Support Vector Regression

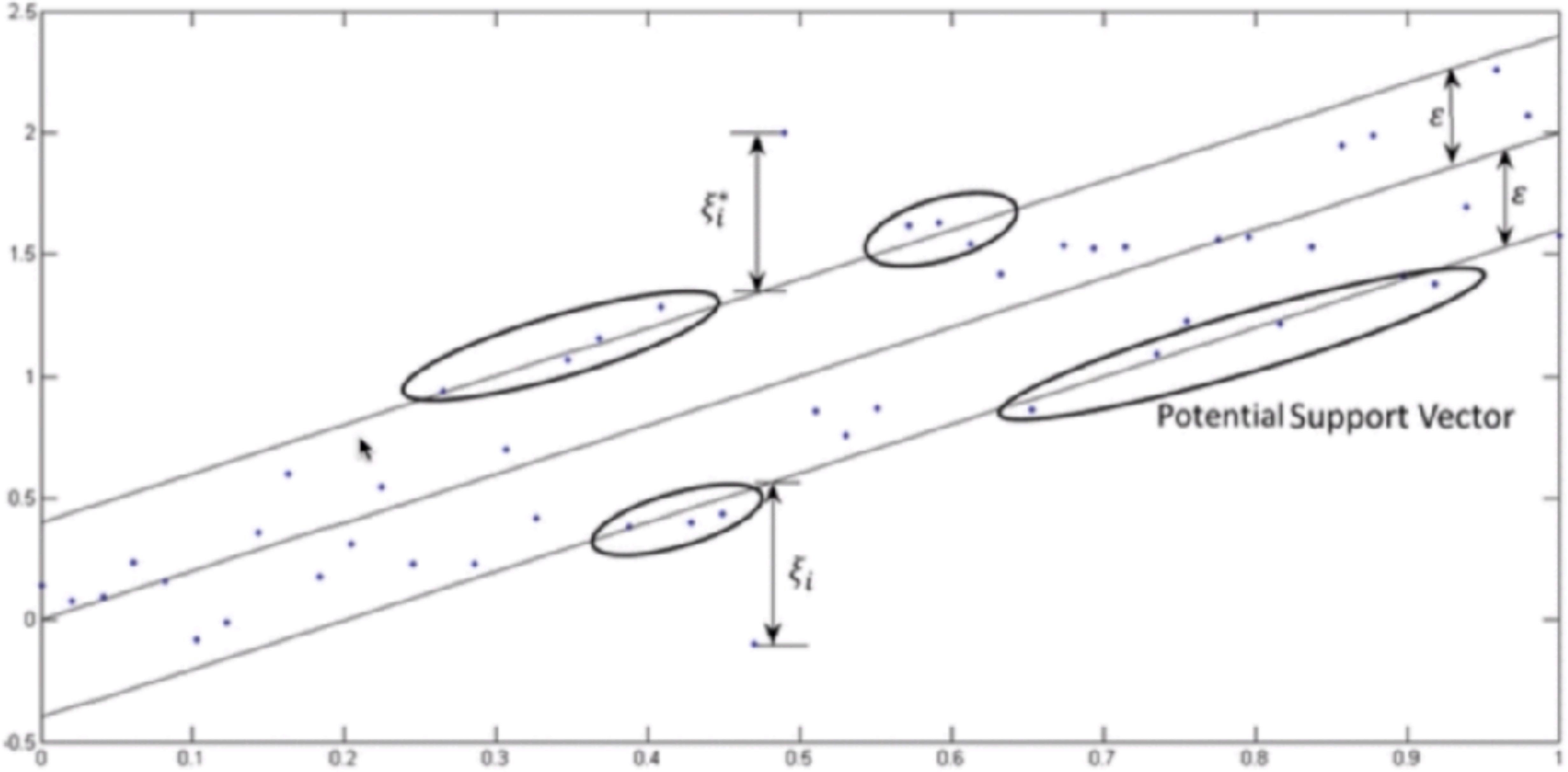
- Support Vector Machines support linear and nonlinear regression that we can refer to as SVR
- Instead of trying to fit the largest possible street between two classes while limiting margin violations, SVR tries to fit as many instances as possible on the street while limiting margin violations.
- The width of the street is controlled by a hyper parameter Epsilon.

- SVR performs linear regression in a higher (dimensional space).

- We can think of SVR as if each data point in the training represents it’s own dimension. When you evaluate your kernel between a test point and a point in the training set the resulting value gives you the coordinate of your test point in that dimension.

- The vector we get when we evaluate the test point for all points in the training set, $\vec{k}$ is the representation of the test point in the higher dimensional space.

- Once you have that vector you the use it to perform a linear regression.



2.5
2
1.5
1
0.5
0
-0.5
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
ξ_i^*
ξ_i
ε
ε
Potential Support Vector

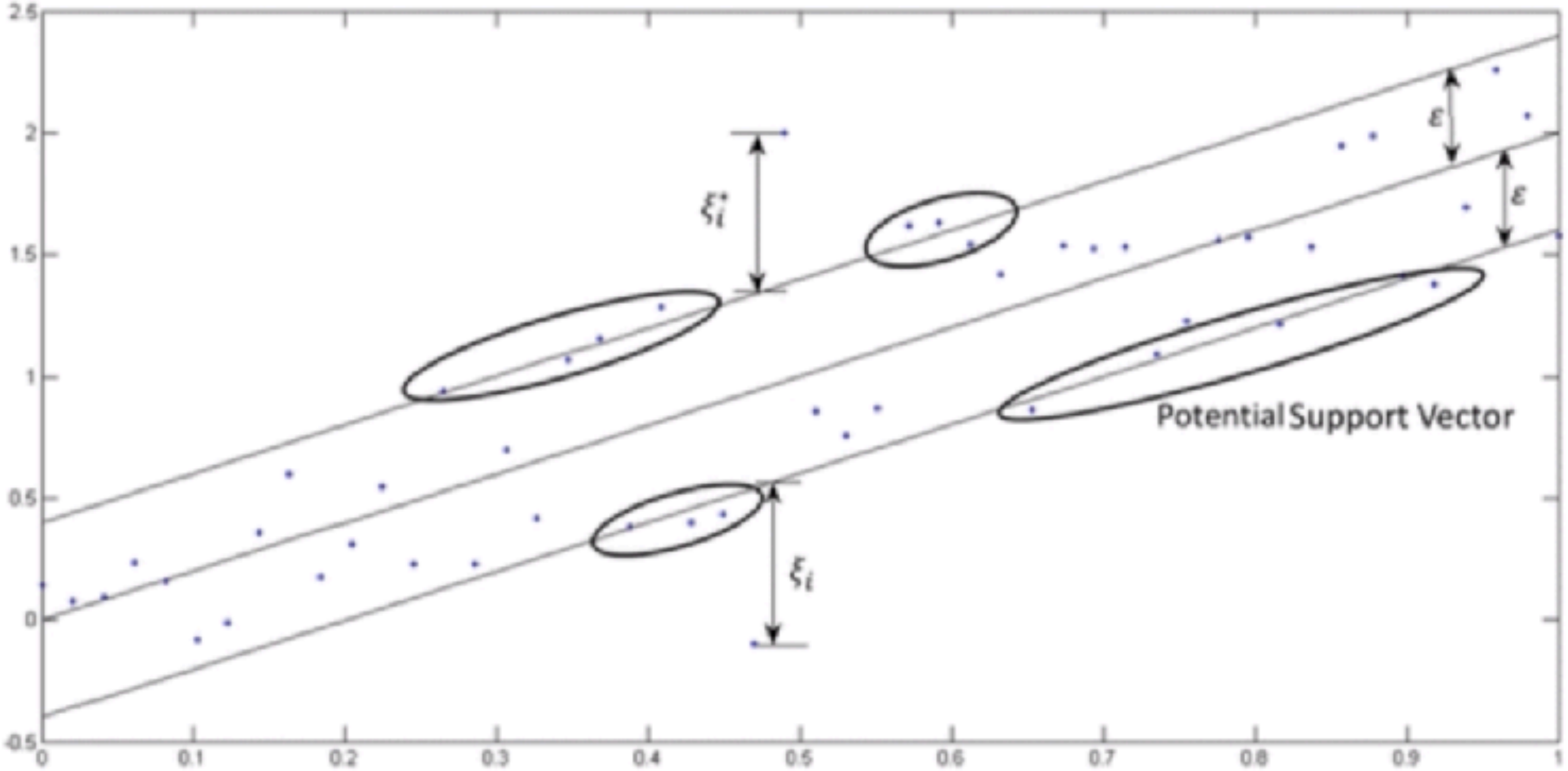
It requires a training set: $\mathcal{T} = \{\vec{X}, \vec{Y}\}$ which covers the domain of interest and is accompanied by solutions on that domain.

The work of the SVM is to approximate the function we used to generate the training set

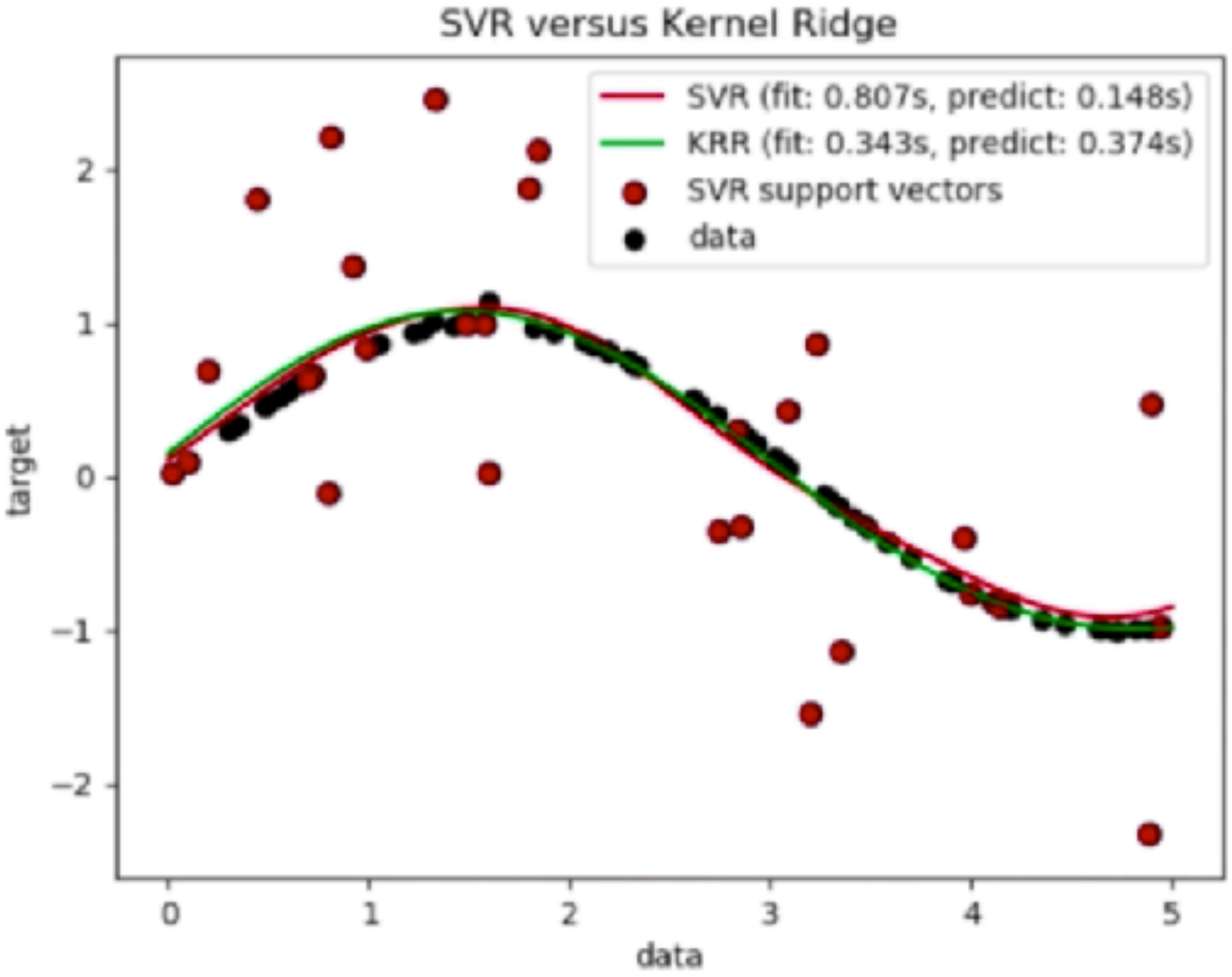
$$F(\vec{X}) = \vec{Y}$$

In a classification problem, the vectors $\vec{X}$ are used to define
a hyperplane that separates the two different classes in your solution.

These vectors are used to perform linear regression. The vectors closest to the test point are referred to as support vectors. We can evaluate our function anywhere so any vectors could be closest to our test evaluation location.


2.5
2
1.5
1
0.5
0
-0.5
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
ξ_i^*
ξ_i
ε
ε
Potential Support Vector

How does it work?



Building a SVR

1. Collect a training set $\tau = \{\vec{X}, \vec{Y}\}$

2. Choose a kernel and it’s parameters as well as any regularization needed.

3. Form the correlation matrix, $\vec{K}$

4. Train your machine, exactly or approximately, to get contraction coefficients $\vec{\alpha} = \{\alpha_i\}$

5. Use those coefficients, create your estimator $f(\vec{X}, \vec{\alpha}, x^\star) = y^\star$

Next step is to choose a kernel

- Gaussian

Regularization

- Noise

The main part of the algorithm $\bar{K}\vec{\alpha} = \vec{y}$

$\vec{y}$ is the vector of values corresponding to your training set,

$\bar{K}$ is your correlation matrix

$\vec{\alpha}$ is a set of unknowns we need to solve for.

$$\vec{\alpha} = \bar{K}^{-1}\vec{y}$$

- Once $\vec{a}$ parameters known - form the estimator
- we use the coefficients we found during the optimization step and the kernel we started off with.

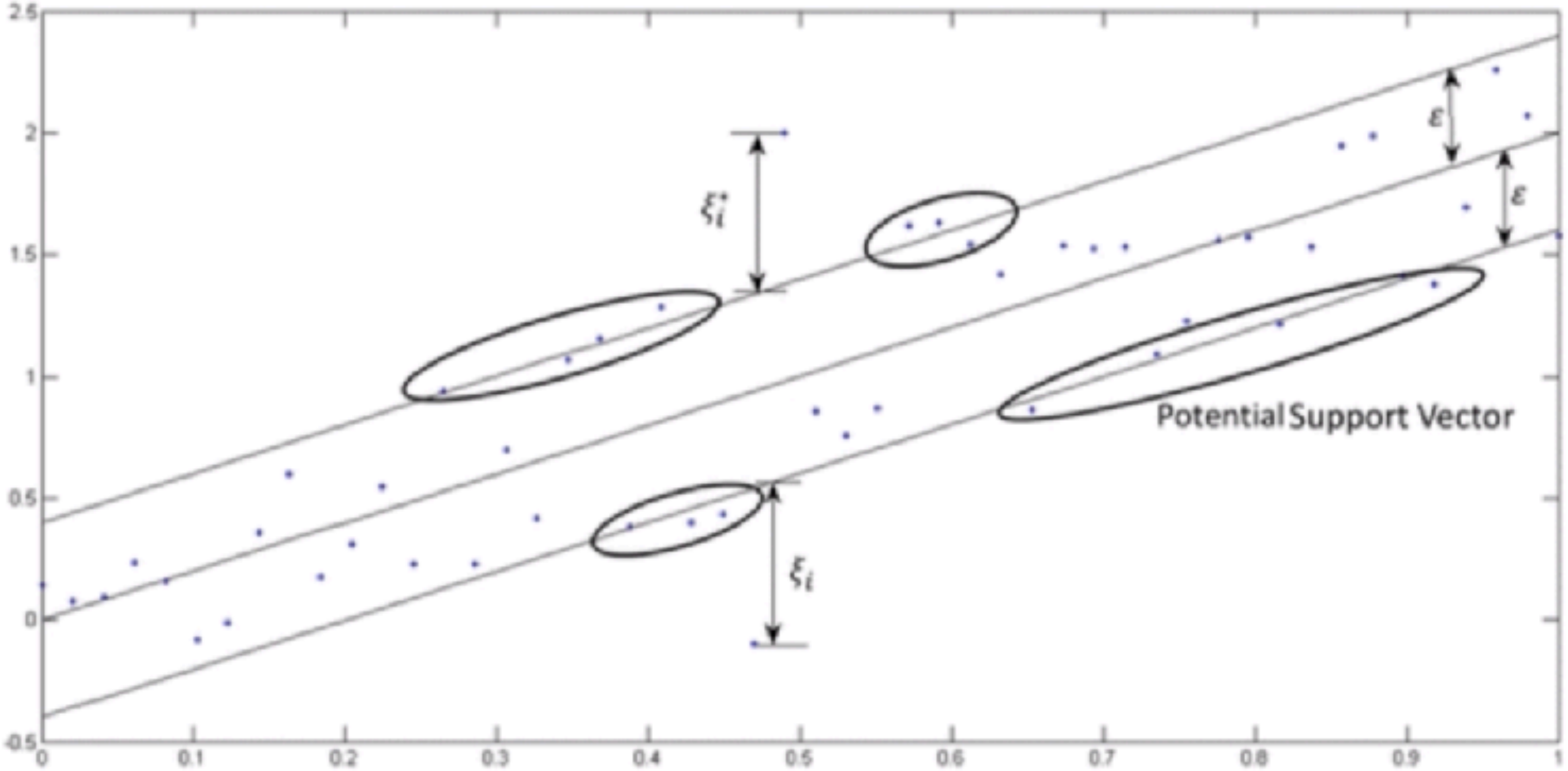
- To estimate the value y^* for a test point, $\vec{x^*}$ - compute the correlation vector $\vec{k}$,

- $y^* = \vec{a} \cdot \vec{k}$

$$k_i = \exp\left(\sum_k \theta_k |x_k^i - x_k^\star|^2 \right)$$

Let's try to simplify everything:

SVR has a different regression goal compared to linear regression. In linear regression we are trying to minimize between the prediction and data. In SVR our goal is to make sure that errors do not exceed the threshold.



2.5
2
1.5
1
0.5
0
-0.5
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1
ξ_i^*
ξ_i
ε
ε
Potential Support Vector