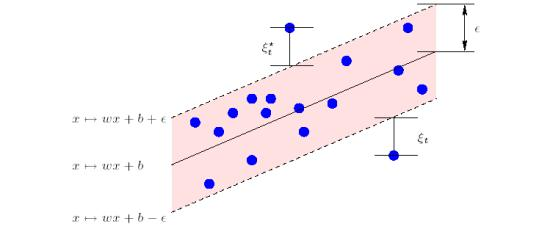
Traditional ϵ-SVR works with the epsilon-insensitive hinge loss(hinge loss penalizes predictions y < 1, corresponding to the notion of a margin in a support vector machine). The value of ϵ defines a margin of tolerance where no penalty is given to errors.

Remember the support vectors are the instances across the margin, i.e. the samples being penalized, which slack variables are non-zero.

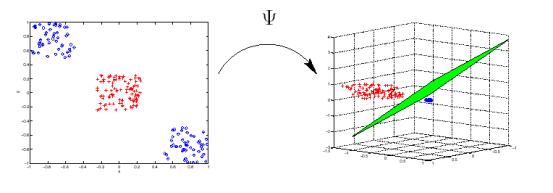
The larger ϵ is, the larger errors you admit in your solution. By contrast, if ϵ→0+, every error is penalized: you end with many (tending to the total number of instances) support vectors to sustain that.

  
Figure 1: Detailed picture of epsilon band with slack variables and selected data points

In SVM regression, the input  is first mapped onto a *m*-dimensional feature space using some fixed (nonlinear) mapping, and then a linear model is constructed in this feature space. Using mathematical notation, the linear model (in the feature space)  is given by



where,  denotes a set of nonlinear transformations, and *b* is the “bias” term. Often the data are assumed to be zero mean (this can be achieved by pre-processing), so the bias term is dropped.

  
Figure 2: Non-linear mapping of input examples into high dimensional feature space. (Classification case, however the same stands for regression as well).

MathWorks - <https://www.mathworks.com/help/stats/understanding-support-vector-machine-regression.html>

Support vector machine (SVM) analysis is a popular machine learning tool for classification and regression, first identified by Vladimir Vapnik and his colleagues in 1992[[5]](https://www.mathworks.com/help/stats/understanding-support-vector-machine-regression.html" \l "buytaw5). SVM regression is considered a nonparametric technique because it relies on kernel functions.

Statistics and Machine Learning Toolbox™ implements linear epsilon-insensitive SVM (ε-SVM) regression, which is also known as L1 loss. In ε-SVM regression, the set of training data includes predictor variables and observed response values. The goal is to find a function f(x) that deviates from yn by a value no greater than ε for each training point x, and at the same time is as flat as possible.

#### **Linear SVM Regression: Primal Formula**

Suppose we have a set of training data where xn is a multivariate set of N observations with observed response values yn.

To find the linear function

*f*(*x*)=*x*′*β*+*b*,

and ensure that it is as flat as possible, find f(x) with the minimal norm value (β′β). This is formulated as a convex optimization problem to minimize

*J*(*β*)=12*β*′*β*

subject to all residuals having a value less than ε; or, in equation form:

∀*n*:|*yn*−(*xn*′*β*+*b*)|≤*ε* .

It is possible that no such function f(x) exists to satisfy these constraints for all points. To deal with otherwise infeasible constraints, introduce slack variables ξn and ξ\*n for each point. This approach is similar to the “soft margin” concept in SVM classification, because the slack variables allow regression errors to exist up to the value of ξn and ξ\*n, yet still satisfy the required conditions.

Including slack variables leads to the objective function, also known as the primal formula[[5]](https://www.mathworks.com/help/stats/understanding-support-vector-machine-regression.html" \l "buytaw5):

*J*(*β*)=12*β*′*β*+*CN**n*=1(*ξn*+*ξ*∗*n*) ,

subject to:

∀*n*:*yn*−(*xn*′*β*+*b*)≤*ε*+*ξn*

∀*n*:(*xn*′*β*+*b*)−*yn*≤*ε*+*ξ*∗*n*

∀*n*:*ξ*∗*n*≥0

∀*n*:*ξn*≥0 .

The constant C is the box constraint, a positive numeric value that controls the penalty imposed on observations that lie outside the epsilon margin (ε) and helps to prevent overfitting (regularization). This value determines the trade-off between the flatness of f(x) and the amount up to which deviations larger than ε are tolerated.

The linear ε-insensitive loss function ignores errors that are within ε distance of the observed value by treating them as equal to zero. The loss is measured based on the distance between observed value y and the ε boundary. This is formally described by

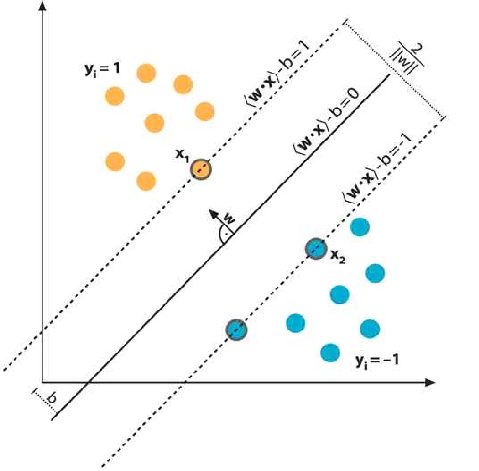
*Lε* = {0if |*y*−*f*(*x*)| ≤ *ε*

|*y−f(x)|−ε* otherwise

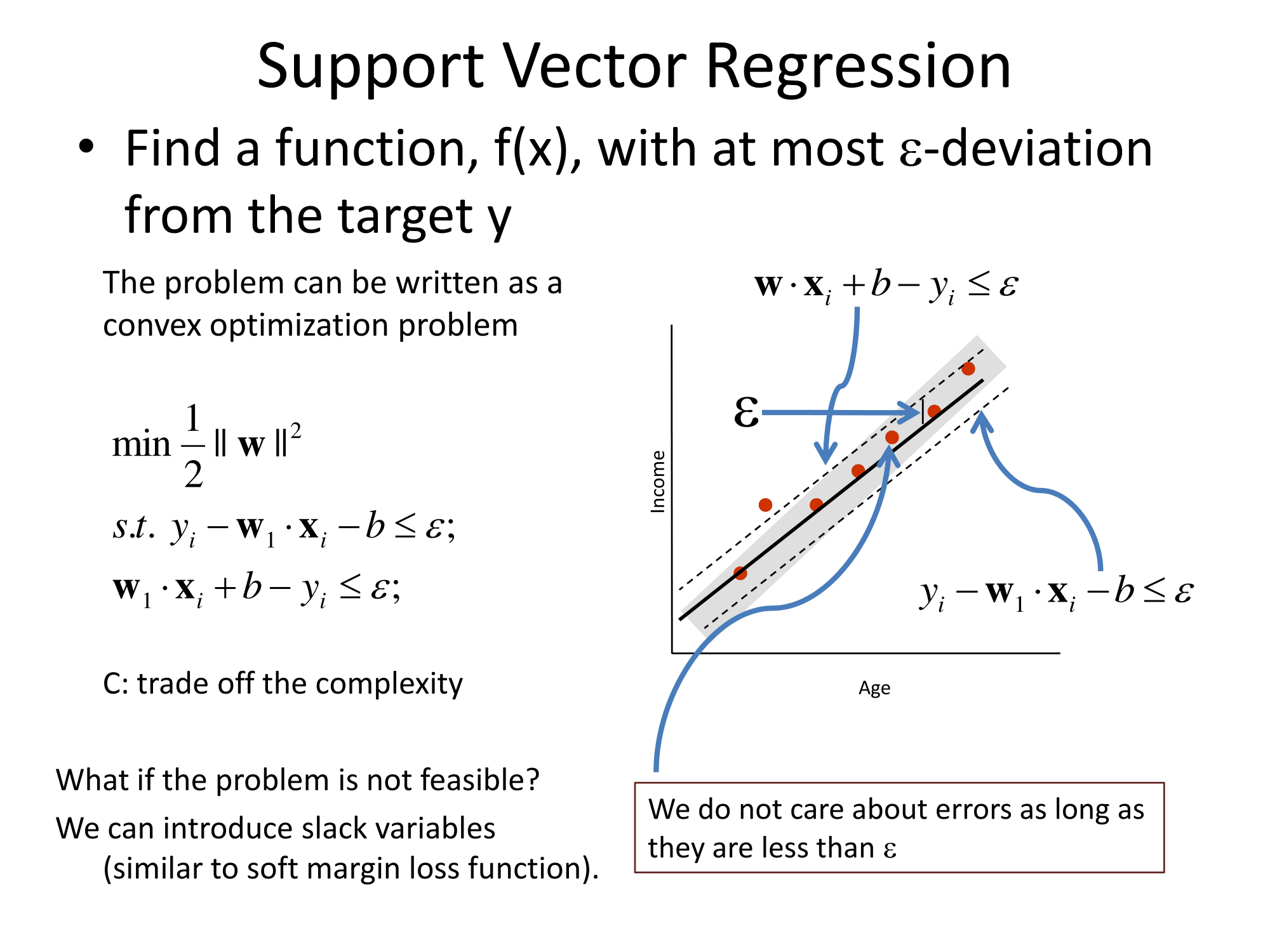
Easy Explanation <https://medium.com/coinmonks/support-vector-regression-or-svr-8eb3acf6d0ff>

UDEMY :

* SVM – Tries to fit the **largest possible Street between two classes** while limiting margin violation



* SVR – Tries to fit **as many instances as possible** while limiting margin violation



* The width of the street is controlled by a hyper parameter **Epsilon *ε***
* SVR performs linear regression in a higher dimensional space
* When you evaluate your kernel between a test point and a point from 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 , k is the representation of the test point in the higher dimensional space.
* Once you have the vector you use it to perform a linear regression.

