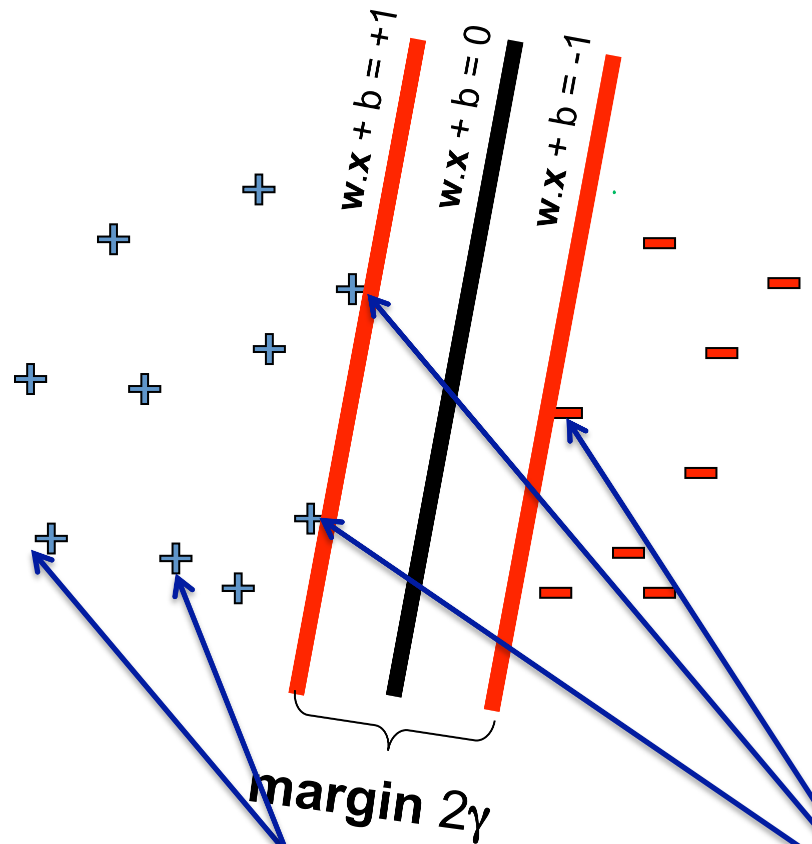
Support Vector Machines

Part 3: Key Takeaways

(Hard margin) Support Vector Machines



Minimize $||\mathbf{w}||^2$
subject to:

$$y^{(i)}(\mathbf{w} \bullet \mathbf{x}^{(i)} + b) \geq 1 \text{ for all } i$$

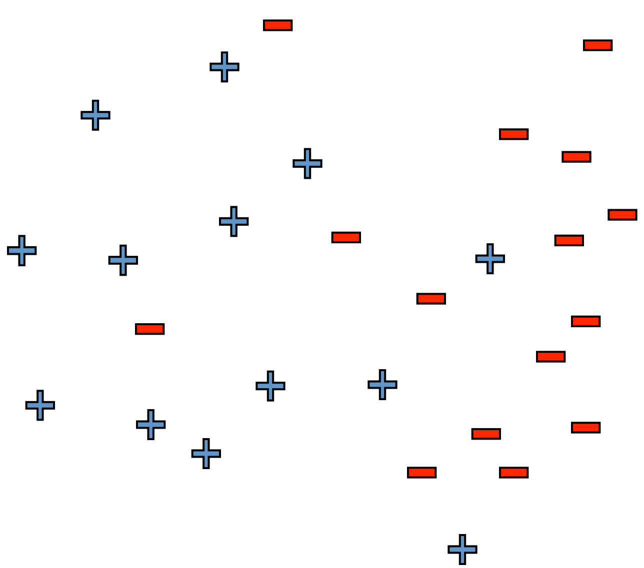
Example of a **convex optimization** problem

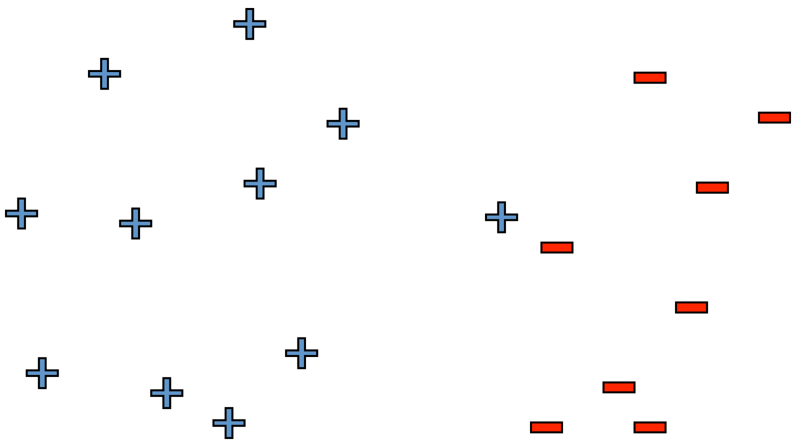
- A quadratic program
- Polynomial-time algorithms to solve!

Non-support Vectors:
- everything else
- moving them will not change **w**

Support Vectors:
- data points on the margin lines

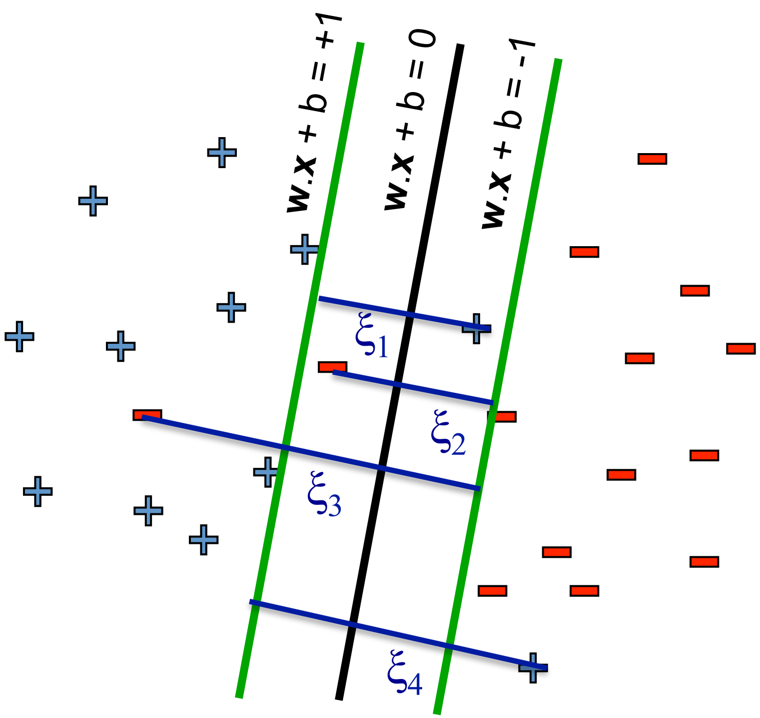
But what if you have:





non-separable or separable

"Soft margin SVM"



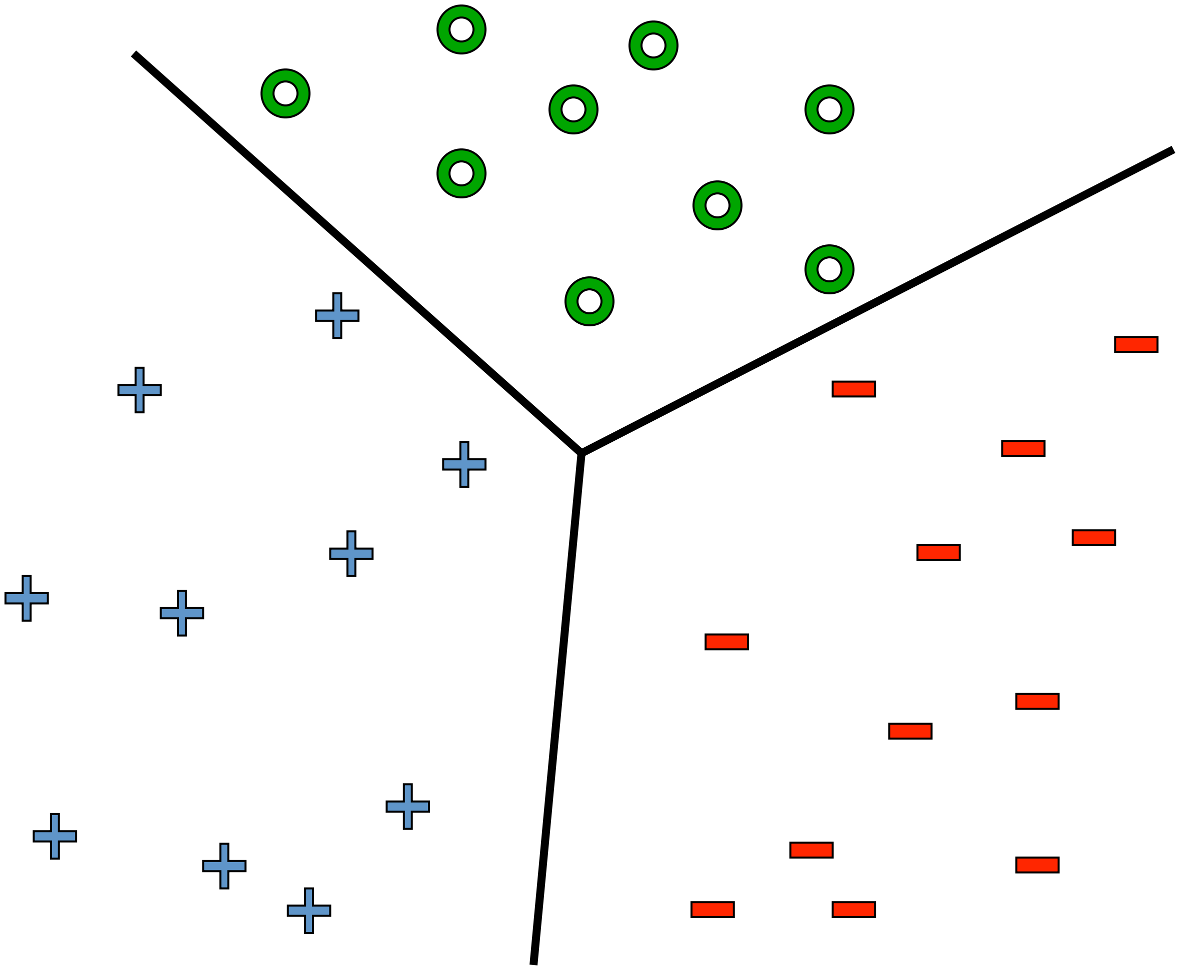
Minimize$_{\mathbf{w},b,\xi}$ $\mathbf{w}\bullet\mathbf{w} + C\,\Sigma_i\, \xi^{(i)}$
subject to
$(\mathbf{w}\bullet\mathbf{x}^{(i)}+b)y^{(i)} \geq 1 - \xi^{(i)}$ for all i

$\xi^{(i)} \geq 0$ for all i

Slack penalty $C > 0$:

- Want to find **w**, b so that the the margin is large and the # of errors is small.
- Want large margin to prevent overfitting.
- Solve optimization problem for different values of C. Choose the C that gives the smallest validation error.

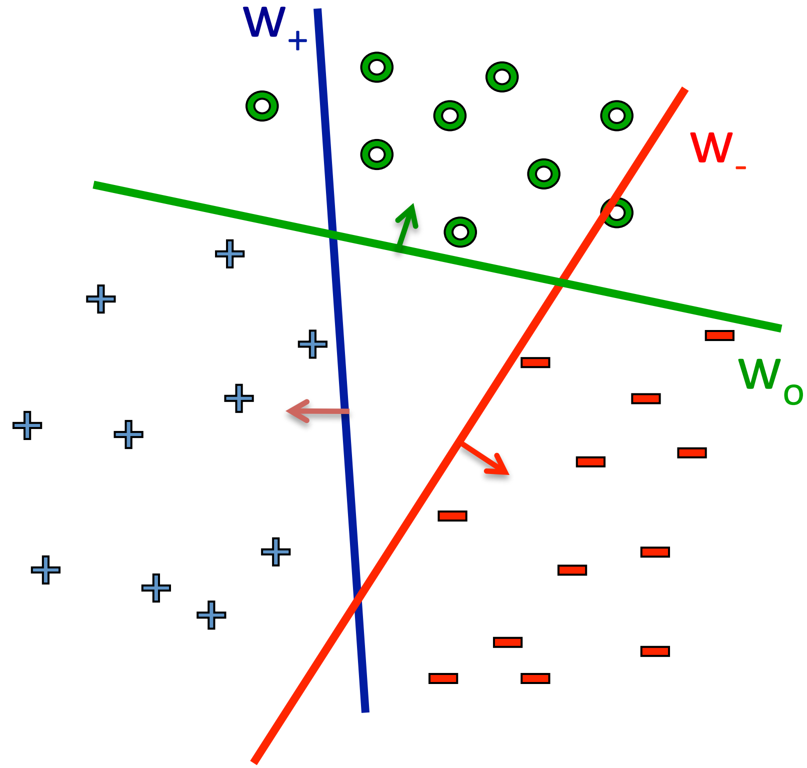
How do we do multi-class classification?



Multi-class SVM

Each example is now labeled either y = +, -, o.

We will simultaneously learn 3 sets of weights: $\mathbf{w}_+, \mathbf{w}_-, \mathbf{w}_o$ and three biases: b_+, b_-, b_o



Ideally, for each example, the "score" of the correct class will be better than the "score" of wrong classes, e.g, for a + examples, want:

$$\mathbf{w}_+ \bullet \mathbf{x}^{(i)} + b_+ > \mathbf{w}_- \bullet \mathbf{x}^{(i)} + b_- \quad \mathbf{w}_+ \bullet \mathbf{x}^{(i)} + b_+ > \mathbf{w}_- \bullet \mathbf{x}^{(i)} + b_- \quad \text{for} \quad y^{(i)} = +$$

Multi-class SVM

- May not be a feasible solution
- But we can allow for slack, and try to maximize the margin as before:

Minimize$_{\mathbf{w},b,\xi}$

$$\mathbf{w}_+ \bullet \mathbf{w}_+ + \mathbf{w}_- \bullet \mathbf{w}_- + \mathbf{w}_o \bullet \mathbf{w}_o + C \sum_i \xi^{(i)}$$

subject to

$$\mathbf{w}_{y(i)} \bullet \mathbf{x}^{(i)} + b_{y(i)} \geq \mathbf{w}_{y'} \bullet \mathbf{x}^{(i)} + b_{y'} + 1 - \xi^{(i)} \quad \text{for all } y' \neq y(i), \text{ for all } i$$

To predict, we use:

$$\hat{y} \leftarrow \arg\max_k \; w_k \cdot x + b_k$$

Dual Formulation of Soft-Margin SVM

Maximize:

$\Sigma_i \, \alpha_i - ½ \, \Sigma_{i,j} y^{(i)} y^{(j)} \, \alpha_i \, \alpha_j \, \mathbf{x}^{(i)} \bullet \mathbf{x}^{(j)}$

s.t. $\Sigma_i \, \alpha_i \, y^{(i)} = 0$

$0 \le \alpha_i \le C$ for i

- α_i's are now the variables in the optimization problem
- m variables
- m+1 constraints

$y \leftarrow \text{sign} \, [\, \Sigma_i \, \alpha_i y^{(i)} \mathbf{x} \bullet \mathbf{x}^{(i)} + b]$

Soft SVM with kernels

Maximize:

$\Sigma_i \alpha_i - ½ \Sigma_{i,j} y^{(i)} y^{(j)} \alpha_i \alpha_j K(\mathbf{x}^{(i)}, \mathbf{x}^{(j)})$

s.t. $\Sigma_i \alpha_i y^{(i)} = 0$

$0 \le \alpha_i \le C \quad$ for i

- Can replace $\mathbf{x} \bullet \mathbf{z}$ with more general function $K(\mathbf{x},\mathbf{z})$
- With the proper choice of function, can give much better results
  - Corresponds to non-linear decision region in original feature space
- But can't use just any function $K(\mathbf{x},\mathbf{z})$
  - Must be able to write $K(\mathbf{x},\mathbf{z})$ as $K(\mathbf{x},\mathbf{z}) := \boldsymbol{\phi}(\mathbf{x}) \bullet \boldsymbol{\phi}(\mathbf{z})$ where $\boldsymbol{\phi}(\mathbf{x})$ is some vector function of $\mathbf{x}$

Common kernels

- Polynomials of degree exactly *d*

$$K(\mathbf{x}, \mathbf{z}) = (\mathbf{x} \cdot \mathbf{z})^d$$

- Polynomials of degree up to *d*

$$K(\mathbf{x}, \mathbf{z}) = (\mathbf{x} \cdot \mathbf{z} + 1)^d$$

- Gaussian kernels

$$K(\mathbf{x}, \mathbf{z}) = \exp(-\frac{\|\mathbf{x} - \mathbf{z}\|_2^2}{2\sigma^2})$$

- And many others!