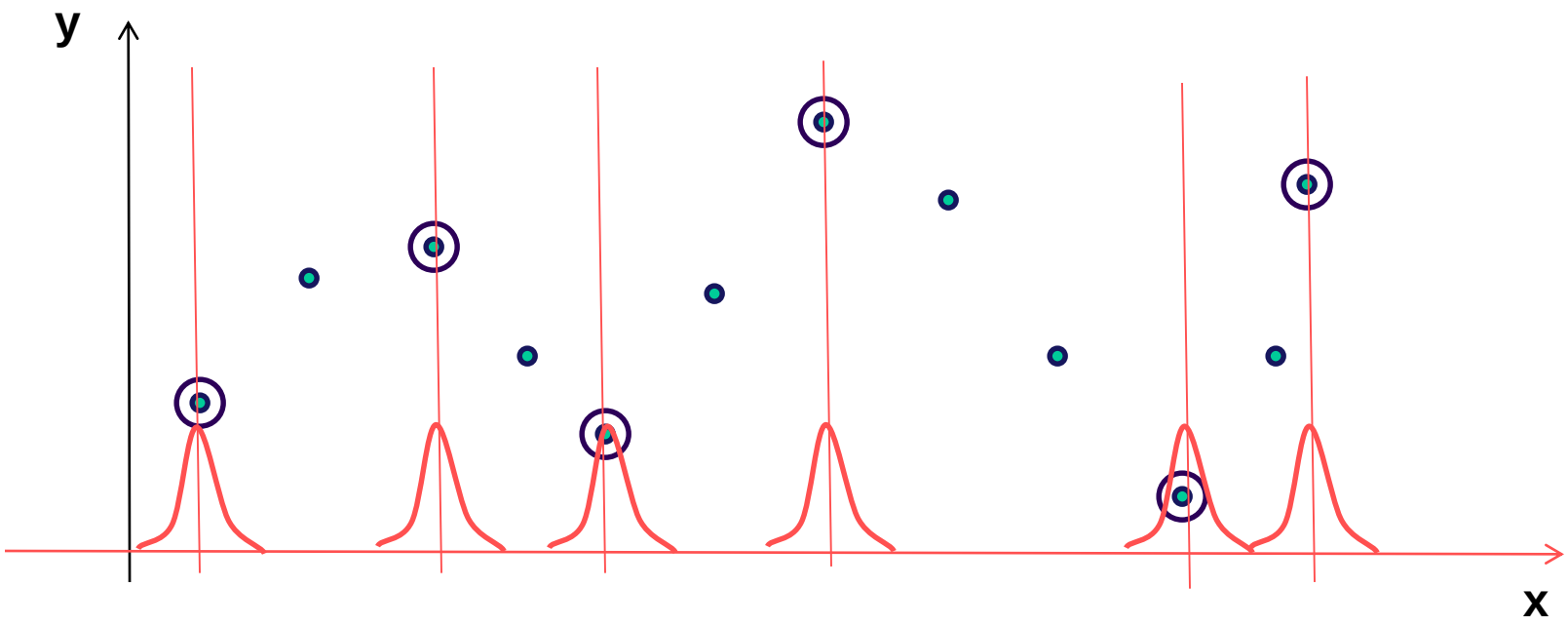
If one uses RBF Kernel, *M* un-normalized isotropic Gaussians centered on each training datapoint.

Support Vector Regression

The solution is given by:

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

Kernel places a Gaussian function on each SV



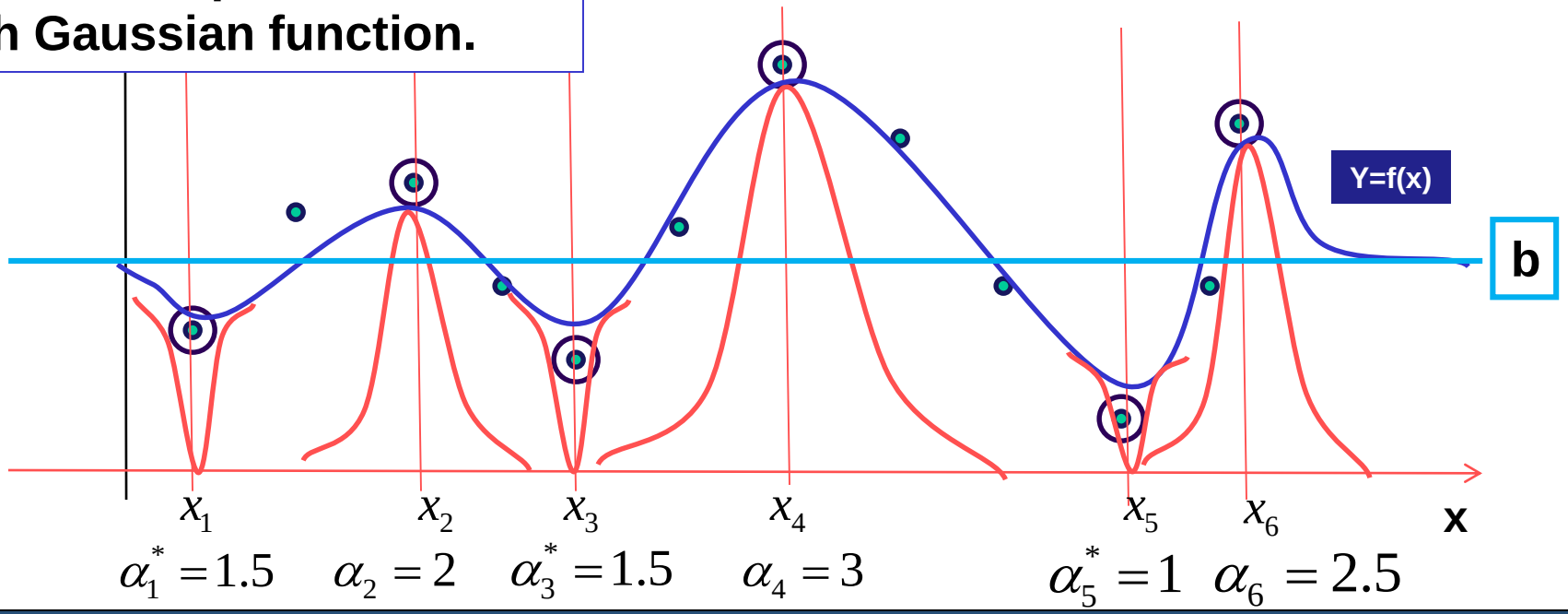
Support Vector Regression

The solution is given by:

Converges to b when SV effect vanishes.

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

The Lagrange multipliers define the importance of each Gaussian function.



Support Vector Regression: Exercise I

SVR gives the following estimate for each pair of datapoints $\{y^j, x^j\}$

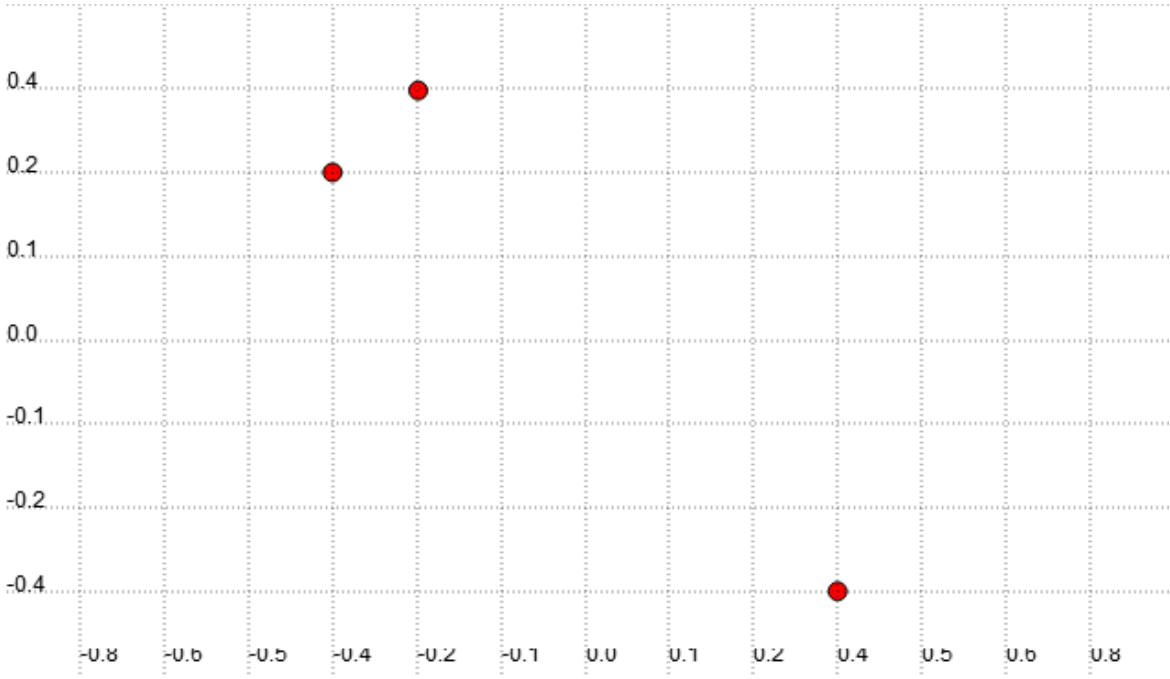
$$y^j = \sum_{i=1}^{M} (\alpha_i - \alpha_i^*) k(x^j, x^j) + b, \quad \text{i} = 1...M$$

For the set of 3 points drawn below,

a) compute an estimate of b using: $b = \frac{1}{M} \sum_{j=1}^{M} \left(y^j - \sum_{i=1}^{M} (\alpha_i - \alpha_i^*) k(x^i, x^j) \right)$

with RBF kernel, and plot b.

b) Plot the regressive curve and show how it varies depending on the kernel width and ε.



Support Vector Regression: Exercise I Solution

a) To answer this question, you must first determine the number of SVs.

This depends on epsilon. If we assume a small epsilon, all 3 points become SVs.

b is then influenced by the value of the kernel width. With a very small kernel width

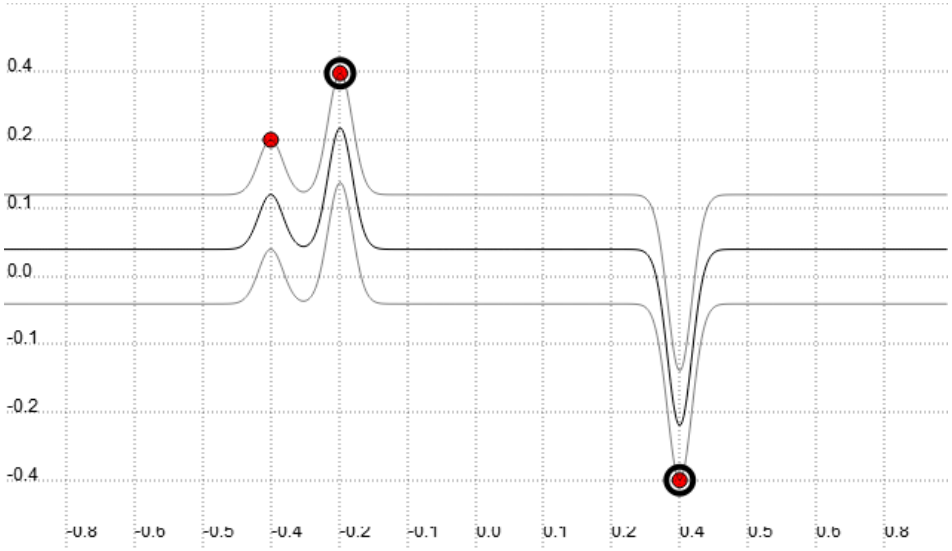
$k\left(x^j, x^i\right) \cong 0 \ \forall x^j \neq x^i$ and then, $b = \frac{1}{M}\sum_{j=1}^{M}\left(y^j - \underbrace{\sum_{i=1}^{M}\left(\alpha_i - \alpha_i^*\right)1}_{=0}\right)$ is the mean of the data.

As the kernel width grows, *b* is pulled toward the SVs modulated by the kernel

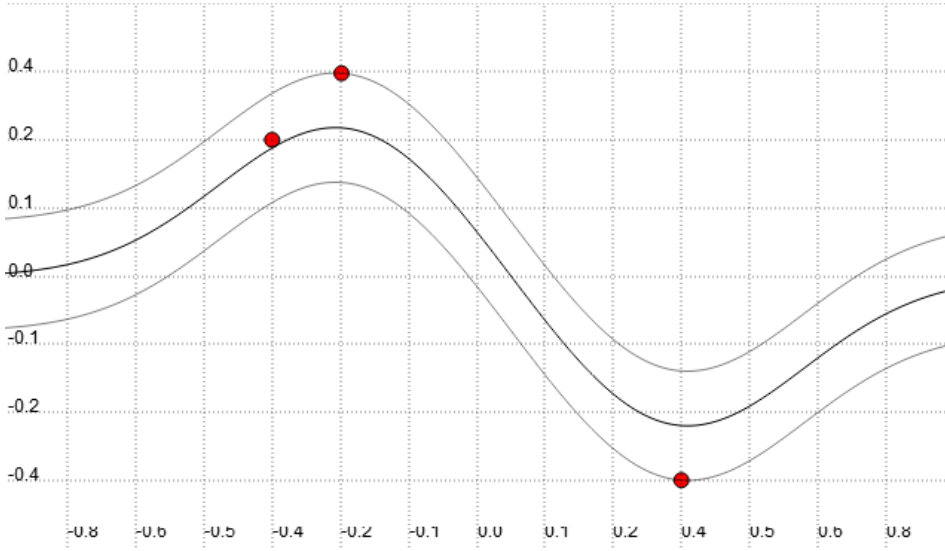
$$b = \frac{1}{M}\sum_{j=1}^{M}\left(y^j - \sum_{i\neq j}^{M}\left(\alpha_i - \alpha_i^*\right)k\left(x^i, x^j\right)\right)$$

With only 2 SVs, the influence of the SVs cancels out and we are back to the mean of the data.

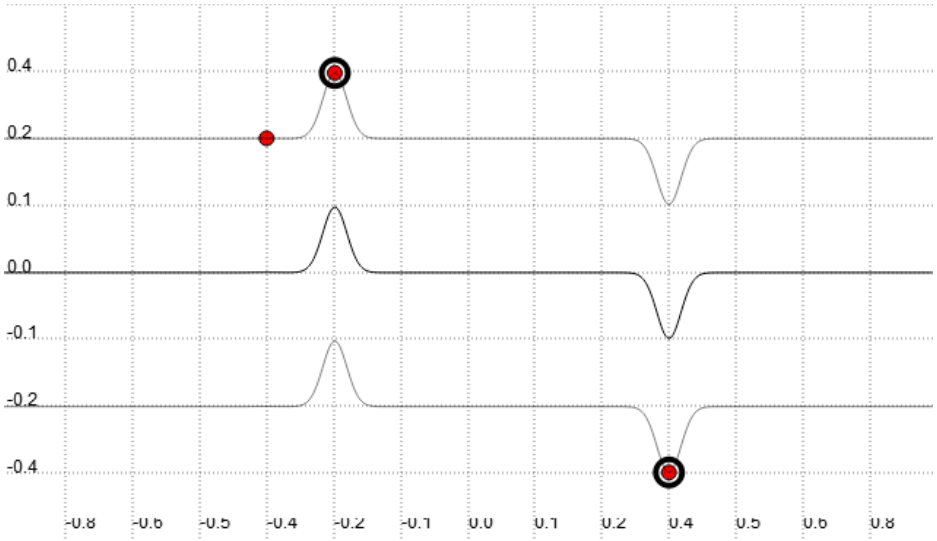
Support Vector Regression: Exercise I Solution



Kernel width = 0.001 , epsilon = 0.1



Kernel width = 0.1 , epsilon = 0.1



Kernel width = 0.001 , epsilon = 0.24

With small kernel width, the effect of each SV is well separated and the curve comes back to b in-between two SVs.

With a large kernel width, b changes and the curve yields a smooth interpolation in-between SVs.

With a large epsilon, one point is absorbed in the epsilon tube and is no longer a SV.

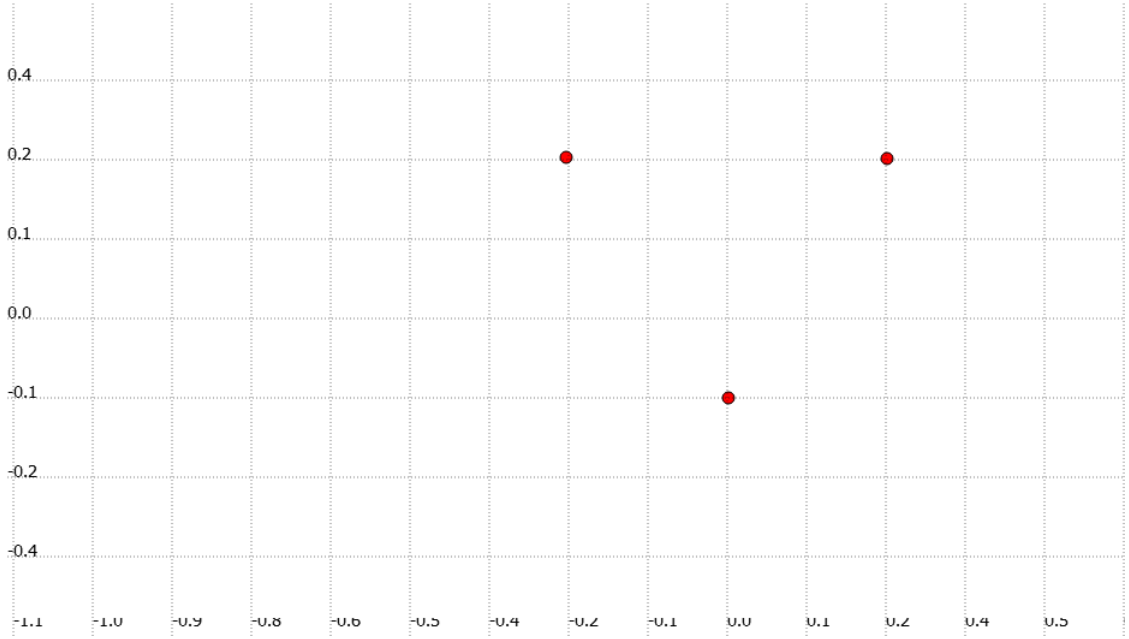
Support Vector Regression: Exercise II

Recall the solution to SVM:

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

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

b) What minimum order of a homogeneous polynomial kernel do you need to achieve good regression on the set of 3 points below?



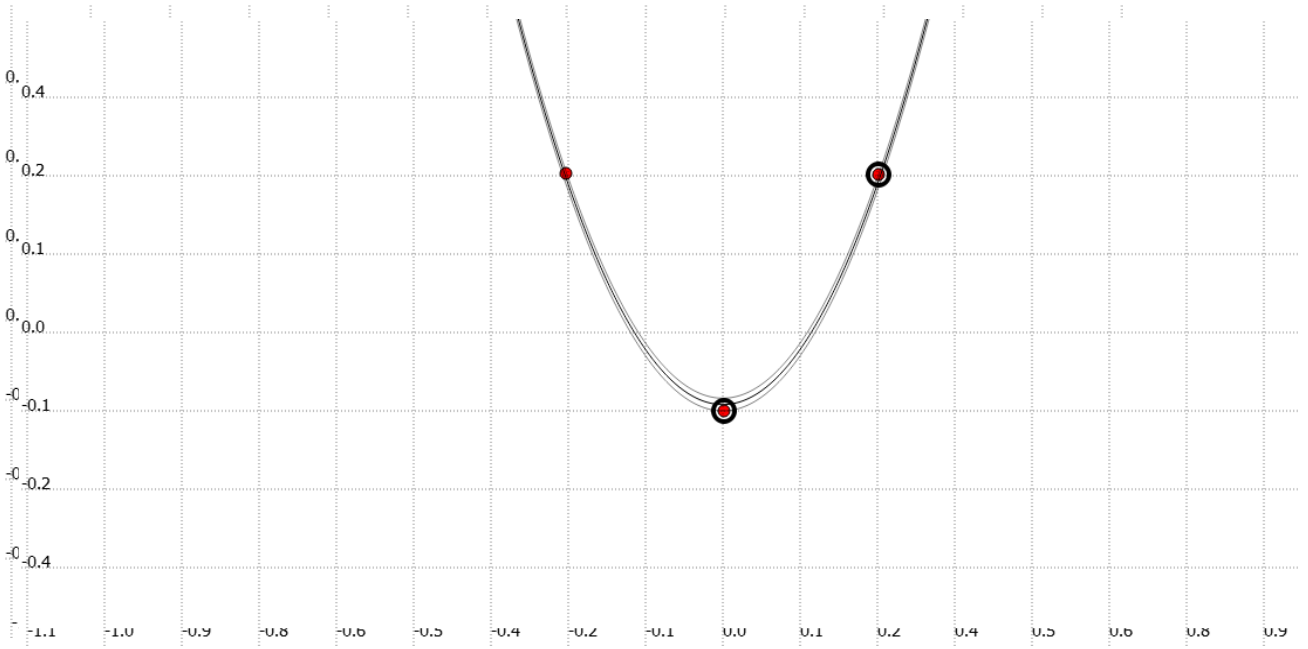
Support Vector Regression: Solutions Exercise II

The equation for a homogeneous polynomial in the 1-D case below is:

$$y = \sum_i (\alpha_i - \alpha_i^*)(x_1^i x)^p + b : \text{ single term scaled \& shifted polynomial}$$

For the set of points below, we need at minimum p=2 and 2 SVs.

See also the supplementary exercises posted on the class's website!



ε–SVR: Hyperparameters

The solution to SVR we just saw is referred to as ε–SVR

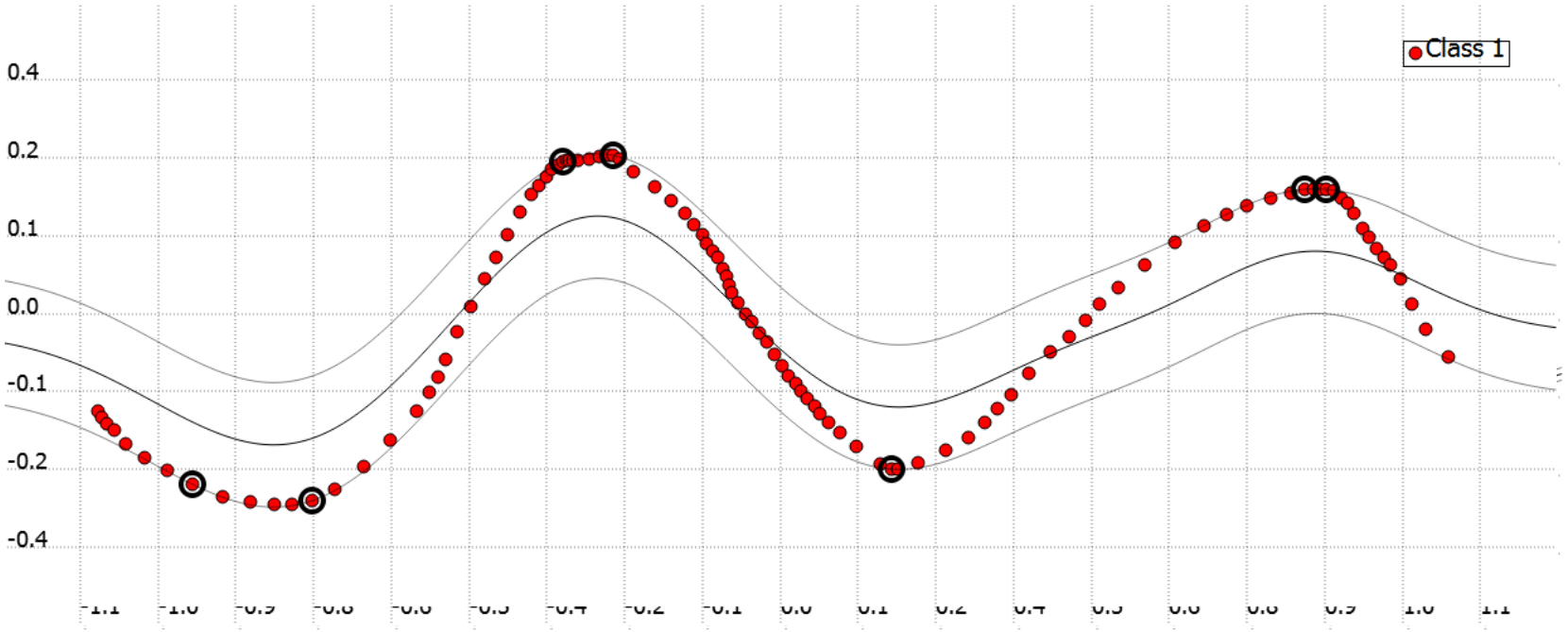
Two Hyperparameters

$$\text{minimize } \frac{1}{2}\|w\|^2 + \frac{C}{M}\sum_{i=1}^{M}\left(\xi_i + \xi_i^*\right)$$

$$\text{subject to } \begin{cases} \langle w, x^i \rangle + b - y^i \leq \varepsilon + \xi_i \\ y^i - \langle w, x^i \rangle - b \leq \varepsilon + \xi_i^* \\ \xi_i \geq 0, \quad \xi_i^* \geq 0 \end{cases}$$

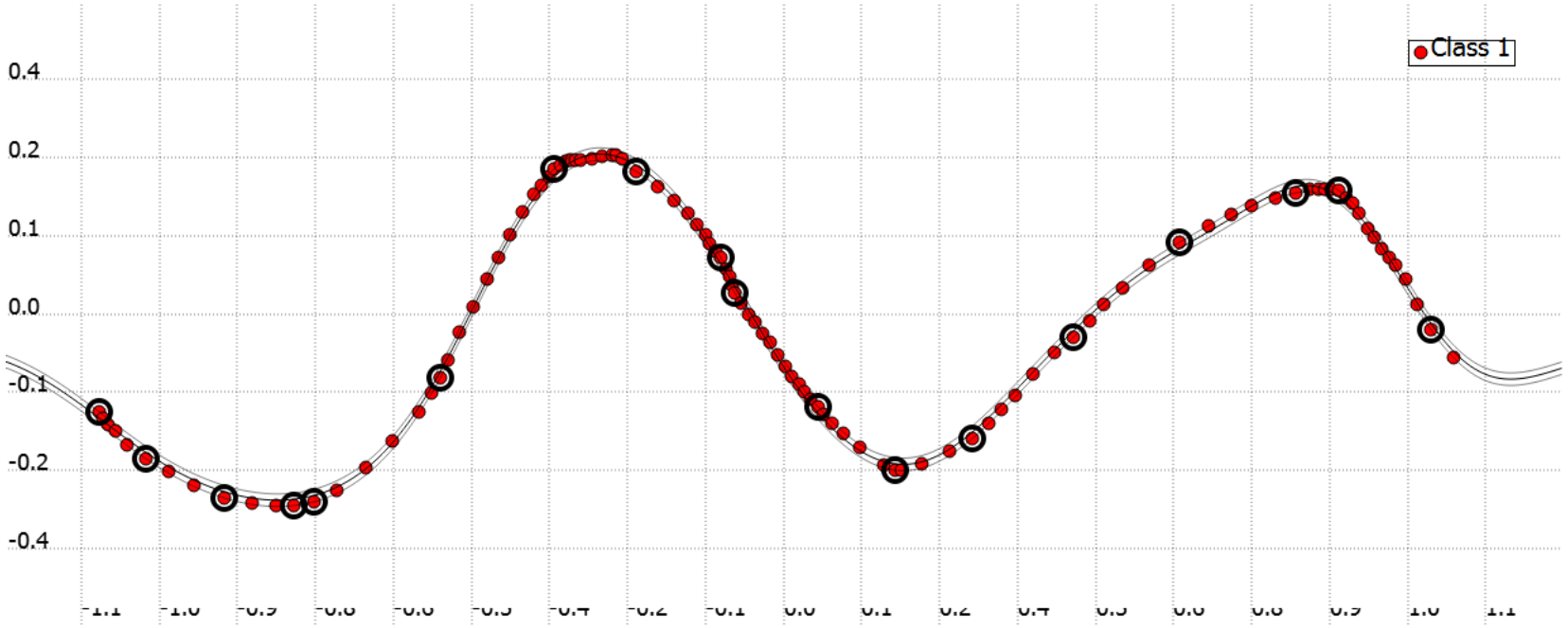
C controls the penalty term on poor fit.
ε determines the minimal required precision.

ε–SVR: Effect of Hyperparameters



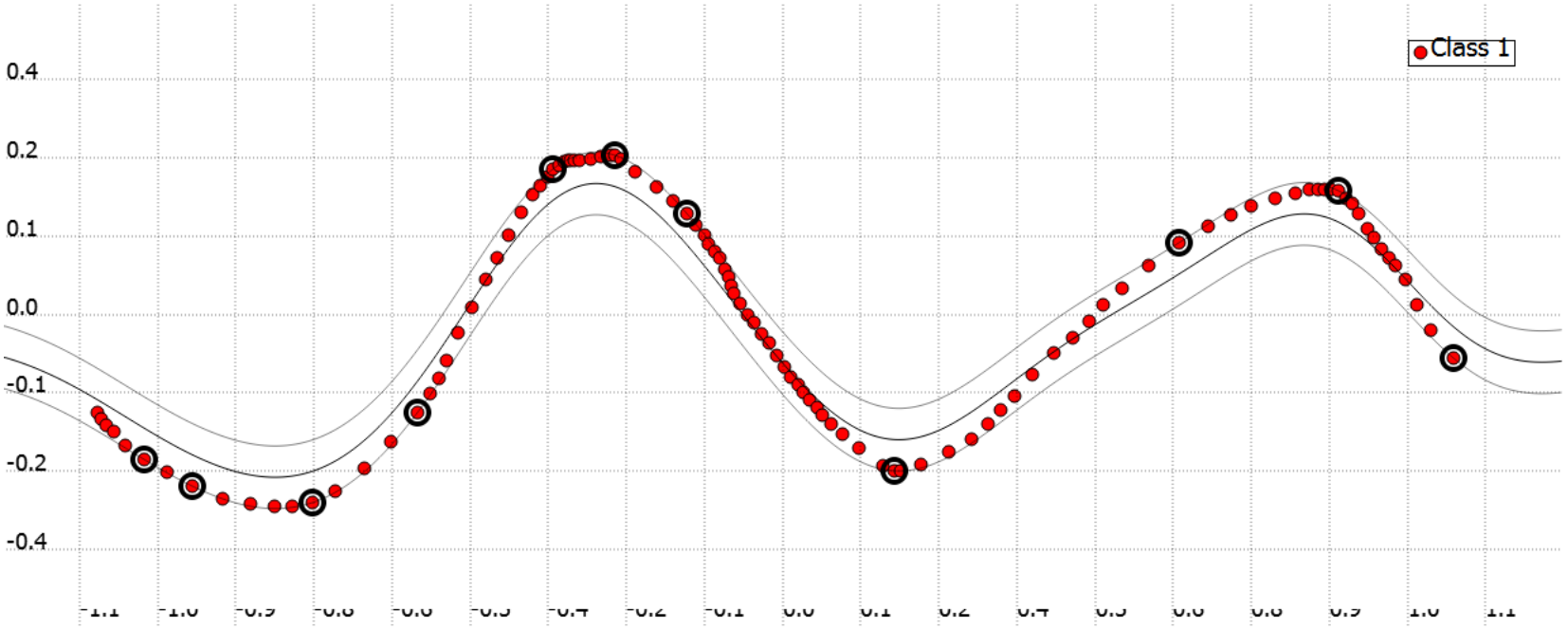
Effect of the RBF kernel width on the fit. Here fit using C=100, ε=0.1, kernel width=0.01.

ε–SVR: Effect of Hyperparameters



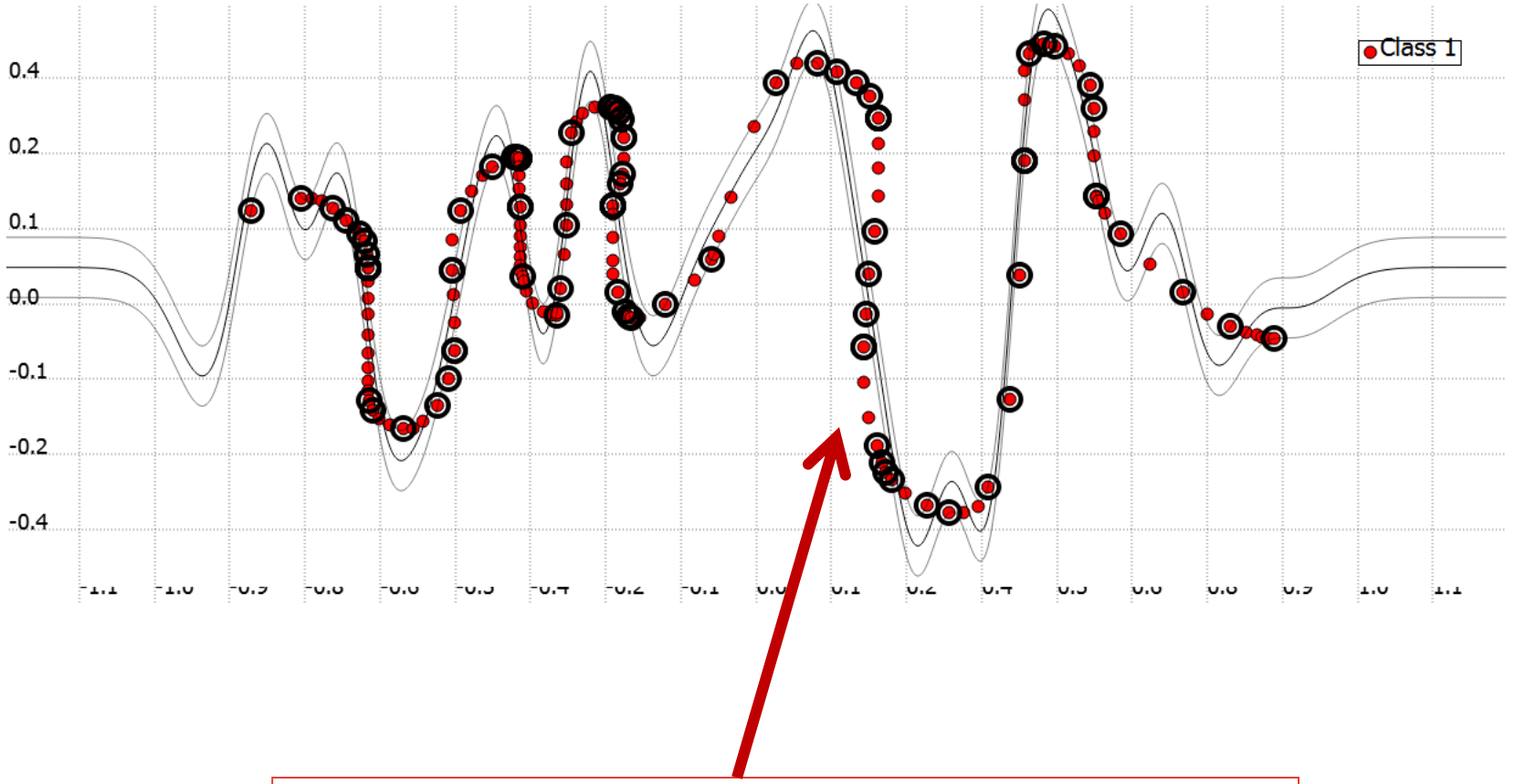
Effect of the RBF kernel width on the fit. Here fit using C=100, ε=0.01, kernel width=0.01 → Overfitting

ε–SVR: Effect of Hyperparameters



Effect of the RBF kernel width on the fit. Here fit using C=100, ε=0.05, kernel width=0.01 ***Reduction of the effect of the kernel width on the fit by choosing appropriate hyperparameters*****.**

ε–SVR: Effect of Hyperparameters



Mldemos does not display the support vectors if there is more than one point for the same x!

ε–SVR: Hyperparameters

The solution to SVR we just saw is referred to as ε–SVR

Two Hyperparameters

$$\text{minimize } \frac{1}{2}\|w\|^2 + \frac{C}{M}\sum_{i=1}^{M}\left(\xi_i + \xi_i^*\right)$$

$$\text{subject to } \begin{cases} \left\langle w, x^i \right\rangle + b - y^i \leq \varepsilon + \xi_i \\ y^i - \left\langle w, x^i \right\rangle - b \leq \varepsilon + \xi_i^* \\ \xi_i \geq 0, \quad \xi_i^* \geq 0 \end{cases}$$

C controls the penalty term on poor fit
ε determines the minimal required precision

Extensions of SVR

As in the classification case, the optimization framework used for support vector regression is extended with:

– ν-SVR: yielding a sparser version of SVR and relaxing the constraint of choosing ε, the width of the ε–insensitive tube.

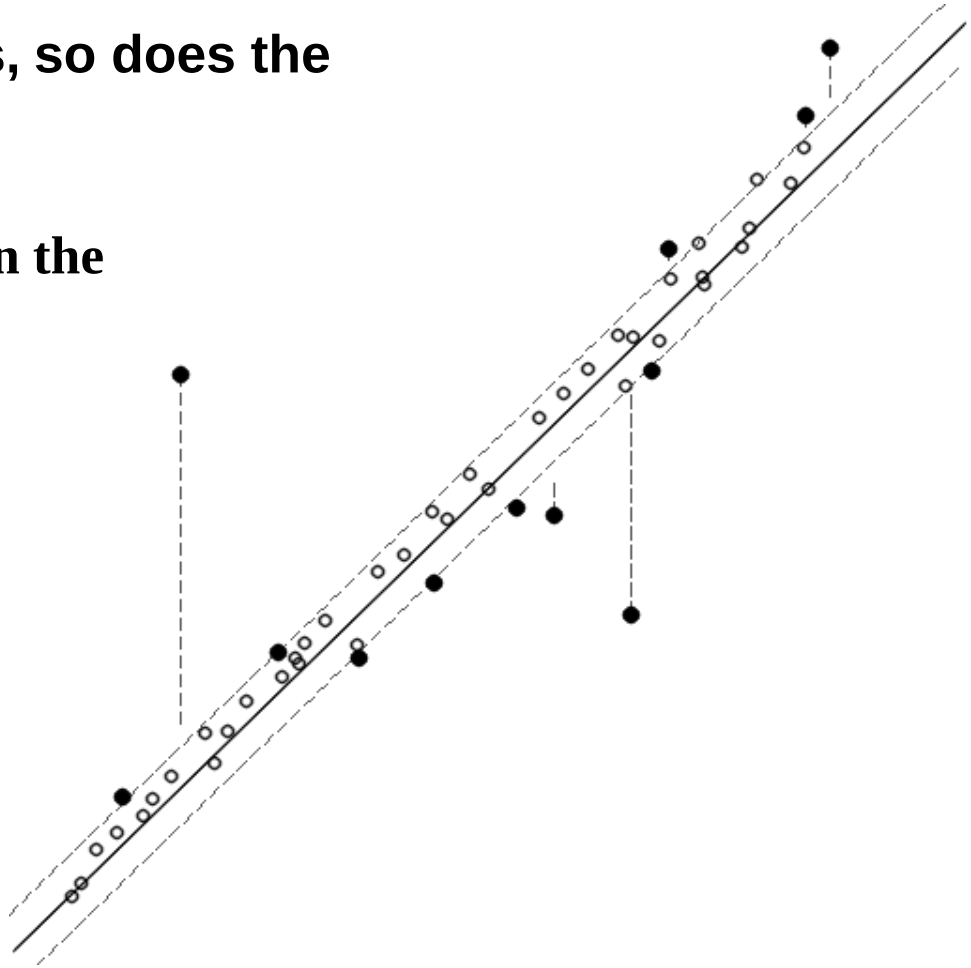
- Relevance Vector Regression: the regression version of RVM, which provides also a sparser version of SVM and offers a probabilistic interpretation of the solution.
  (see Tipping 2011, supplementary material to the class)

Support Vector Regression: ν-SVR

As the number of data grows, so does the number of support vectors.

ν−SVR puts a lower bound on the fraction of support vectors (see previous case for SVM)

$$\nu \in [0,1]$$



Support Vector Regression: ν-SVR

As for ν-SVM, one can rewrite the problem as a convex optimization expression:

$$\min_{w,\xi,\rho}\left(\frac{1}{2}\|w\|^2 + C\left[\nu\varepsilon + \frac{1}{M}\sum_{j=1}^{M}\left(\xi_j + \xi_j^*\right)\right]\right) \quad \text{under constraints}$$

$$\left(w^T \cdot x^j + b\right) - y^j \geq \varepsilon + \xi_j,$$

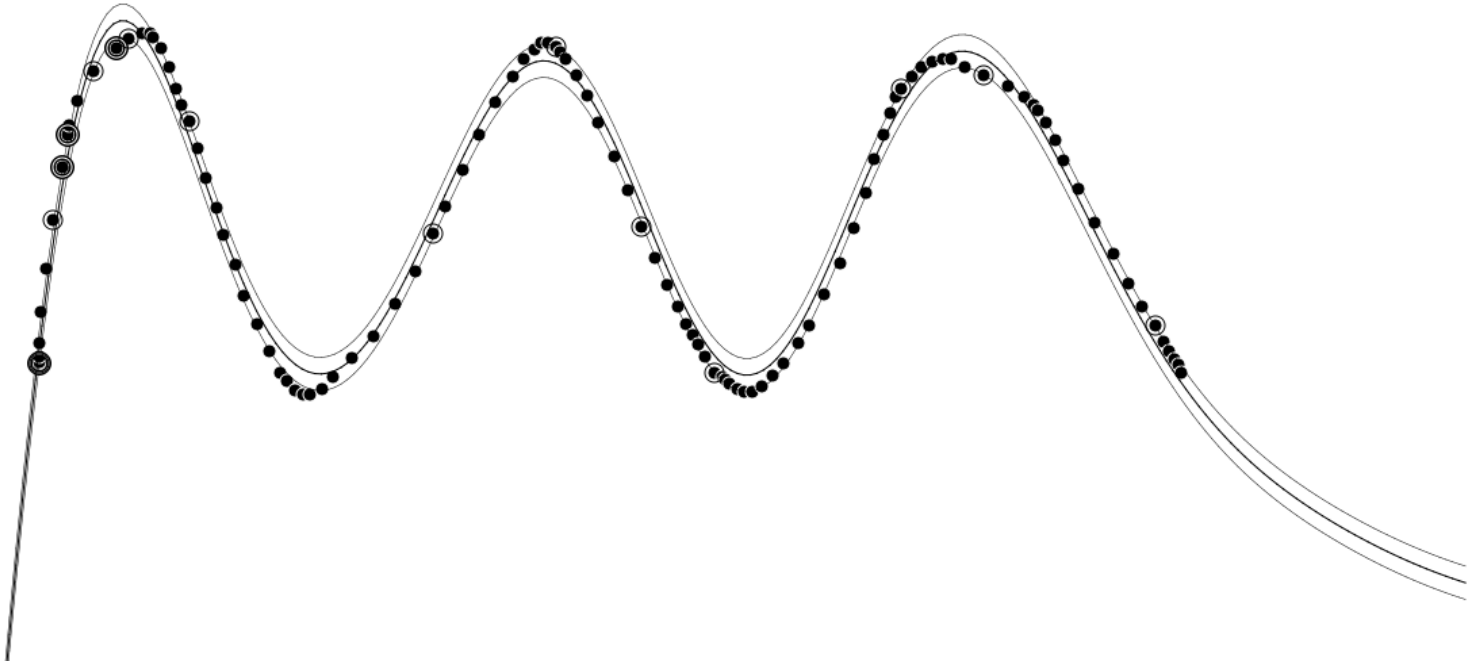
$$y^j - \left(w^T \cdot x^j + b\right) \geq \varepsilon + \xi_j^*,$$

$$\varepsilon \geq 0, \quad 0 \leq \nu \leq 1, \quad \xi_j \geq 0, \ \xi_j^* \geq 0.$$

The margin error is given by all the data points for which $\xi_j > 0$.

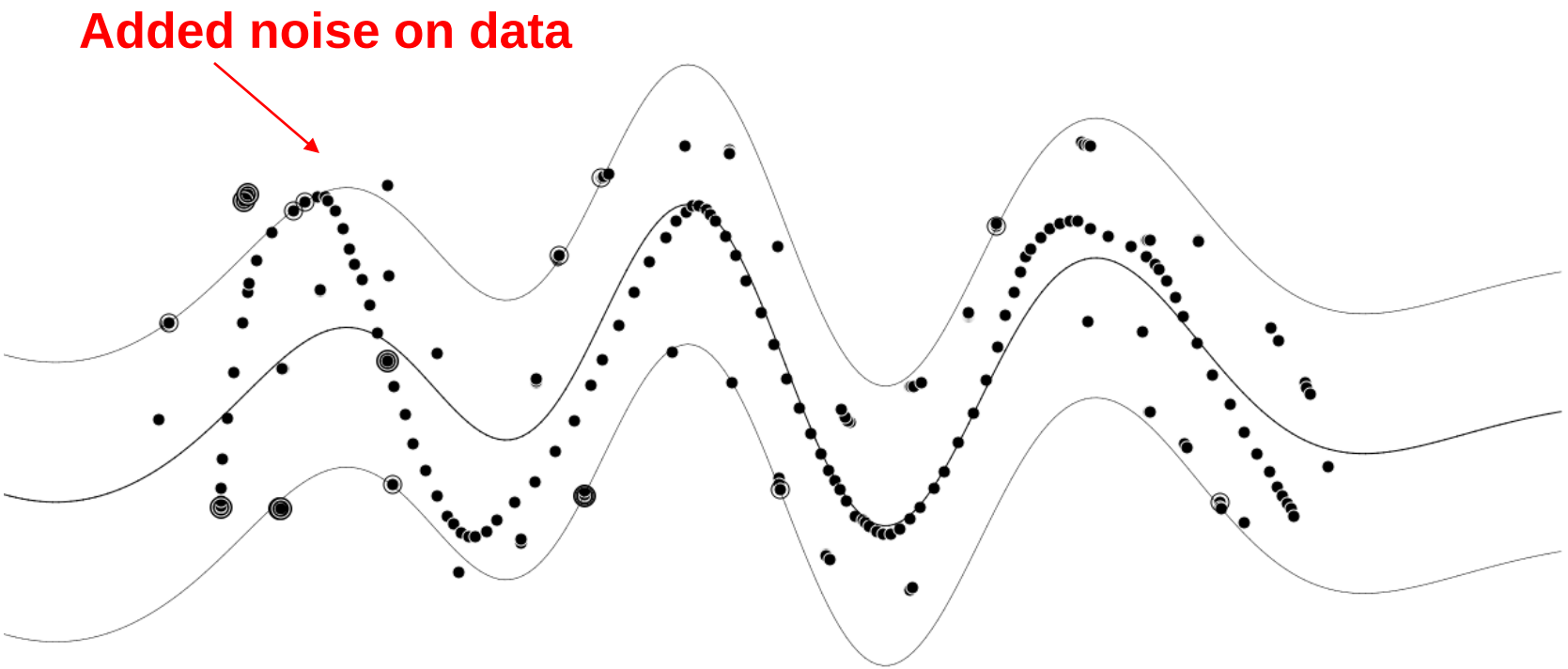
ν is an upper bound on the fraction of training error and a lower bound on the fraction of support vectors.

ν–SVR: Example



Effect of the automatic adaptation of ε using ν-SVR

ν–SVR: Example



Effect of the automatic adaptation of ε using ν-SVR

Relevance Vector Regression (RVR)

Same principle as that described for RVM (see slides on SVM and extensions). The derivation of the parameters however differ (see Tipping 2011 for details).

To recall, we start from the solution of SVM.

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

Rewrite the solution of SVM as a linear combination over M basis functions

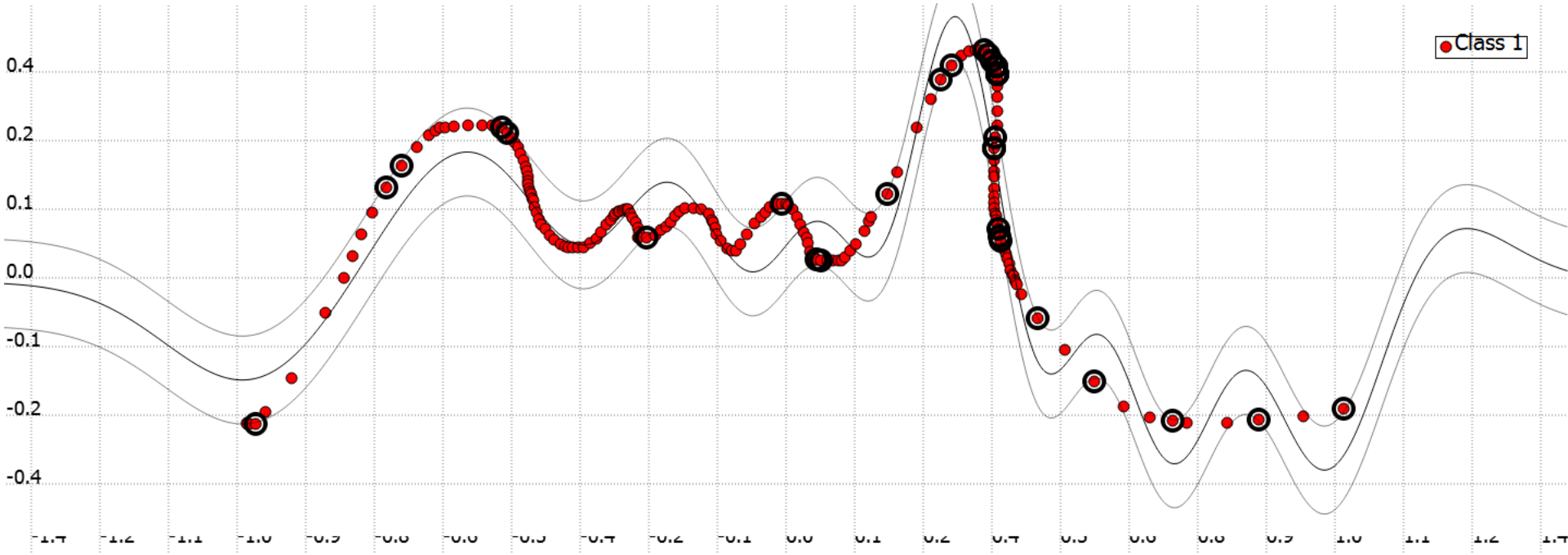
A sparse solution has a majority of entries with alpha zero.

$$\alpha = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ . \\ . \\ . \\ \alpha_M \end{bmatrix} = \begin{bmatrix} \alpha_1 \\ 0 \\ \alpha_3 \\ 0 \\ 0 \\ . \\ \alpha_M \end{bmatrix}$$

In the (binary) classification case, $y \in [0;1]$.

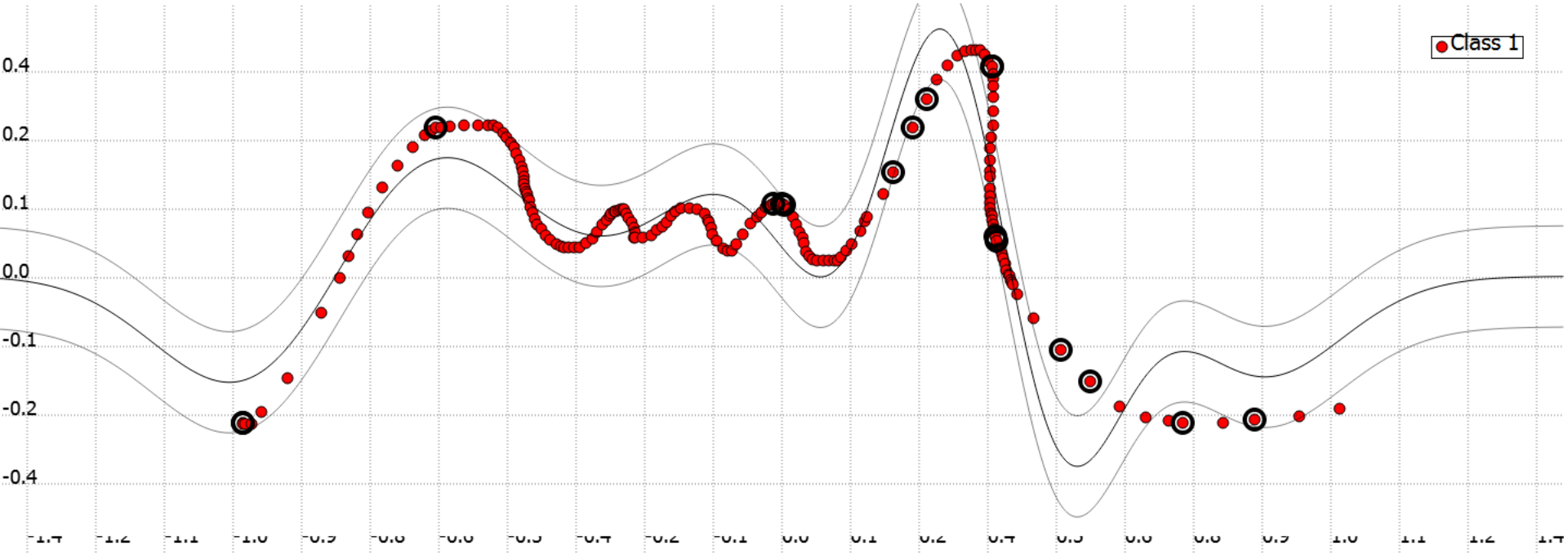
In the regression case, $y \in \mathbb{R}$.

Comparison ε-SVR, ν-SVR, RVR



Solution with **ε-SVR:**
RBF kernel , C=3000, ε=0.08, σ=0.05, 37 support vectors

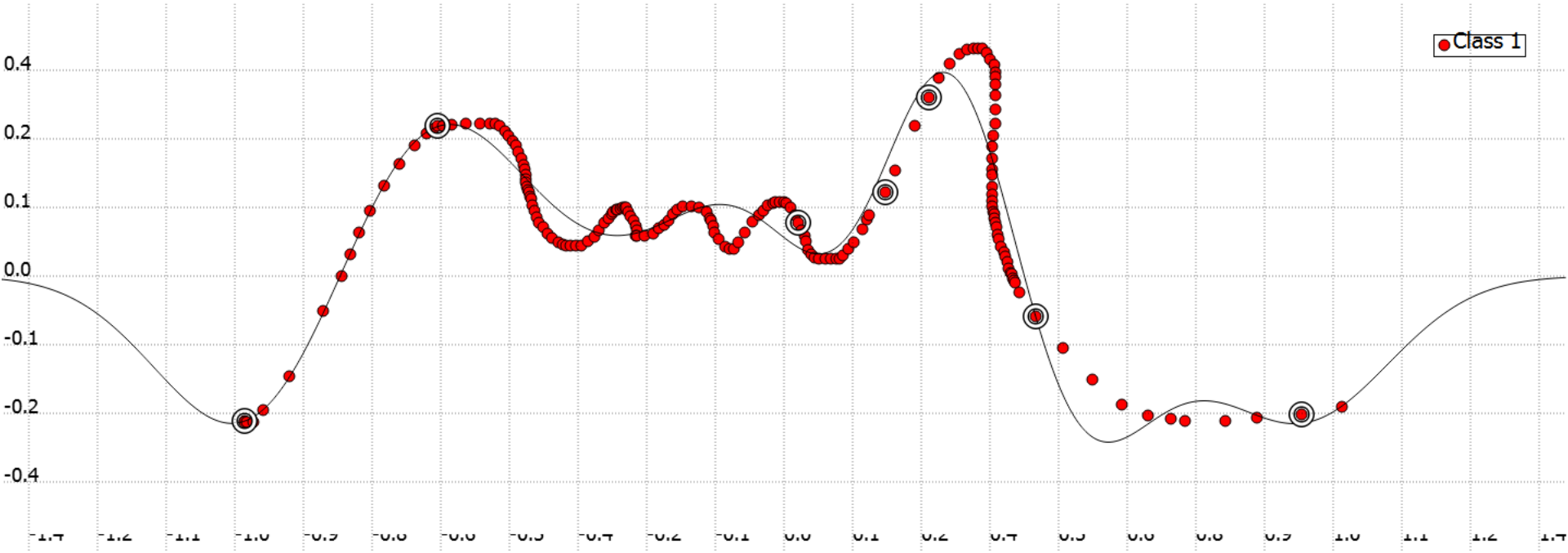
Comparison ε-SVR, ν-SVR, RVR



Solution with **ν-SVR:**

RBF kernel , C=3000, ν=0.04, σ=0.001, **17 support vectors**

Comparison ε-SVR, ν-SVR, RVR

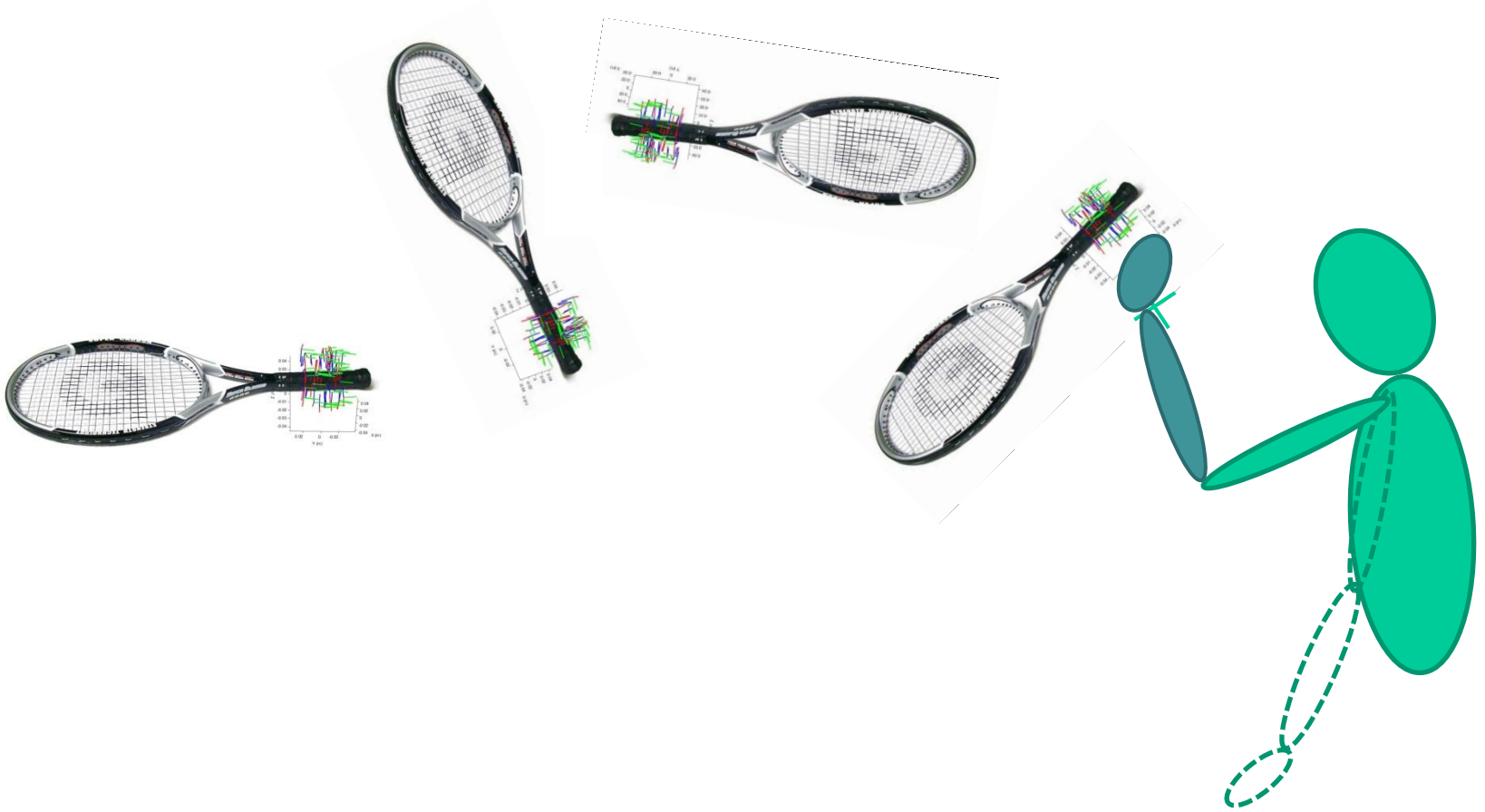


Solution with **RVR:**
RBF kernel , ε=0.08, σ=0.05, **7 support vectors**

SVR:
Examples of Applications

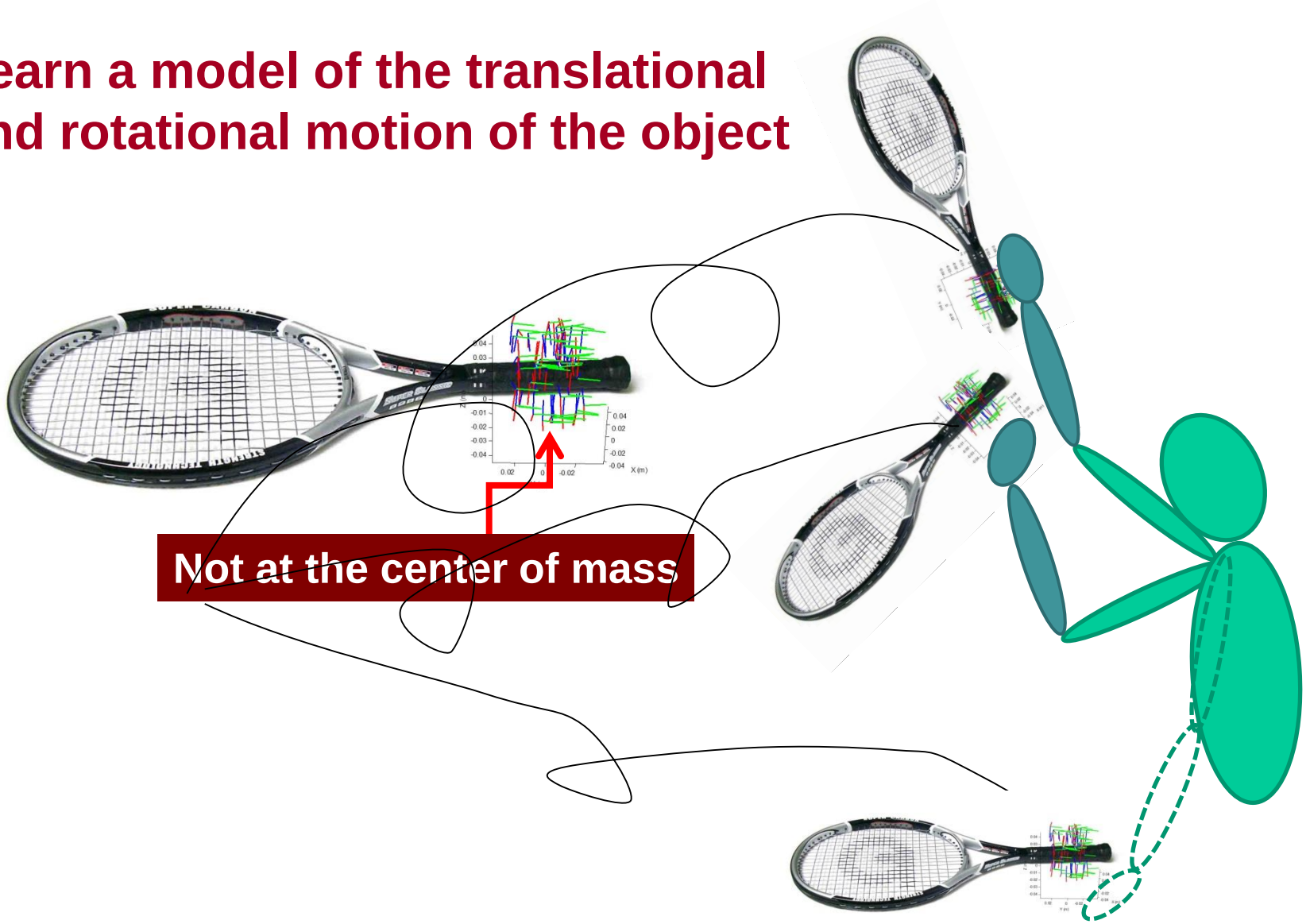
Catching Object in Flight

Extremely fast computation (object flies in half a second); re-estimation of arm motion to adapt to noisy visual detection of object.

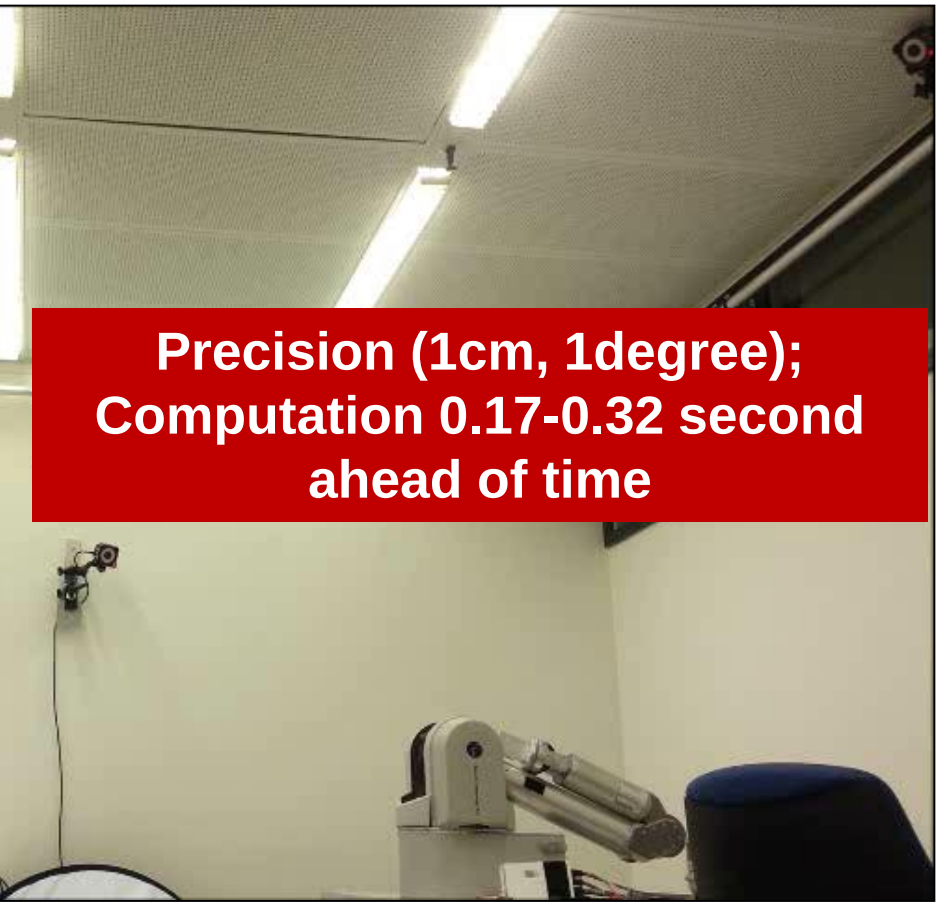


Catching Object in Flight

Learn a model of the translational and rotational motion of the object



Catching Object in Flight



Precision (1cm, 1degree); Computation 0.17-0.32 second ahead of time

Gather Demonstrations of free flying object

Build model of dynamics using Support Vector Regression

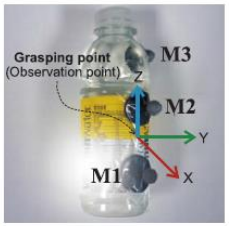
$$\ddot{x} = \sum_{i=1}^{M} \alpha_i k\left(\left[x^i \dot{x}^i\right]^T, \left[x\ \dot{x}\right]^T\right) + b$$

Compute derivative (closed form)

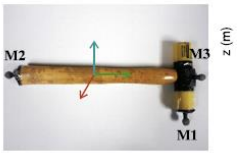
Use model in Extended Kalman Filter for real-time tracking

Kim, S. and Billard, A. (2012) Estimating the non-linear dynamics of free-flying objects. *Robotics and Autonomous Systems*, Volume 60, Issue 9, P. 1108–1122..

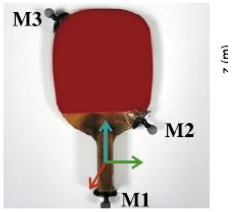
(a) A ball.



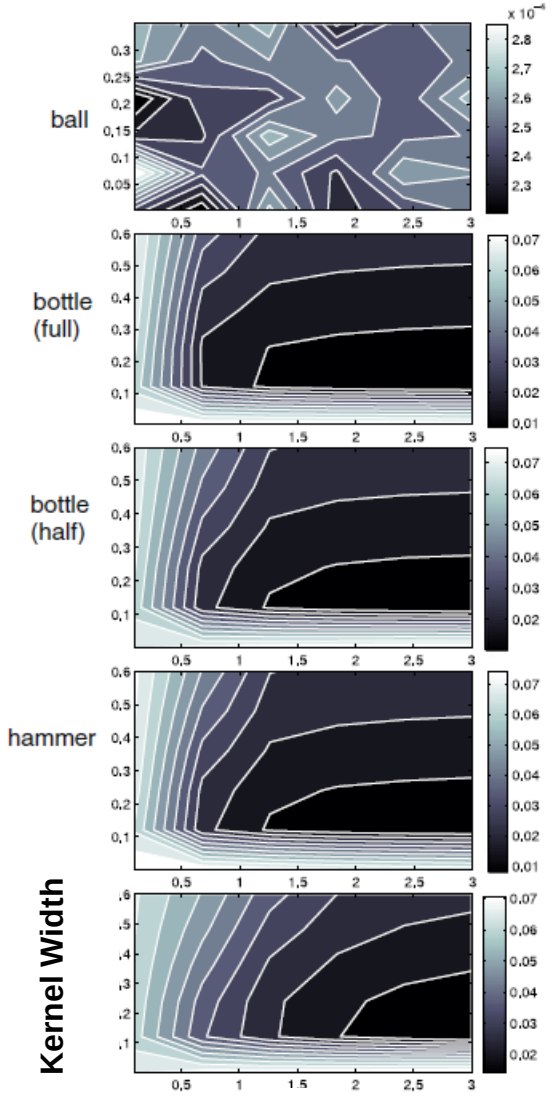
(b) A full-filled bottle.



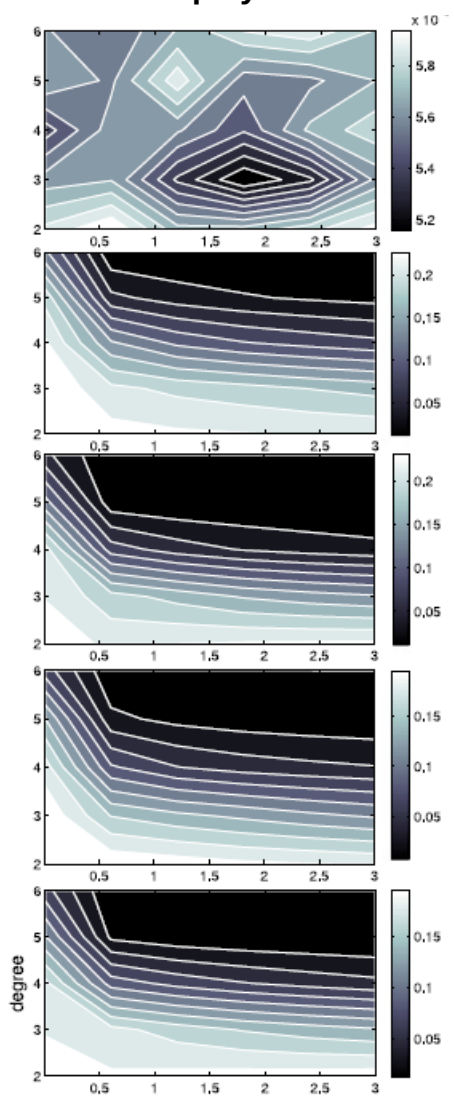
(d) A hammer.



(e) A racket.



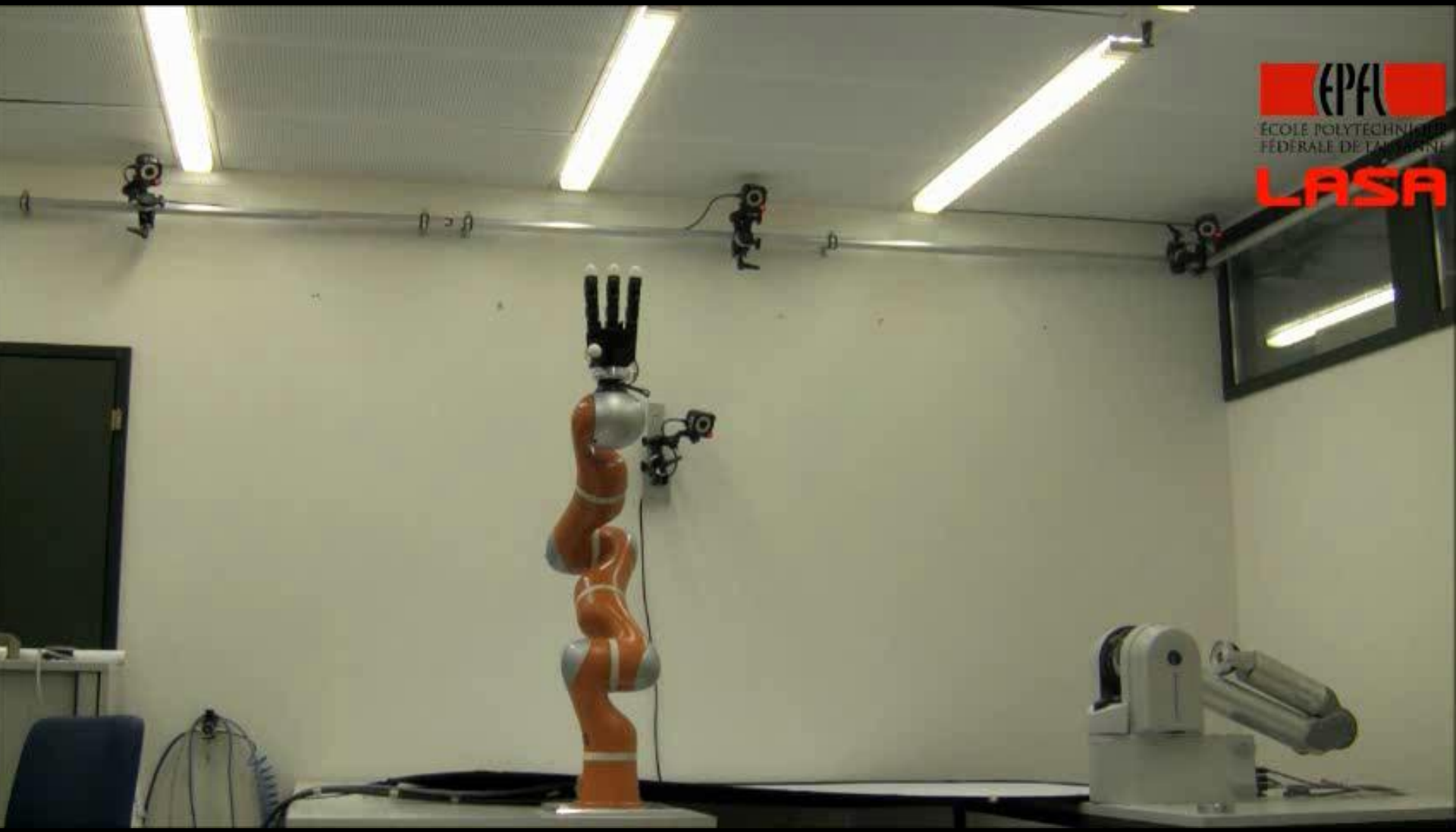
(a) $f_{\text{SVR-RBF}}$.



(b) $f_{\text{SVR-POLY}}$.

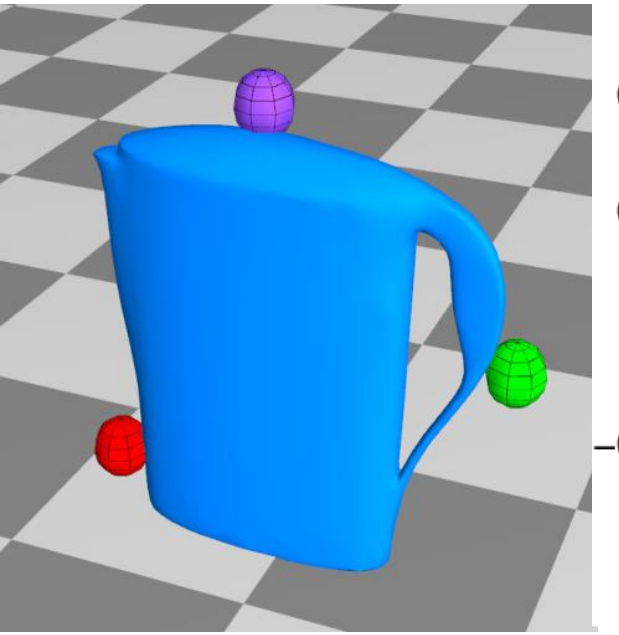
Systematic assessment of sensitivity to choice of hyperparameters and choice of kernel.

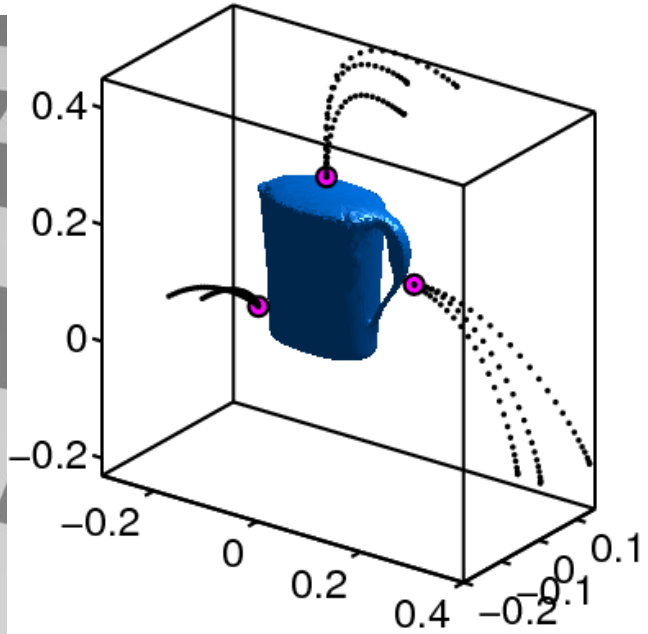
Kim, Shukla, Billard, IEEE Transaction on Robotics, 2015 *IEEE Transactions on Robotics King-Sun Fu Memorial Best Paper Award*

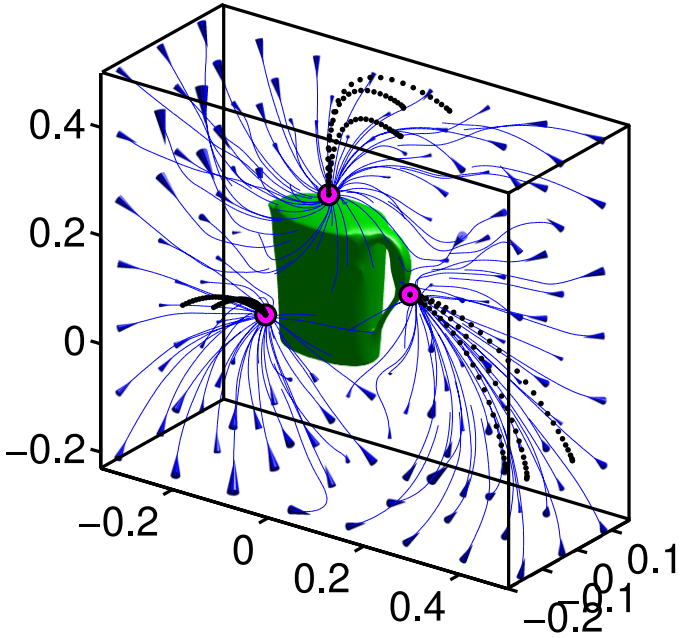


Kim, Shukla, Billard, IEEE Transaction on Robotics, 2015 *IEEE Transactions on Robotics King-Sun Fu Memorial Best Paper Award*

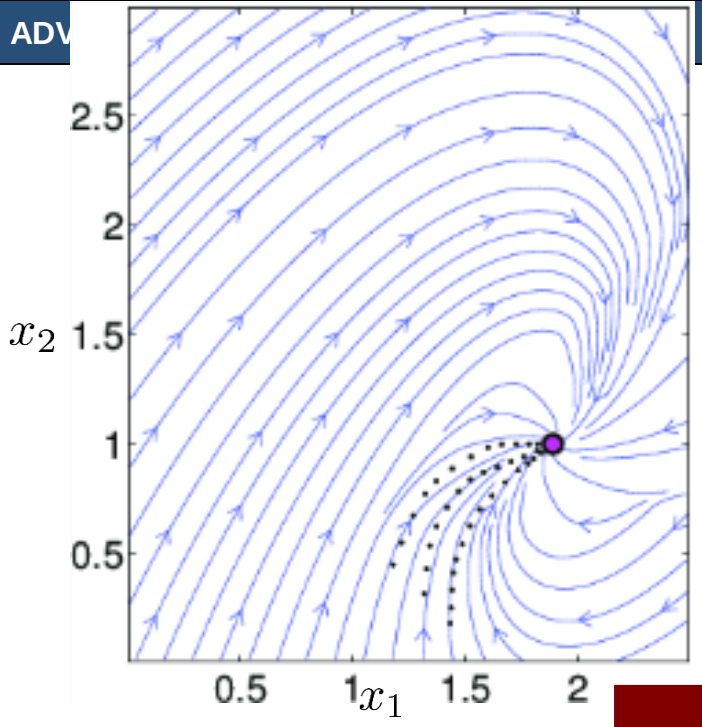
Learning a Multi-Attractor System

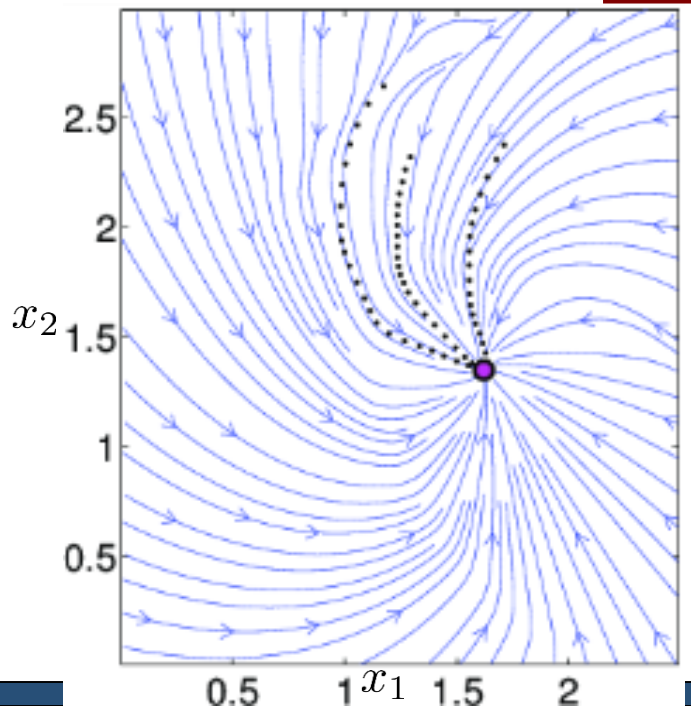


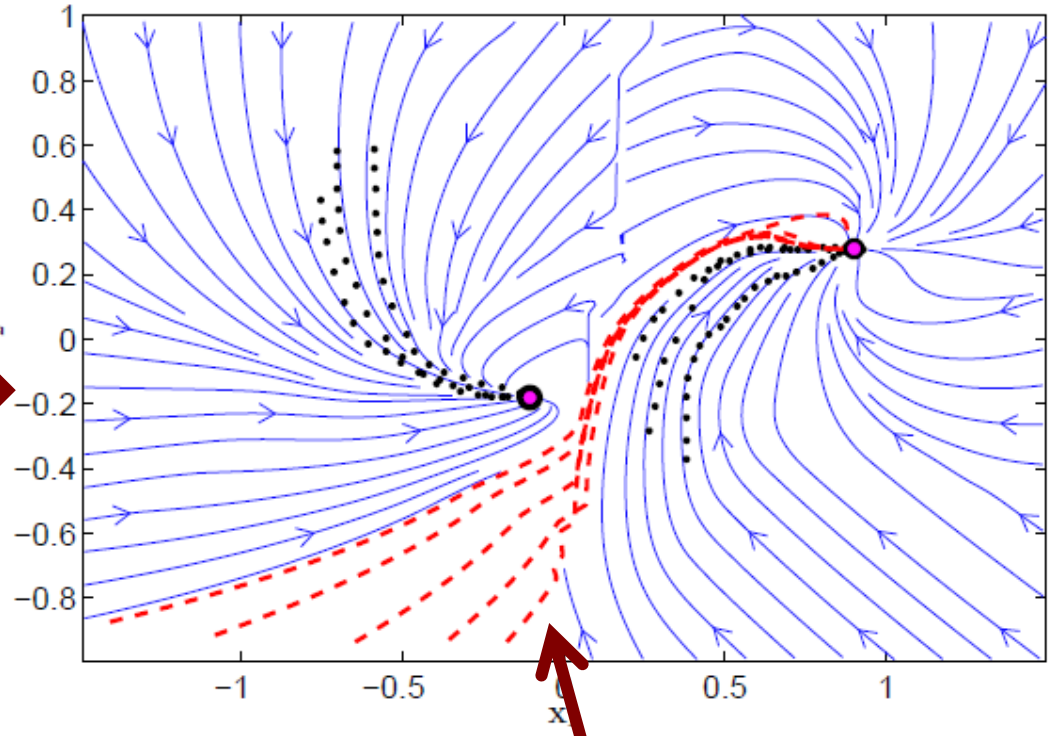




Shukla and Billard, NIPS 2012 - http://asvm.epfl.ch



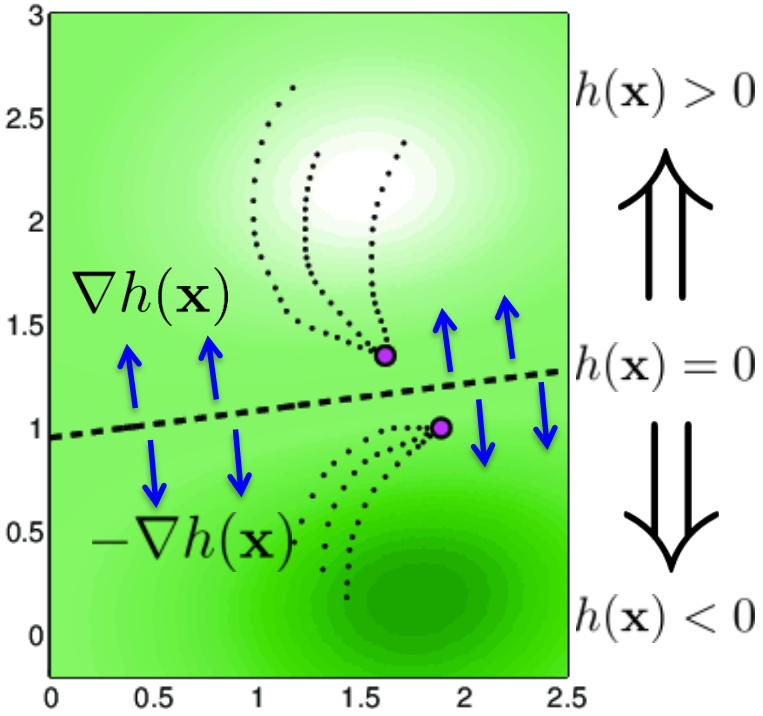




Crossing over

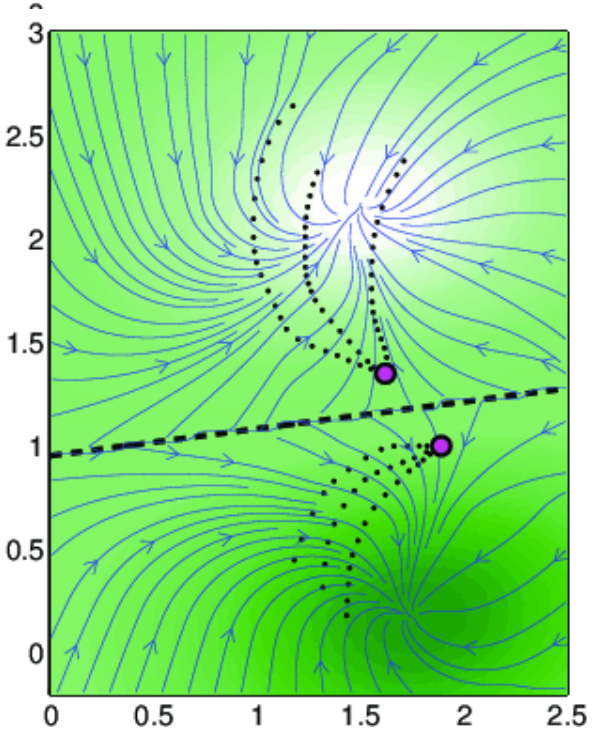
Learning a Multi-Attractor System

Build a partition with support Vector Machine (SVM)



Learning a Multi-Attractor System

Build a partition with support Vector Machine (SVM)



Attractors not located at the right place

$$\nabla h(\mathbf{x}) = \mathbf{0}$$

Learning a Multi-Attractor System

Extend the SVM optimization framework with new constraints

Maximize classification margin

$$\min_{\mathbf{w},\xi_i} \frac{1}{2}\|\mathbf{w}\|^2 + c\sum_{i=1}^{M}\xi_i \quad \textbf{subject to}$$

All points correctly classified

Follow dynamics

Stability at attractor

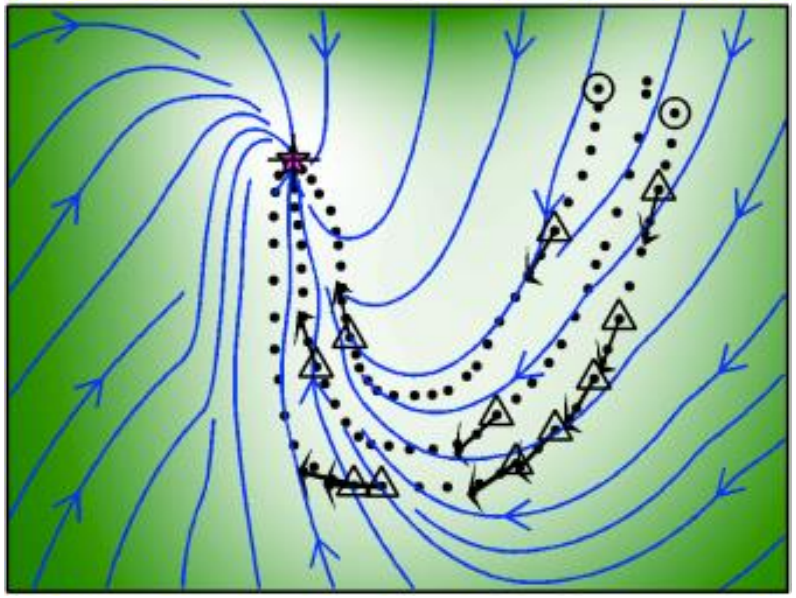
$$\begin{aligned}
y_i\left(\mathbf{w}^T\boldsymbol{\phi}(\mathbf{x}_i)+b\right) &\geq 1 & \forall i = 1\cdots M\\
y_i\mathbf{w}^T\mathrm{J}(\mathbf{x}_i)\hat{\dot{\mathbf{x}}}_i + \xi_i &> 0 & \forall i = 1\cdots M\\
\xi_i &> 0 & \forall i = 1\cdots M\\
\mathbf{w}^T\mathrm{J}(\mathbf{x}^*)\mathbf{e}_i &= 0 & \forall i = 1\cdots N
\end{aligned}$$

$$\{\mathbf{e}_i\} \rightarrow \text{Cannonical basis of } \mathbb{R}^N$$

Shukla and Billard, NIPS 2012 - http://asvm.epfl.ch

Learning a Multi-Attractor System

$$\odot \equiv \alpha - \text{SV}$$
$$\Delta \equiv \beta - \text{SV}$$



$$f(x) = \sum_{i=1}^{M} \alpha_i y_i k(\mathbf{x}, \mathbf{x}_i) + \sum_{i=1}^{M} \beta_i \hat{\mathbf{x}}_i^{\mathrm{T}} \frac{\partial k(\mathbf{x}, \mathbf{x}_i)}{\partial \mathbf{x}_i} - \sum_{i=1}^{N} \gamma_i \mathbf{e}_i^{\mathrm{T}} \frac{\partial k(\mathbf{x}, \mathbf{x}^*)}{\partial \mathbf{x}^*} + b$$

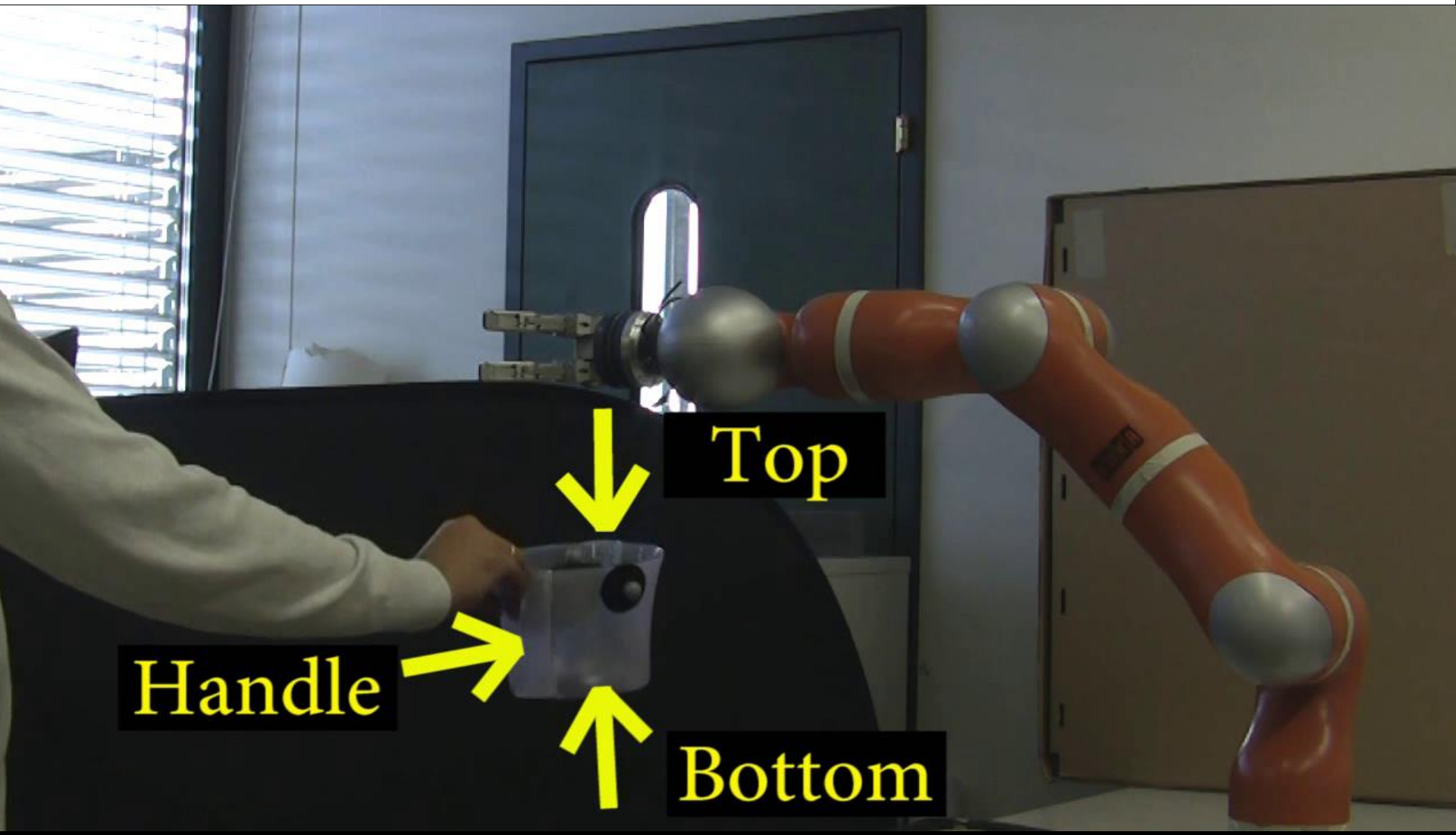
Standard SVM α- SVs

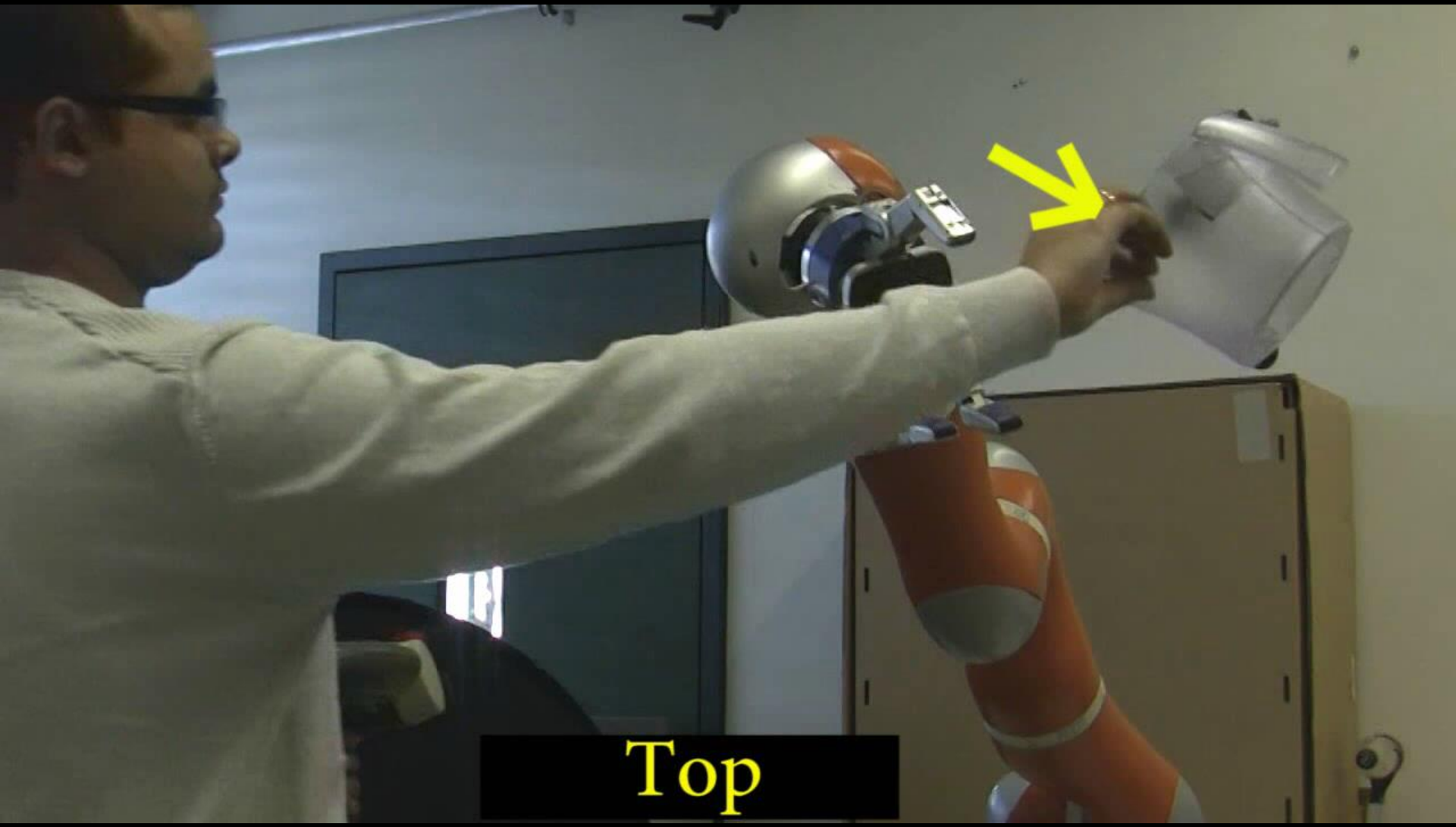
New β- SVs

Non-linear bias

Const. bias

Several possible grasping points




Top

Multiple Attractor Systems

The robot switches between the two attractors *on-the-fly*


LASA
ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE