CS 412 Introduction to Machine Learning

Support Vector Machine

Instructor: Wei Tang

Department of Computer Science
University of Illinois at Chicago
Chicago IL 60607

https://tangw.people.uic.edu
tangw@uic.edu

Slides credit: Xinhua Zhang

Support Vector Machine (SVM)

- Discriminant-based: No need to estimate densities first
- Define the discriminant in terms of support vectors
- Convex optimization problems with a unique solution

Hyperplane that correctly separates

$$\mathcal{X} = \{\mathbf{x}^t, r^t\}_t \text{ where } r^t = \begin{cases} +1 & \text{if } \mathbf{x}^t \in C_1 \\ -1 & \text{if } \mathbf{x}^t \in C_2 \end{cases}$$

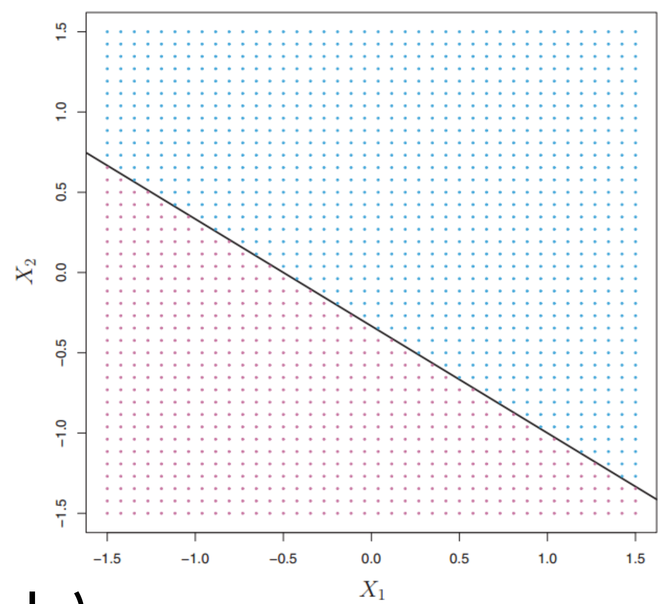
find $\mathbf{w}$ and w_0 such that

$$\mathbf{w}^T\mathbf{x}^t + w_0 \geq 0 \text{ for } r^t = +1$$

$$\mathbf{w}^T\mathbf{x}^t + w_0 \leq 0 \text{ for } r^t = -1$$

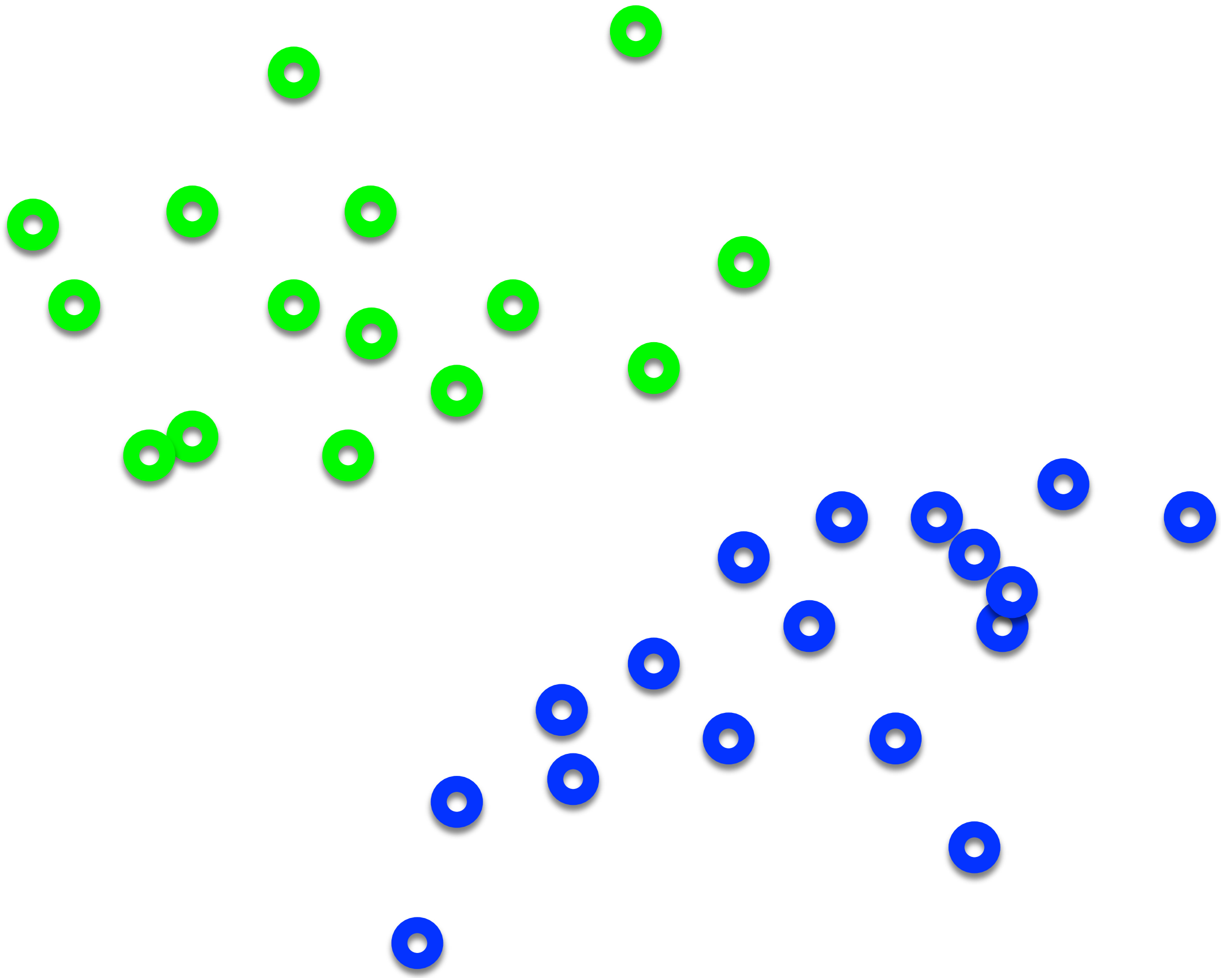
which can be rewritten as

$$r^t\left(\mathbf{w}^T\mathbf{x}^t + w_0\right) \geq 0$$

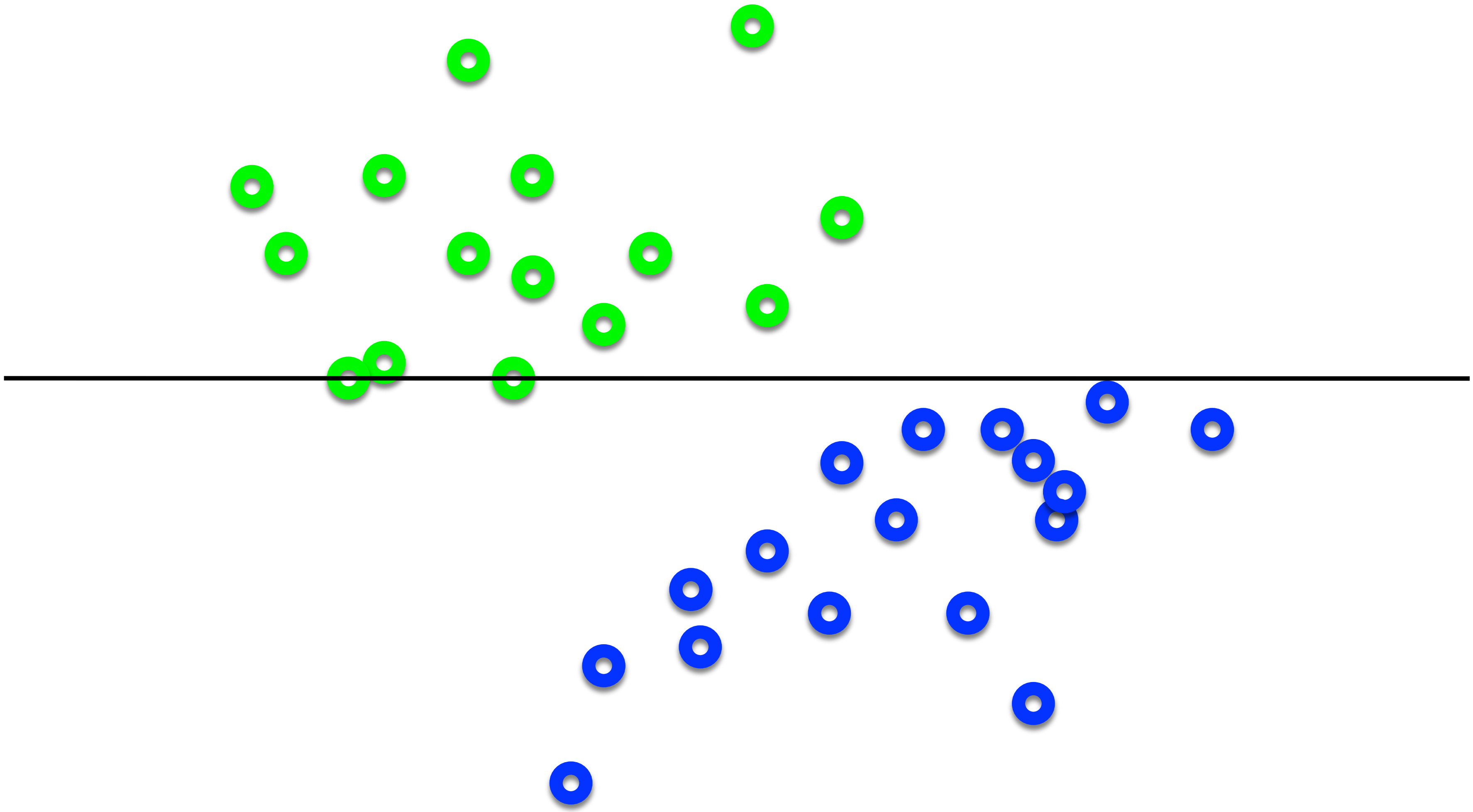


- Usually no solutions (not linearly separable)
- But…assume there is a solution, then what?

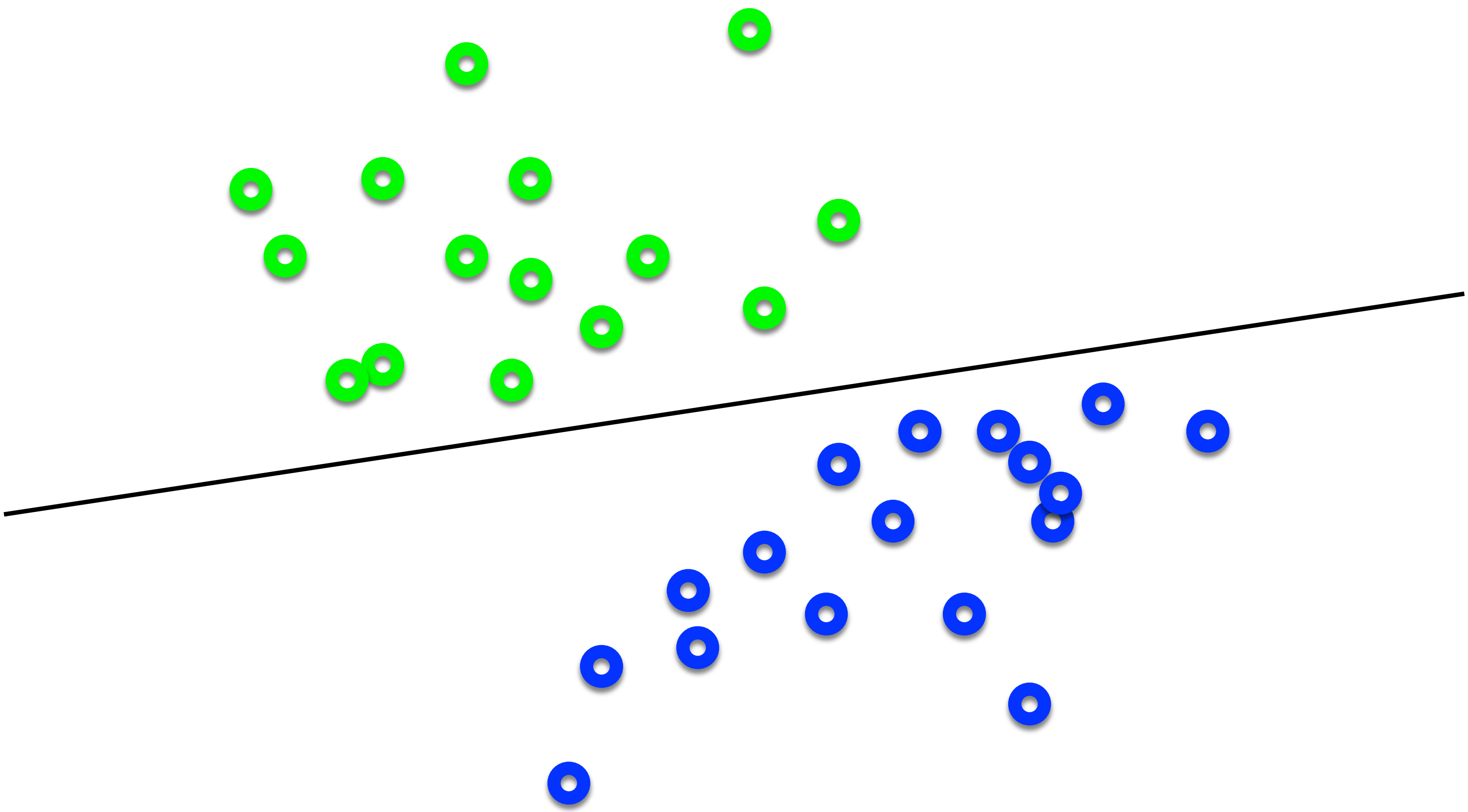
What’s the best **w**?



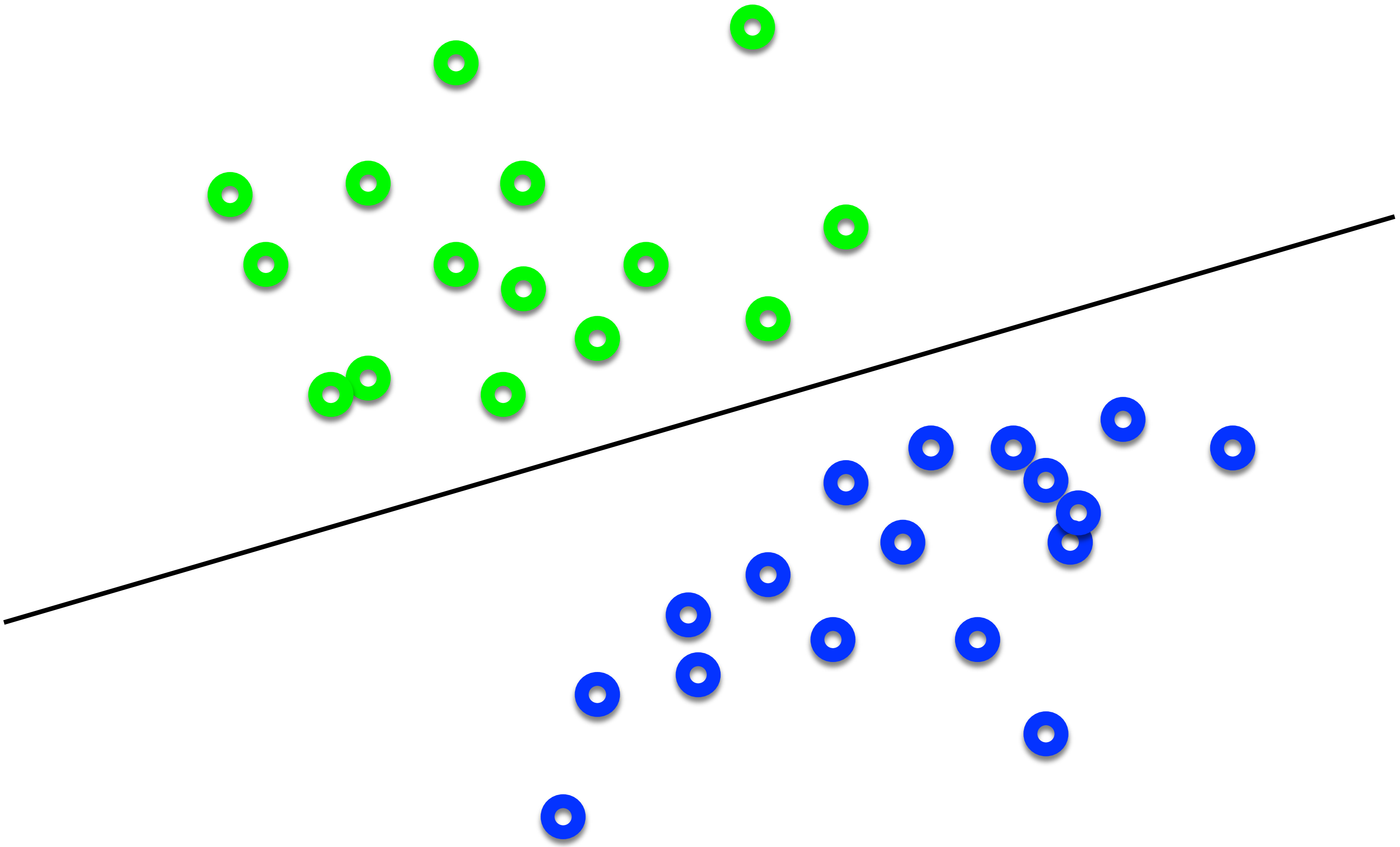
What’s the best **w**?



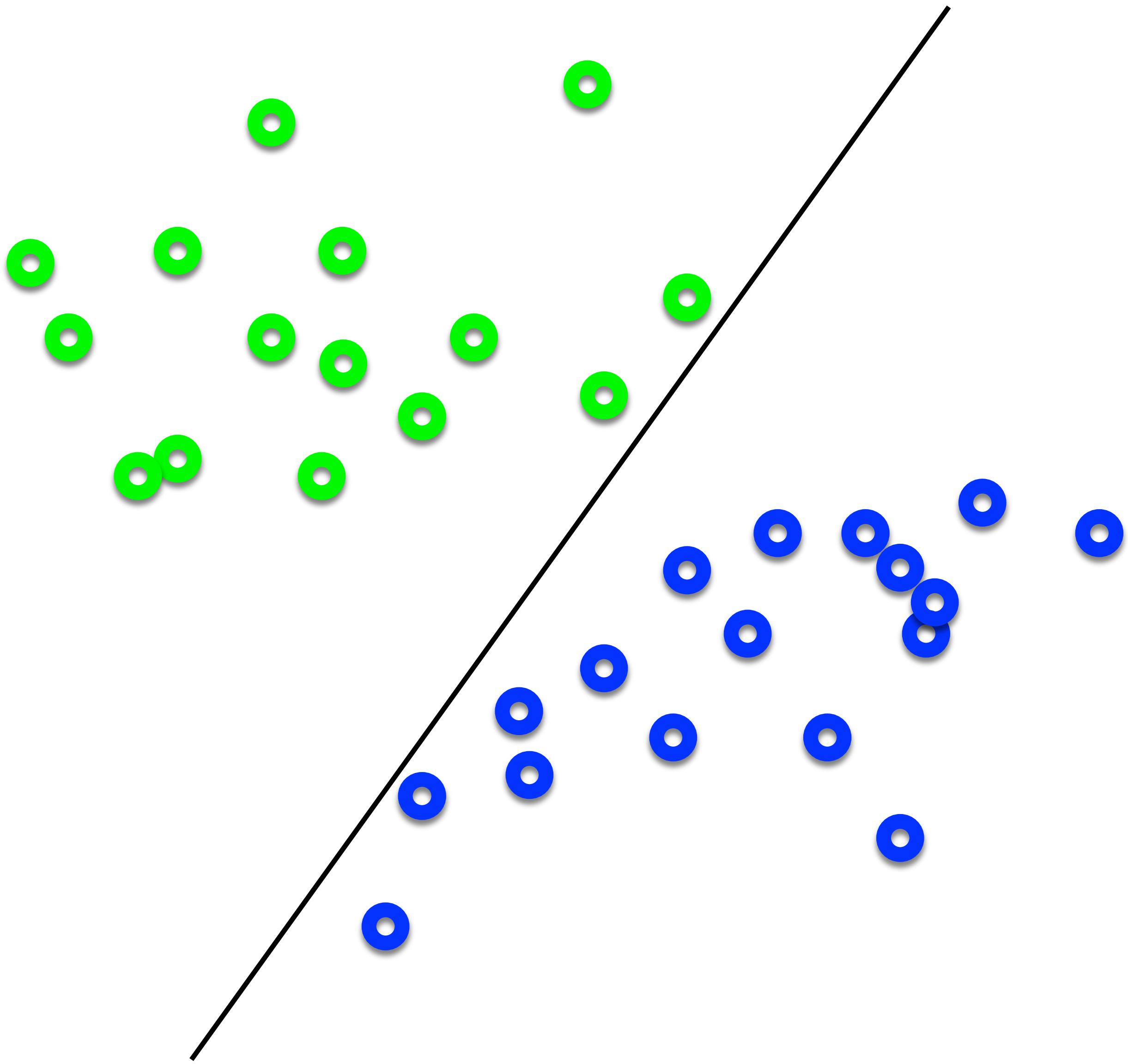
What's the best **w**?



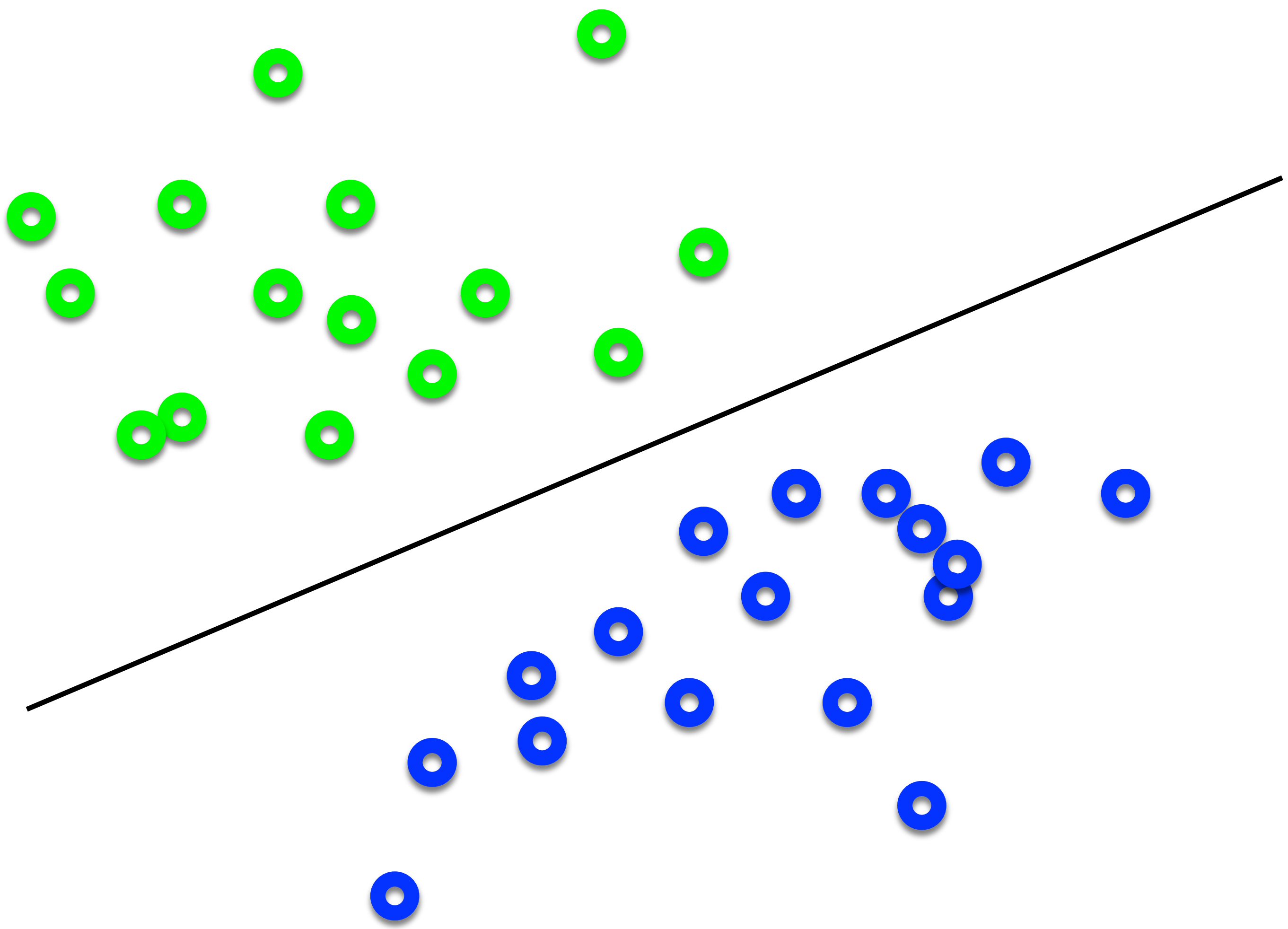
What's the best **w**?



What's the best **w**?

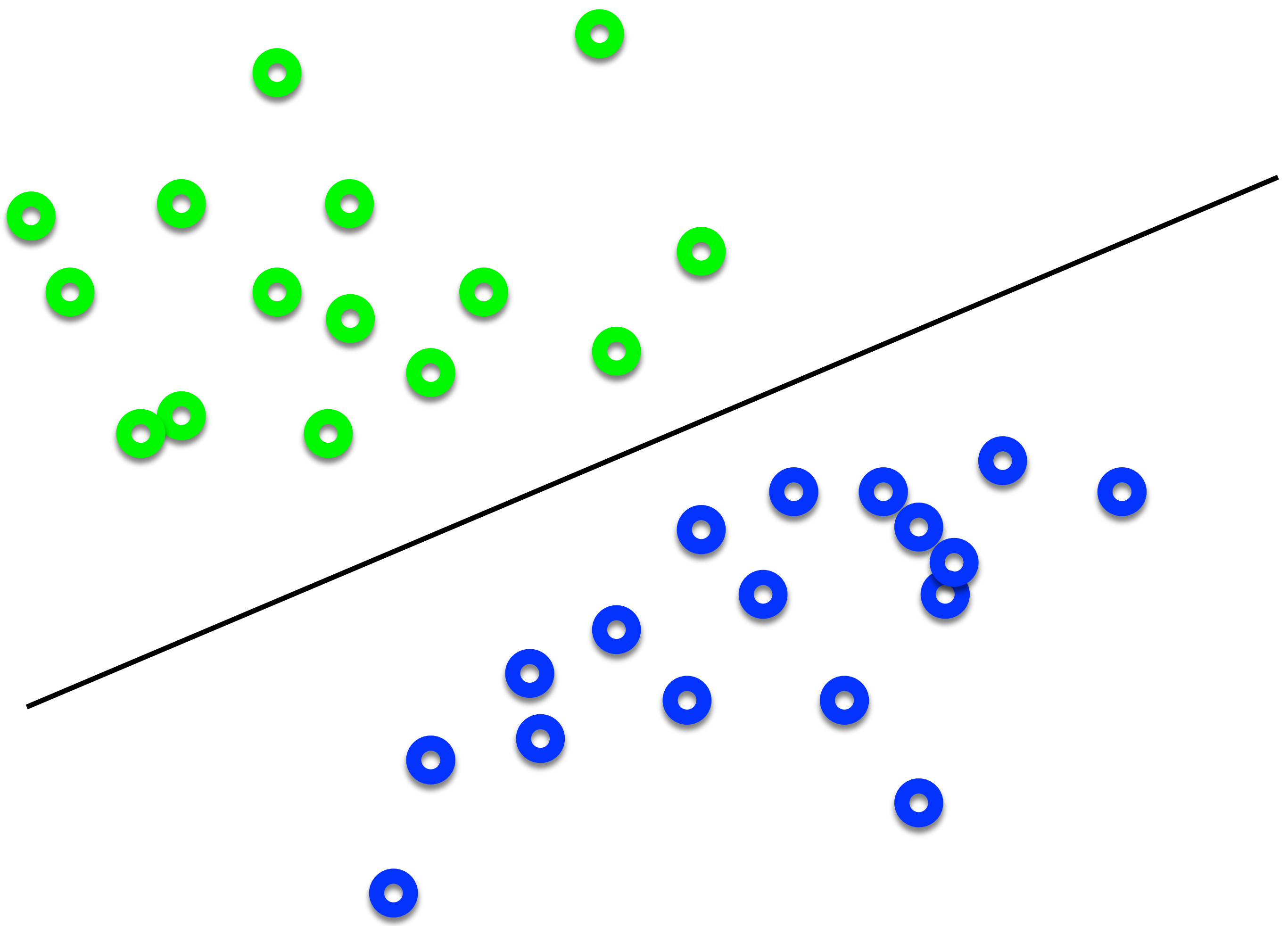


What’s the best **w**?



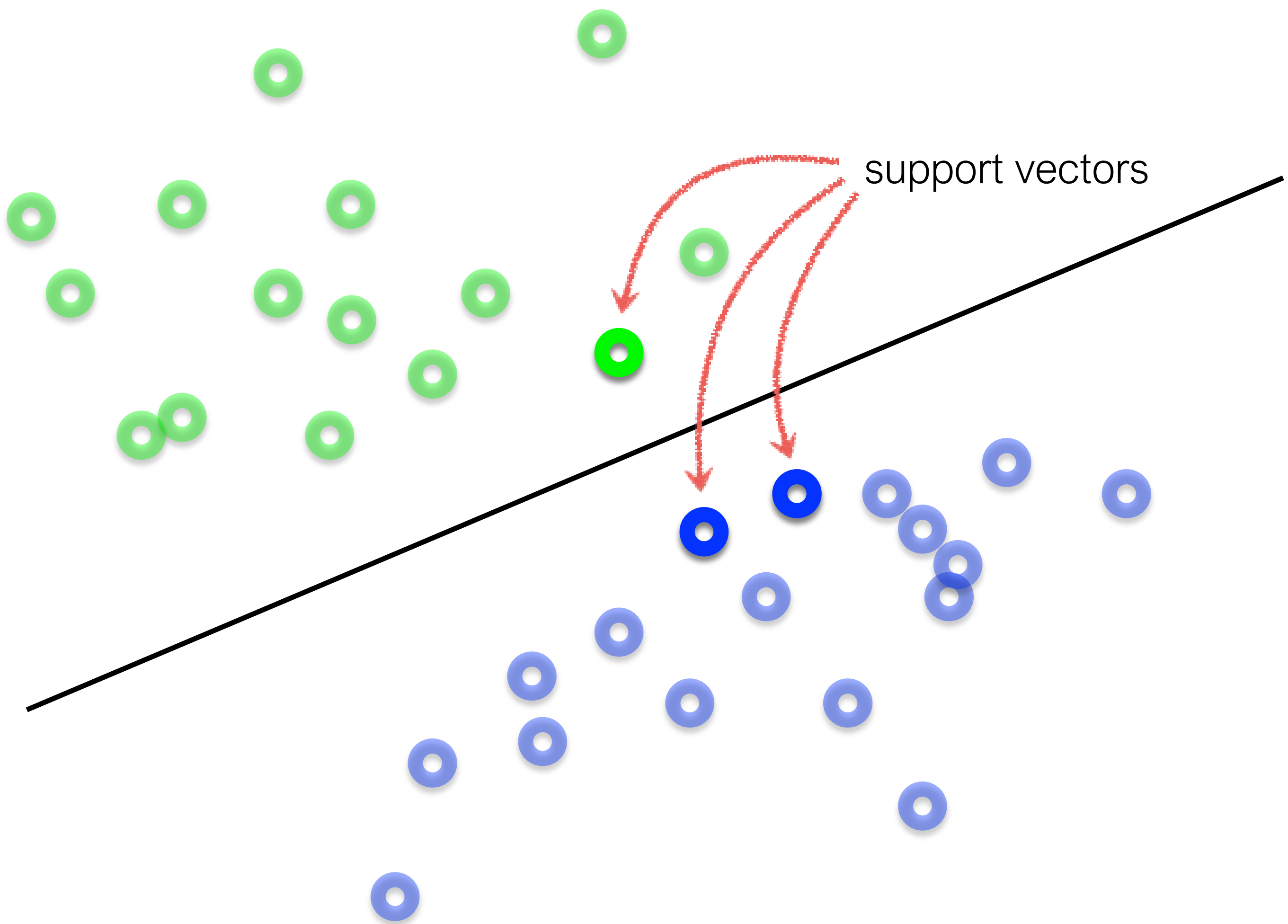
Intuitively, the line that the one that represents the largest separation, or margin, between the two classes

What's the best **w**?



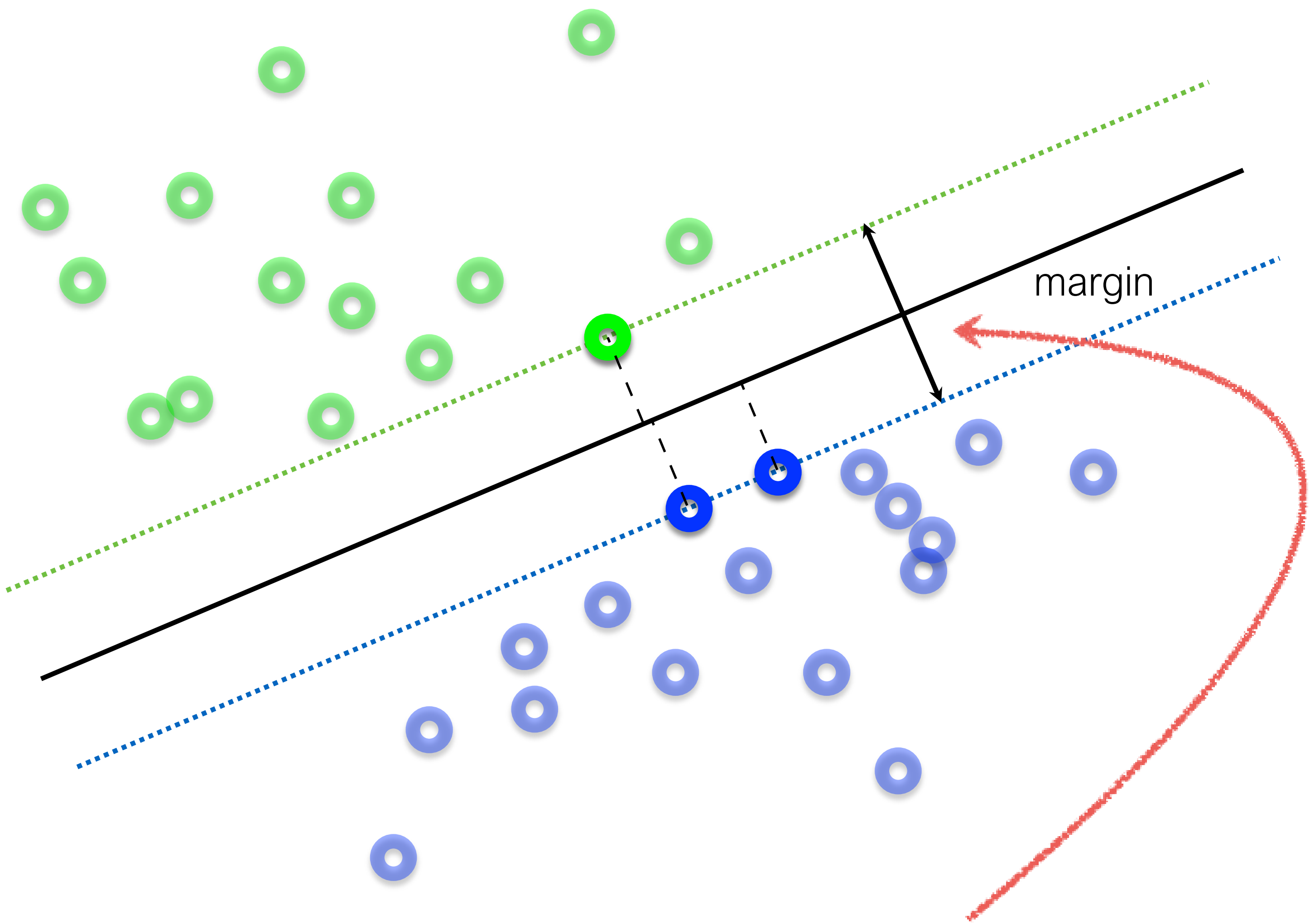
Maximum Margin solution: most stable to perturbations of data

What's the best **w**?



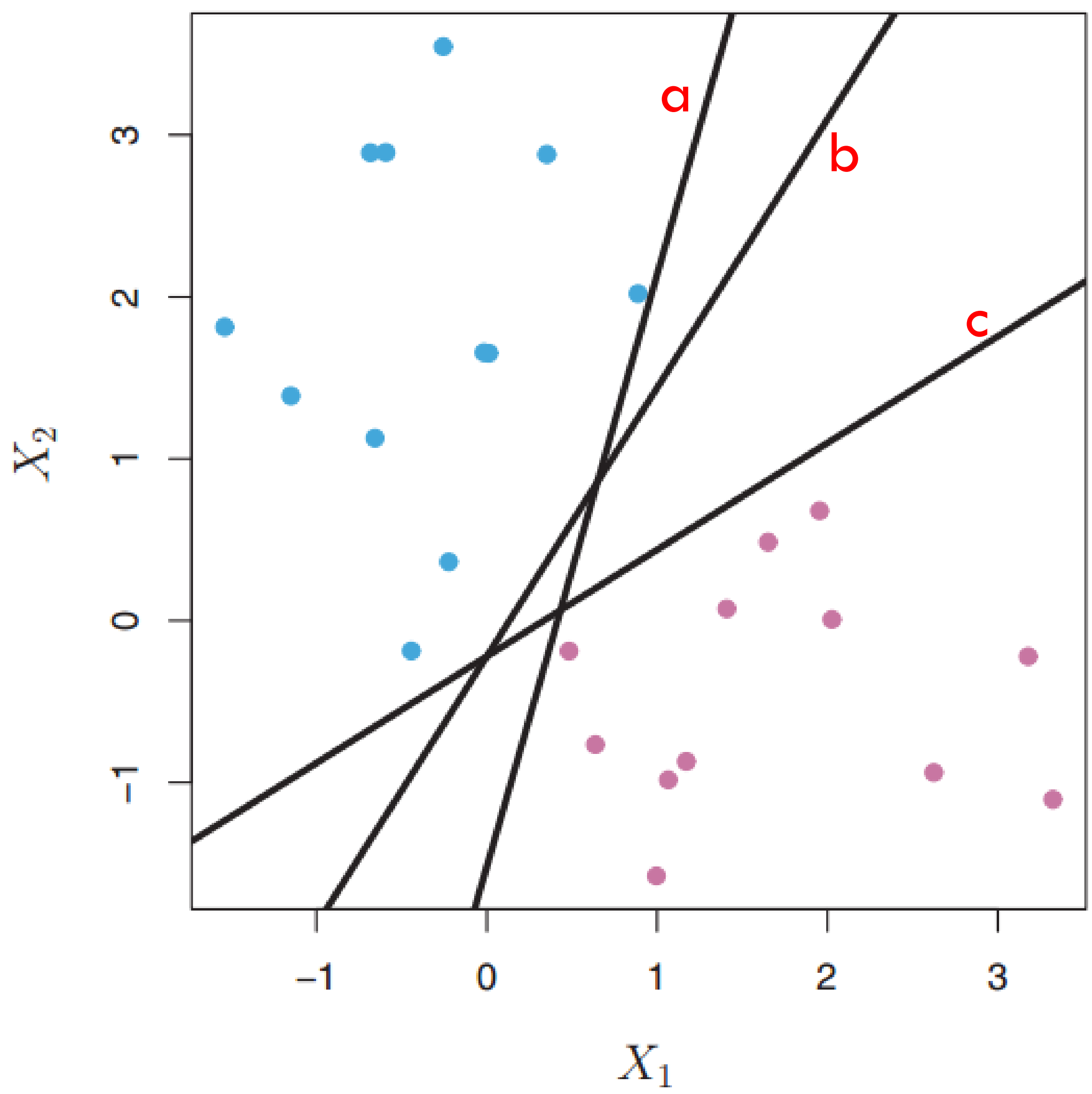
Want a hyperplane that represents the largest separation, or margin, between the two classes

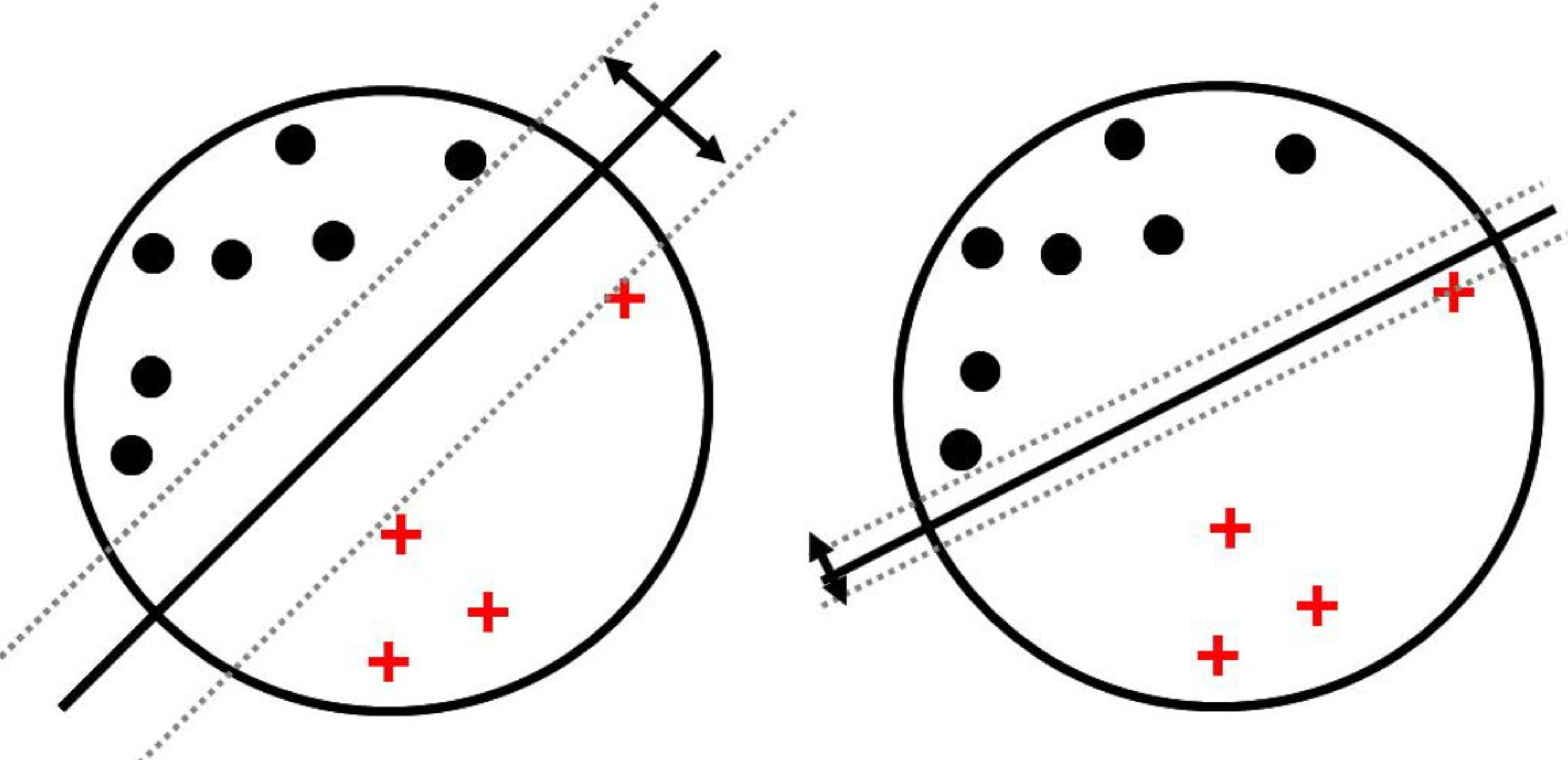
Find hyperplane **w** such that …



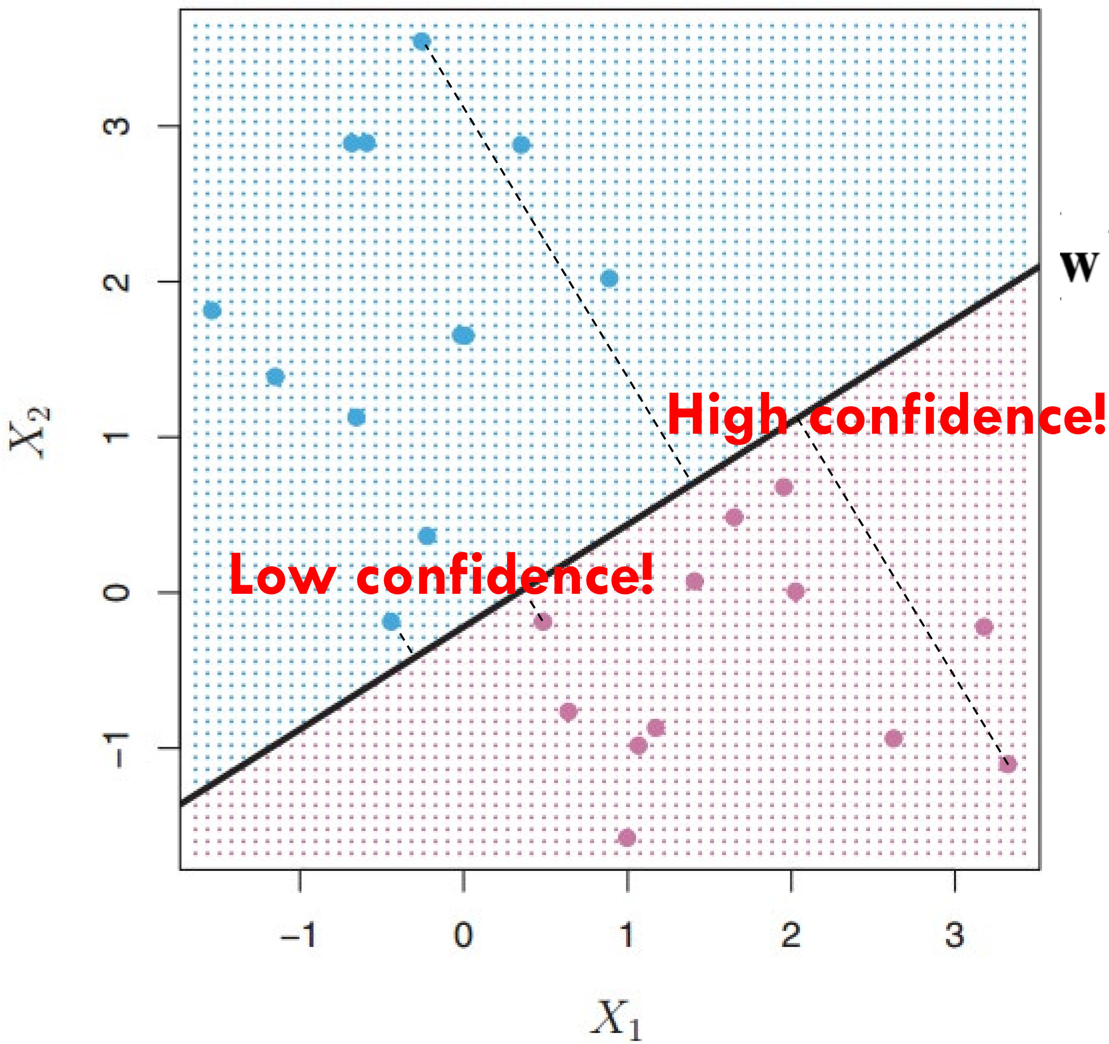
the gap between parallel hyperplanes is maximized

Linear classifiers: Which hyperplane is best?





“Confidence” of Predictions



$\mathbf{w}^T\mathbf{x} + w_0 = 0$

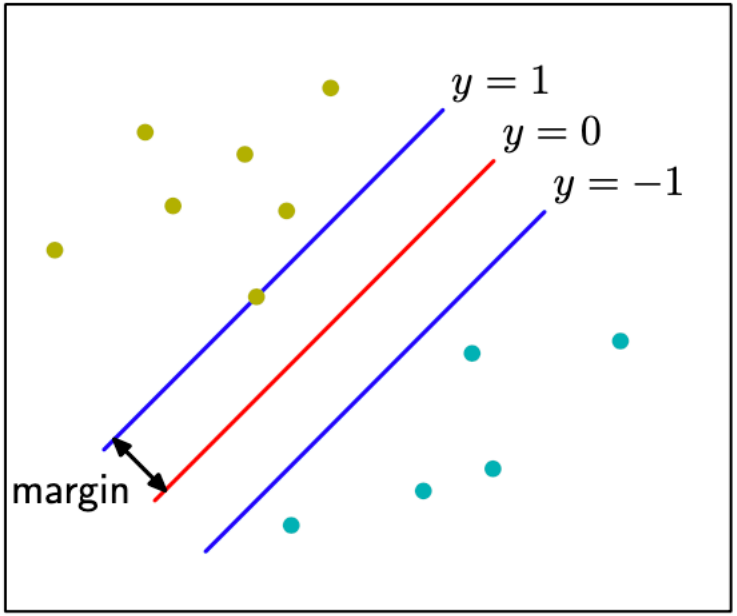
“Confidence” =

$$r^t\left(\mathbf{w}^T\mathbf{x}^t + w_0\right)$$

What about multiplying $\mathbf{w}$ and w_0 by 2 or 100?

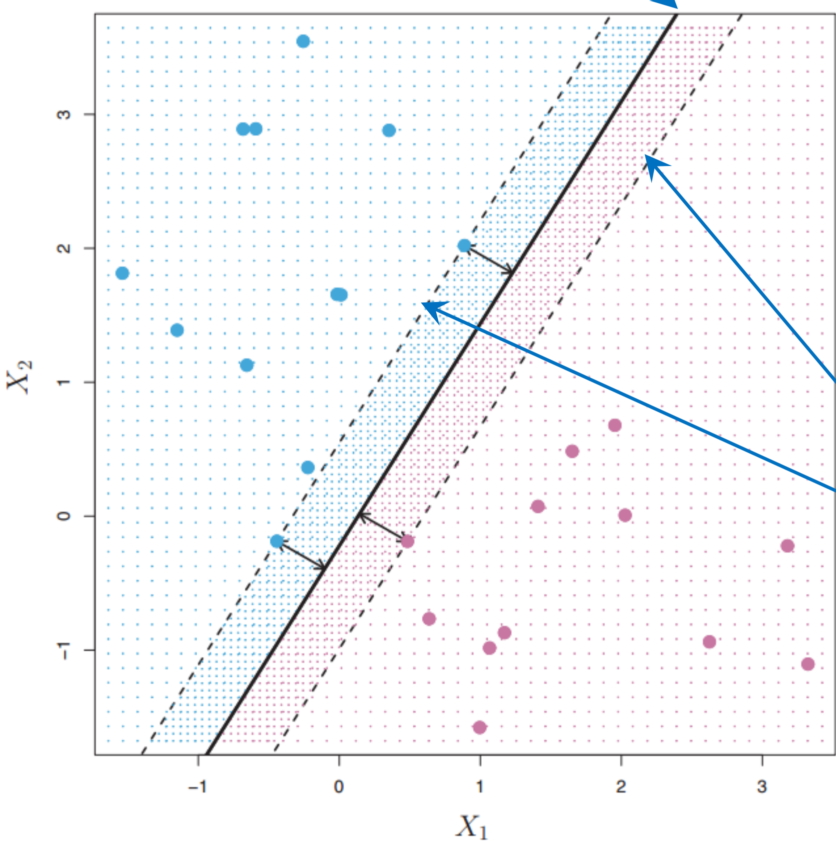
Margin

- Perpendicular distance between between boundary and the closest data point



Pick the one with the largest margin!

$\mathbf{w}^T\mathbf{x} + w_0 = 0$ separation boundary

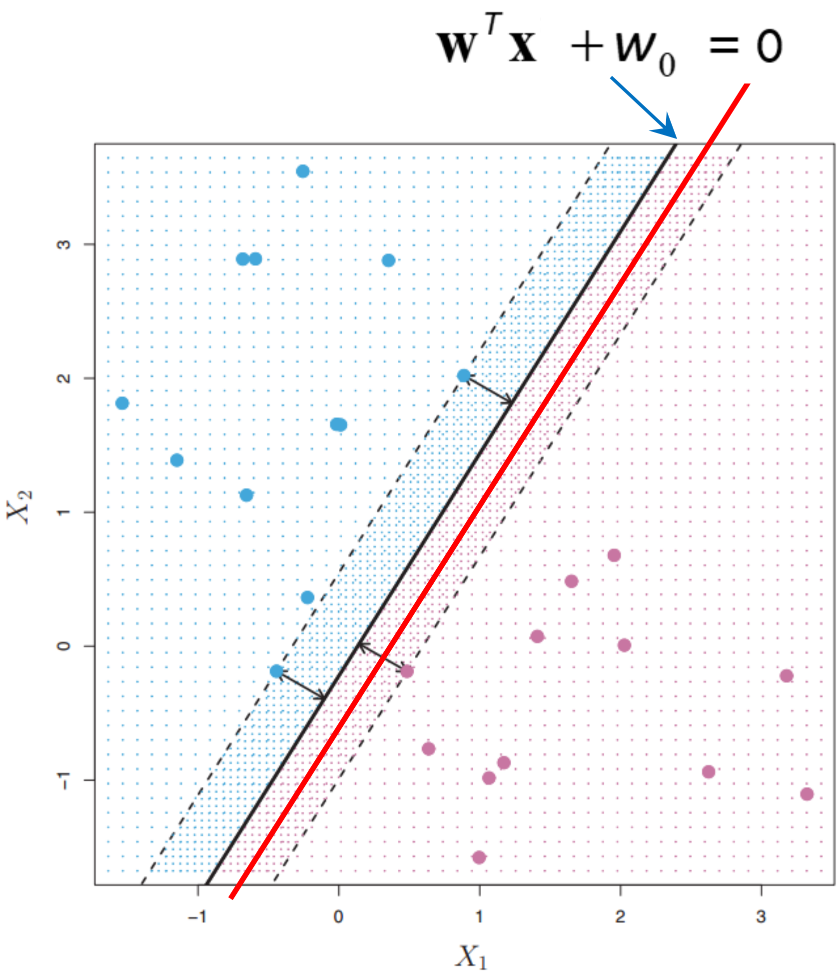


Points on the margin boundary have the lowest "confidence" over all points

Let's maximize this!

margin boundary

Pick the one with the largest margin!



Points on the margin boundary have the lowest "confidence" over all points

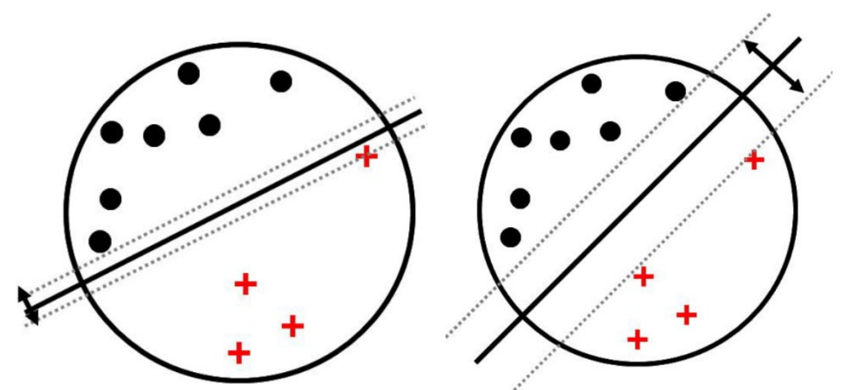
Let's maximize this!

Naturally, we want the margin to be the same for pos and neg

Hard margin SVM (linearly separable)

- Maximize the distance from the discriminant to the closest instances on either side
- Distance of x^t to the hyperplane is $\dfrac{r^t\left(\mathbf{w}^T\mathbf{x}^t + w_0\right)}{\|\mathbf{w}\|}$
- Margin of the dataset $\min\limits_{t} \dfrac{r^t\left(\mathbf{w}^T\mathbf{x}^t + w_0\right)}{\|\mathbf{w}\|}$
- Find the (w, w_0) hyperplane that maximizes the margin

$$\max_{w,w_0} \min_{t} \frac{r^t\left(\mathbf{w}^T\mathbf{x}^t + w_0\right)}{\|\mathbf{w}\|}$$



Hard margin SVM (linearly separable)

- Find the (w, w_0) hyperplane that maximizes the margin

$$\max_{w,w_0} \min_t \frac{r^t\left(\mathbf{w}^T\mathbf{x}^t + w_0\right)}{\|\mathbf{w}\|} = \max_{w,w_0} \frac{\min_t r^t(w^T x^t + w_0)}{||w||}$$

- Key idea: restrict the search on (w, w_0) to those such that

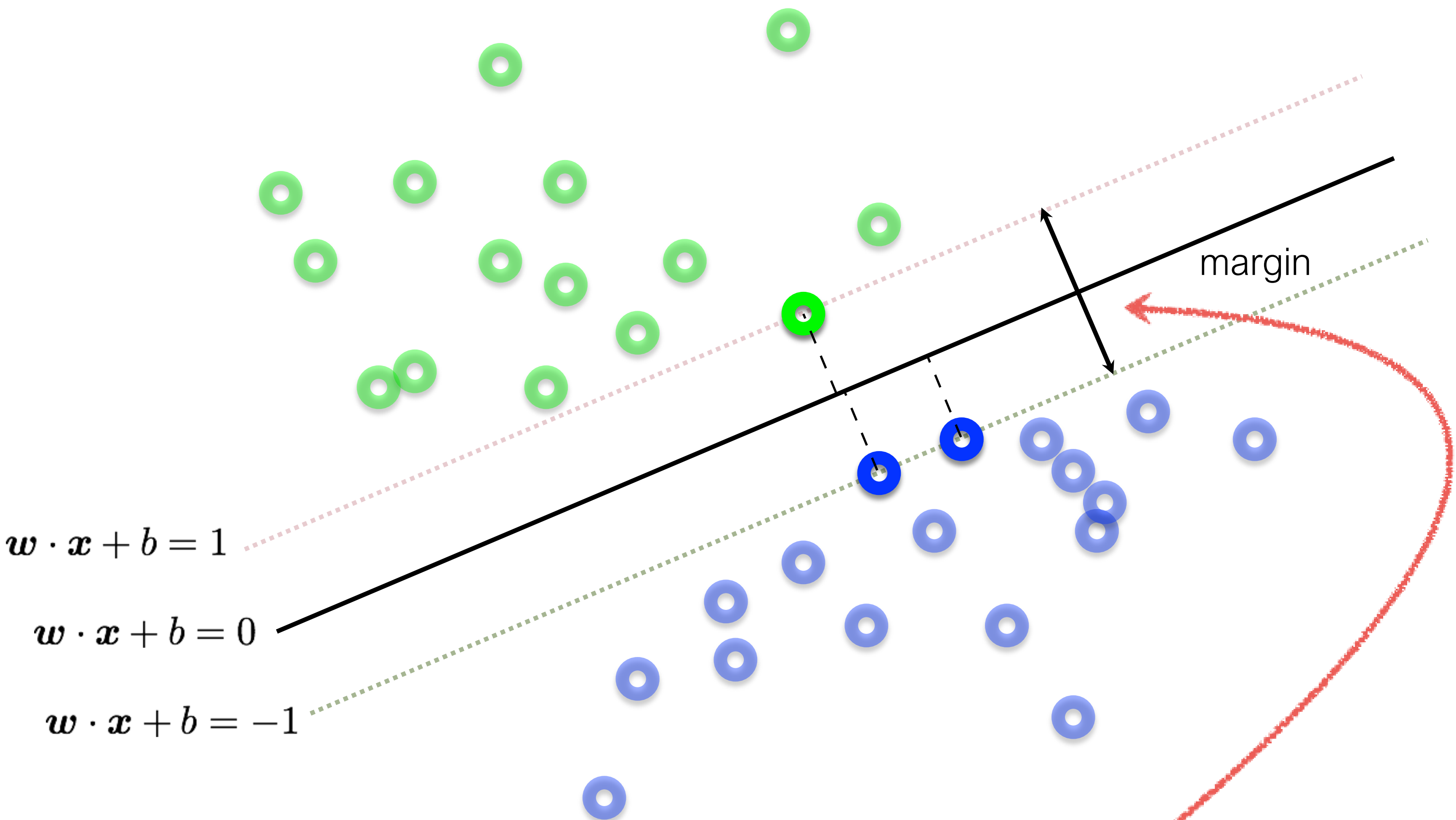
$$\min_t r^t(w^T x^t + w_0) = 1$$

find $\mathbf{w}$ and w_0 such that

$$\boxed{\begin{aligned} &\mathbf{w}^T\mathbf{x}^t + w_0 \geq +1 \text{ for } r^t = +1 \\ &\mathbf{w}^T\mathbf{x}^t + w_0 \leq -1 \text{ for } r^t = -1 \end{aligned}}$$

- Eventually, $\max_{w,w_0} \frac{1}{||w||} \Leftrightarrow \min_{w,w_0} ||w||$
  - subject to the constraints in red box

Find hyperplane **w** such that …

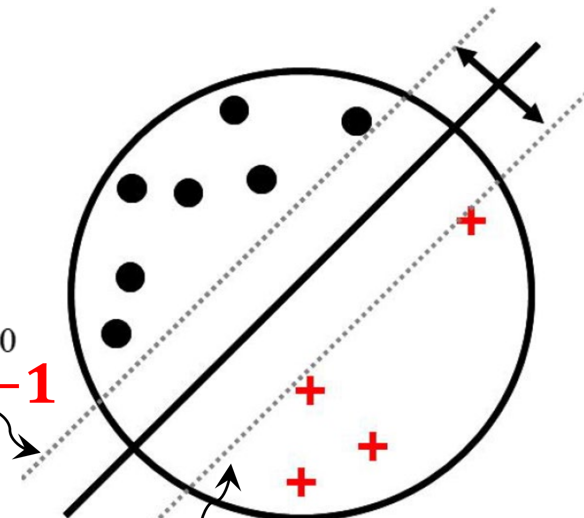


the gap between parallel hyperplanes $\frac{2}{\|\boldsymbol{w}\|}$ is maximized

Hard margin SVM (linearly separable)

- Key idea: restrict the search on (w, w_0) to those such that

$$\min_t r^t(w^T x^t + w_0) = 1$$



- Eventually, $\max_{w,w_0} \frac{1}{||w||} \Leftrightarrow \min_{w,w_0} ||w||$
- Putting together

$$\min_{w,w_0} \frac{1}{2}\|\mathbf{w}\|^2 \text{ subject to } \boxed{r^t\left(\mathbf{w}^T\mathbf{x}^t + w_0\right) \geq +1, \forall t}$$

  - At the optimal, $\min_t r^t(w^T x^t + w_0)$ will be exactly 1, not > 1

Margin and support vectors

- Margin ρ

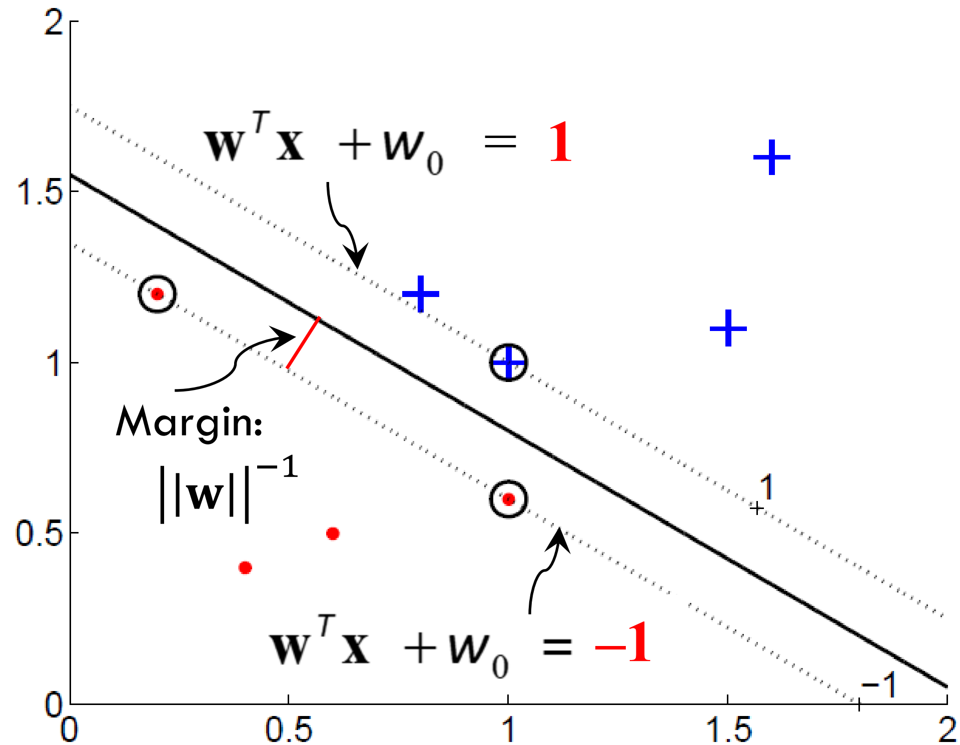
$$\min_t \frac{r^t\left(\mathbf{w}^T\mathbf{x}^t + w_0\right)}{\|\mathbf{w}\|} = \frac{1}{||\mathbf{w}||}$$

- Marginal hyperplanes

$$\mathbf{w}^T\mathbf{x} + w_0 = -1$$
$$\mathbf{w}^T\mathbf{x} + w_0 = 1$$

- Separating hyperplane

$$\mathbf{w}^T\mathbf{x} + w_0 = 0$$



- Support vectors: points lying on the marginal hyperplanes
    - All the examples t with $r^t\left(w^T x^t + w_0\right) = 1$
    - NO change of solution if: remove all other points and retrain support vectors

Learning

$$\min_{w, w_0} \frac{1}{2}\|\mathbf{w}\|^2 \text{ subject to } r^t\left(\mathbf{w}^T\mathbf{x}^t + w_0\right) \geq +1, \forall t$$

- Convex optimization: global optimum!
- Quadratic programming: a bunch of packages available
  - CVXOPT, CVXPY, Gurobi, MOSEK, quadprog …

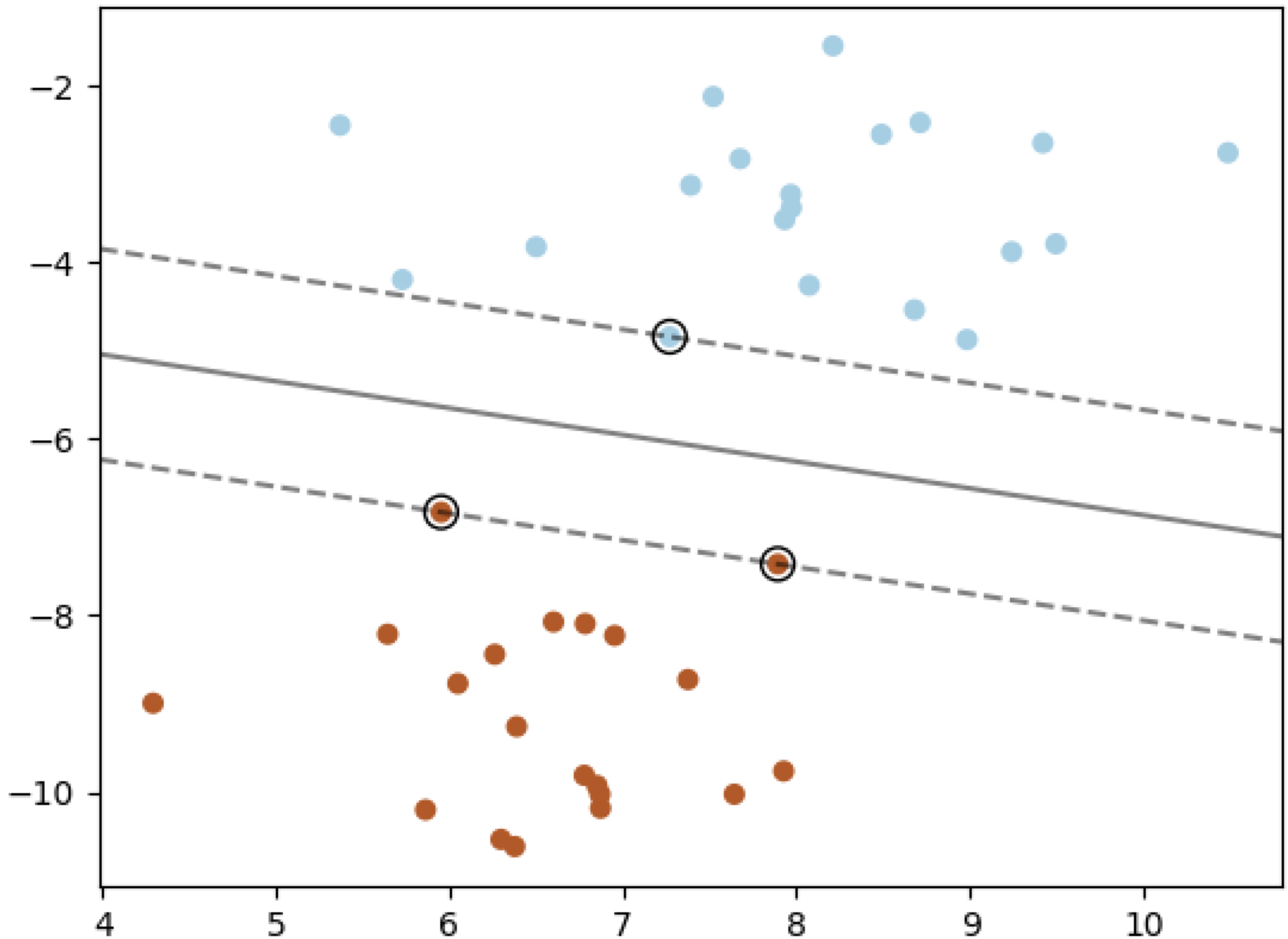
`cvxopt.solvers.qp(`*P, q [, G, h [, A, b [, solver [, initvals]]]]*`)`

Solves the pair of primal and dual convex quadratic programs

$$\begin{array}{ll} \text{minimize} & (1/2)x^T P x + q^T x \\ \text{subject to} & Gx \preceq h \\ & Ax = b \end{array}$$

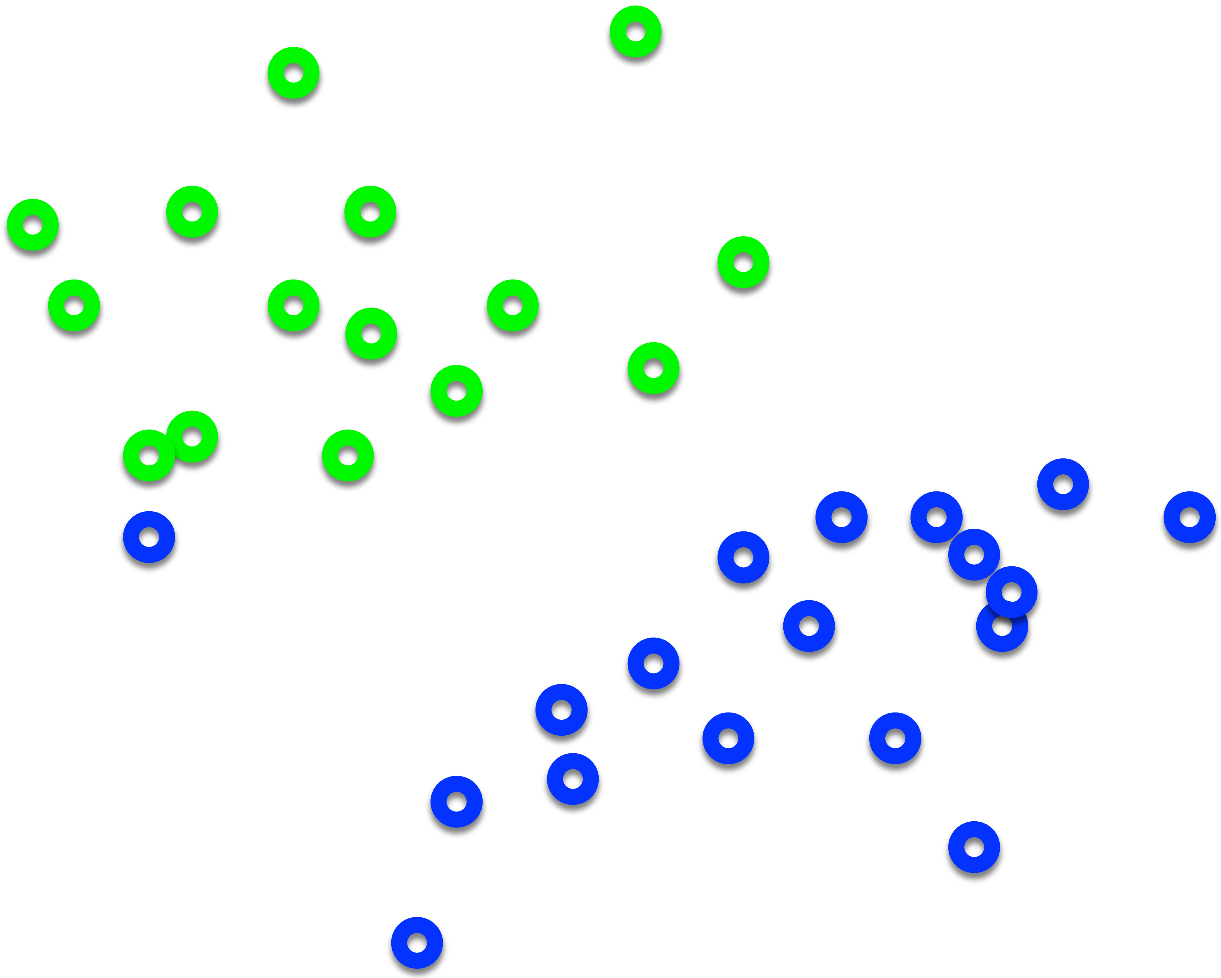
SVM in scikit-learn

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn import svm
from sklearn.datasets import make_blobs

# we create 40 separable points
X, y = make_blobs(n_samples=40, centers=2, random_state=6)

# fit the model, don't regularize for illustration purposes
clf = svm.SVC(kernel='linear')
clf.fit(X, y)
```

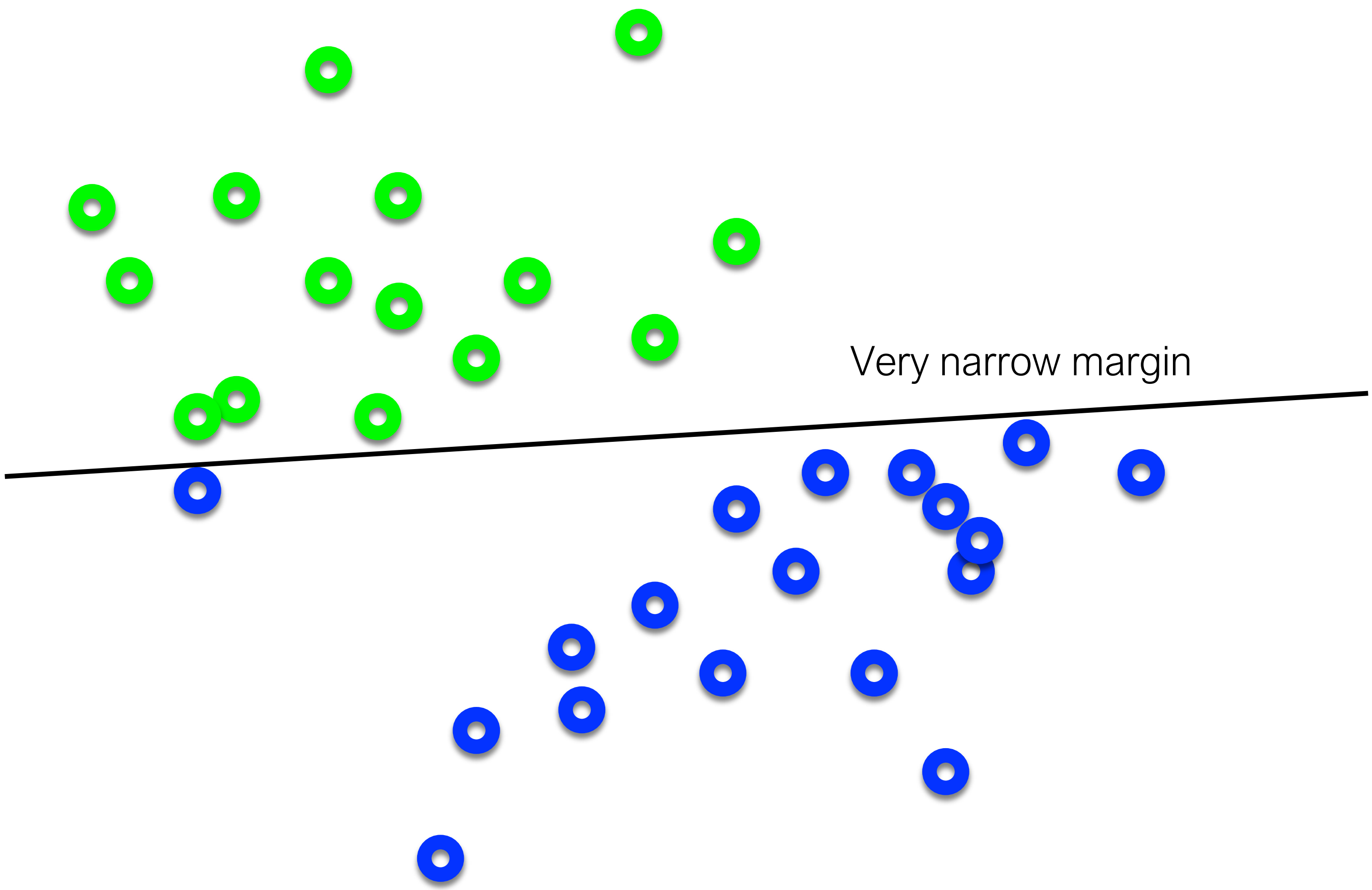


−2
−4
−6
−8
−10
4
5
6
7
8
9
10

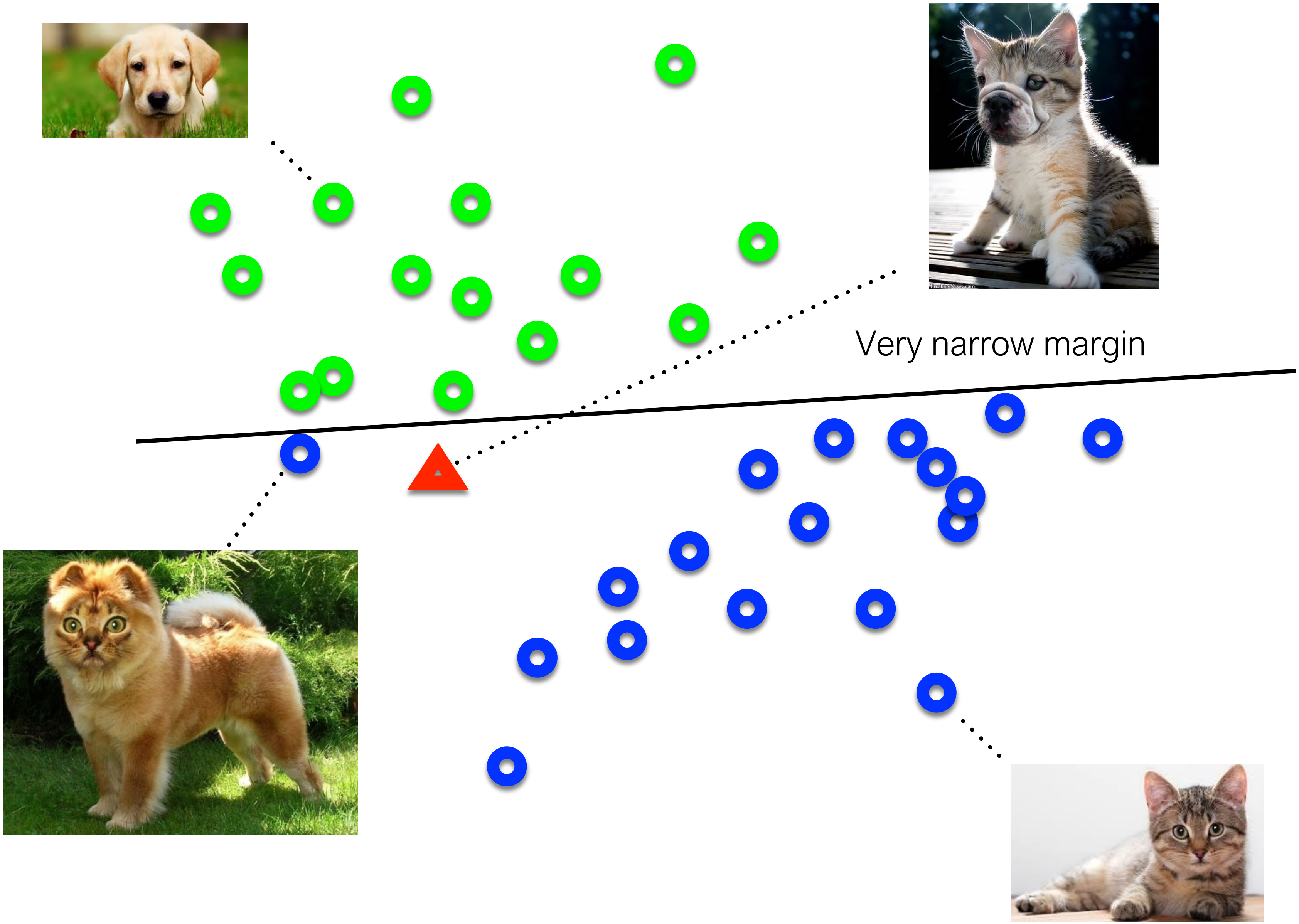
What's the best **w**?



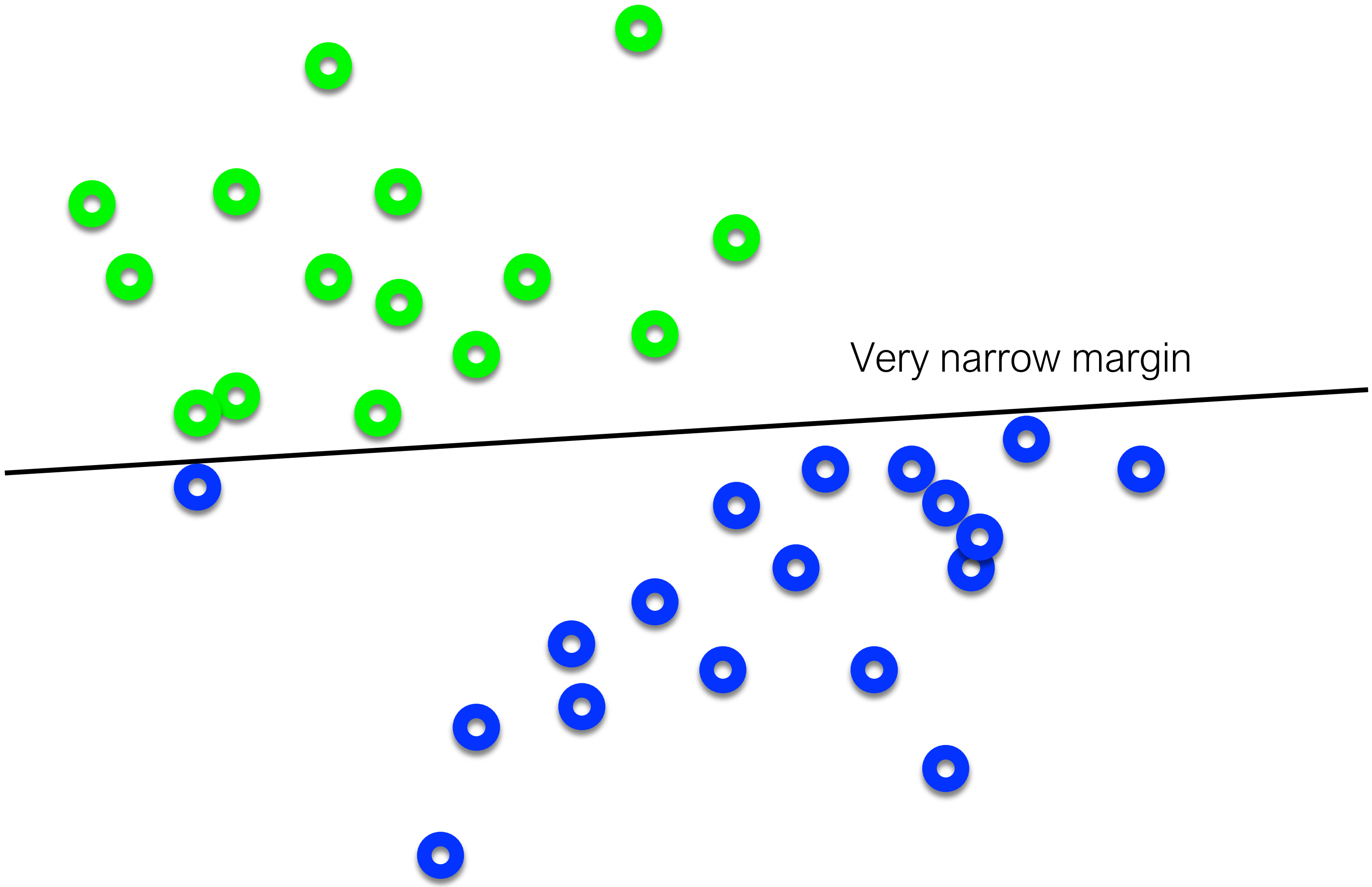
What's the best **w**?



Separating cats and dogs

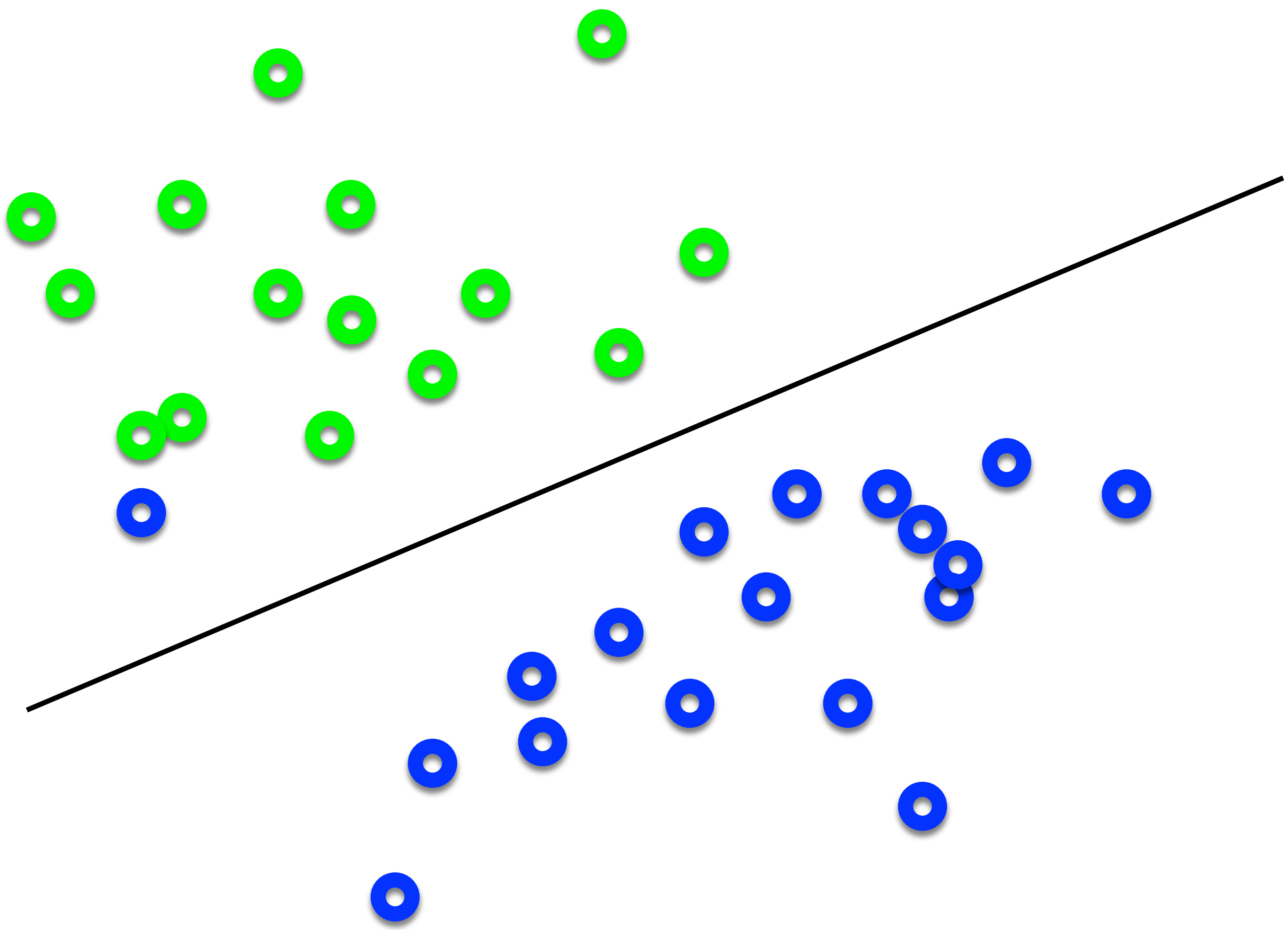


What's the best **w**?



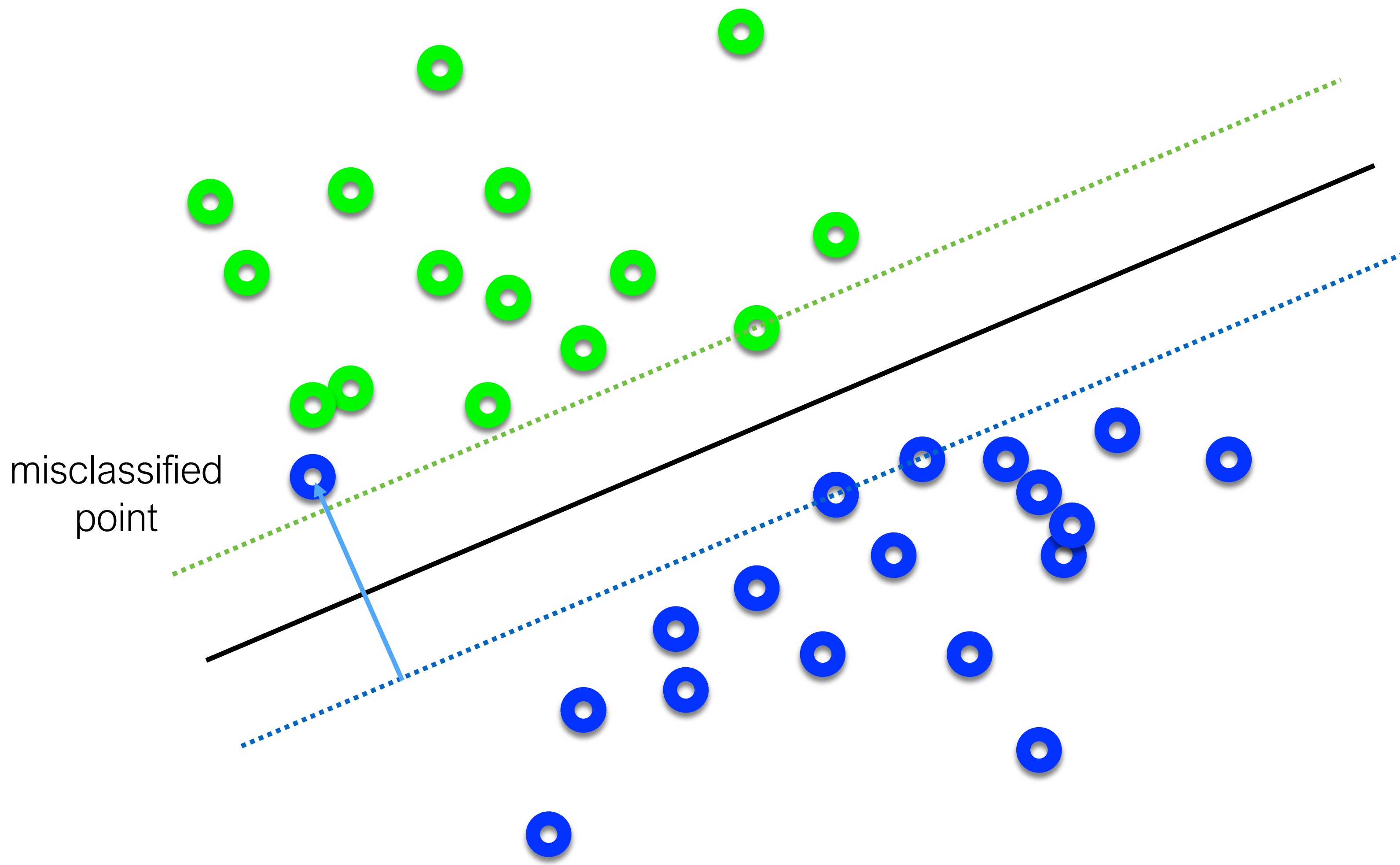
Intuitively, we should allow for some misclassification if we can get more robust classification

What's the best **w**?



Trade-off between the MARGIN and the MISTAKES
(might be a better solution)

Adding slack variables to relax the hard constraint $\xi_i \geq 0$



‘soft’ margin

objective

$$\min_{\boldsymbol{w},\boldsymbol{\xi}} \|\boldsymbol{w}\|^2 + C\sum_i \xi_i$$

subject to

$$y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) \geq 1 - \xi_i$$

for $i = 1, \ldots, N$

‘soft’ margin

objective

$$\min_{\boldsymbol{w},\boldsymbol{\xi}} \|\boldsymbol{w}\|^2 + C\sum_i \xi_i$$

subject to

$$y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) \geq 1 - \xi_i$$

for $i = 1, \ldots, N$

The slack variable allows for mistakes,
as long as the inverse margin is minimized.

‘soft’ margin

objective

$$\min_{\boldsymbol{w},\boldsymbol{\xi}} \|\boldsymbol{w}\|^2 + C\sum_i \xi_i$$

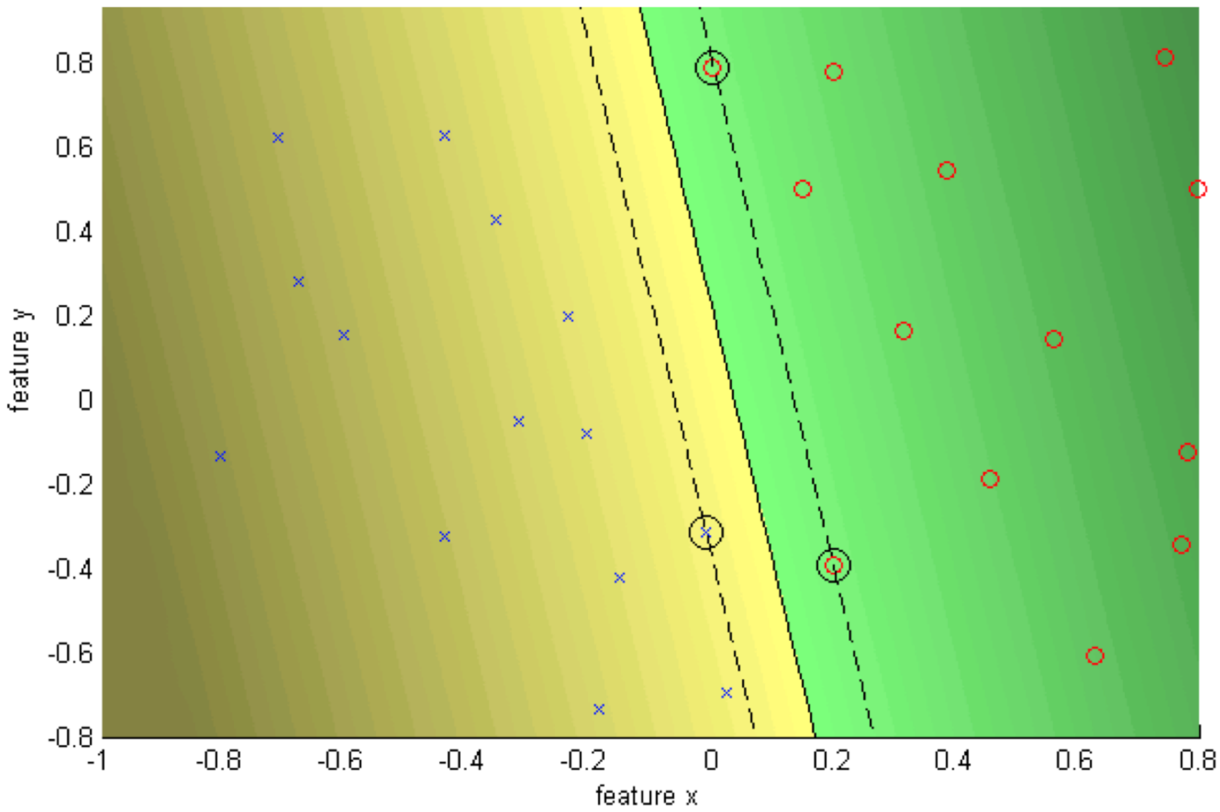
subject to

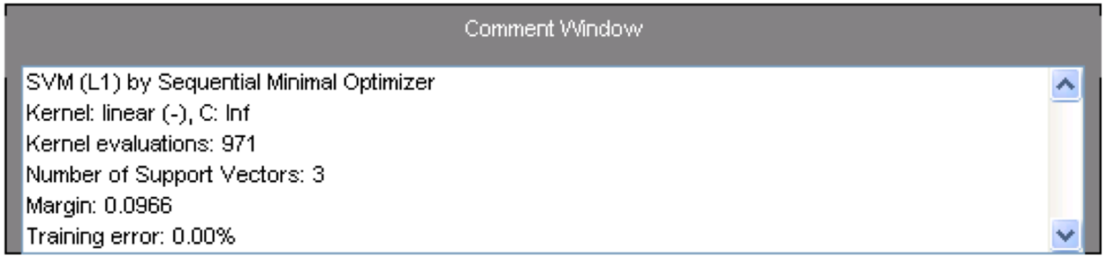
$$y_i(\boldsymbol{w}^\top \boldsymbol{x}_i + b) \geq 1 - \xi_i$$

$$\text{for} \quad i = 1, \ldots, N$$

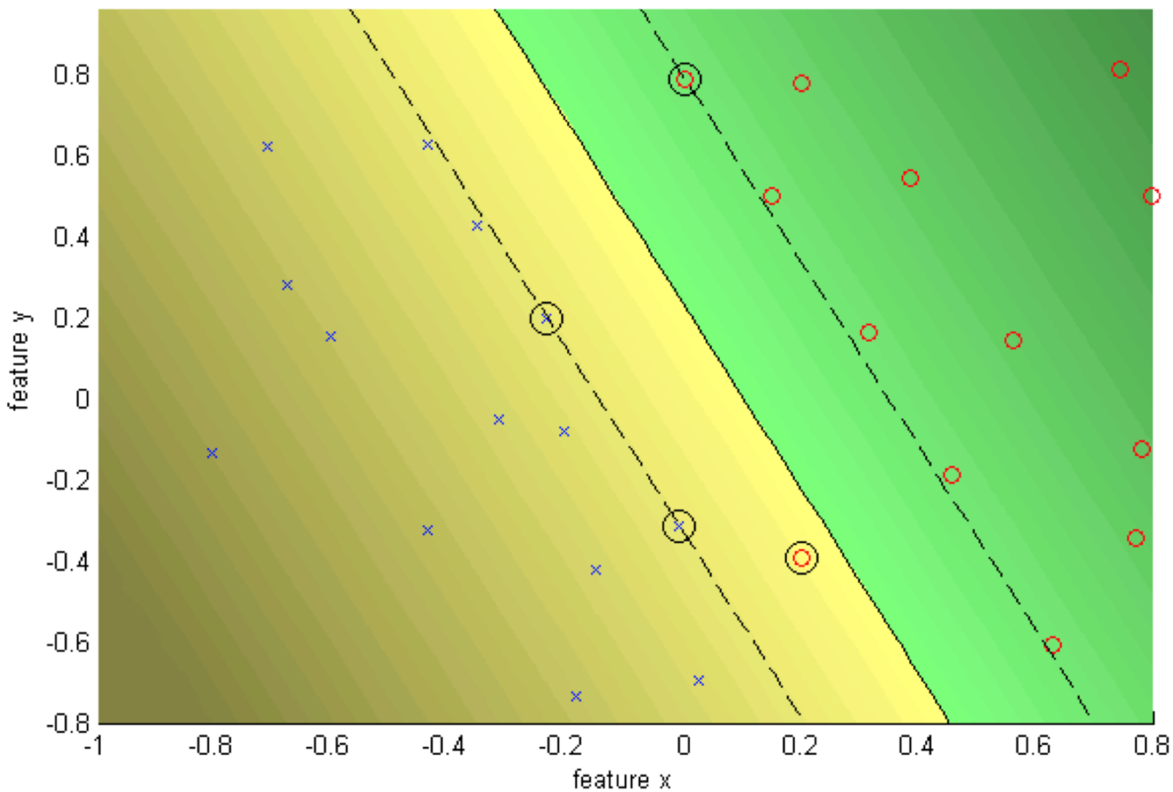
- Every constraint can be satisfied if slack is large
- C is a regularization parameter
  - Small C: ignore constraints (larger margin)
  - Big C: constraints (small margin)
- Still QP problem (unique solution)

C = Infinity   hard margin





C = 10 soft margin



Comment Window

SVM (L1) by Sequential Minimal Optimizer
Kernel: linear (-), C: 10.0000
Kernel evaluations: 2645
Number of Support Vectors: 4
Margin: 0.2265
Training error: 3.70%

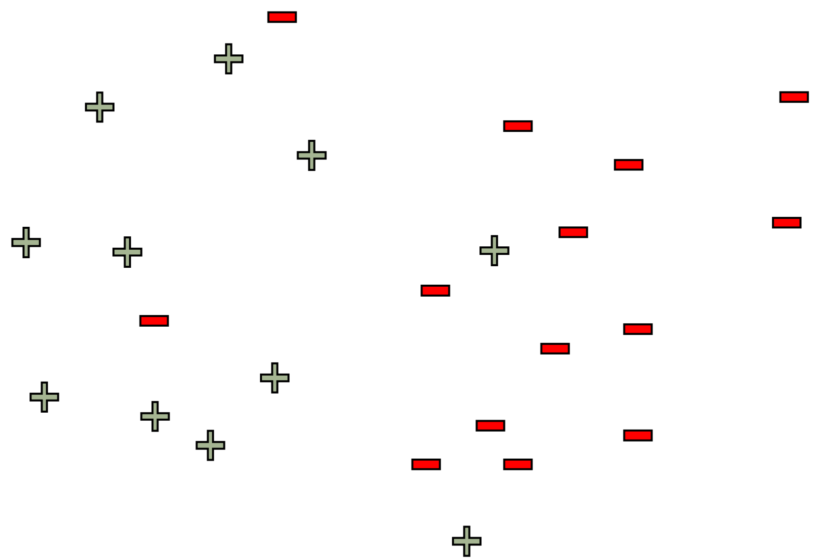
Soft Margin Hyperplane

- Linear separable:

$$\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}\|^2 \text{ subject to } r^t\left(\mathbf{w}^T\mathbf{x}^t + w_0\right) \geq +1, \forall t$$

- Not linearly separable
  - Add slack variable

$$r^t\left(\mathbf{w}^T x^t + w_0\right) \geq 1 - \xi^t$$



- Soft error $\sum_t \xi^t$
- New (primal) objective is

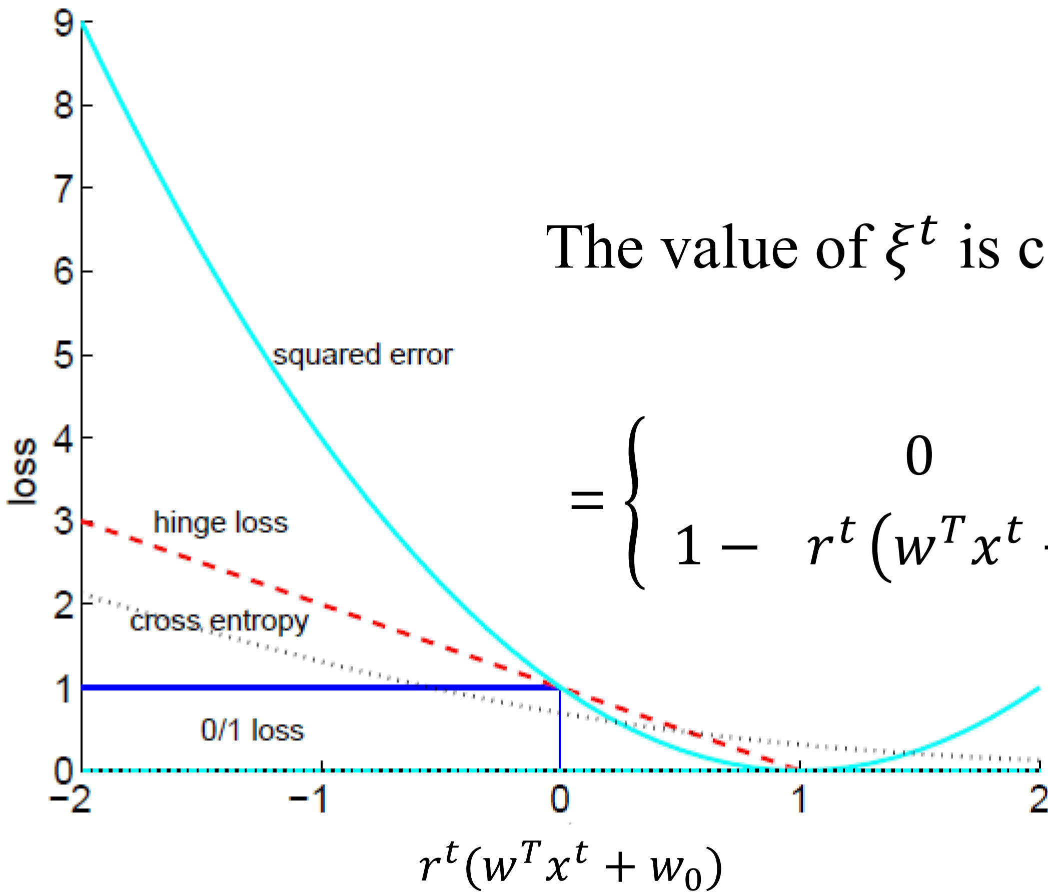
$$\min_{w, w_0, \{\xi^t\}} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_t \xi^t \quad \text{subject to} \quad r^t\left(\mathbf{w}^T x^t + w_0\right) \geq 1 - \xi^t, \quad \xi^t \geq 0$$

trade off between loss and regularization

Hinge Loss

$$\min_{w, w_0, \{\xi^t\}} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_t \xi^t \quad \text{subject to} \quad r^t\left(\mathbf{w}^T x^t + w_0\right) \geq 1 - \xi^t$$

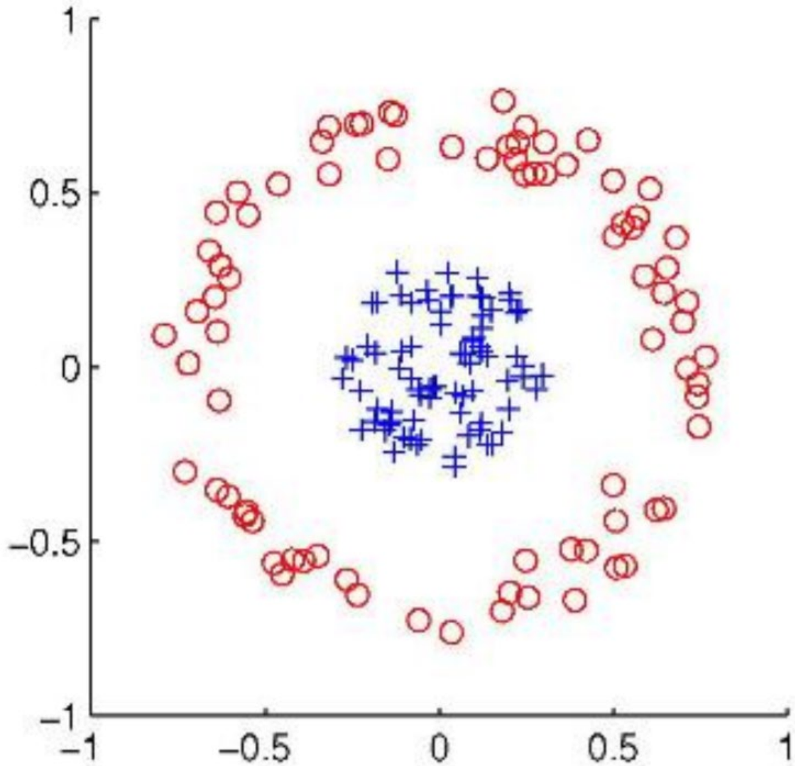
$$\xi^t \geq 0$$



The value of ξ^t is called hinge loss:

$$= \begin{cases} 0 & \text{if } r^t\left(w^T x^t + w_0\right) \geq 1 \\ 1 - r^t\left(w^T x^t + w_0\right) & \text{otherwise} \end{cases}$$

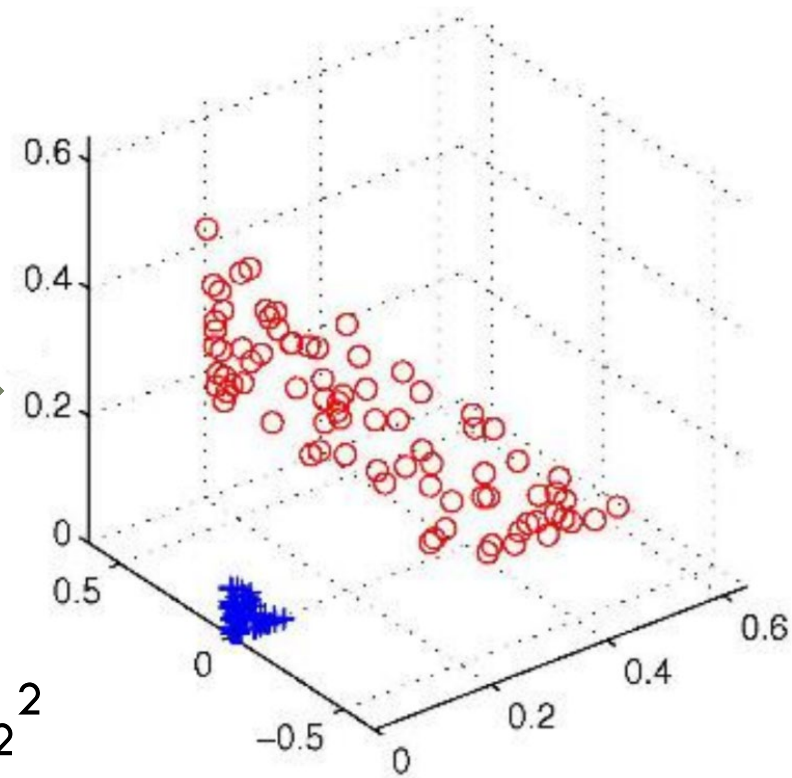
What if the data is not linearly separable?



x_1^2

x_2^2

$x_1^2 + x_2^2$



Solving the optimization

Introduce Lagrange multipliers with one multiplier α^t for each constraint

$$\min_{\mathbf{w}} \frac{1}{2}\|\mathbf{w}\|^2 \text{ subject to } r^t\left(\mathbf{w}^T\mathbf{x}^t + w_0\right) \geq +1, \forall t$$

Lagrange function

$$L_p = \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{t=1}^{N} \alpha^t \left[r^t\left(\mathbf{w}^T\mathbf{x}^t + w_0\right) - 1\right]$$

$$= \frac{1}{2}\|\mathbf{w}\|^2 - \sum_{t=1}^{N} \alpha^t r^t \left(\mathbf{w}^T\mathbf{x}^t + w_0\right) + \sum_{t=1}^{N} \alpha^t$$

$$\frac{\partial L_p}{\partial \mathbf{w}} = 0 \Rightarrow \boxed{\mathbf{w} = \sum_{t=1}^{N} \alpha^t r^t \mathbf{x}^t}$$

$$\frac{\partial L_p}{\partial w_0} = 0 \Rightarrow \sum_{t=1}^{N} \alpha^t r^t = 0$$

https://en.wikipedia.org/wiki/Lagrange_multiplier

Solving the optimization

$$L_d = \frac{1}{2}\left(\mathbf{w}^T\mathbf{w}\right) - \mathbf{w}^T \sum_t \alpha^t r^t \mathbf{x}^t - w_0 \sum_t \alpha^t r^t + \sum_t \alpha^t$$

$$= -\frac{1}{2}\left(\mathbf{w}^T\mathbf{w}\right) + \sum_t \alpha^t$$

$$\boxed{\mathbf{w} = \sum_{t=1}^{N} \alpha^t r^t \mathbf{x}^t}$$

$$= -\frac{1}{2}\sum_t \sum_s \alpha^t \alpha^s r^t r^s \left(\mathbf{x}^t\right)^T \mathbf{x}^s + \sum_t \alpha^t$$

subject to $\sum_t \alpha^t r^t = 0$ and $\alpha^t \geq 0, \forall t$

Sparsity:

- Most α^t are 0, and
- Only a small number have $\alpha^t > 0$ (they are the support vectors)

Kernel Trick

- Preprocess input **x** by basis functions

$$\boldsymbol{z} = \boldsymbol{\varphi}(\boldsymbol{x}) \qquad g(\boldsymbol{z}) = \boldsymbol{w}^T\boldsymbol{z}$$

$$g(\boldsymbol{x}) = \boldsymbol{w}^T \boldsymbol{\varphi}(\boldsymbol{x})$$

- The SVM solution

$$\mathbf{w} = \sum_t \alpha^t r^t \mathbf{z}^t = \sum_t \alpha^t r^t \boldsymbol{\varphi}\left(\mathbf{x}^t\right)$$

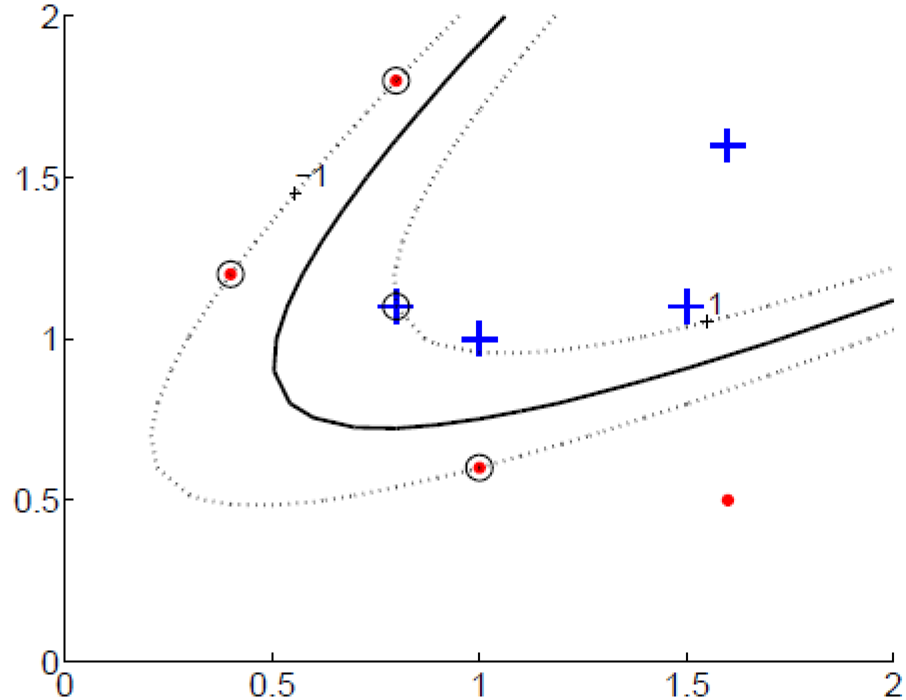
$$g(\mathbf{x}) = \mathbf{w}^T \boldsymbol{\varphi}(\mathbf{x}) = \sum_t \alpha^t r^t \boxed{\boldsymbol{\varphi}\left(\mathbf{x}^t\right)^T \boldsymbol{\varphi}(\mathbf{x})}$$

$$g(\mathbf{x}) = \sum_t \alpha^t r^t \boxed{K\left(\mathbf{x}^t, \mathbf{x}\right)}$$

Vectorial Kernels

- Polynomials of degree q:

$$K\left(\mathbf{x}^t, \mathbf{x}\right) = \left(\mathbf{x}^T \mathbf{x}^t + 1\right)^q$$



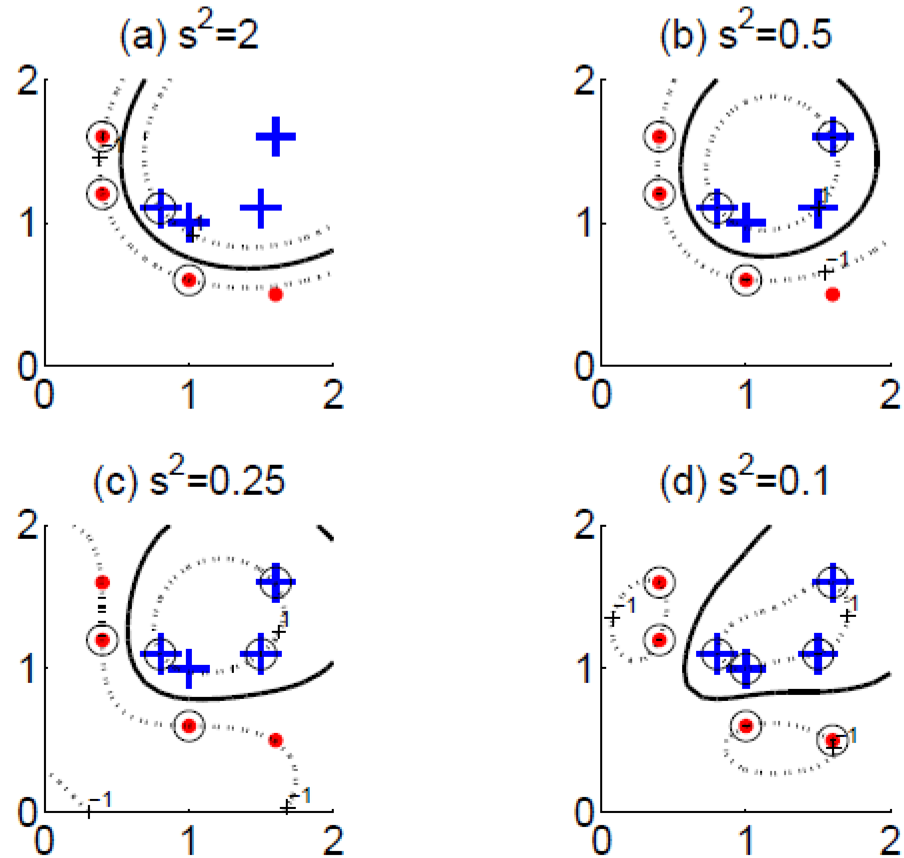
$$\begin{aligned} K(\mathbf{x}, \mathbf{y}) &= \left(\mathbf{x}^T \mathbf{y} + 1\right)^2 \\ &= \left(x_1 y_1 + x_2 y_2 + 1\right)^2 \\ &= 1 + 2x_1 y_1 + 2x_2 y_2 + 2x_1 x_2 y_1 y_2 + x_1^2 y_1^2 + x_2^2 y_2^2 \end{aligned}$$

$$\phi(\mathbf{x}) = \left[1, \sqrt{2}x_1, \sqrt{2}x_2, \sqrt{2}x_1 x_2, x_1^2, x_2^2\right]^T$$

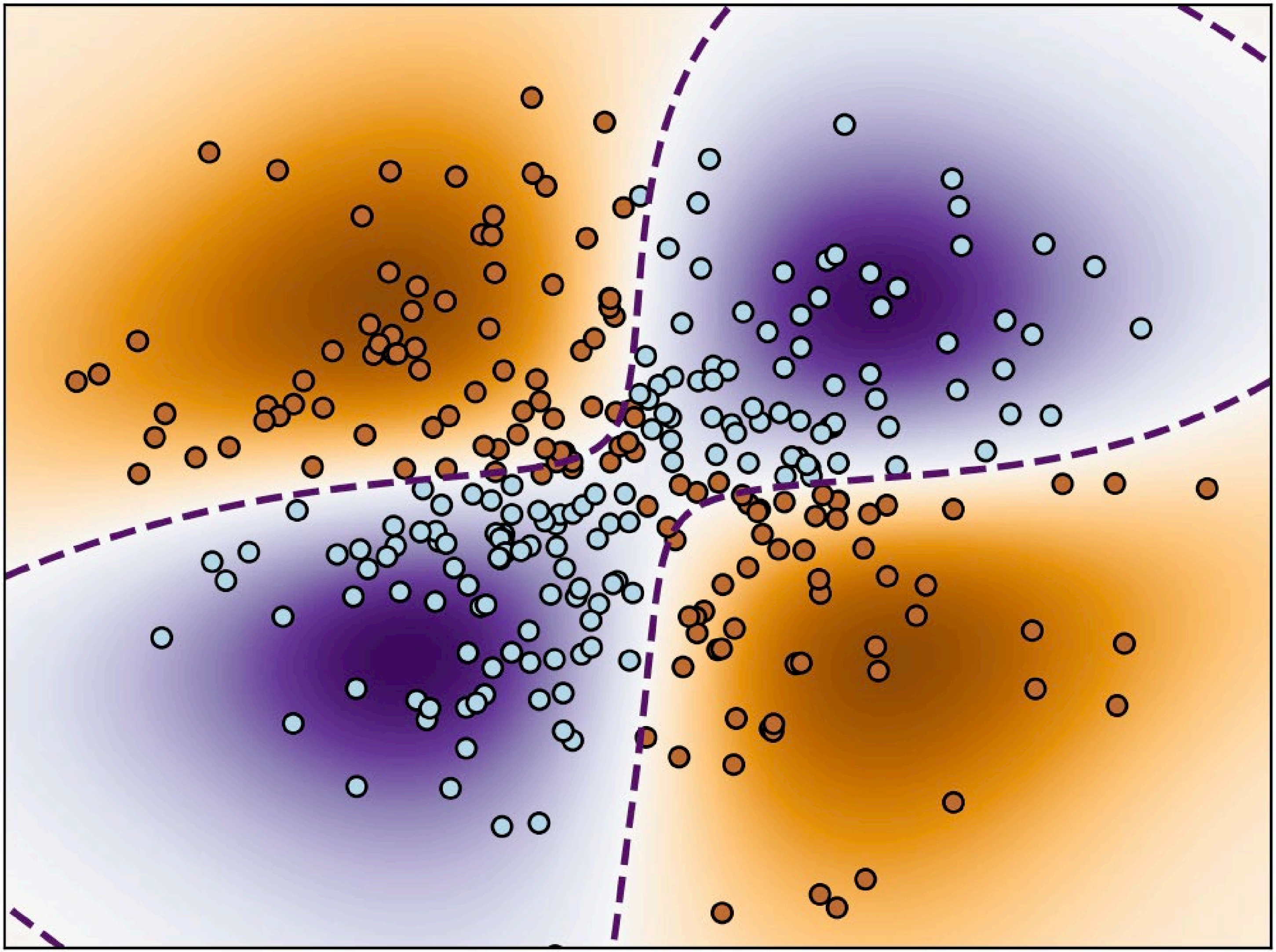
Vectorial Kernels

- Radial-basis functions:

$$K\left(\mathbf{x}^t, \mathbf{x}\right) = \exp\left[-\frac{\left\|\mathbf{x}^t - \mathbf{x}\right\|^2}{2s^2}\right]$$



```python
np.random.seed(0)
X = np.random.randn(300, 2)
Y = np.logical_xor(X[:, 0] > 0, X[:, 1] > 0)

# fit the model
clf = svm.NuSVC(gamma='auto')
clf.fit(X, Y)
```



Support Vector Machine Summary

- Margin-based classification
- Slack variables and hinge loss
- Sparse (depends on only some of the data)
- Losses: 0/1 vs. Hinge vs. Log loss (logistic reg.)
- Nonlinear boundary through nonlinear kernels