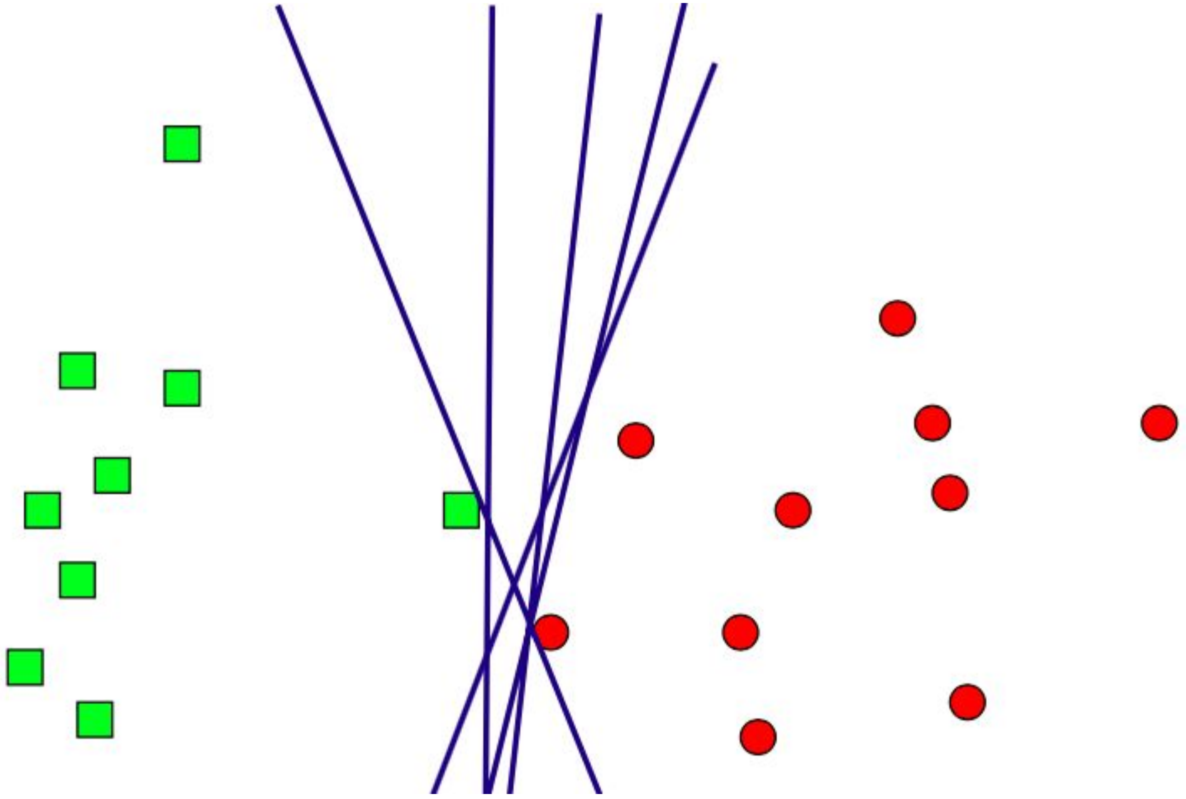
SUPPORT VECTOR MACHINES

Many slides courtesy of Marios Savvides

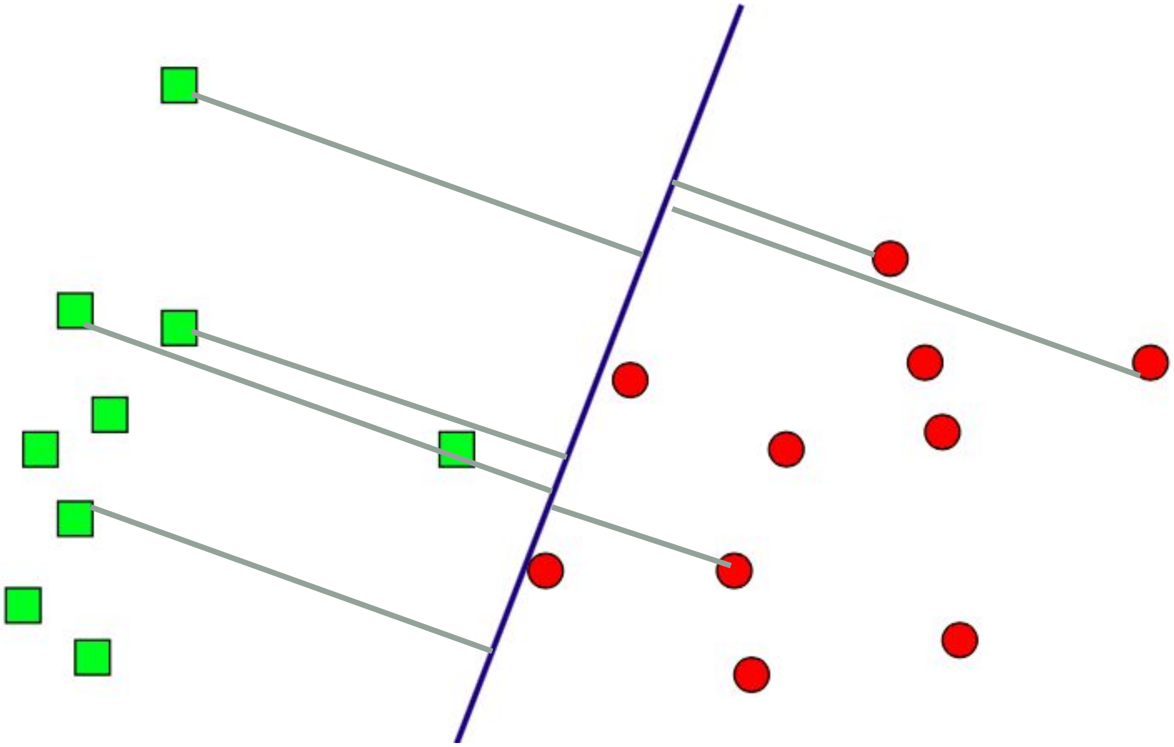
Linear classification problem

- Find a line that separates two classes
- Many solutions exist!
- Which one is the best?

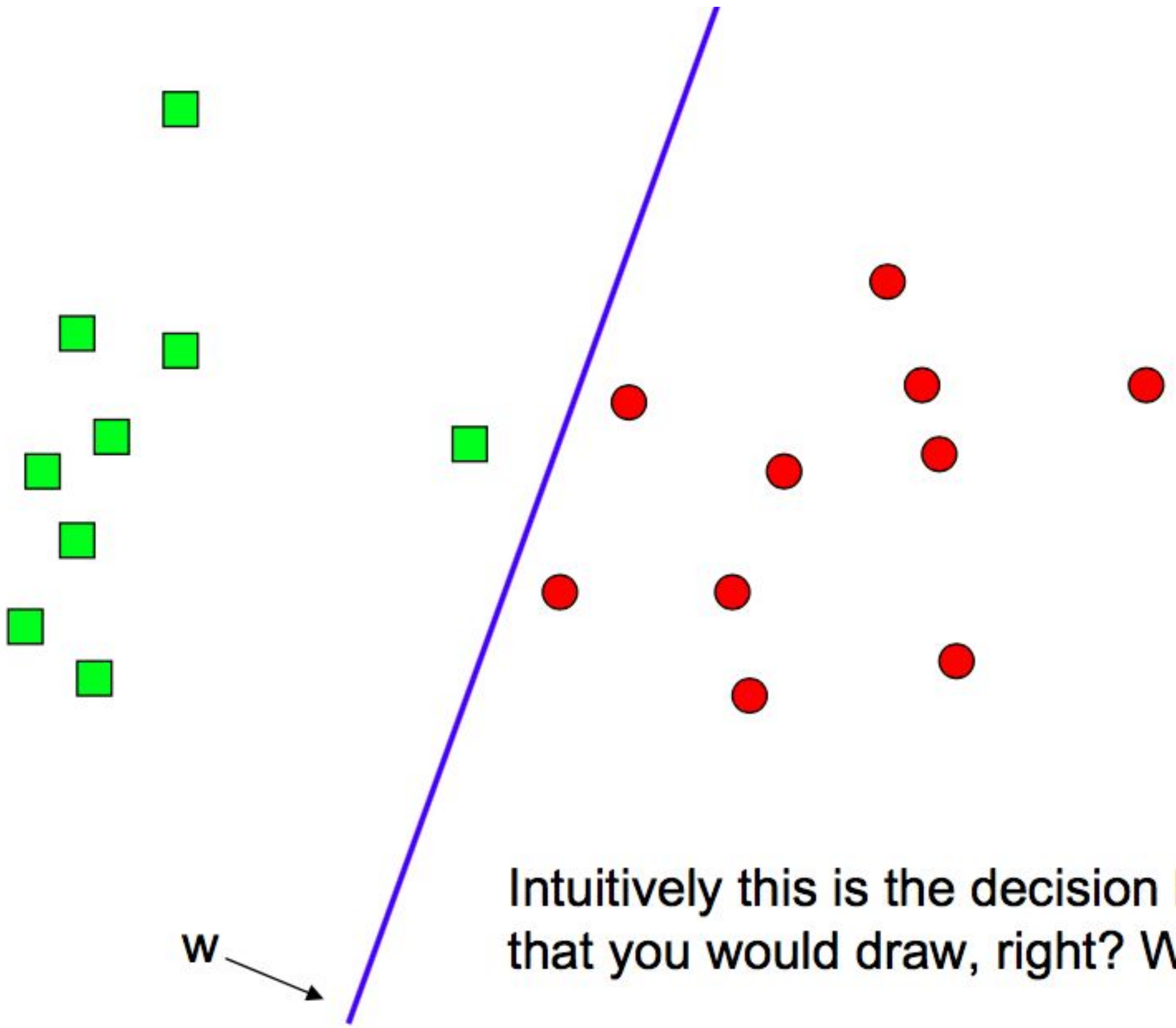


Logistic Regression

- Minimizes sum of L2 distance (square error) between all points to the line
- Also have probabilistic interpretation (assume noise is Gaussian)



Support vectors

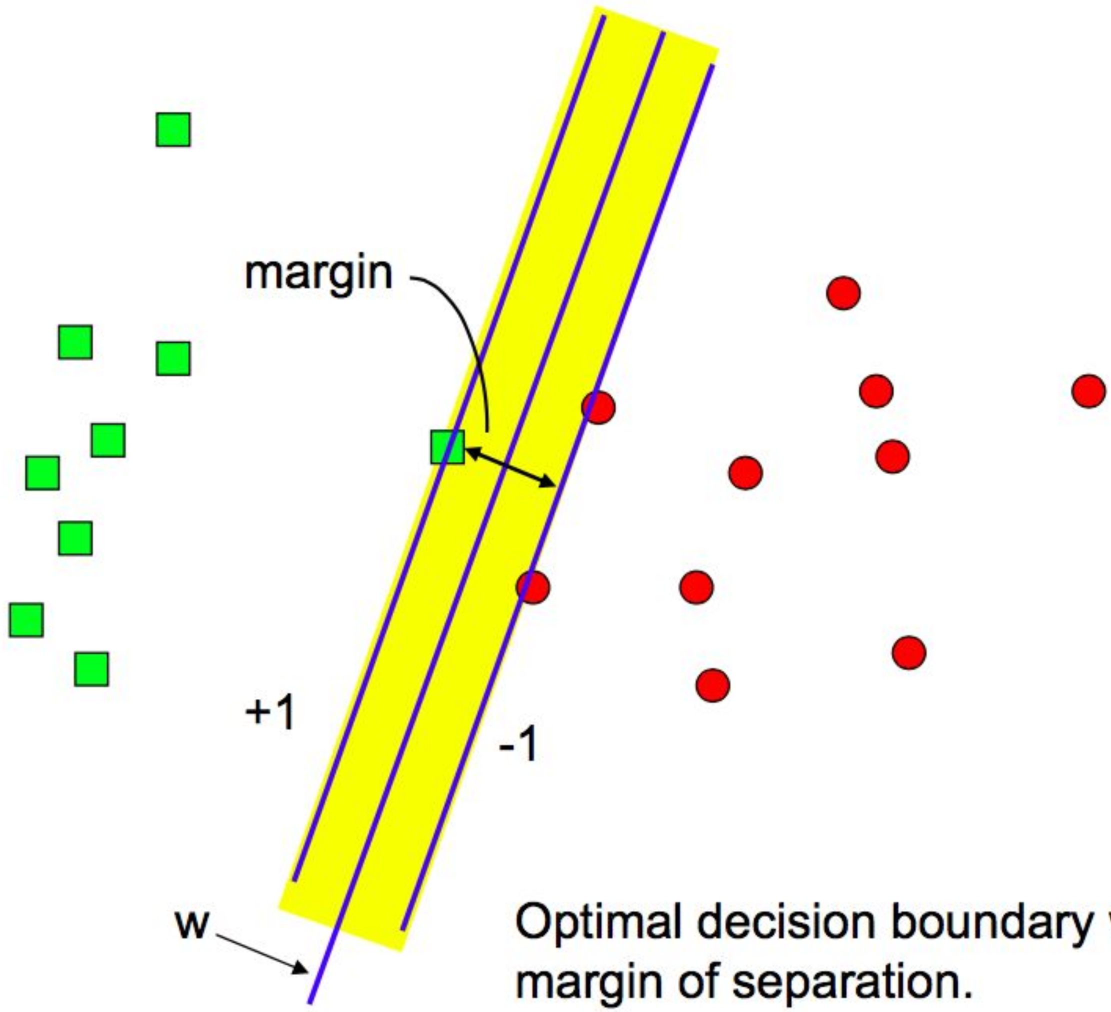


Intuitively this is the decision boundary that you would draw, right? Why?

Support Vector Machines (SVM)

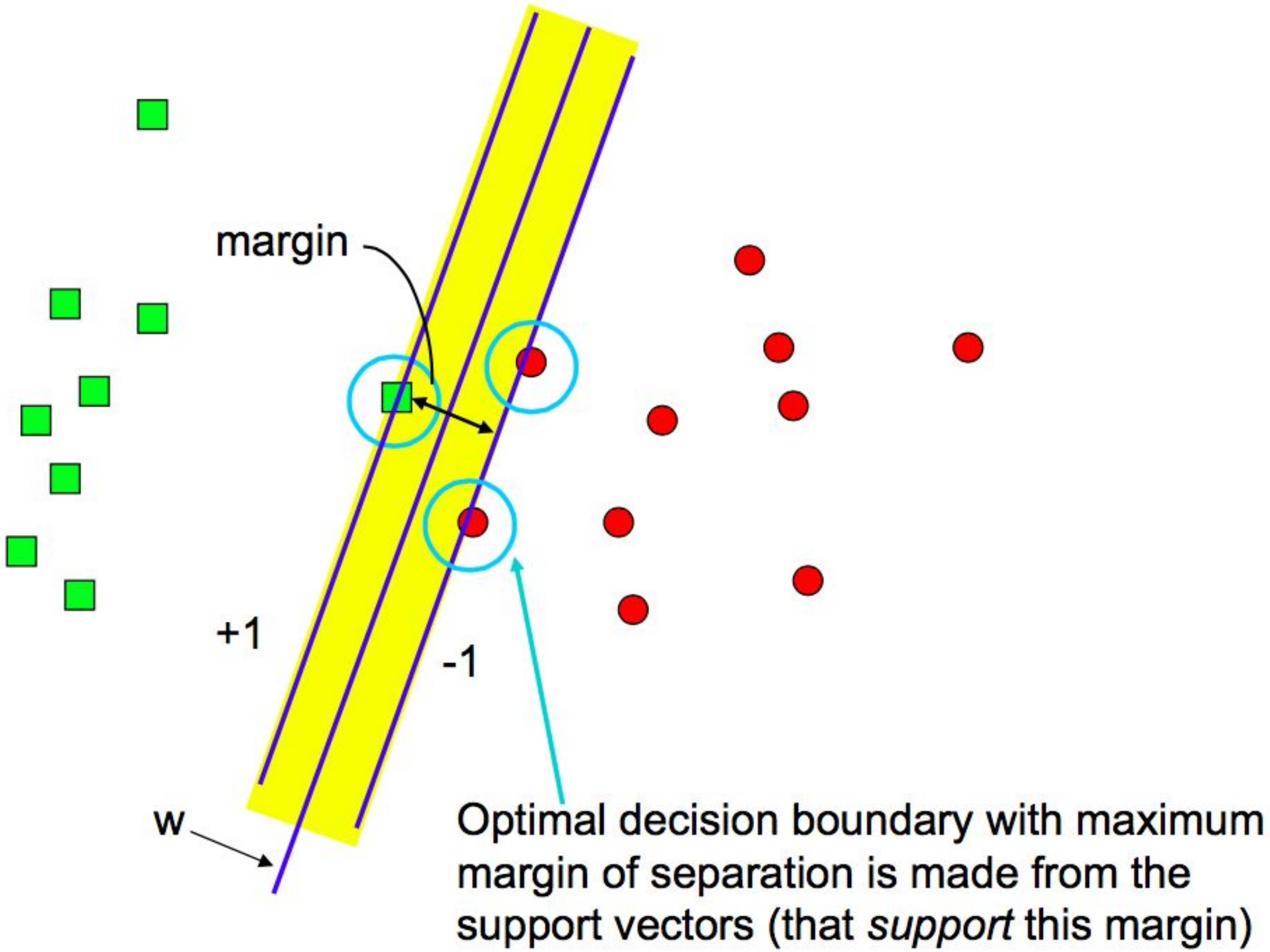
- Goal: improve generalization!
  - Care more about reducing classifier variance than reducing classifier bias
- How?
- Find the decision boundary that gives the most “slack” in classification
  - Don’t care about easy cases, care about borderline cases!
    - Focus on the margin
  - Maximize the “margin of error” between two classes

Support Vectors

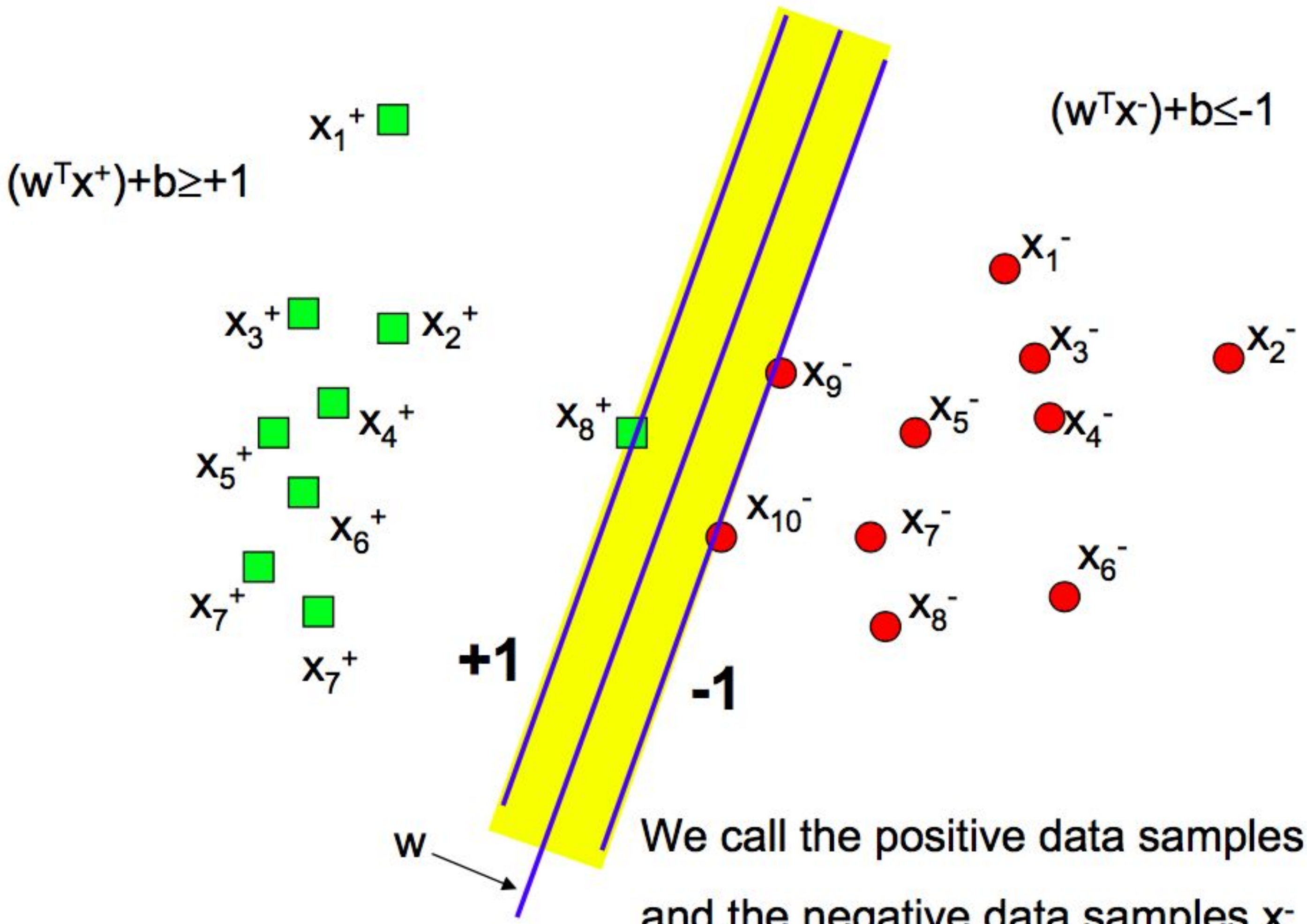


Optimal decision boundary with maximum margin of separation.

Support Vectors



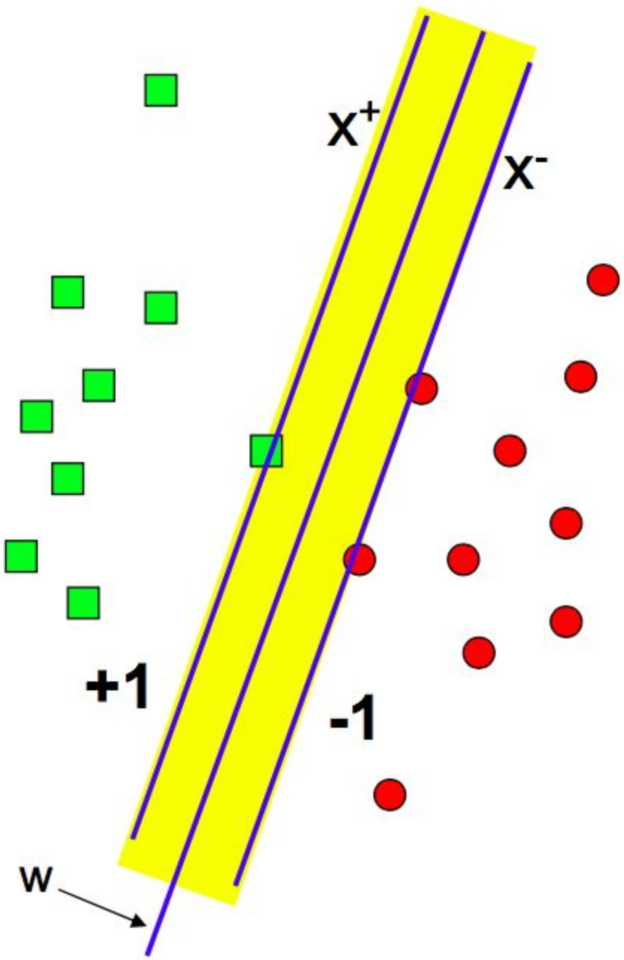
Support Vectors



We call the positive data samples x^+ i.e. $(w^Tx^+)+b\geq+1$

and the negative data samples x^- i.e. $(w^Tx^-)+b\leq-1$

Support Vectors



Let x^+ denote a positive point with functional margin of 1 and x^- denote a negative point respectively.

This implies:

$w^T x^+ + b = +1$

$w^T x^- + b = -1$

The functional margin of the resulting classifier m is

$$m = \left(\left\langle \frac{\mathbf{w}}{\|\mathbf{w}\|}, \mathbf{x}^+ \right\rangle - \left\langle \frac{\mathbf{w}}{\|\mathbf{w}\|}, \mathbf{x}^- \right\rangle \right)$$

$$= \frac{1}{\|\mathbf{w}\|} \left(\langle \mathbf{w}, \mathbf{x}^+ \rangle - \langle \mathbf{w}, \mathbf{x}^- \rangle \right)$$

$$= \frac{2}{\|\mathbf{w}\|}$$

< > denotes dot product

SVM objective function

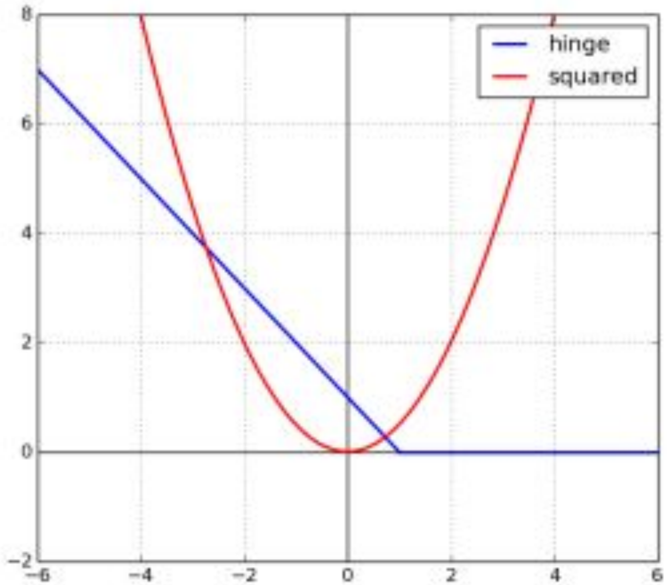
- Minimize $\mathbf{w}^T\mathbf{w}$
- Subject to
  - $y_i(<\mathbf{w}, \mathbf{x}_i> + b) = y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1$

- $y_i = \{+1,-1\}$ depending on the binary class
  - Positive class must fall on the positive side of the boundary
  - Negative class must fall on the negative size

- Convex optimization (No local minimas)
- Can be solved by Quadratic Programing (QP)

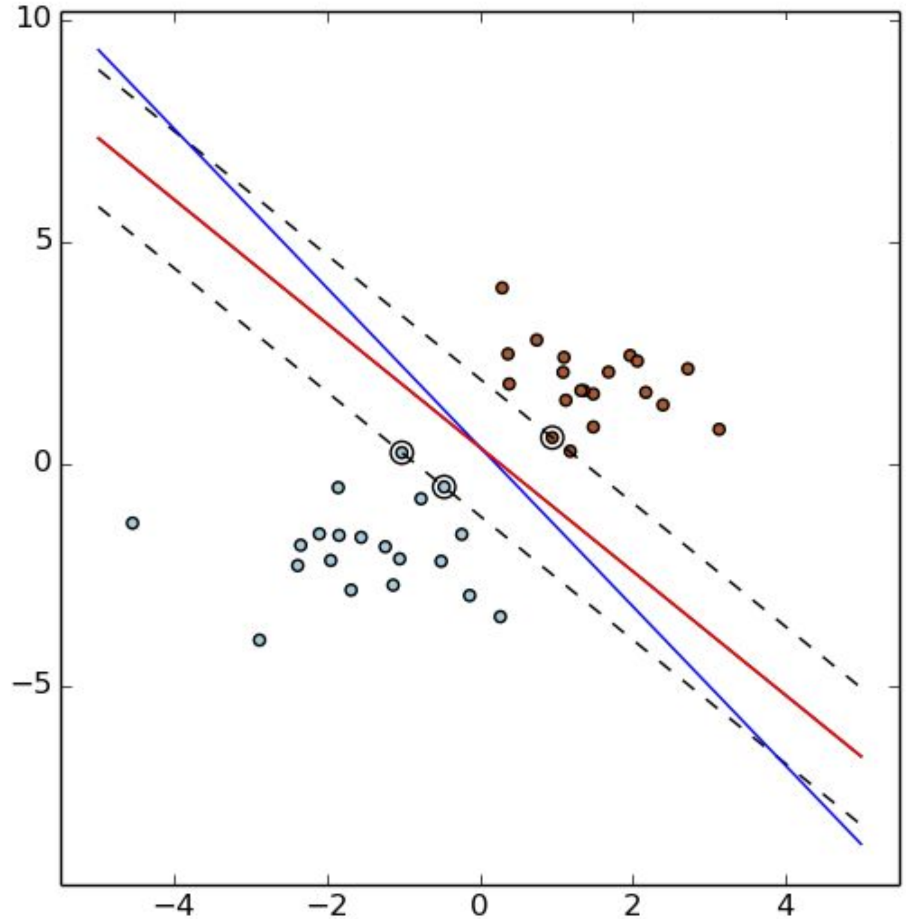
Notes on the Losses

- Linear regression optimizes for the L2 loss (squared loss)
  - Squared distance of data points to boundary $(x- h(x))^2$
- SVM optimize for the hinge loss
  - $y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1$
  - Or $0 \geq 1-y_i(\mathbf{w}^T\mathbf{x}_i + b)$
  - We don't want this inequality to be broken so our effective loss is
    - $\max(0, 1-y_i(\mathbf{w}^T\mathbf{x}_i + b))$



https://cvstuff.wordpress.com/2014/11/29/latex-l_1-versus-latex-l_2-loss-a-svm-example/

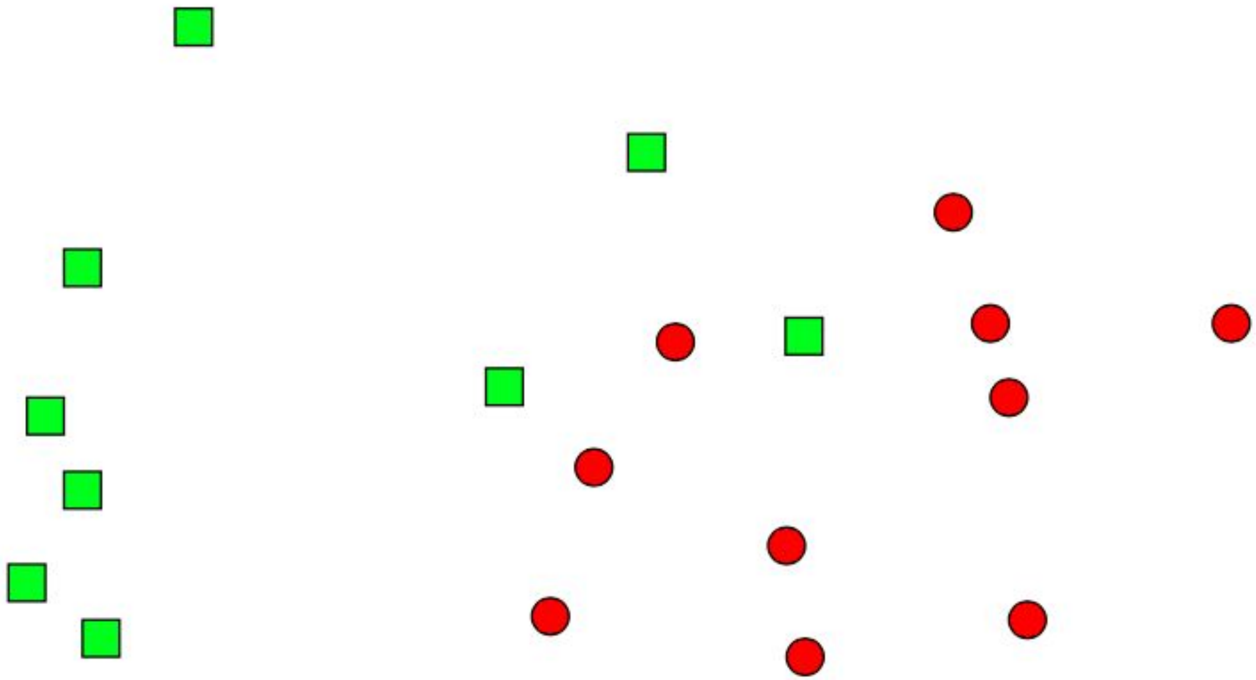
L2 vs hinge loss



https://cvstuff.wordpress.com/2014/11/29/latex-l_1-versus-latex-l_2-loss-a-svm-example/

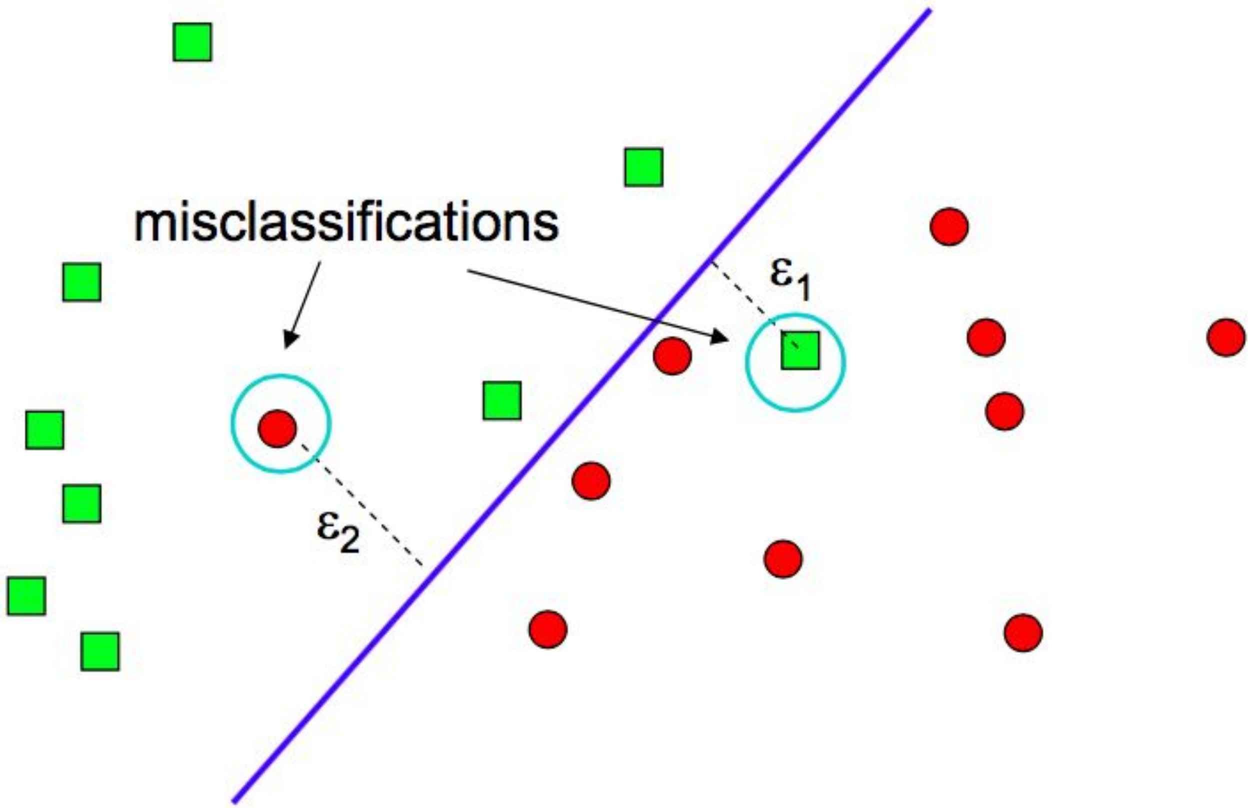
Linearly non-separable

- What happens when you cannot separate the two classes with a linear boundary



Introducing an error term ε

- Aim for a hyperplane that tries to maximize the margin while minimize total error $\Sigma\varepsilon_i$



Slack variables

- We call these error terms “Slack variables”
- Give SVM some slack so that the SVM can do its job.

SVM objective function

- Minimize $\mathbf{w}^T\mathbf{w}$
- Subject to
  - $y_i(<\mathbf{w}, \mathbf{x}_i> + b) = y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1$

- $y_i = \{+1,-1\}$ depending on the binary class
  - Positive class must fall on the positive side of the boundary
  - Negative class must fall on the negative size

- Convex optimization (No local minimas)
- Can be solved by Quadratic Programing (QP)

SVM objective with slack

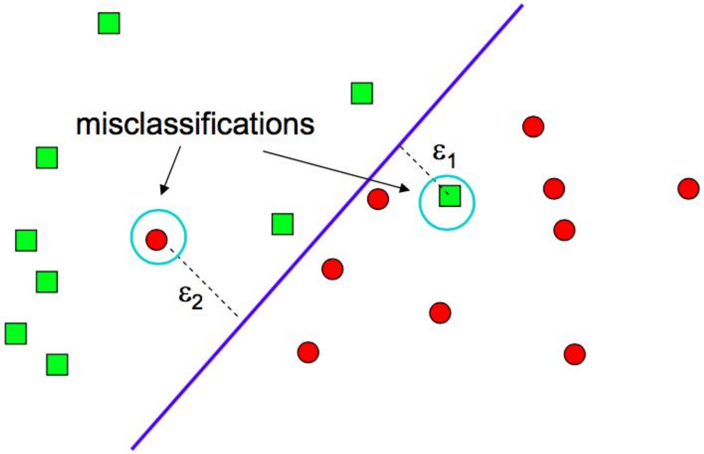
- Minimize $\mathbf{w}^T\mathbf{w} + \Sigma\varepsilon_i / C$
- Subject to

C is a weight parameter, how much we care about slack

$$\mathbf{w}^{\mathbf{T}}\mathbf{x}_i + b \geq 1 - \varepsilon_i \qquad \textit{for} \quad +ve \qquad \textit{class}$$

$$\mathbf{w}^{\mathbf{T}}\mathbf{x}_i + b \leq -1 + \varepsilon_i \qquad \textit{for} \quad -ve \qquad \textit{class}$$

$$\varepsilon_i > 0 \qquad \forall i$$



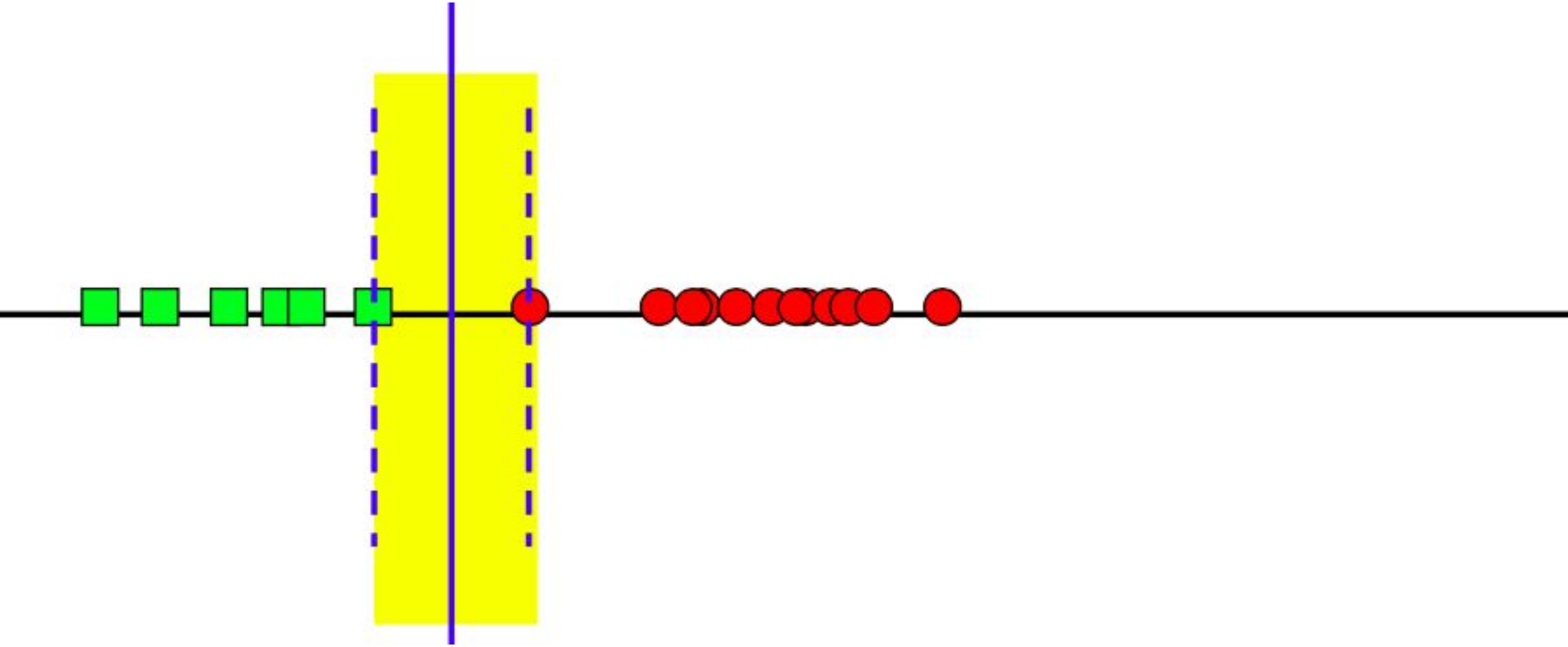
Notes about slacks

- Even if the problem has linear separability we might want some slack still.
  - Missed label points near the boundaries, noise in the data set etc.
  - In this case, we trade-off classifier bias for classifier variance.

  - A form of regularization!

Linear SVMs

- Easy



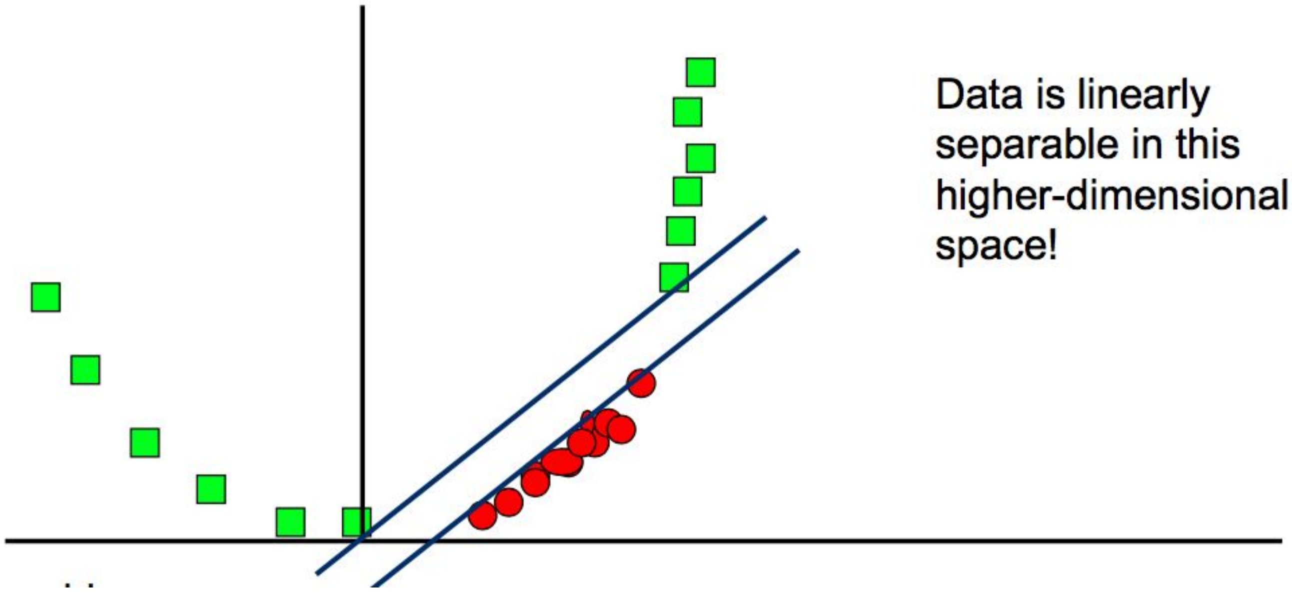
Example SVMs

- ??????

Adding features (non-linear transformation)

- Remember we add non-linear features to linear regression to do non-linear fitting
- Consider as a non-linear transformation to higher dimensional space

$F(x) \rightarrow (x, x^2)$



Data is linearly separable in this higher-dimensional space!

Mapping functions

$$\phi : X \to F$$

- A mapping function that maps to higher dimensional space

Kernel function

- We define the inner product in the mapped space as a kernel function K(x,y)

$$K(\mathbf{x},\mathbf{y}) = \langle \Phi(\mathbf{x}), \Phi(\mathbf{y}) \rangle$$

- Kernel of x -> (x, x^2)
  - $xy + x^2y^2$
- Kernel of x -> (x, x^2, x^3)
  - $xy + x^2y^2 + x^3y$

SVM and kernels

Instead of mapping input to high dimensions, we can use kernel to save computation in SVMs.

Example kernels:

Linear

Polynomial (degree 2,3,4…)

Radial basis functions (RBF)

Radial Basis Kernels

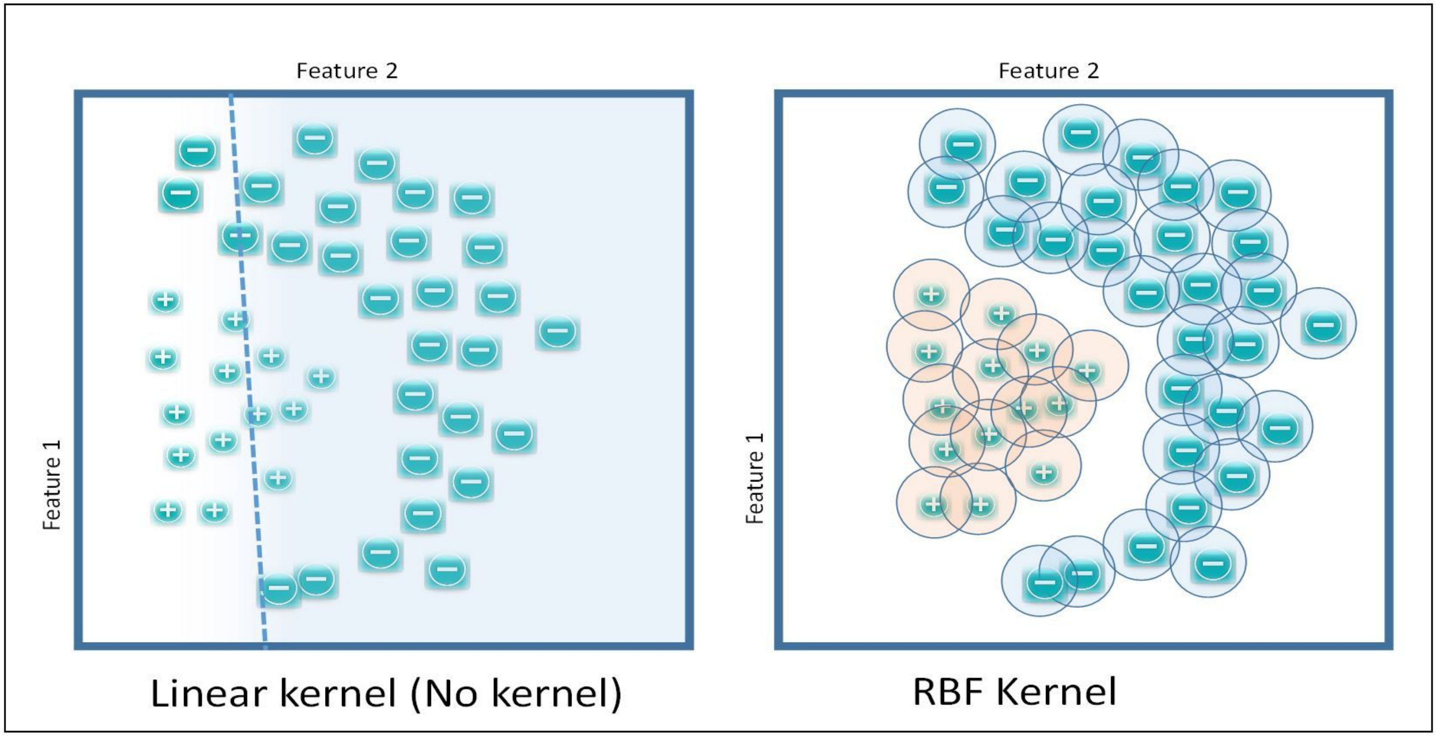
- Most powerful general purpose kernel

$$K(\mathbf{x}, \mathbf{x}') = \exp\left(-\frac{\|\mathbf{x} - \mathbf{x}'\|^2}{2\sigma^2}\right)$$

- Pretty much a Gaussian with mean x' and variance σ^2
  - Variance is a parameter to select
- This kernel comes from a space that has infinite dimensions

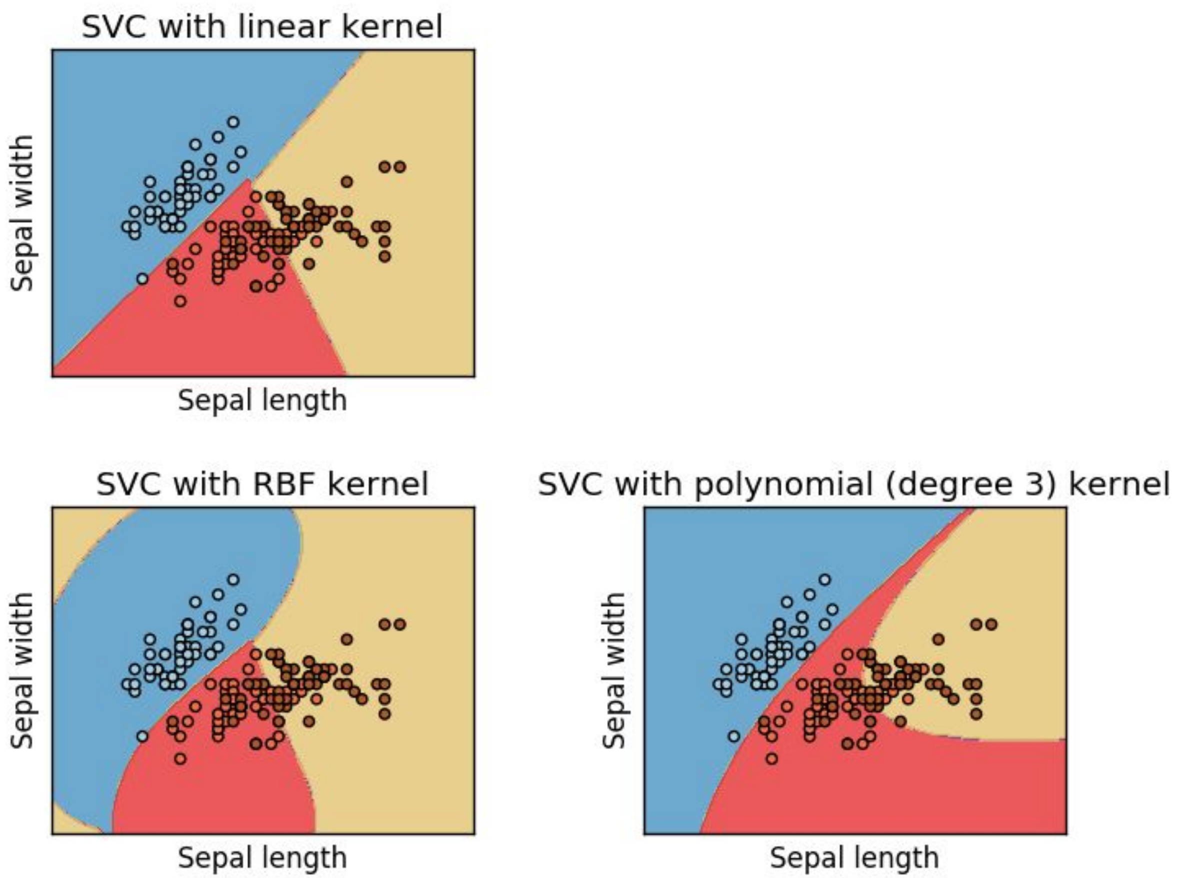
RBF kernels

- Think of RBF as putting Gaussians onto the support vectors



https://www.intechopen.com/books/epilepsy-topics/nonlinear-epilepsy-forewarning-by-support-vector-machines

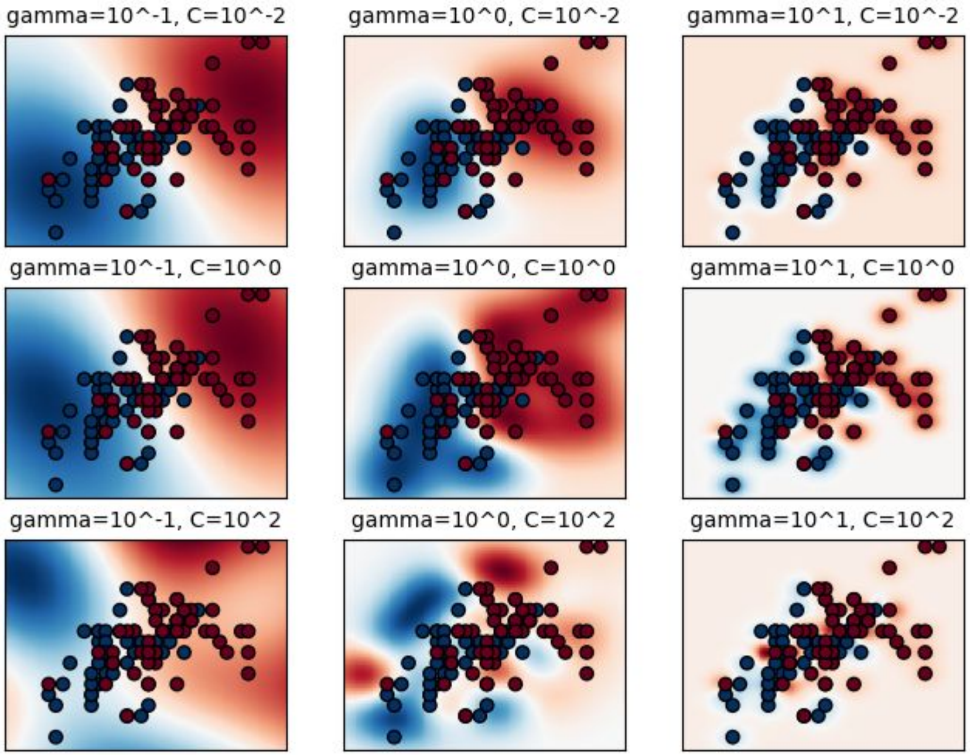
SVM examples



http://lijiancheng0614.github.io/scikit-learn/modules/svm.html

RBF SVM and sci-kit learn

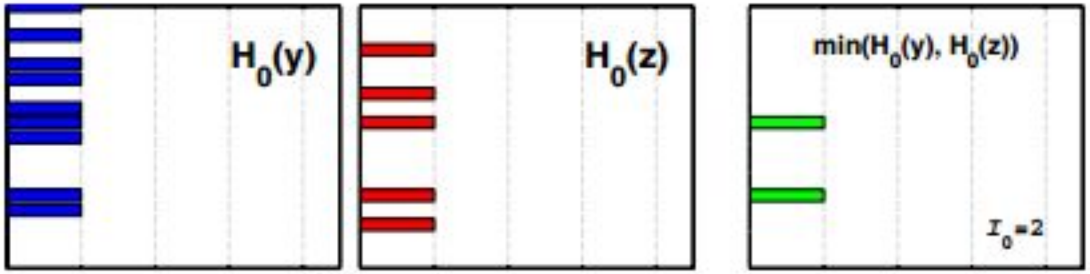
- Gamma is the inverse of the variance
- C is the inverse slack variable weight



http://scikit-learn.org/stable/auto_examples/svm/plot_rbf_parameters.html

Histogram intersection kernels

- Given input features which are histograms
  - Histogram of first data $H_0(y)$. Histogram of second data $H_1(z)$
- The Kernel that counts the intersection of the histograms is a valid kernel.
  - E.g. Sum of $\min(H_0(y), H_1(z))$ for all histogram bins
- (One of the most used kernels in computer vision)



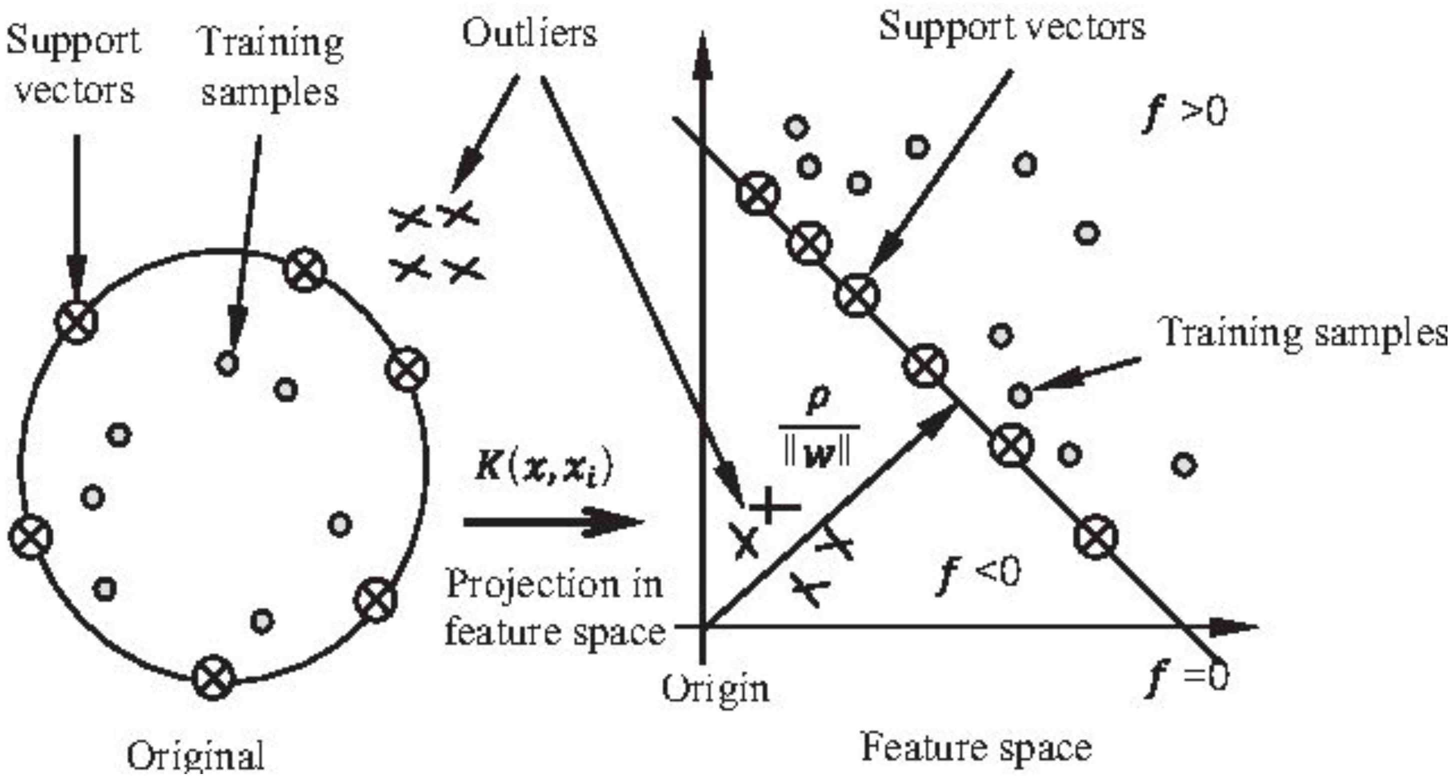
One class SVMs

- Sometimes it is easy to get positive examples but hard to acquire all possible negative examples
  - Email spam filter
    - We kind of know what a good email looks like. And we have lots of examples
    - Hard to model what a spam is. Spammer can change the format and evade detection.

- Solution: train on just the positive class
  - Model what that class looks like
  - Anything that deviates too much from it is considered negative examples

How?

- Separates the data from the “origin” (in mapped space)
- Maximize the distance between data points and the origin



http://rvlasveld.github.io/blog/2013/07/12/introduction-to-one-class-support-vector-machines/
https://www.semanticscholar.org/paper/The-effective-use-of-the-one-class-SVM-classifier-Guerbai-Chibani/ed575b5554e7704140d16af8390e153e26394283

Summary

- SVMs
  - Max margin
  - Slack
  - Kernel (inner product of higher space)
  - RBF kernels
  - One class SVM