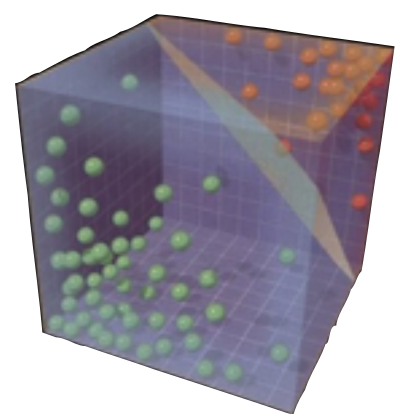
Classification

16-385 Computer Vision (Kris Kitani)

Carnegie Mellon University

typical perception pipeline

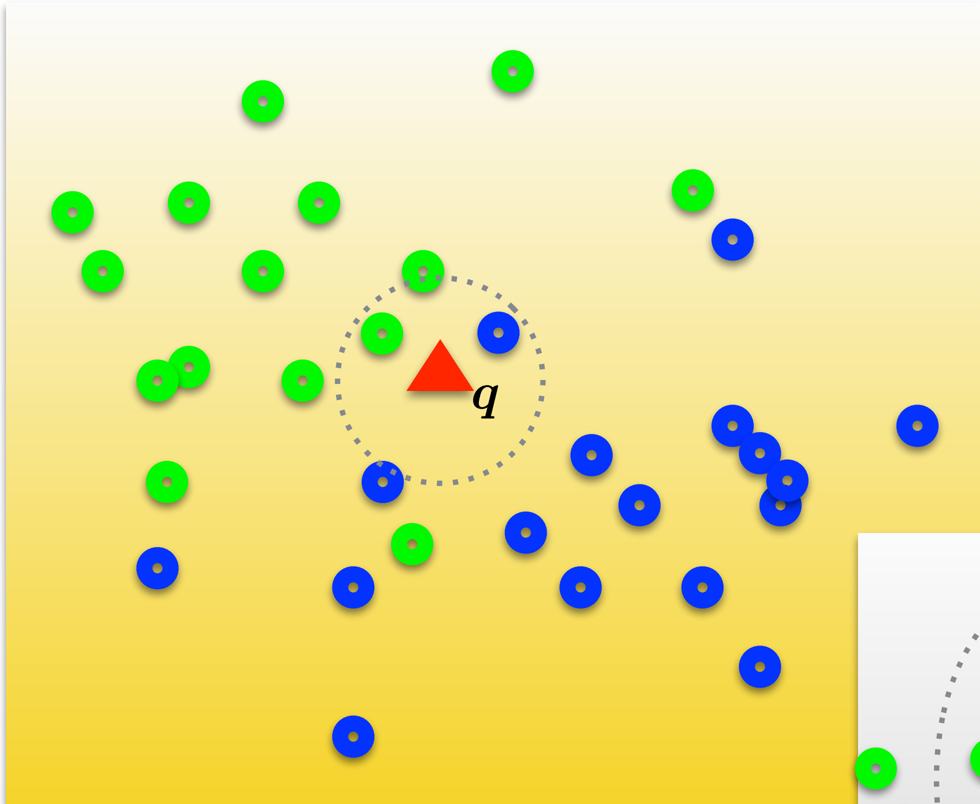
representation



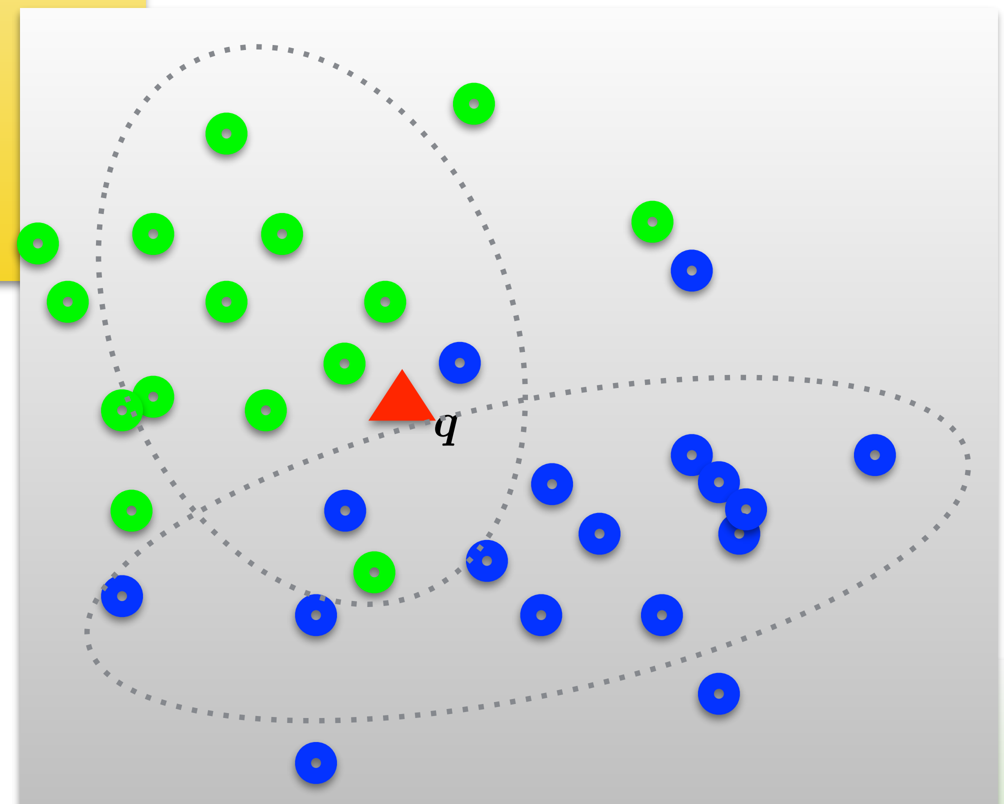
classifier

- Nearest Neighbor classifier
- Naive Bayes classifier
- Support Vector Machine

output

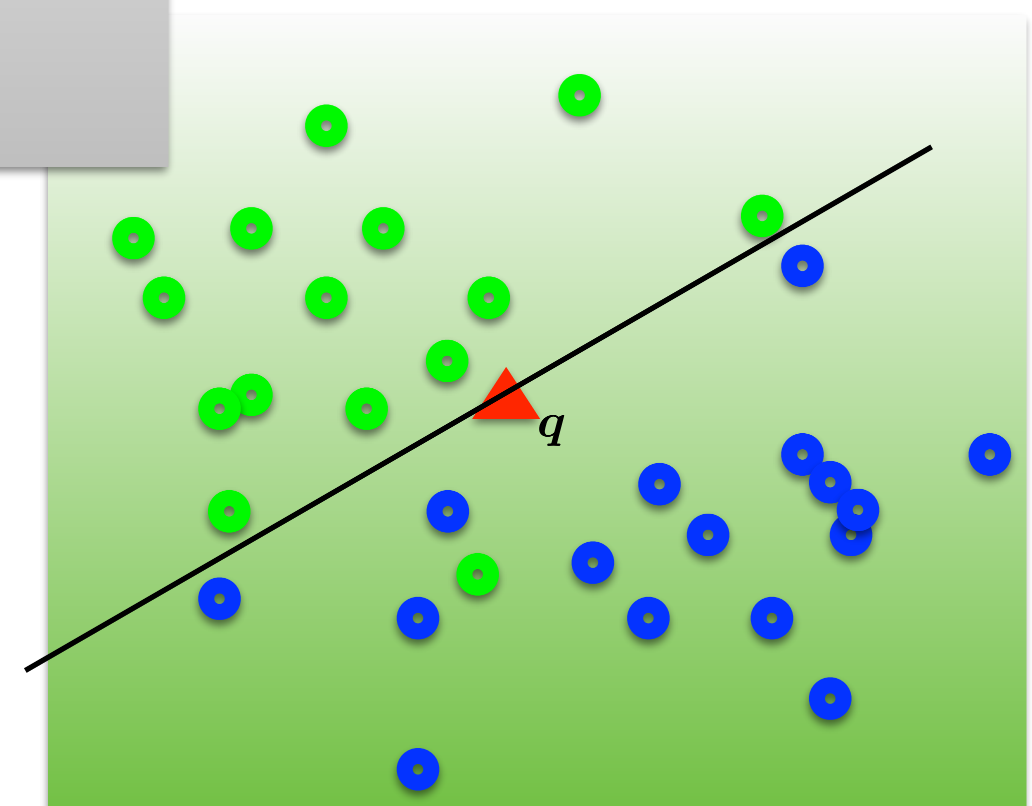


Nearest Neighbor

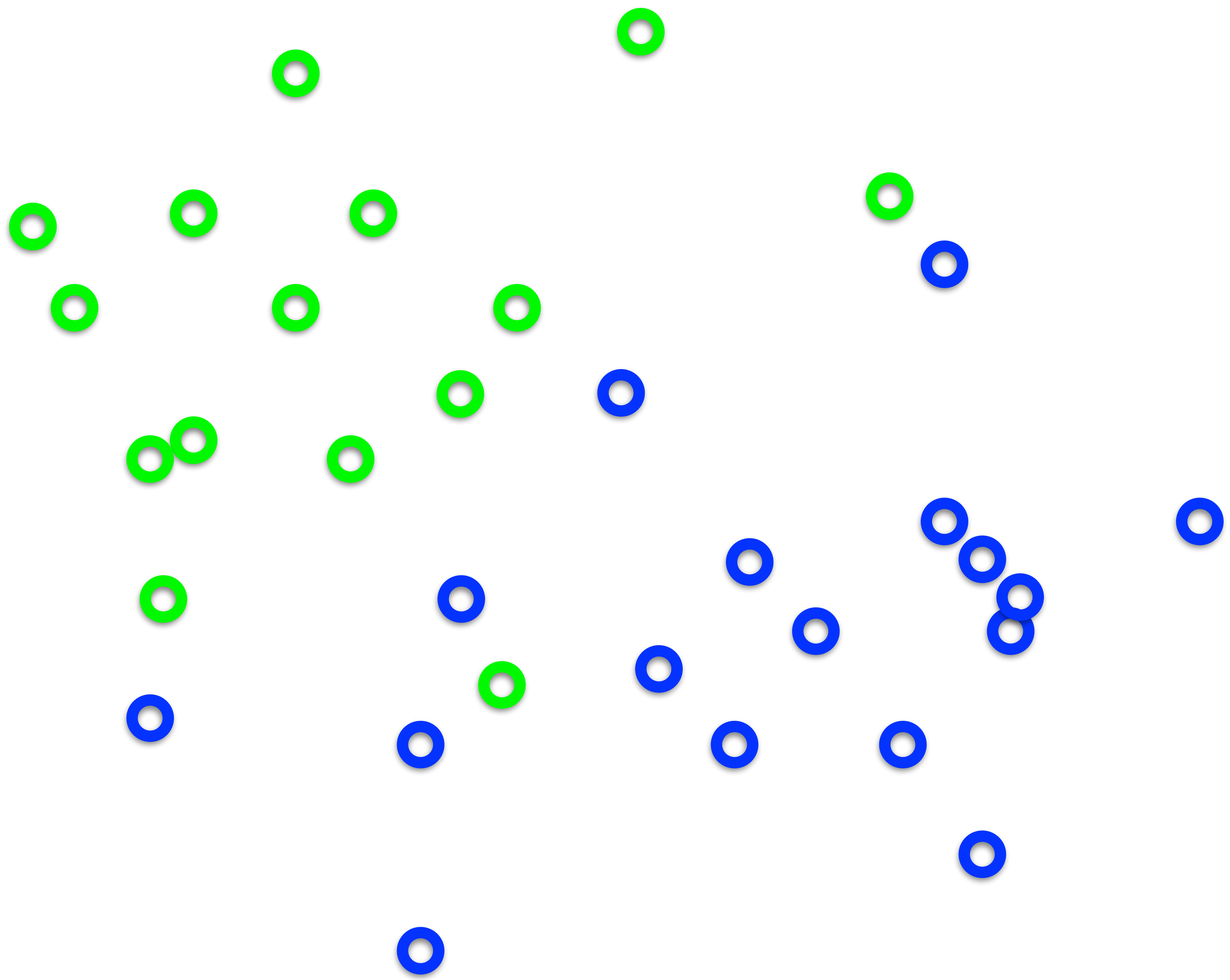


Naive Bayes

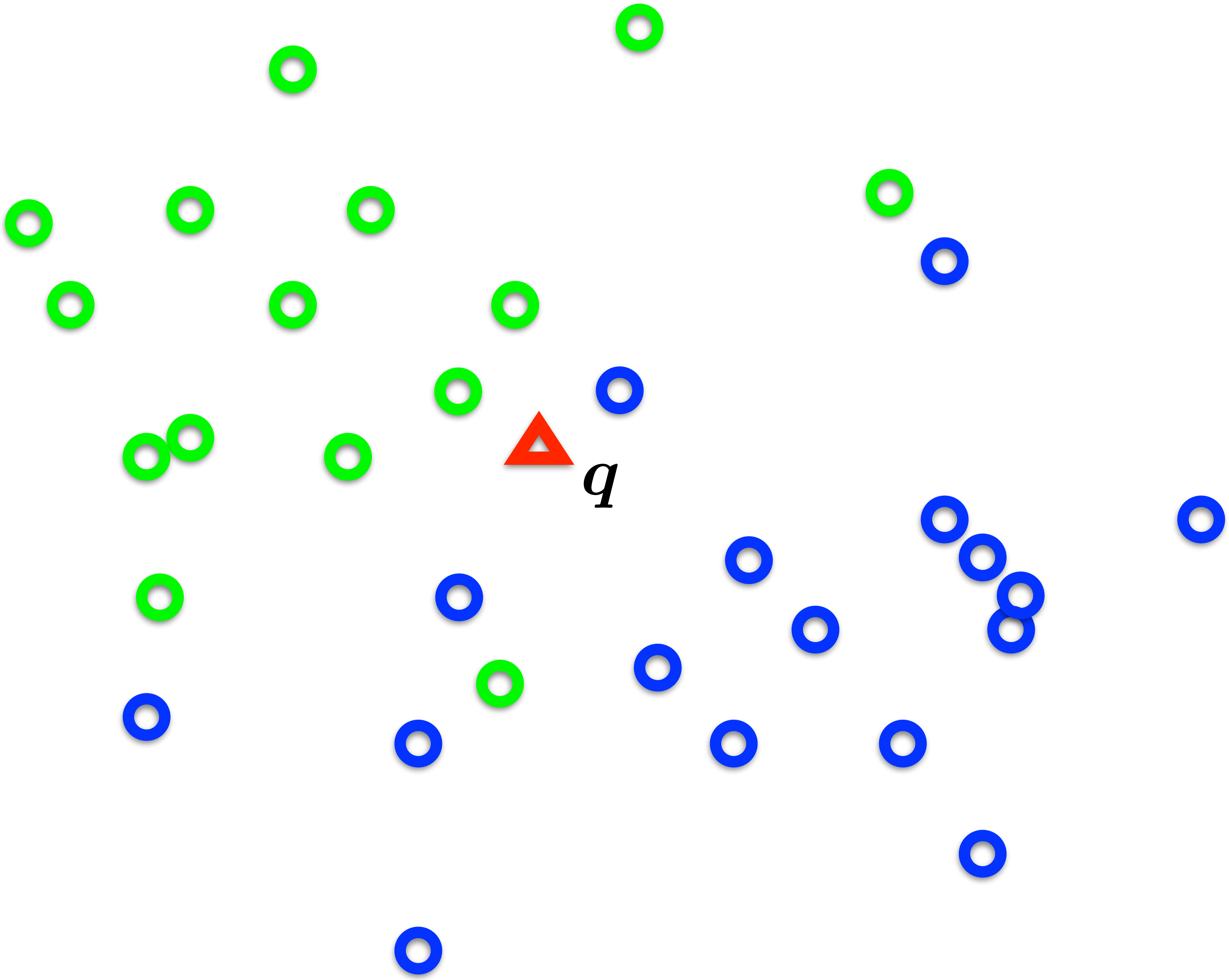
Support Vector Machine



Distribution of data from two classes

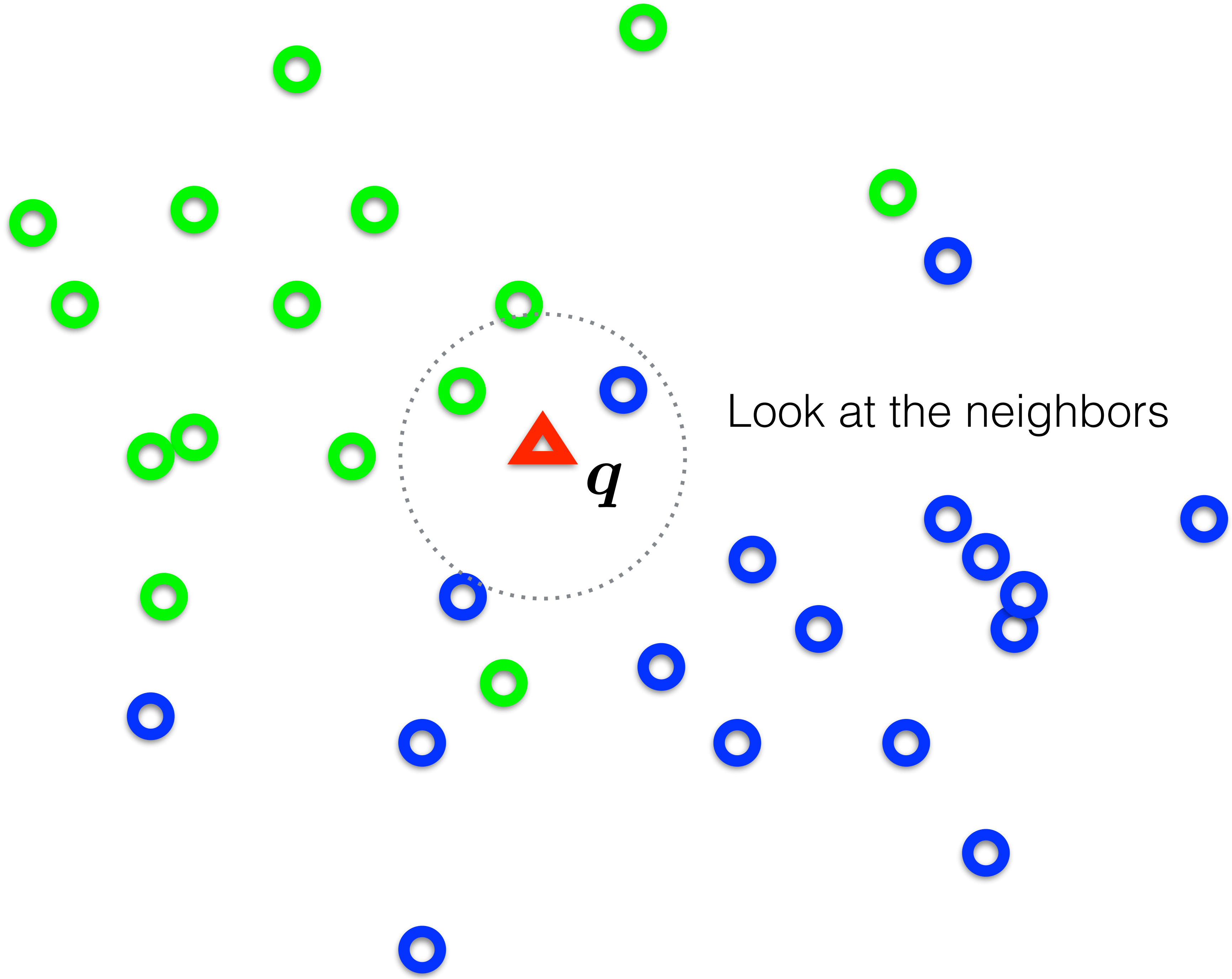


Distribution of data from two classes



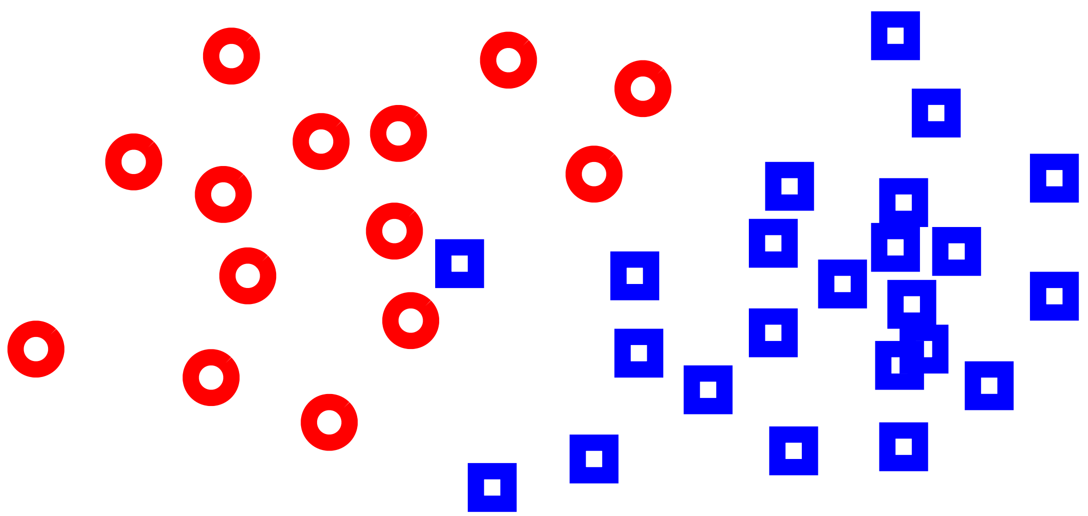
Which class does q belong too?

Distribution of data from two classes



K-nearest neighbor

K-Nearest Neighbor (KNN) Classifier

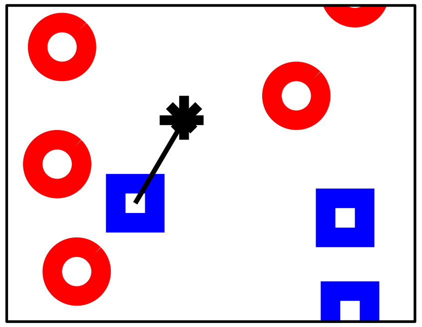


Non-parametric pattern classification approach

Consider a two class problem where each sample consists of two measurements (x,y).

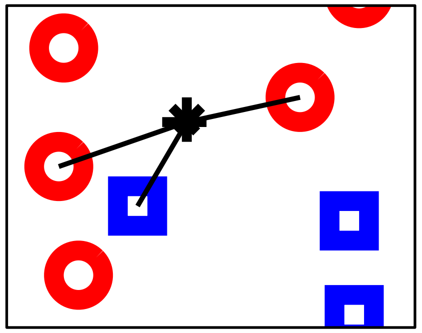
For a given query point q, assign the class of the nearest neighbor

k = 1



Compute the k nearest neighbors and assign the class by majority vote.

k = 3



Nearest Neighbor is competitive

MNIST Digit Recognition

- Handwritten digits
- 28x28 pixel images: d = 784
- 60,000 training samples
- 10,000 test samples

Yann LeCunn

	Test Error Rate (%)
Linear classifier (1-layer NN)	12.0
K-nearest-neighbors, Euclidean	5.0
K-nearest-neighbors, Euclidean, deskewed	2.4
K-NN, Tangent Distance, 16x16	1.1
K-NN, shape context matching	0.67
1000 RBF + linear classifier	3.6
SVM deg 4 polynomial	1.1
2-layer NN, 300 hidden units	4.7
2-layer NN, 300 HU, [deskewing]	1.6
LeNet-5, [distortions]	0.8
Boosted LeNet-4, [distortions]	0.7

Pros

- simple yet effective

Cons

- search is expensive (can be sped-up)
- storage requirements
- difficulties with high-dimensional data

What is the best distance metric between data points?

- Typically Euclidean distance
- Locality sensitive distance metrics
- Important to normalize.
  Dimensions have different scales

How many K?

- Typically k=1 is good
- Cross-validation

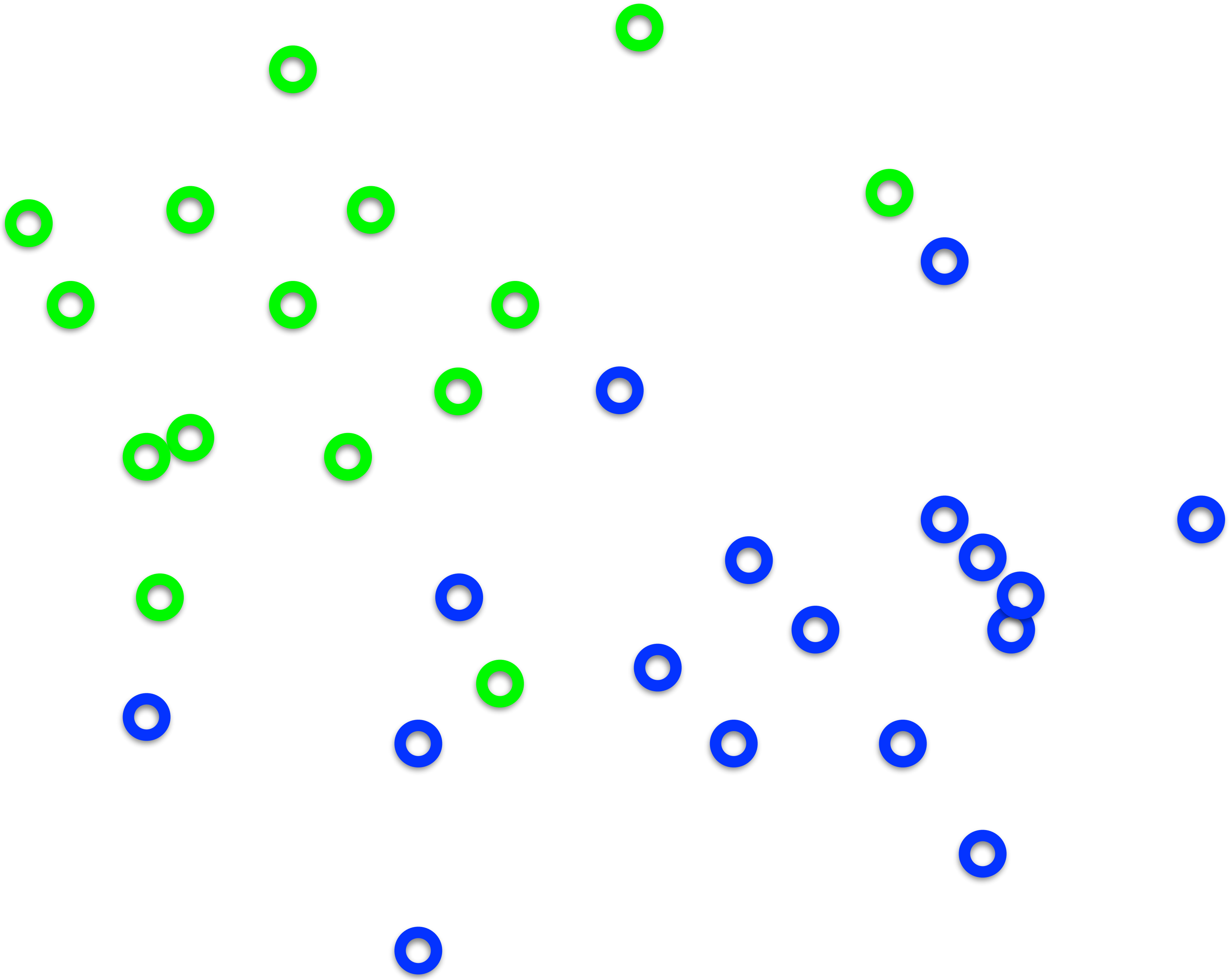
Distance metrics

$$D(\boldsymbol{x}, \boldsymbol{y}) = \sqrt{(x_1 - y_1)^2 + \cdots + (x_N - y_N)^2}$$ Euclidean

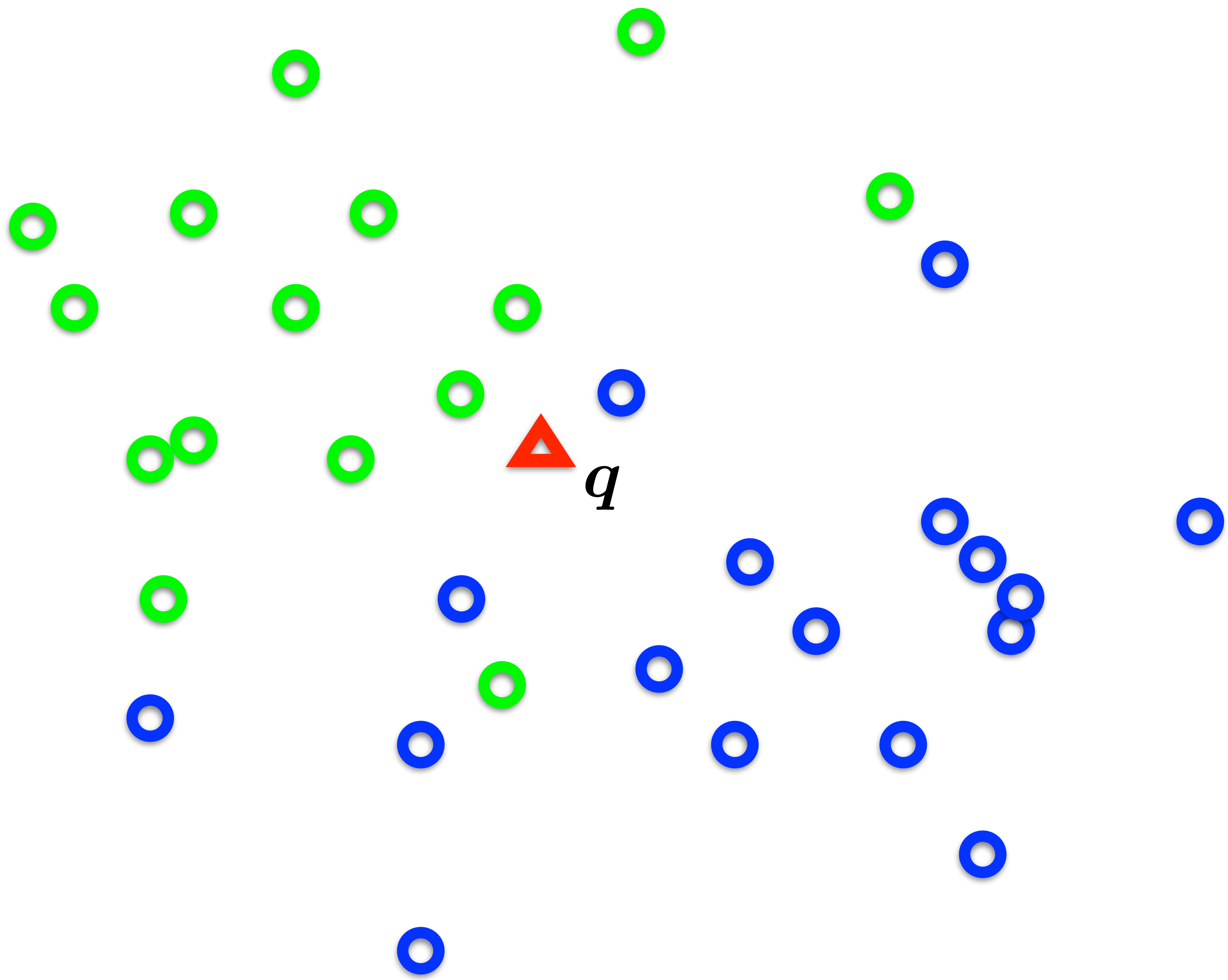
$$D(\boldsymbol{x}, \boldsymbol{y}) = \frac{\boldsymbol{x} \cdot \boldsymbol{y}}{\|\boldsymbol{x}\| \|\boldsymbol{y}\|} = \frac{x_1 y_1 + \cdots + x_N y_N}{\sqrt{\sum_n x_n^2} \sqrt{\sum_n y_n^2}}$$ Cosine

$$D(\boldsymbol{x}, \boldsymbol{y}) = \frac{1}{2} \sum_n \frac{(x_n - y_n)^2}{(x_n + y_n)}$$ Chi-squared

Distribution of data from two classes



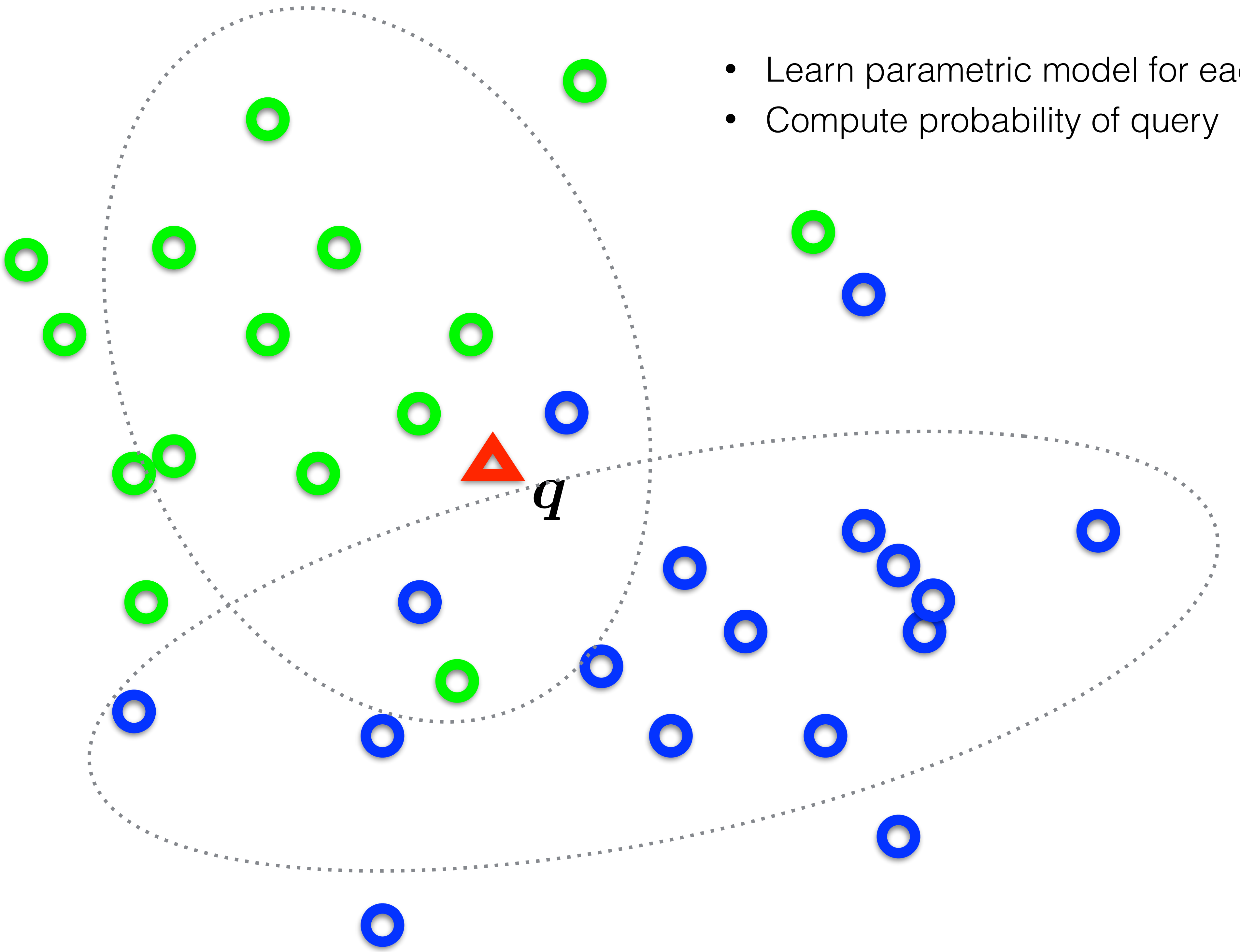
Distribution of data from two classes



Which class does q belong too?

Distribution of data from two classes

- Learn parametric model for each class
- Compute probability of query



Naive Bayes

This is called the posterior:

the probability of a class z given the observed features X

$$p(\boldsymbol{z}|\boldsymbol{X})$$

For classification, z is a discrete random variable
(e.g., car, person, building)

X is a set of observed feature
(e.g., features from a single image)

(it's a function that returns a single probability value)

This is called the posterior:

the probability of a class z given the observed features X

$$p(\boldsymbol{z}|\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N)$$

For classification, z is a discrete random variable
(e.g., car, person, building)

Each x is an observed feature
(e.g., visual words)

(it's a function that returns a single probability value)

Recall:

The posterior can be decomposed according to **Bayes' Rule**

$$\underbrace{p(A|B)}_{\text{posterior}} = \frac{\overbrace{p(B|A)}^{\text{likelihood}}\overbrace{p(A)}^{\text{prior}}}{p(B)}$$

In our context…

$$p(\boldsymbol{z}|\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N) = \frac{p(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N|\boldsymbol{z})p(\boldsymbol{z})}{p(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N)}$$

The naive Bayes' classifier is solving this optimization

$$\hat{z} = \underset{z \in \boldsymbol{Z}}{\arg\max}\, p(z|\boldsymbol{X})$$

MAP (maximum a posteriori) estimate

$$\hat{z} = \underset{z \in \boldsymbol{Z}}{\arg\max}\, \frac{p(\boldsymbol{X}|z)p(z)}{p(\boldsymbol{X})}$$

Bayes' Rule

$$\hat{z} = \underset{z \in \boldsymbol{Z}}{\arg\max}\, p(\boldsymbol{X}|z)p(z)$$

Remove constants

To optimize this…

$$\hat{z} = \arg\max_{z \in \boldsymbol{z}} p(z|\boldsymbol{X})$$

We need to compute this

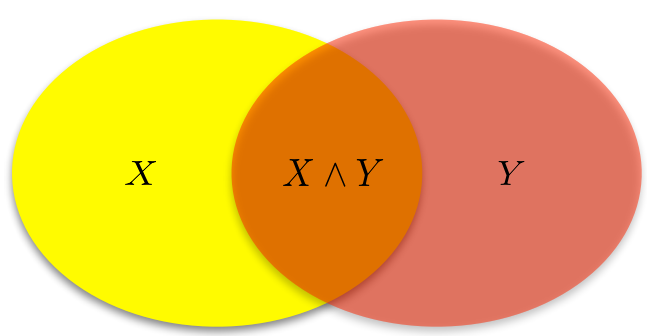
$$p(\boldsymbol{z}|\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N) = \frac{p(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N|\boldsymbol{z})p(\boldsymbol{z})}{p(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N)}$$

Compute the likelihood…

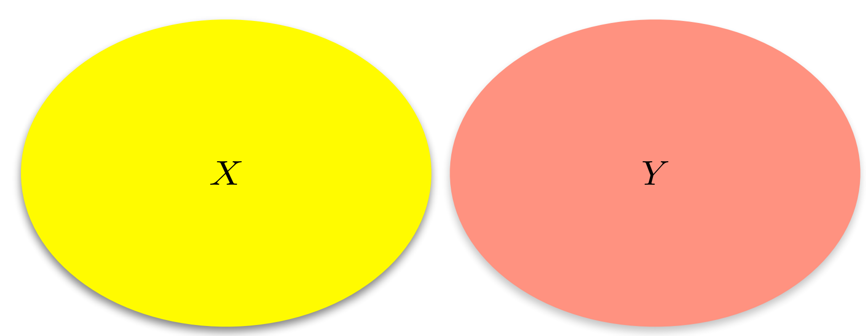
A naive Bayes' classifier assumes all features are ***conditionally independent***

$$
\begin{aligned}
p(\boldsymbol{x}_1, \ldots, \boldsymbol{x}_N | \boldsymbol{z}) &= p(\boldsymbol{x}_1|\boldsymbol{z}) p(\boldsymbol{x}_2, \ldots, \boldsymbol{x}_N|\boldsymbol{z}) \\
&= p(\boldsymbol{x}_1|\boldsymbol{z}) p(\boldsymbol{x}_2|\boldsymbol{z}) p(\boldsymbol{x}_3, \ldots, \boldsymbol{x}_N|\boldsymbol{z}) \\
&= p(\boldsymbol{x}_1|\boldsymbol{z}) p(\boldsymbol{x}_2|\boldsymbol{z}) \cdots p(\boldsymbol{x}_N|\boldsymbol{z})
\end{aligned}
$$

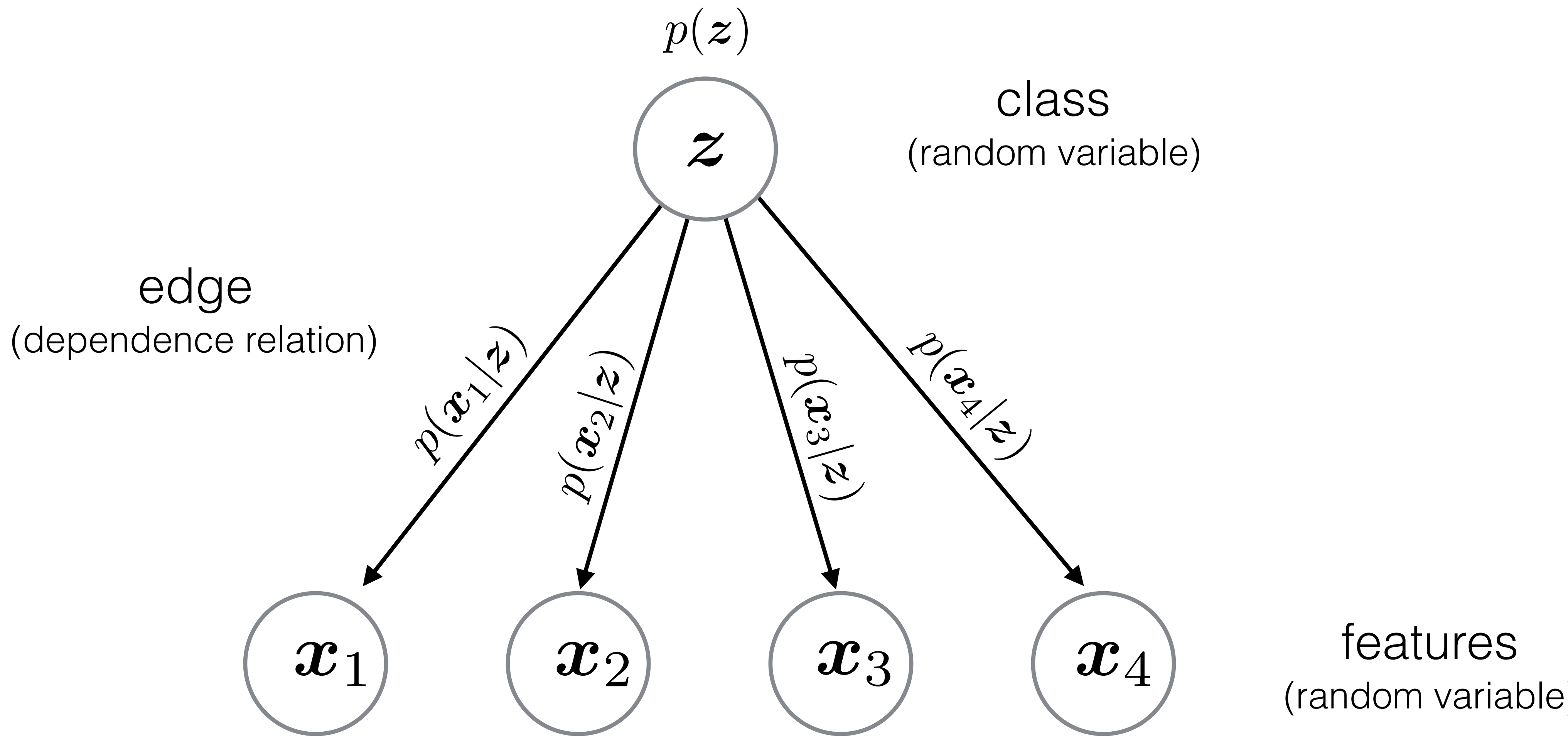
Recall:



$$p(x, y) = p(x|y)p(y)$$



$$p(x, y) = p(x)p(y)$$



Graphical model visualization

To compute the MAP estimate

Given (1) a set of known parameters

$$p(\boldsymbol{z}) \quad p(\boldsymbol{x}|\boldsymbol{z})$$

(2) observations

$$\{x_1, x_2, \ldots, x_N\}$$

Compute which z has the largest probability

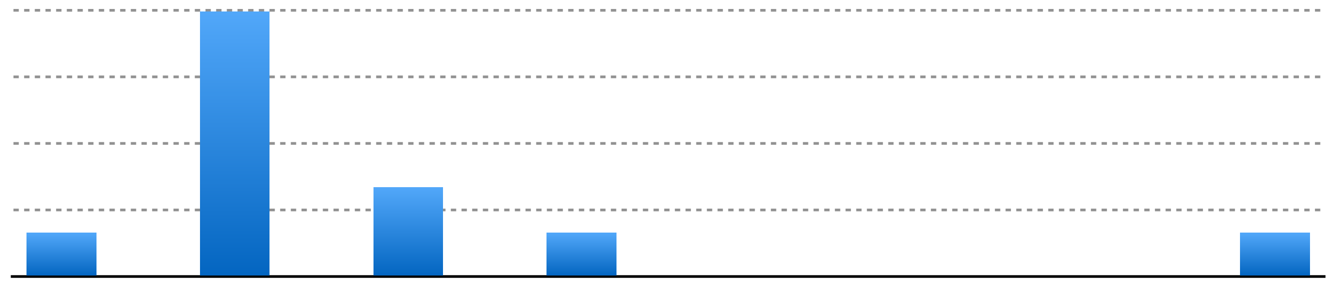
$$\hat{z} = \arg\max_{z \in \boldsymbol{z}} p(z) \prod_n p(x_n|z)$$

The Newspa

Sunday, December 22, 2013

DARPA Selects Carnegie Me

The Tartan Rescue Team from Carnegie Mellon University's National Robotics Engineering Center ranked third among teams competing in the Defense Advanced Research Projects Agency (DARPA) Robotics Challenge Trials this weekend in Homestead, Fla., and was selected by the agency as one of eight teams eligible for DARPA funding to prepare for next December's finals. The team's four-limbed CMU Highly Intelligent Mobile Platform, or CHIMP, robot scored 18 out of a possible 32 points during the two-day trials. It demonstrated its ability to perform such tasks as removing debris, cutting a hole through a wall and closing a series of valves.



count	1	6	2	1	0	0	0	1
word	Tartan	robot	CHIMP	CMU	bio	soft	ankle	sensor
p(x|z)	0.09	0.55	0.18	0.09	0.0	0.0	0.0	0.09

$$p(X|z) = \prod_v p(x_v|z)^{c(w_v)}$$

$$= (0.09)^1 (0.55)^6 \cdots (0.09)^1$$

Numbers get really small so use log probabilities

$$\log p(X|z = \text{`grandchallenge'}) = -2.42 - 3.68 - 3.43 - 2.42 - 0.07 - 0.07 - 0.07 - 2.42 = -14.58$$

$$\log p(X|z = \text{`softrobot'}) = -7.63 - 9.37 - 15.18 - 2.97 - 0.02 - 0.01 - 0.02 - 2.27 = -37.48$$

* typically add pseudo-counts (0.001)

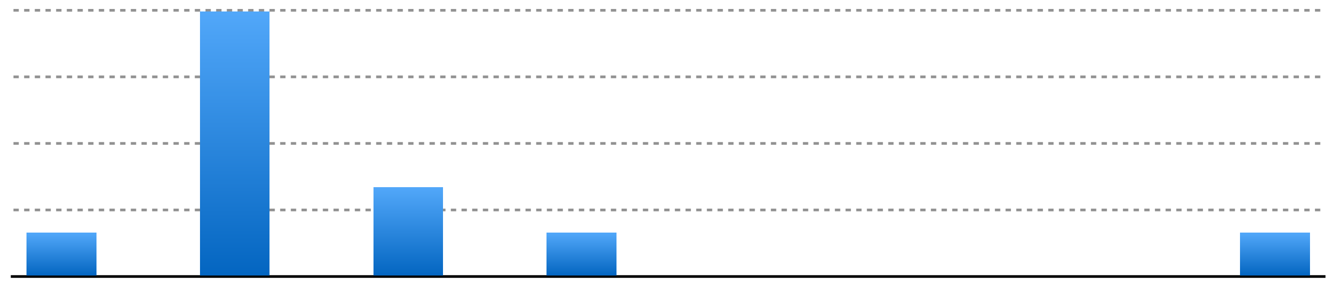
** this is an example for computing the likelihood, need to multiply times **prior** to get posterior


The Newspa

Sunday, December 22, 2013

DARPA Selects Carnegie Me

The Tartan Rescue Team from Carnegie Mellon University's National Robotics Engineering Center ranked third among teams competing in the Defense Advanced Research Projects Agency (DARPA) Robotics Challenge Trials this weekend in Homestead, Fla., and was selected by the agency as one of eight teams eligible for DARPA funding to prepare for next December's finals. The team's four-limbed CMU Highly Intelligent Mobile Platform, or CHIMP, robot scored 18 out of a possible 32 points during the two-day trials. It demonstrated its ability to perform such tasks as removing debris, cutting a hole through a wall and closing a series of valves.



count	1	6	2	1	0	0	0	1
word	Tartan	robot	CHIMP	CMU	bio	soft	ankle	sensor
p(xlz)	0.09	0.55	0.18	0.09	0.0	0.0	0.0	0.09

log p(X|z=grand challenge) = **- 14.58**

log p(X|z=bio inspired) = - 37.48

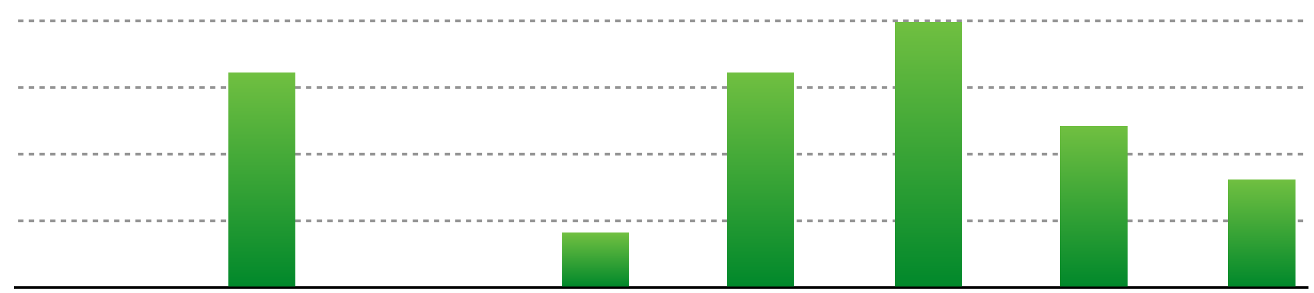

Tartan Tim

Monday, January 20, 2014

Bio-Inspired Robotic Device

PITTSBURGH—A soft, wearable device that mimics the muscles, tendons and ligaments of the lower leg could aid in the rehabilitation of patients with ankle-foot disorders such as drop foot, said Yong-Lae Park, an assistant professor of robotics at Carnegie Mellon University. Park, working with collaborators at Harvard University, the University of Southern California, MIT and BioSensics, developed an active orthotic device using soft plastics and composite materials, instead of a rigid exoskeleton. The soft materials, combined with pneumatic artificial muscles (PAMs), lightweight sensors and advanced control software, made it possible for the robotic device to achieve natural motions in the ankle.

http://www.fodey.com/generators/newspaper/snippet.asp



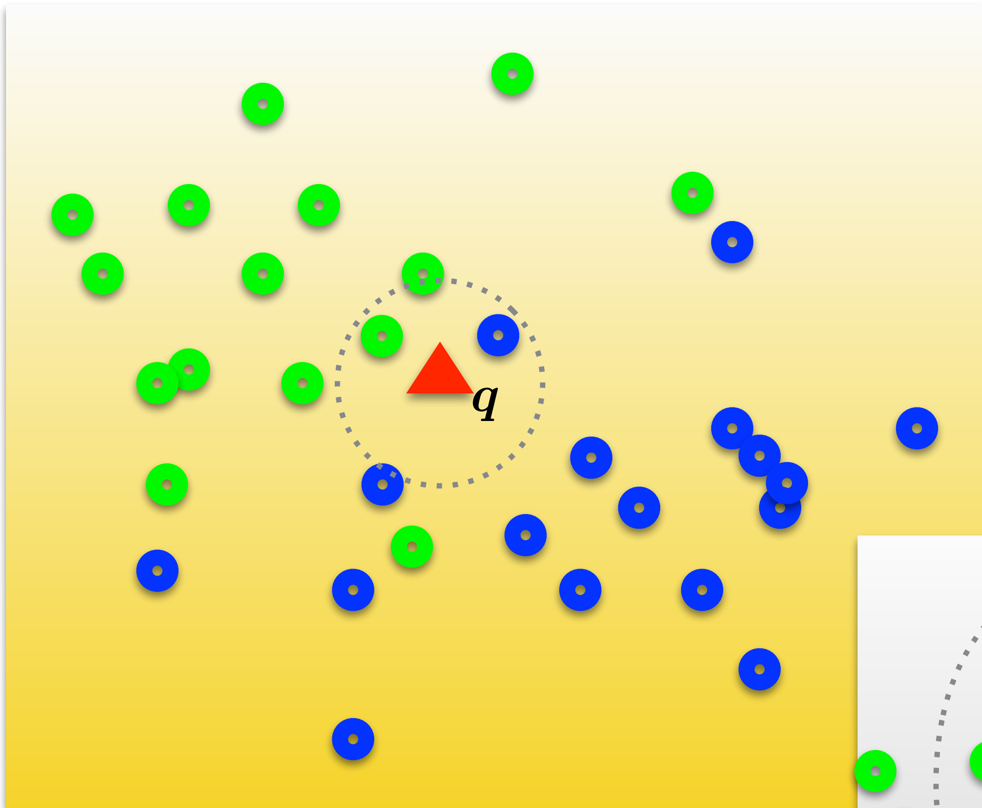
count	0	4	0	1	4	5	3	2
word	Tartan	robot	CHIMP	CMU	bio	soft	ankle	sensor
p(xlz)	0.0	0.21	0.0	0.05	0.21	0.26	0.16	0.11

log p(X|z=grand challenge) = - 94.06

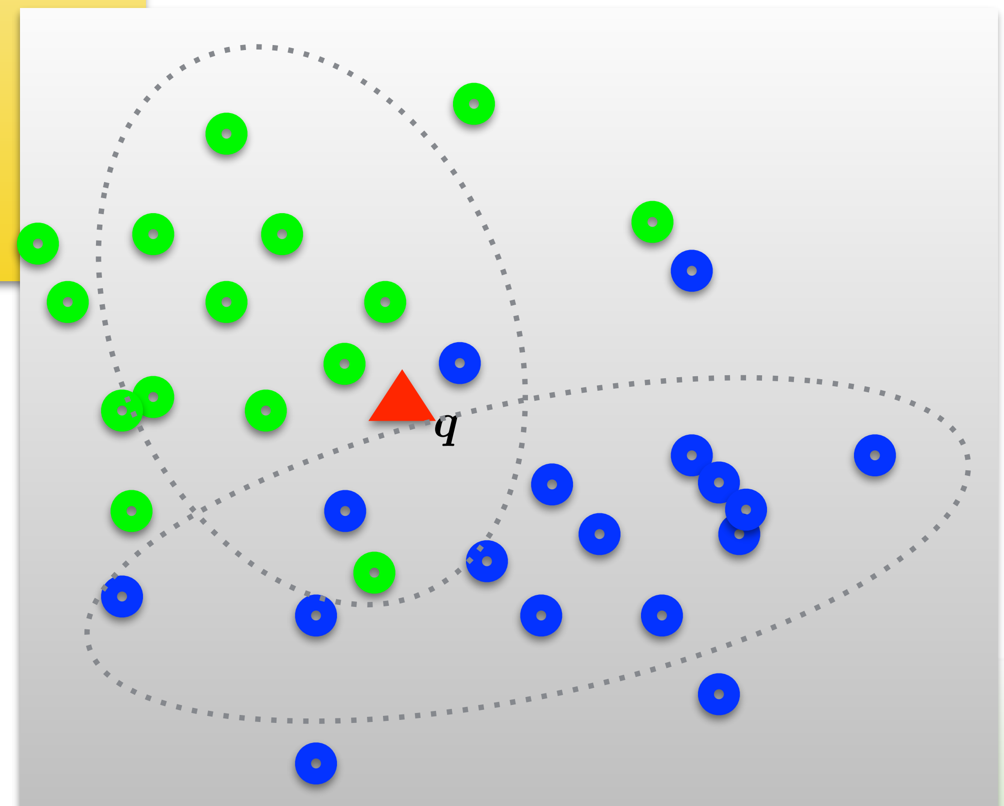
log p(X|z=bio inspired) = **- 32.41**

* typically add pseudo-counts (0.001)

** this is an example for computing the likelihood, need to multiply times prior to get posterior

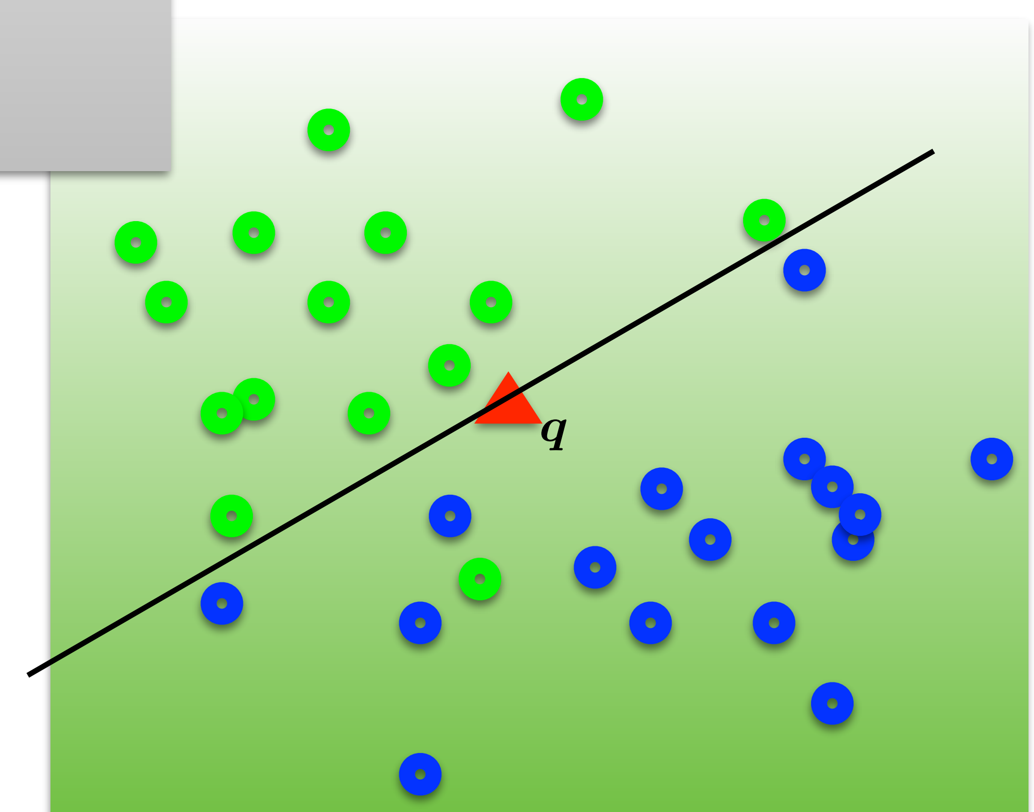


Nearest Neighbor

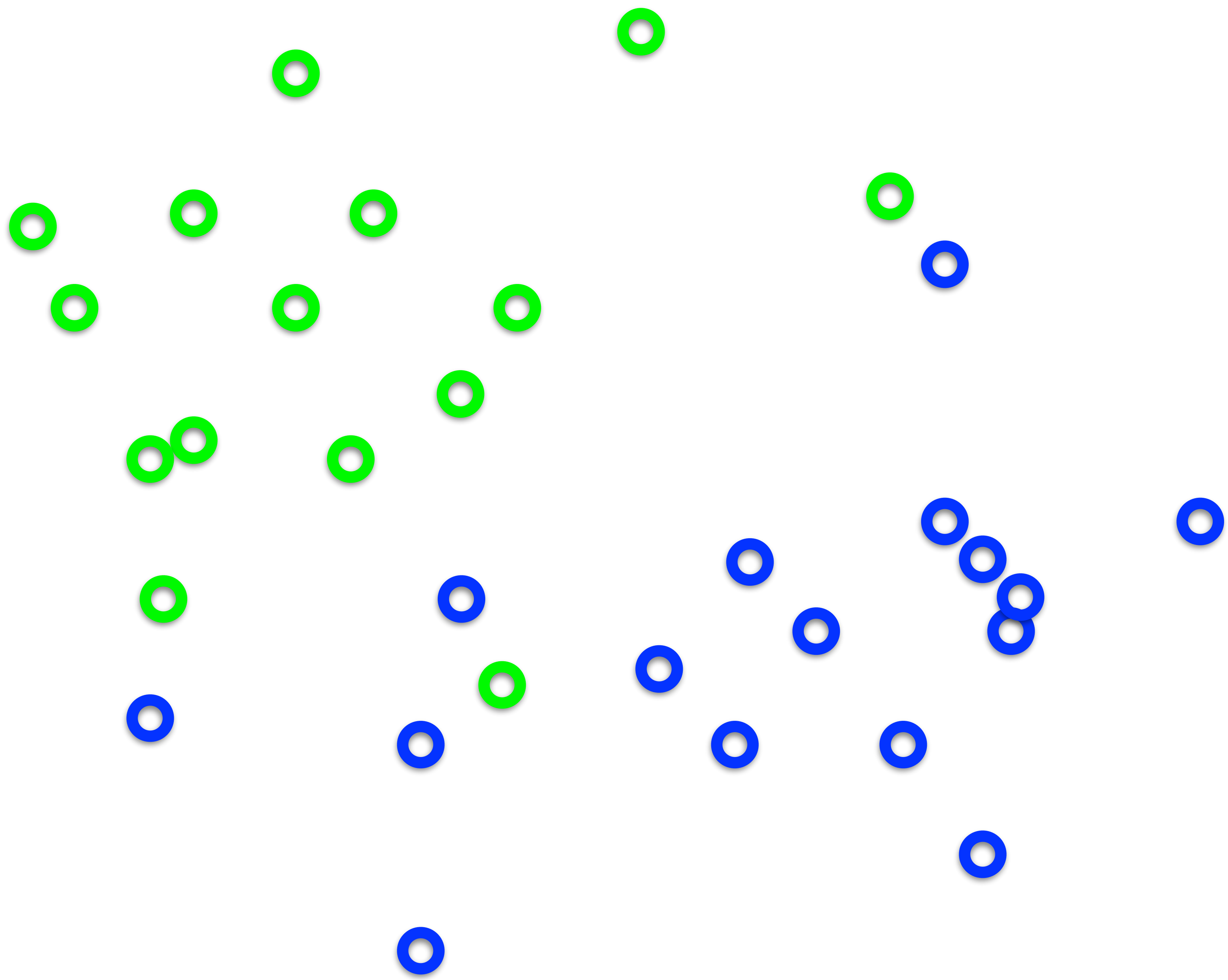


Naive Bayes

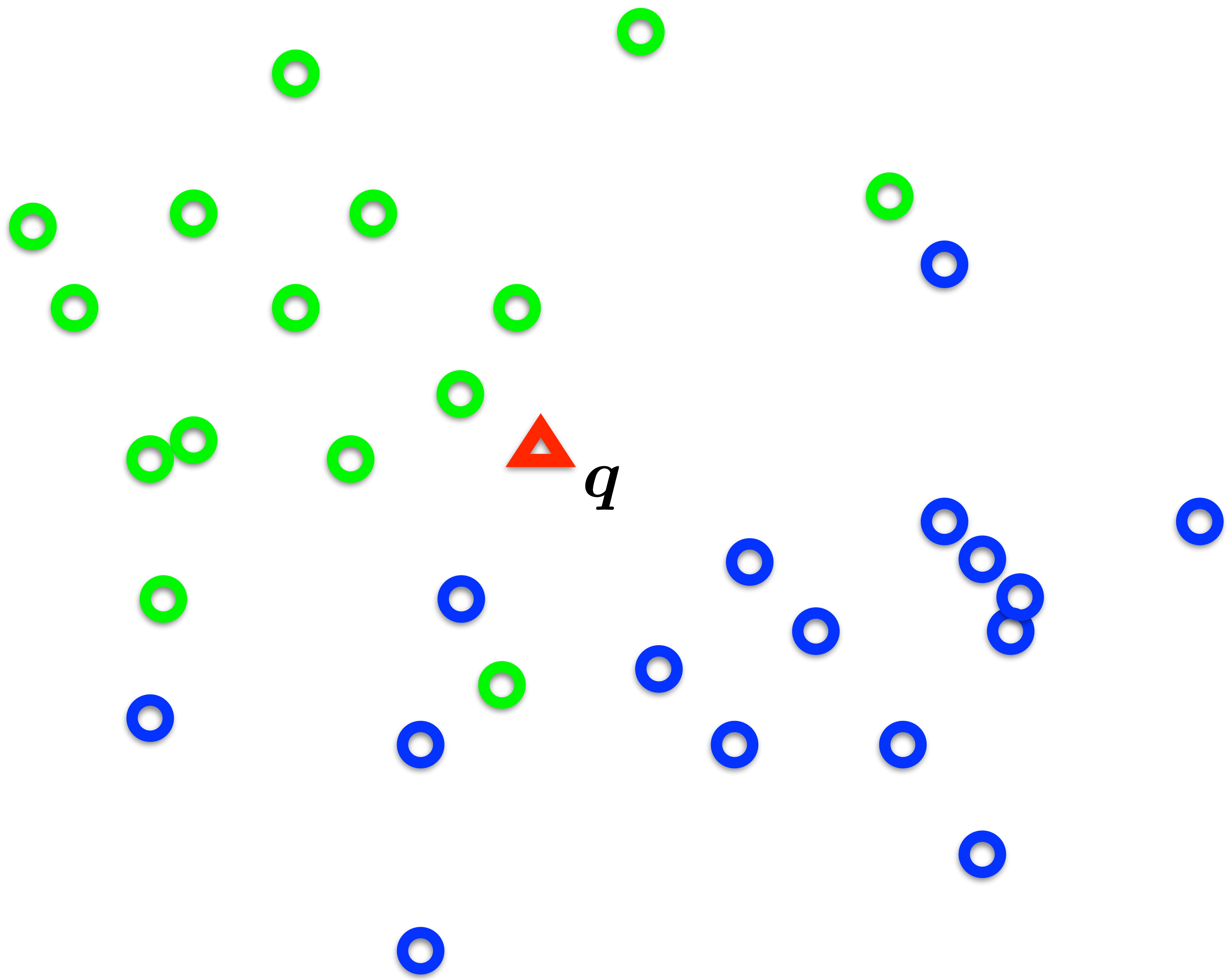
Support Vector Machine



Distribution of data from two classes

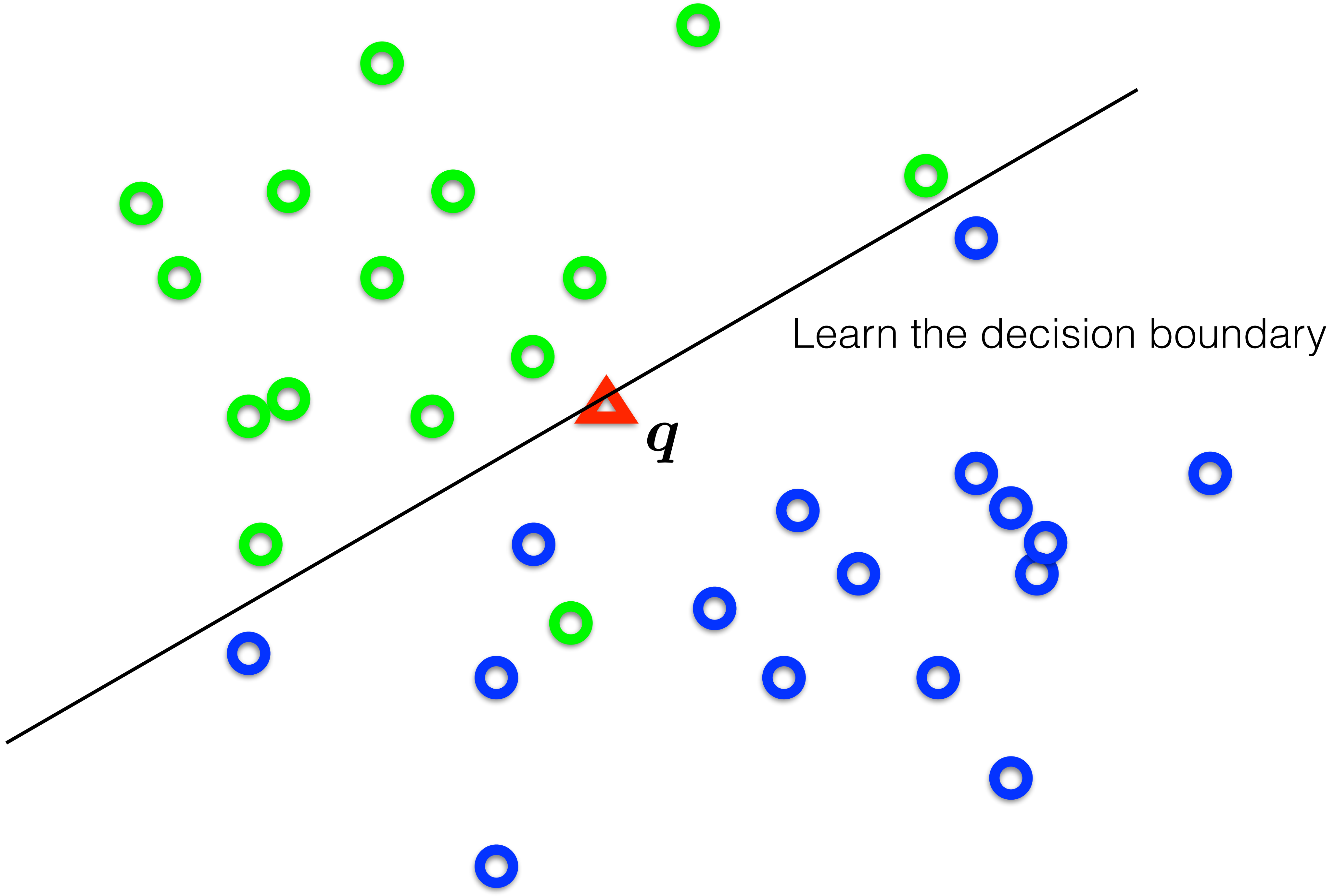


Distribution of data from two classes



Which class does q belong too?

Distribution of data from two classes

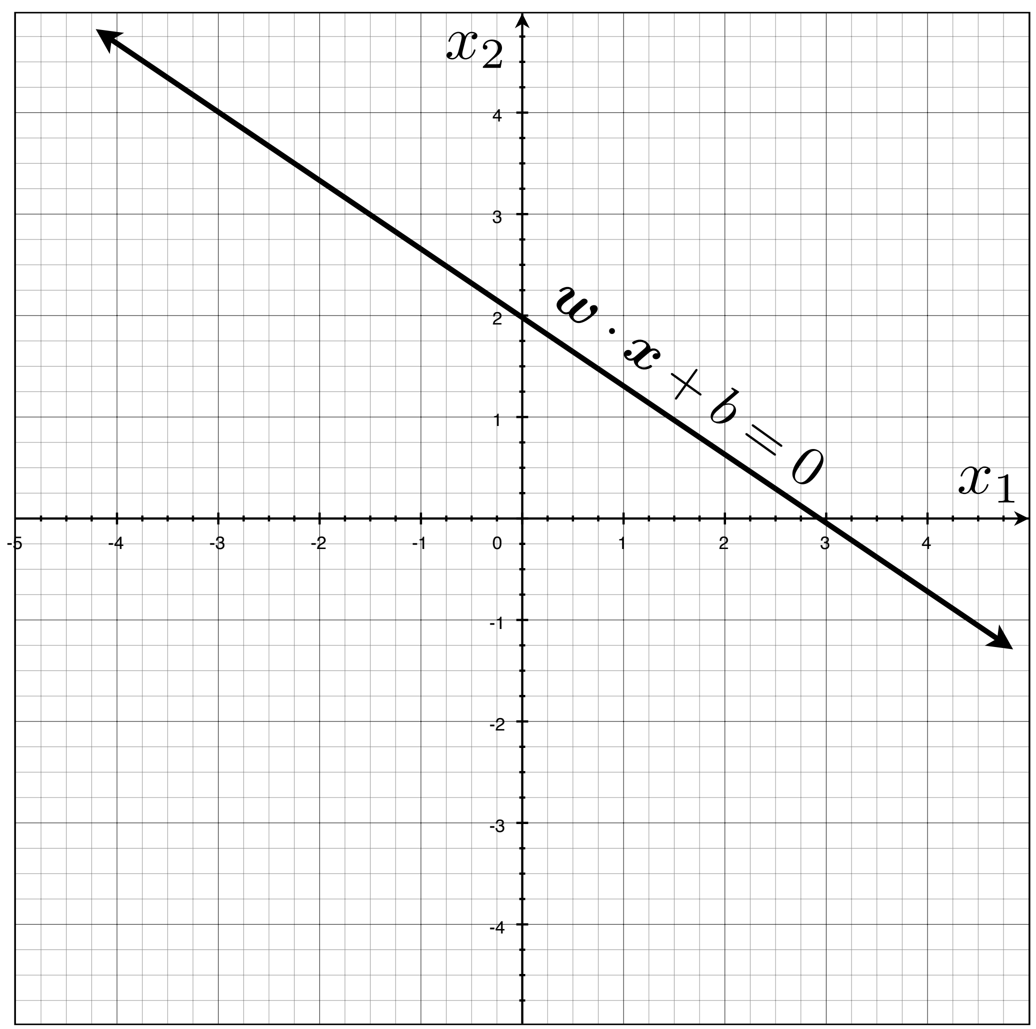


Support Vector Machine

First we need to understand hyperplanes…

Hyperplanes (lines) in 2D

$$w_1 x_1 + w_2 x_2 + b = 0$$



a line can be written as dot product plus a bias

$$\boldsymbol{w} \cdot \boldsymbol{x} + b = 0$$

$$\boldsymbol{w} \in \mathcal{R}^2$$

another version, add a weight 1 and push the bias inside

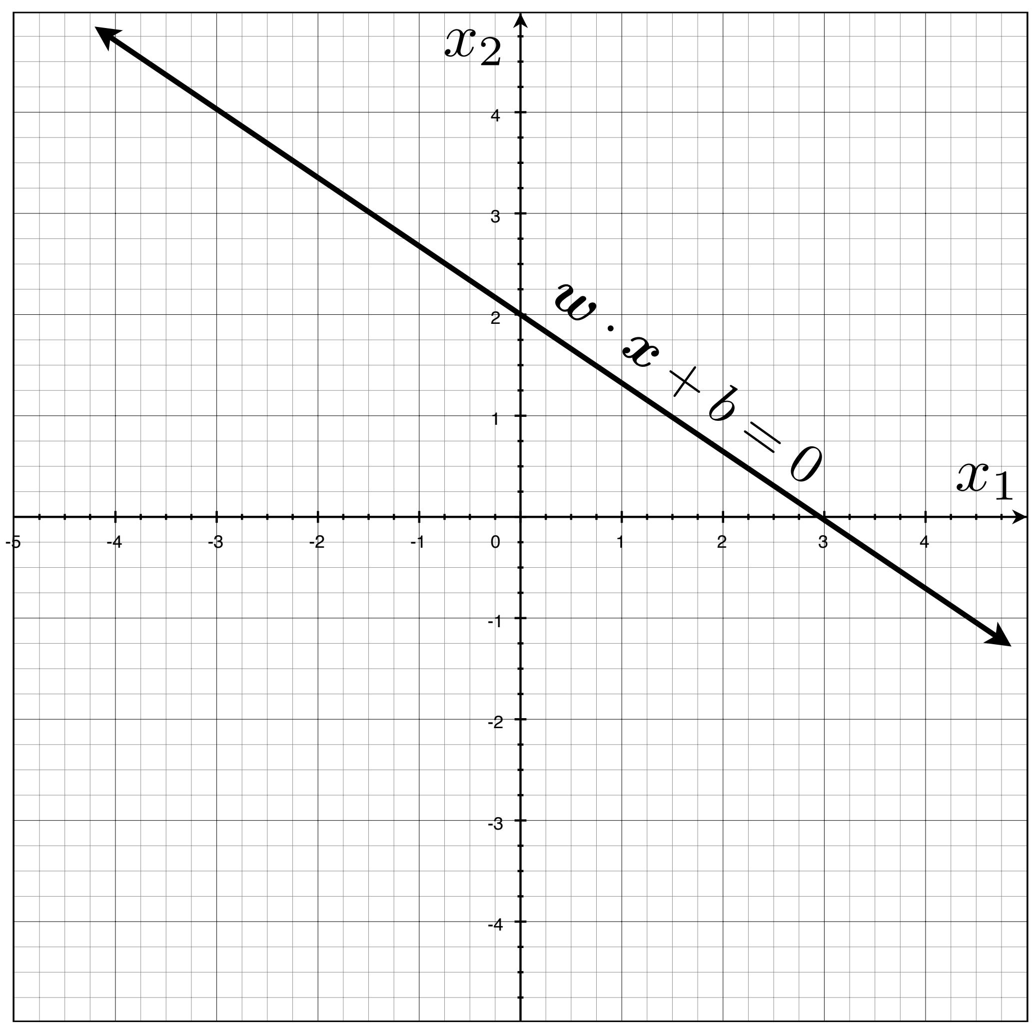
$$\boldsymbol{w} \cdot \boldsymbol{x} = 0$$

$$\boldsymbol{w} \in \mathcal{R}^3$$

Hyperplanes (lines) in 2D

$\boldsymbol{w} \cdot \boldsymbol{x} + b = 0$ (offset/bias outside) $\boldsymbol{w} \cdot \boldsymbol{x} = 0$ (offset/bias inside)

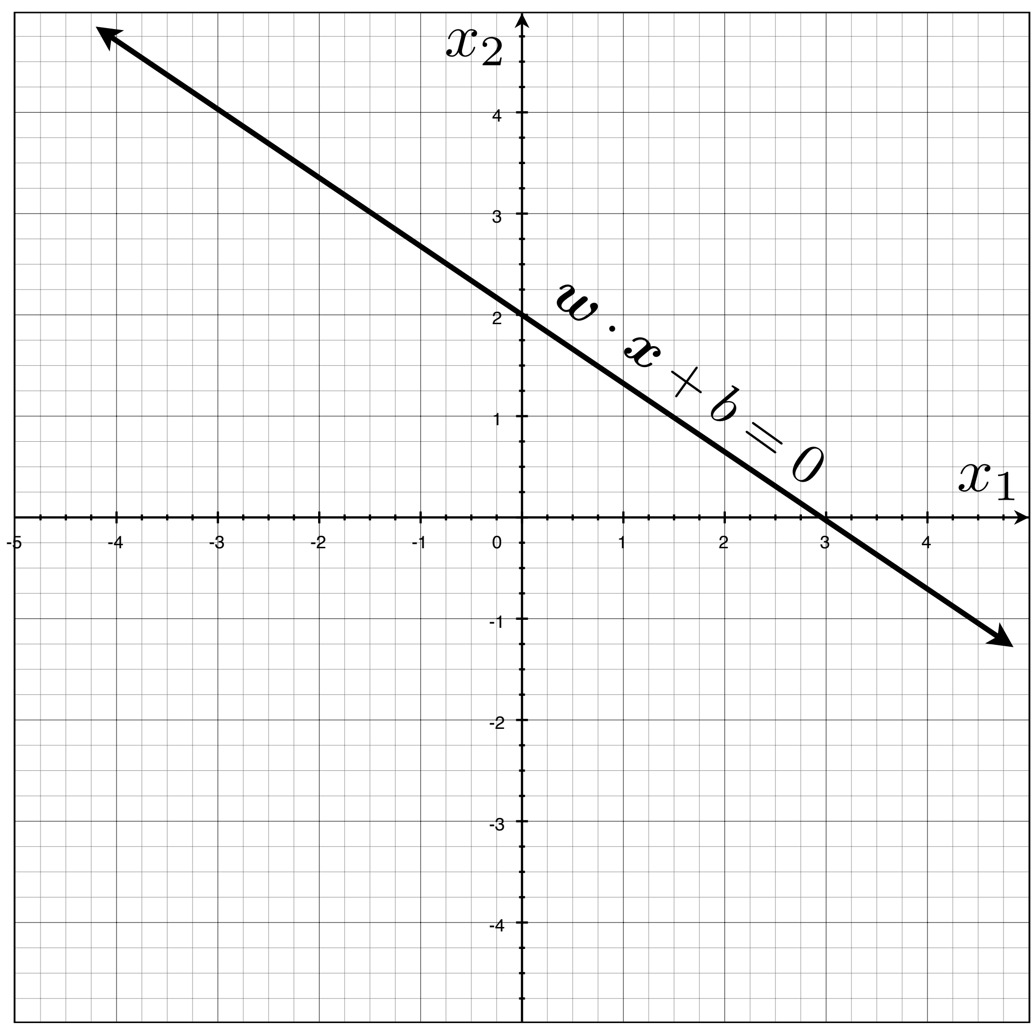
$$w_1 x_1 + w_2 x_2 + b = 0$$



Hyperplanes (lines) in 2D

$\boldsymbol{w} \cdot \boldsymbol{x} + b = 0$ (offset/bias outside) $\boldsymbol{w} \cdot \boldsymbol{x} = 0$ (offset/bias inside)

$$w_1 x_1 + w_2 x_2 + b = 0$$



Important property:

Free to choose any normalization of w

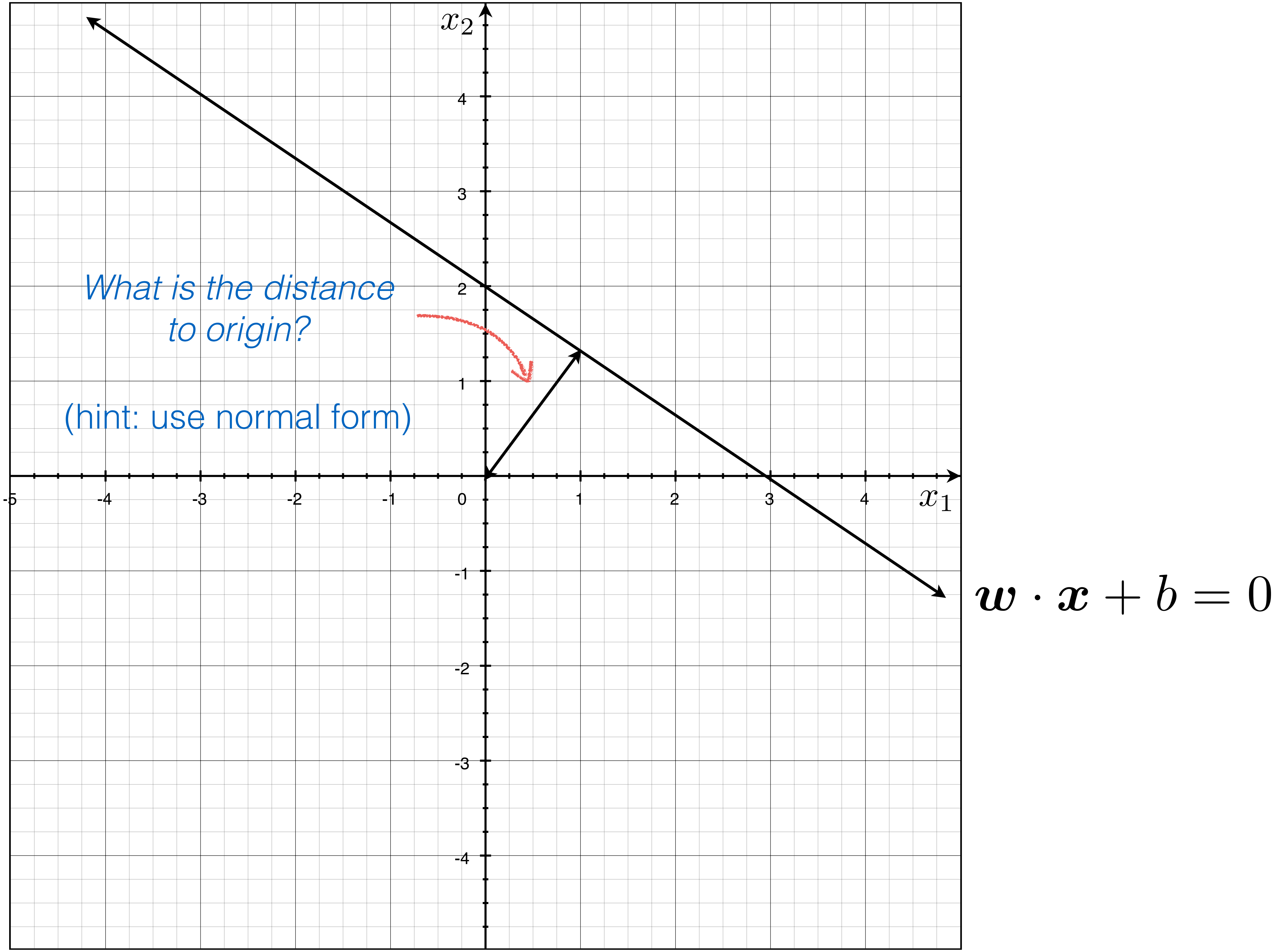
The line

$$w_1 x_1 + w_2 x_2 + b = 0$$

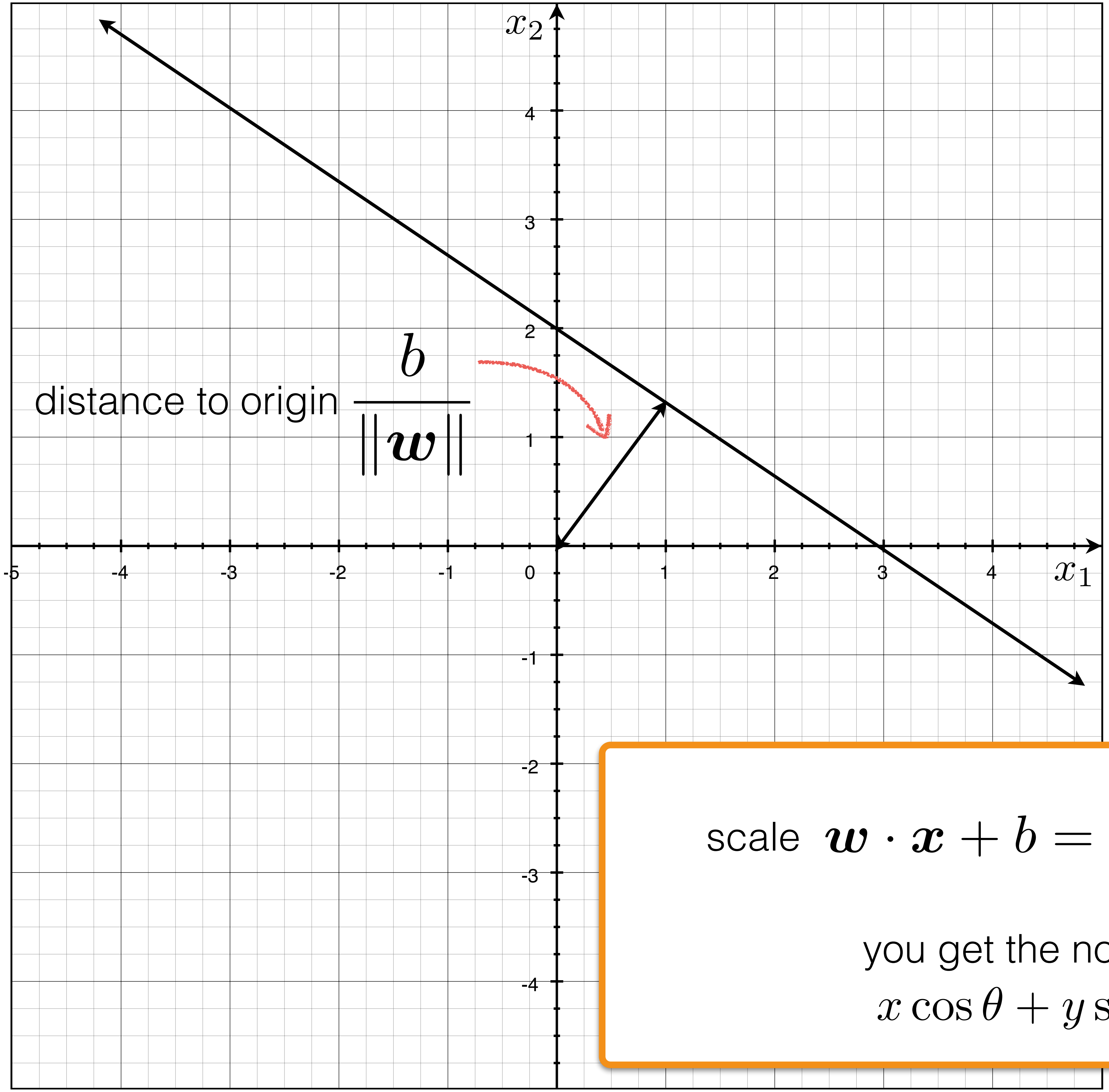
and the line

$$\lambda(w_1 x_1 + w_2 x_2 + b) = 0$$

define the same line



x_2
4
3
2
1
0
-1
-2
-3
-4
-5
-4
-3
-2
-1
1
2
3
4
x_1
What is the distance to origin?
(hint: use normal form)
$\boldsymbol{w} \cdot \boldsymbol{x} + b = 0$

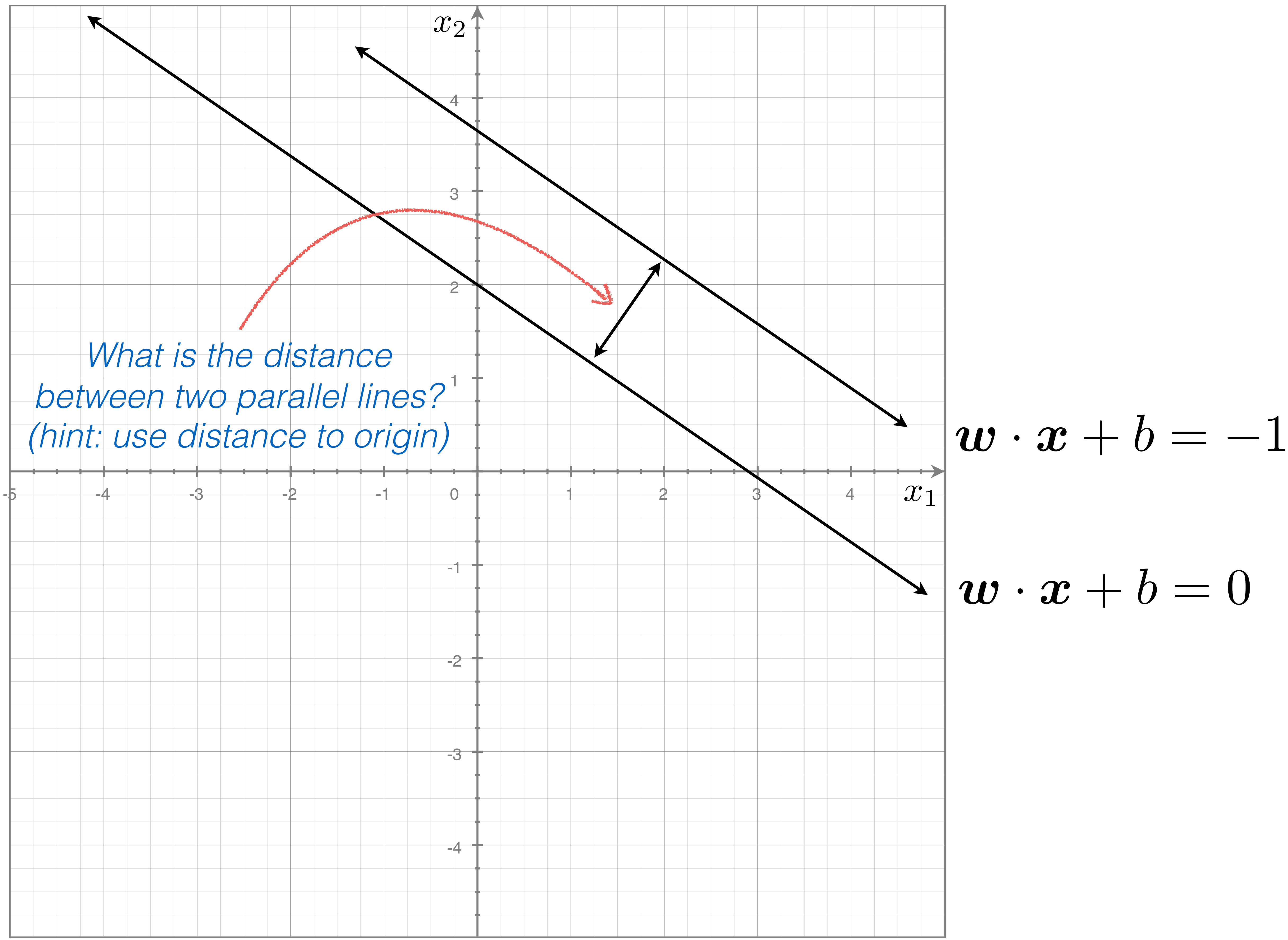


$$\boldsymbol{w} \cdot \boldsymbol{x} + b = 0$$

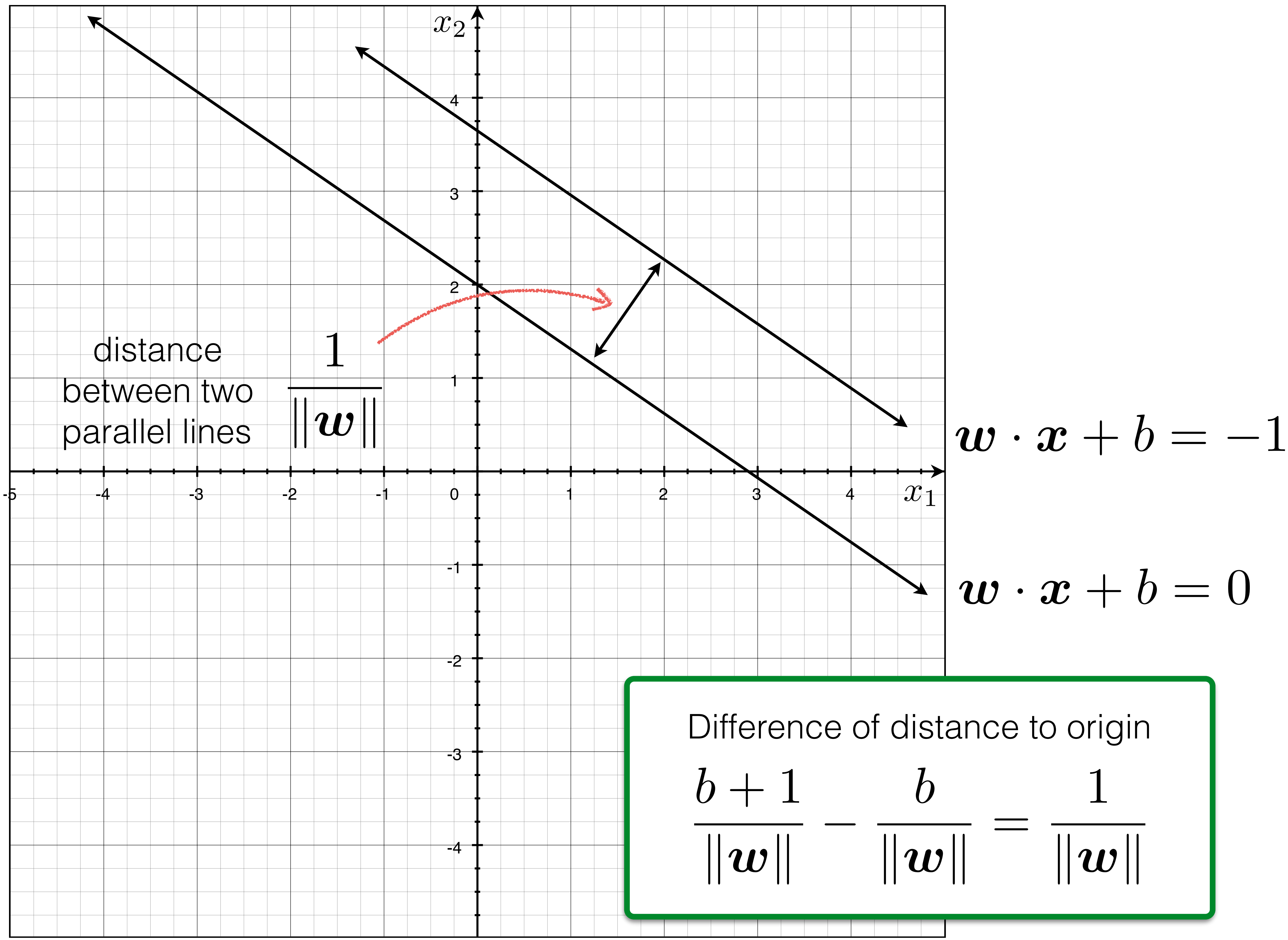
scale $\boldsymbol{w} \cdot \boldsymbol{x} + b = 0$ by $\frac{1}{\|\boldsymbol{w}\|}$

you get the normal form

$$x \cos\theta + y \sin\theta = \rho$$



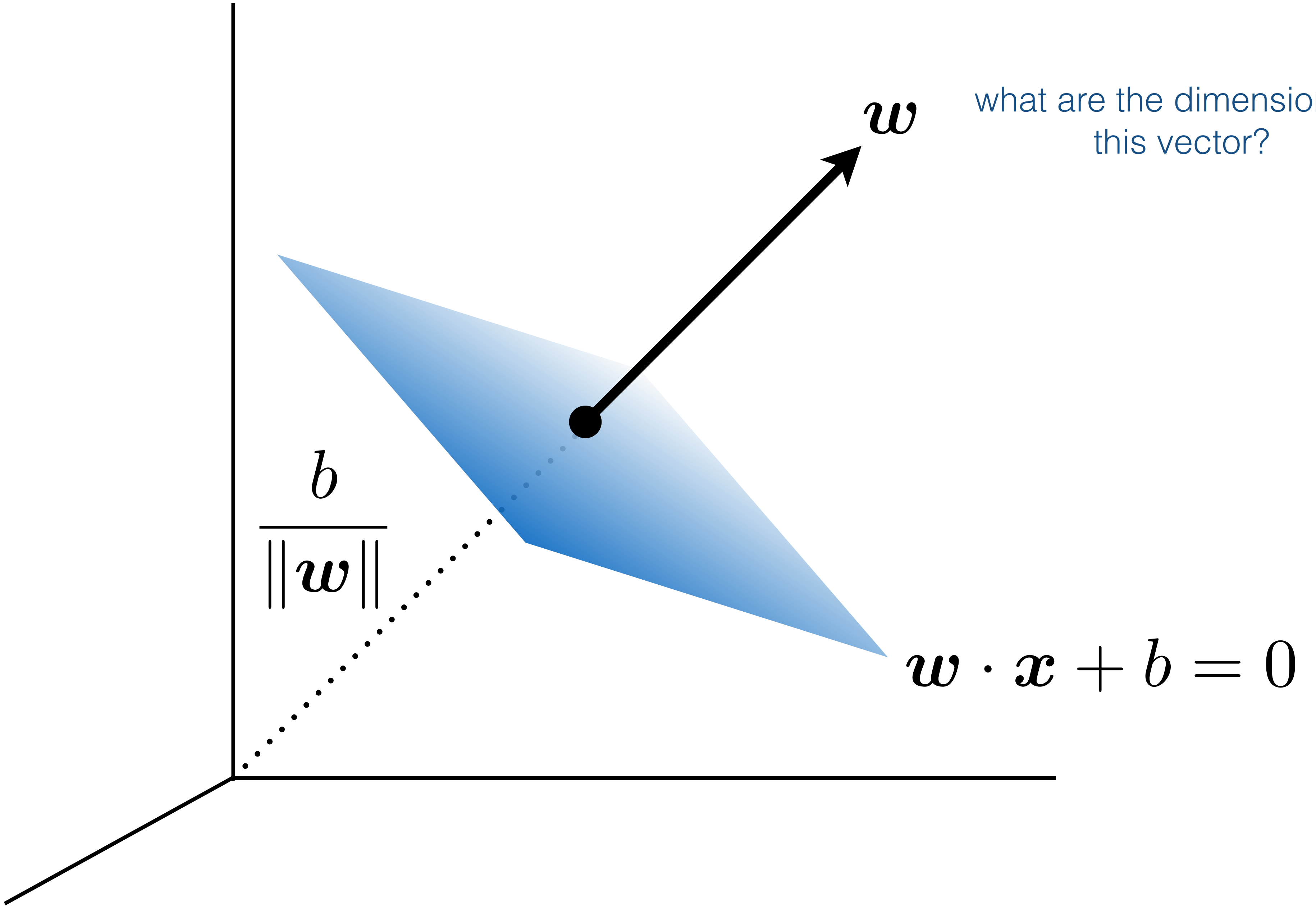
x_2
x_1
What is the distance
between two parallel lines?
(hint: use distance to origin)
$\boldsymbol{w} \cdot \boldsymbol{x} + b = -1$
$\boldsymbol{w} \cdot \boldsymbol{x} + b = 0$



x_2
x_1
distance between two parallel lines
$\frac{1}{\|\boldsymbol{w}\|}$
$\boldsymbol{w} \cdot \boldsymbol{x} + b = -1$
$\boldsymbol{w} \cdot \boldsymbol{x} + b = 0$
Difference of distance to origin
$\frac{b+1}{\|\boldsymbol{w}\|} - \frac{b}{\|\boldsymbol{w}\|} = \frac{1}{\|\boldsymbol{w}\|}$

Now we can go to 3D …

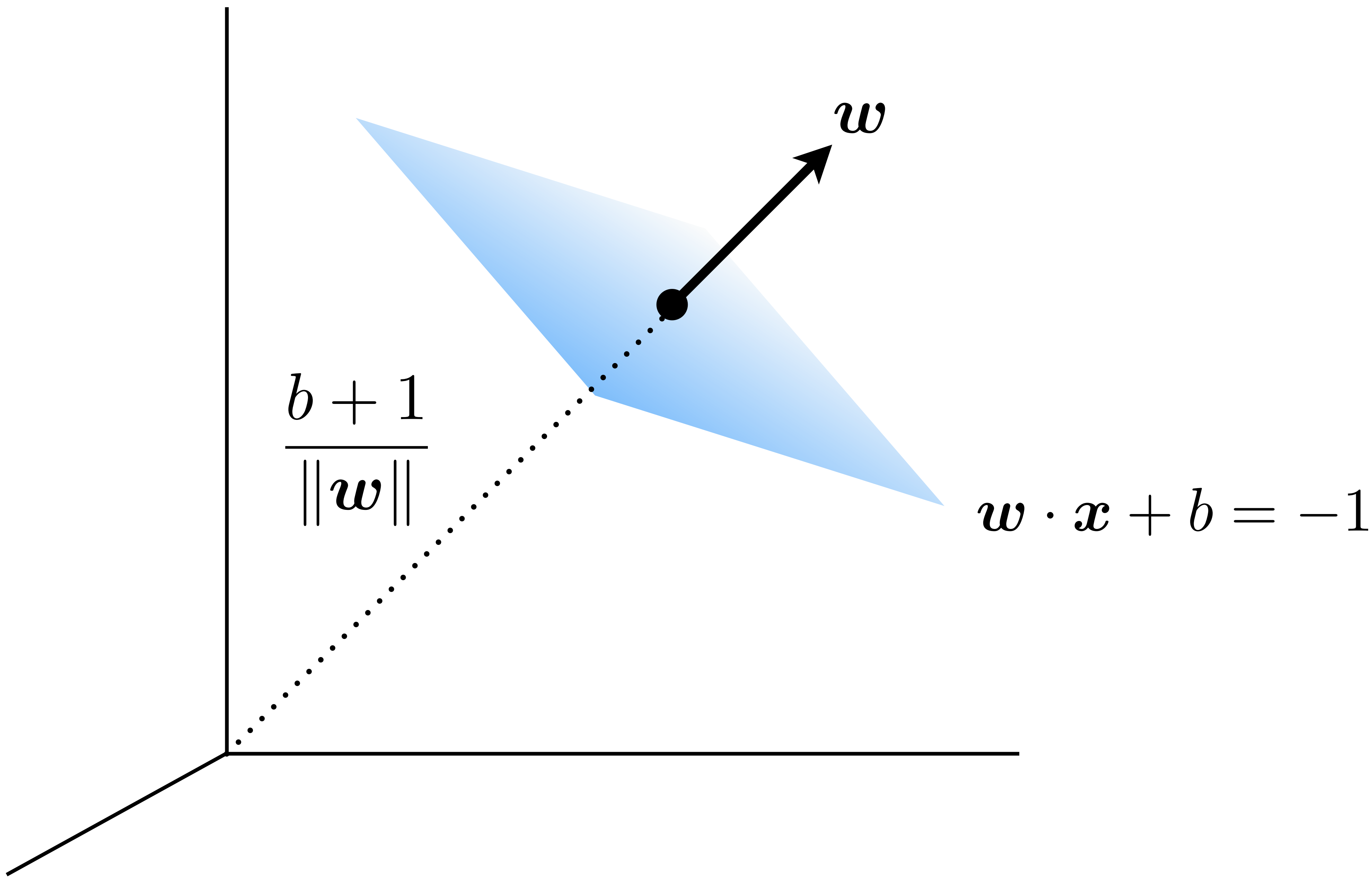
Hyperplanes (planes) in 3D



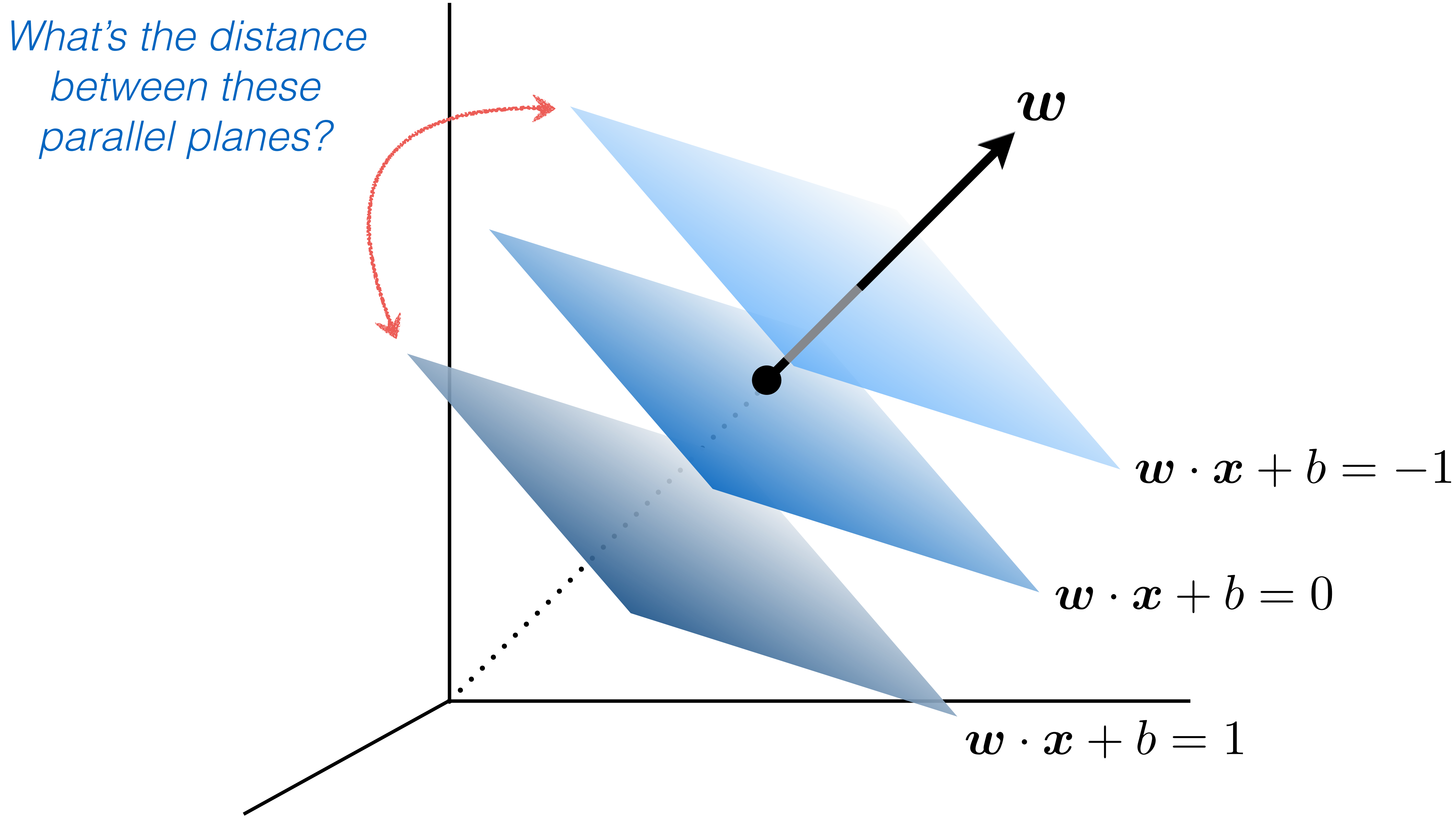
what are the dimensions of this vector?

*What happens if you change **b**?*

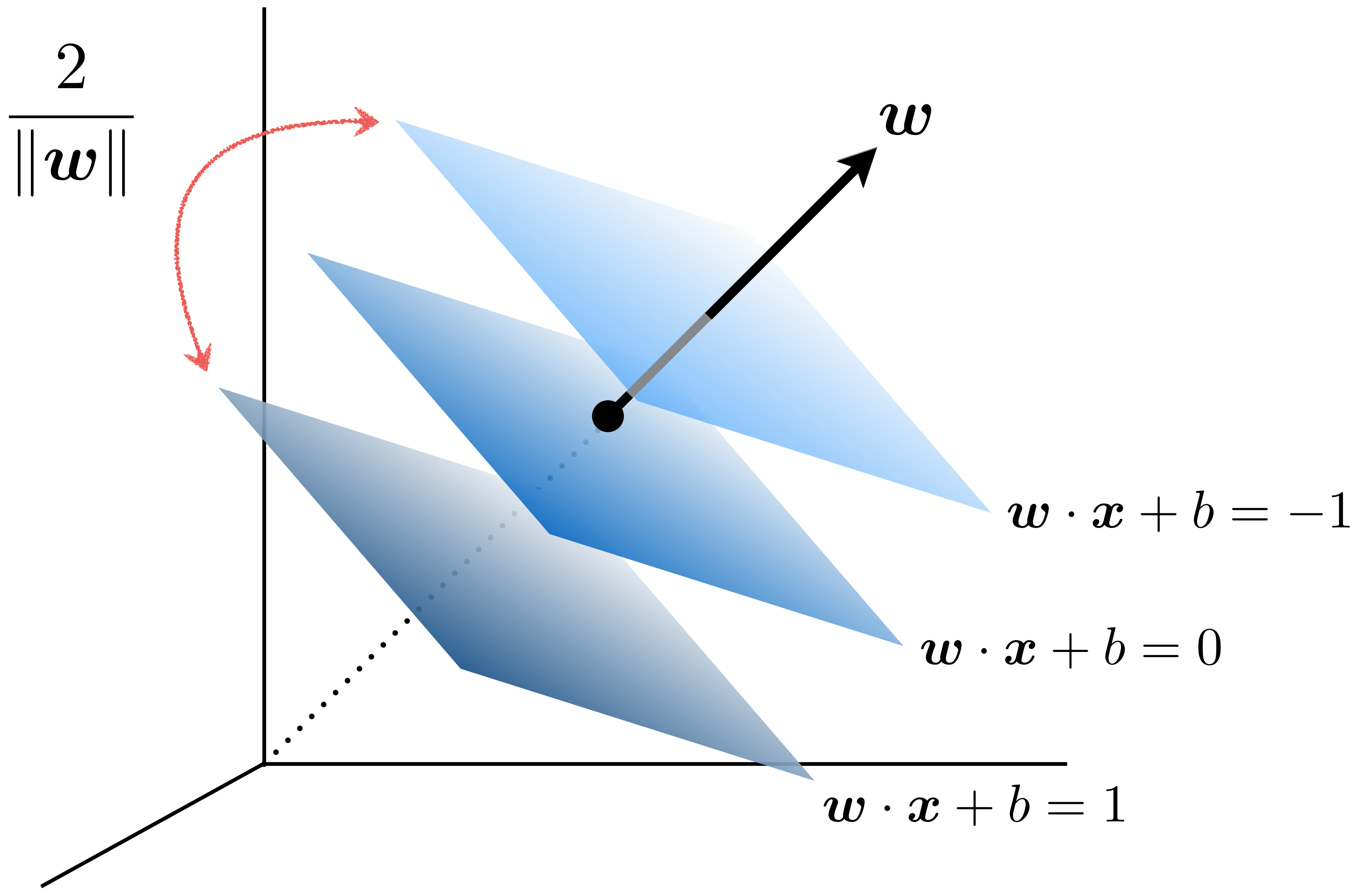
Hyperplanes (planes) in 3D



Hyperplanes (planes) in 3D

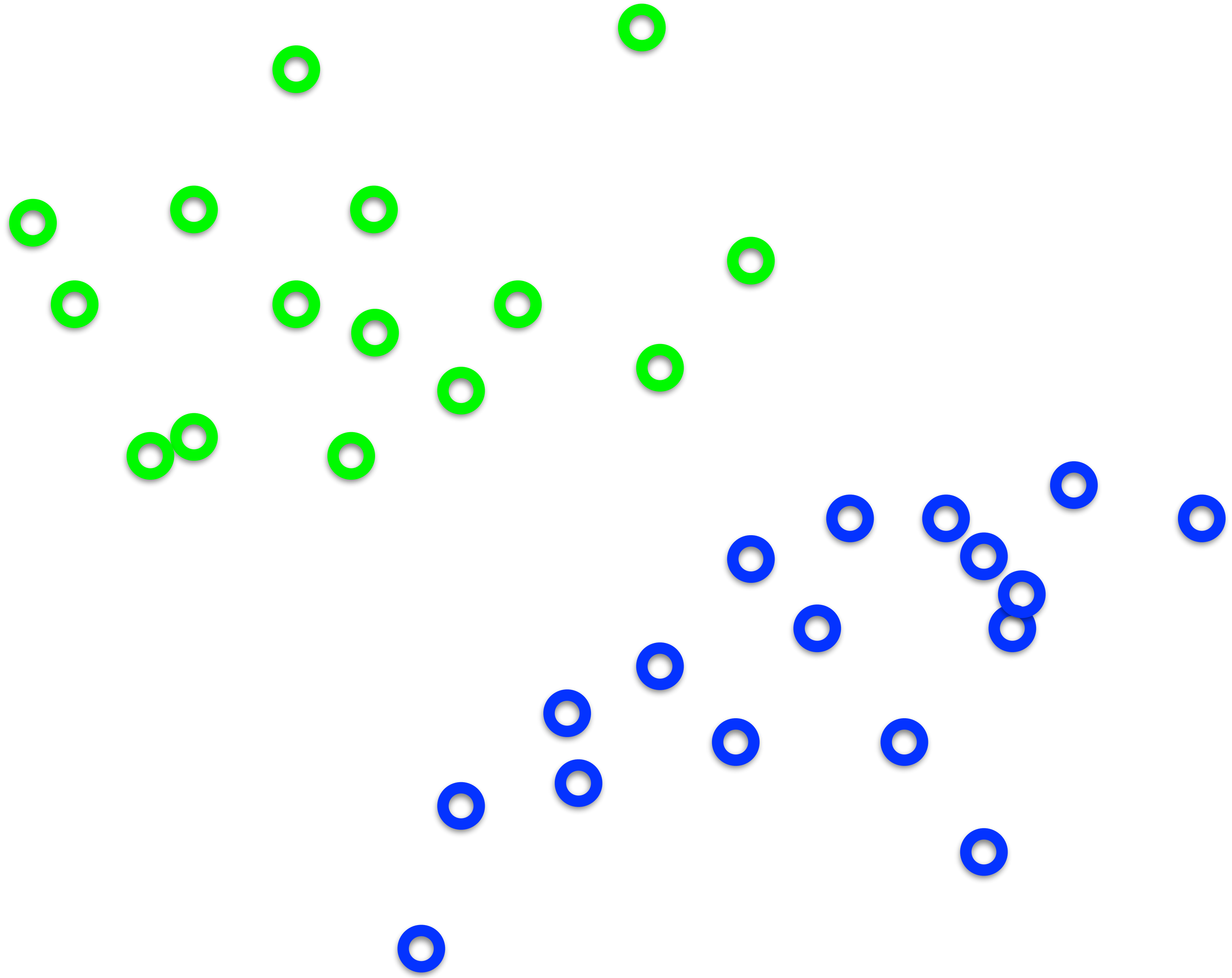


Hyperplanes (planes) in 3D

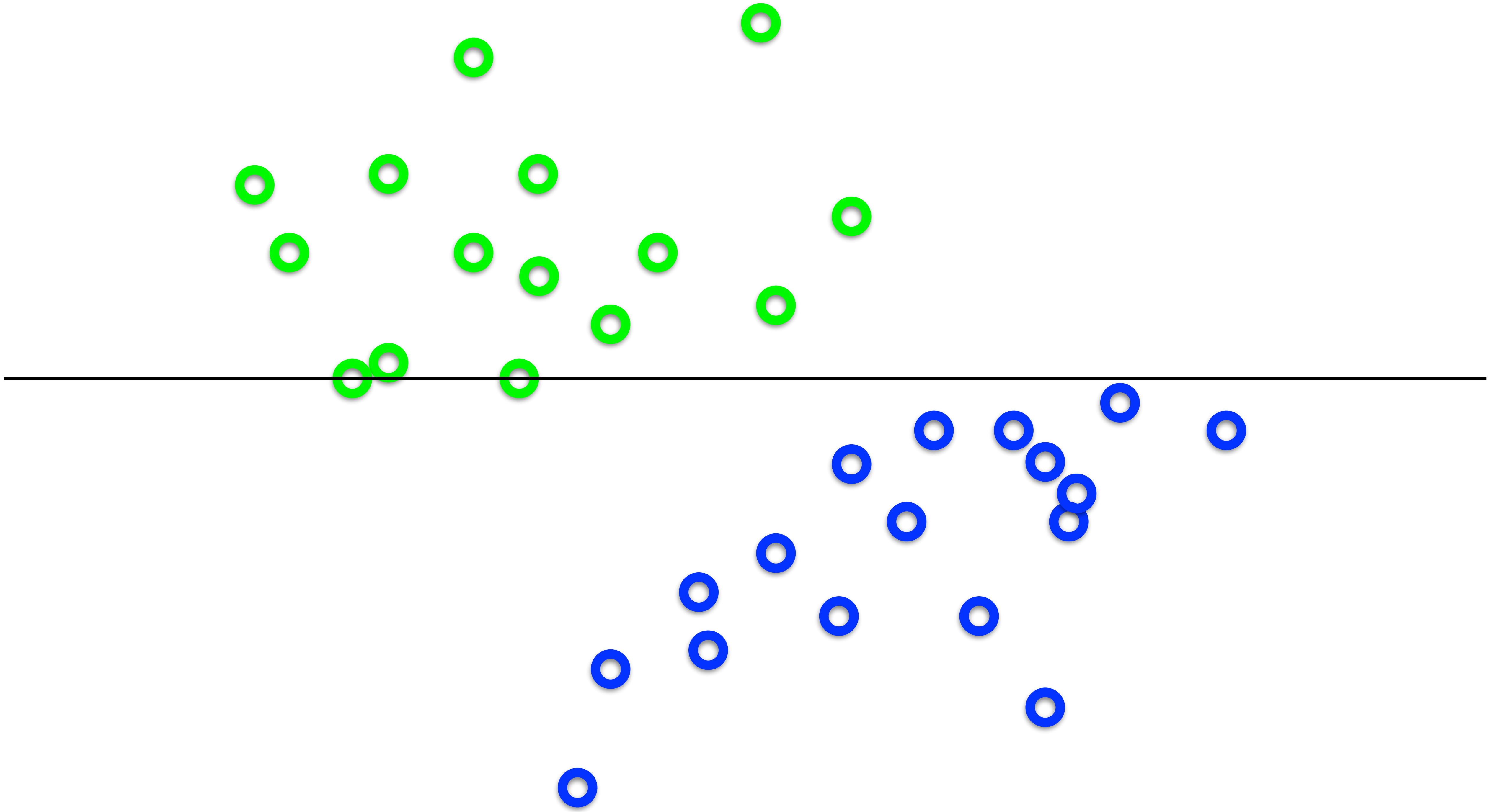


Support Vector Machine

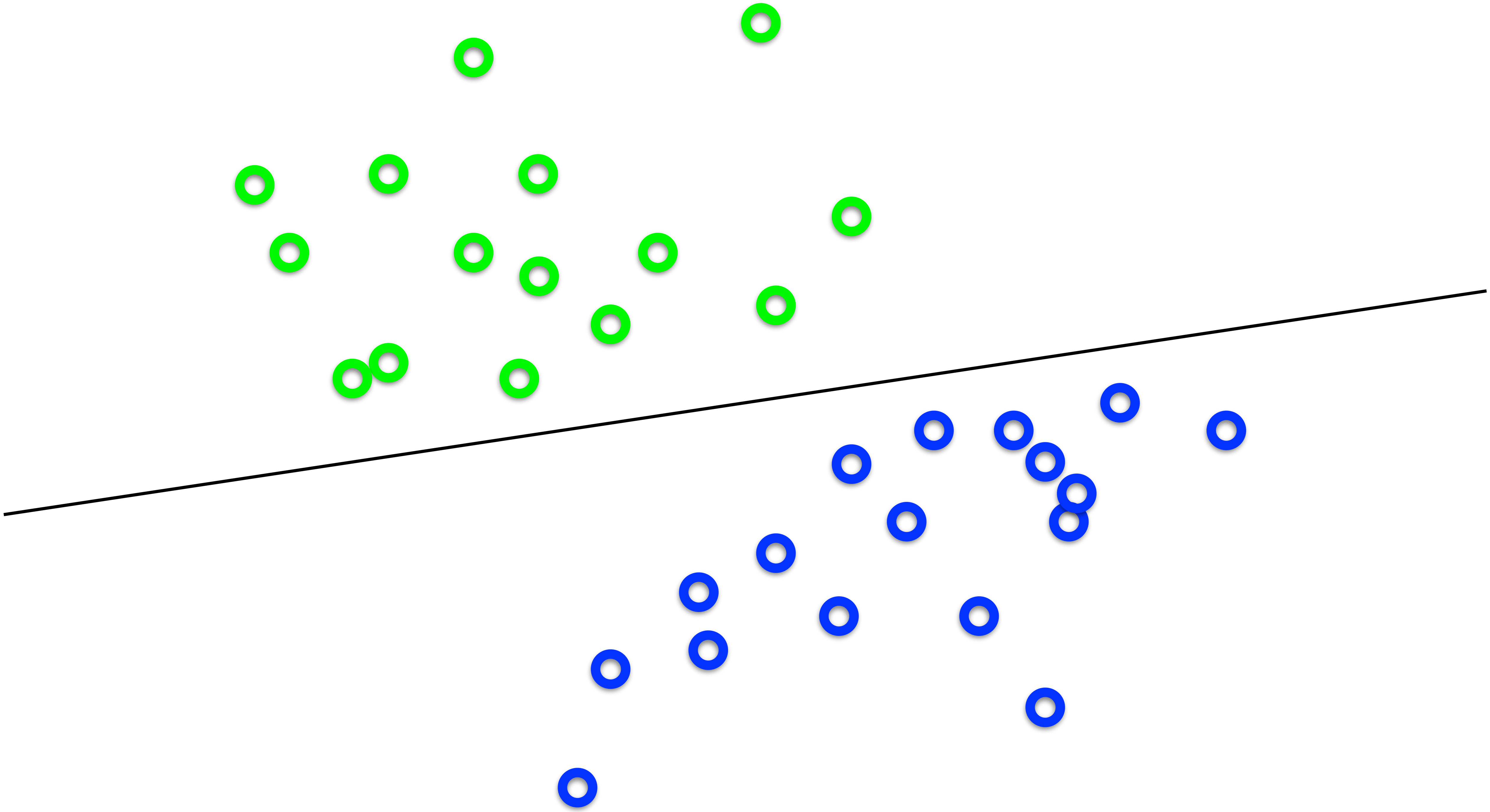
What's the best **w**?



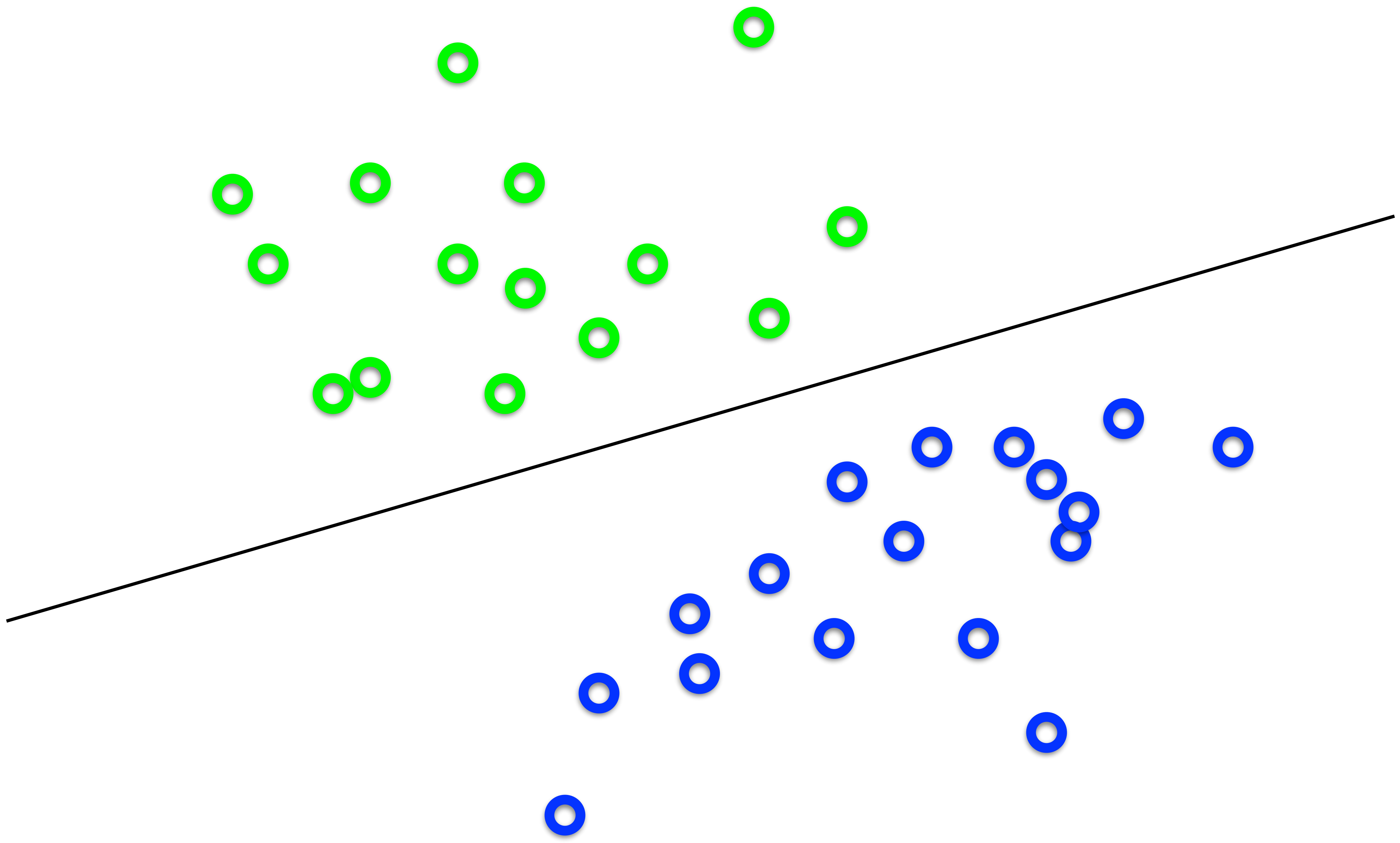
What’s the best **w**?



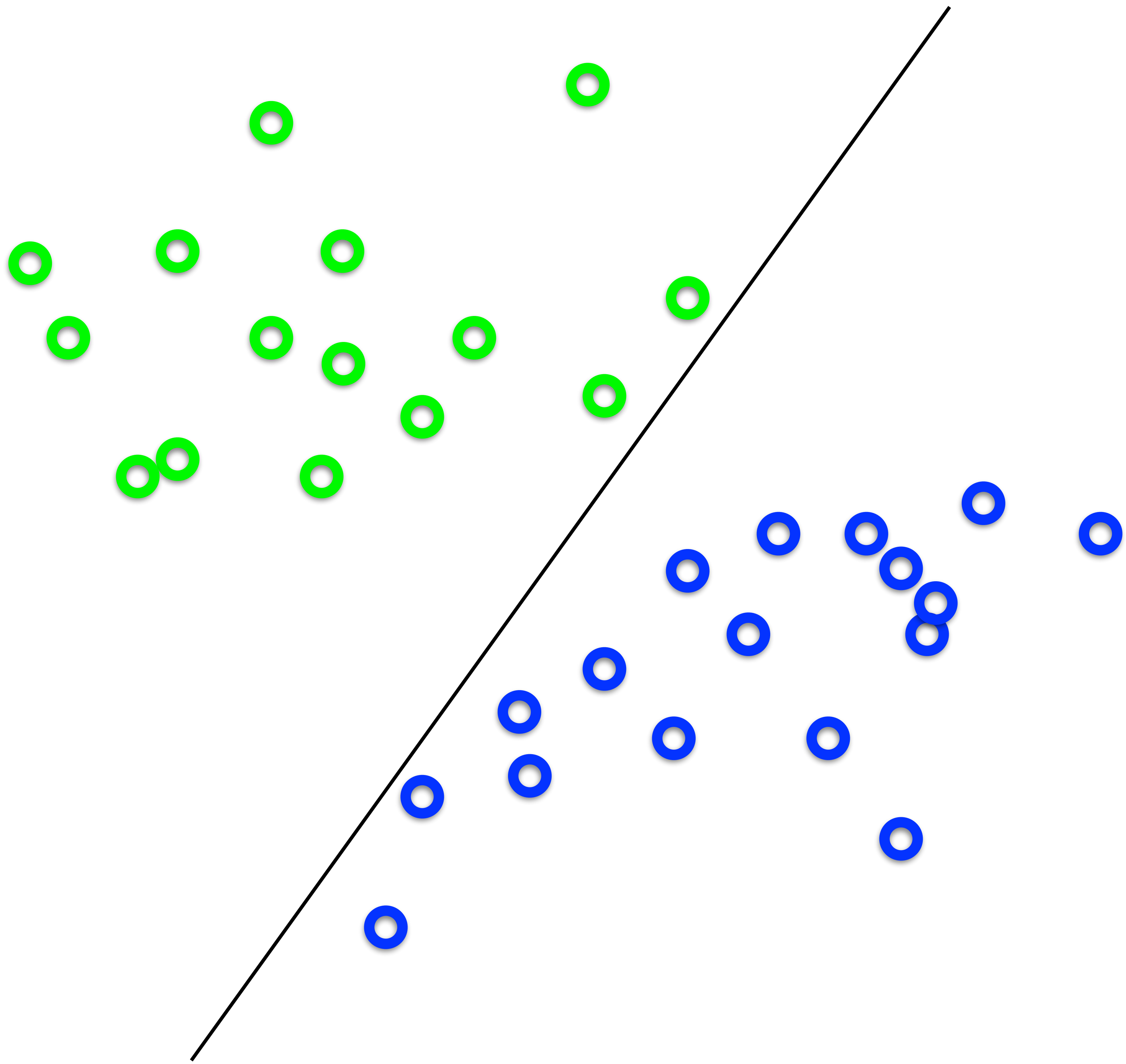
What’s the best **w**?



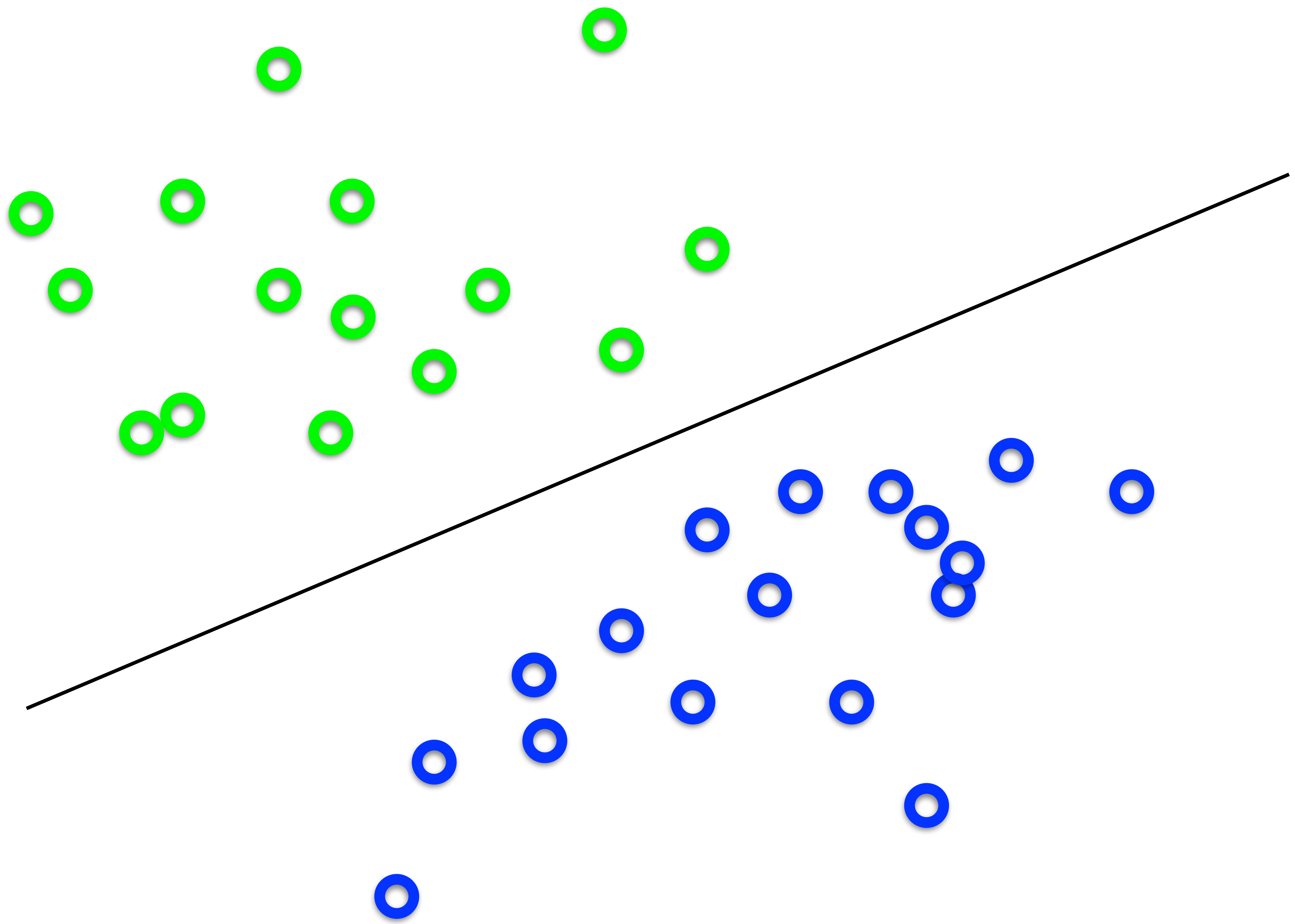
What’s the best **w**?



What's the best **w**?

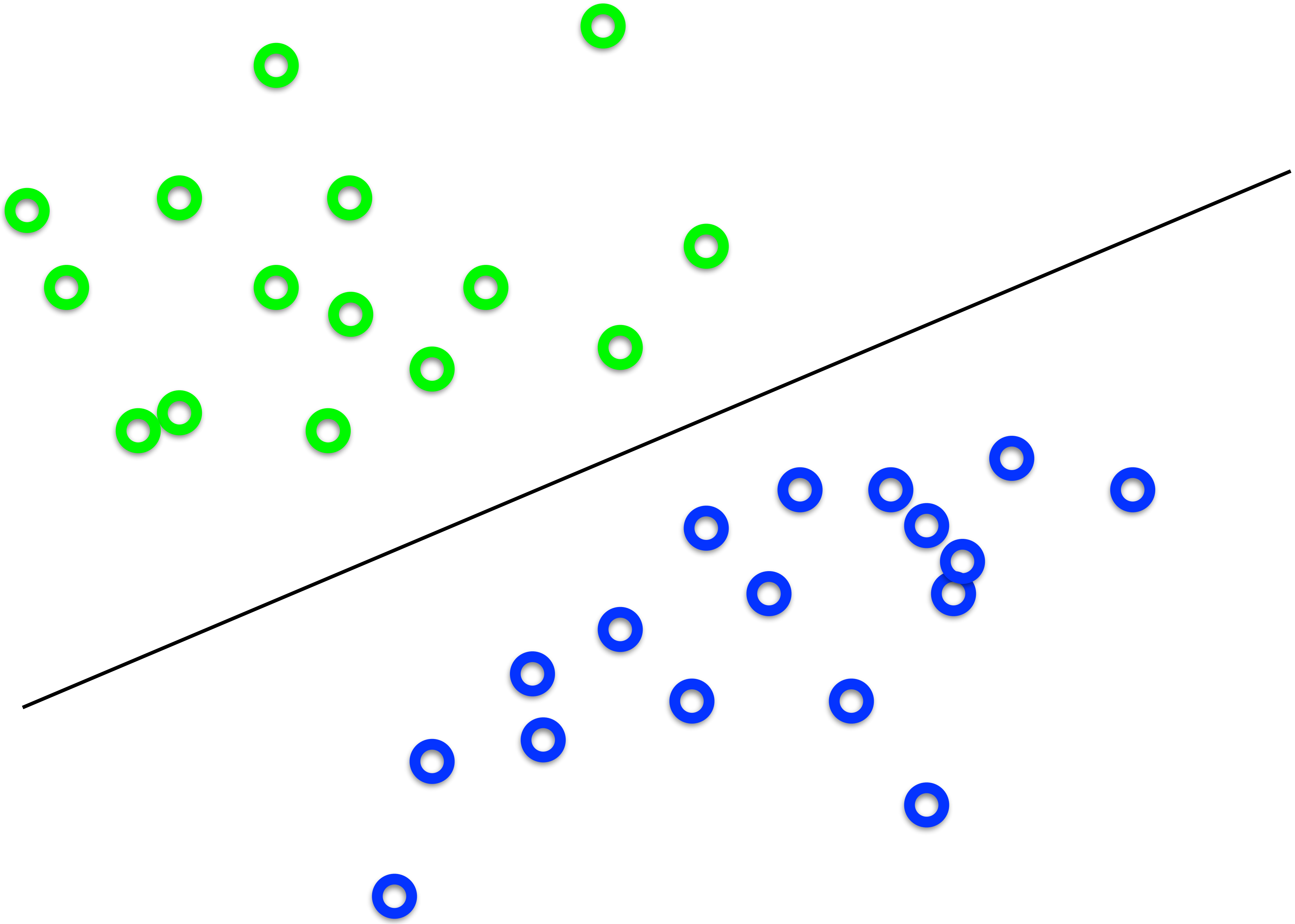


What's the best **w**?



Intuitively, the line that is the farthest from all interior points

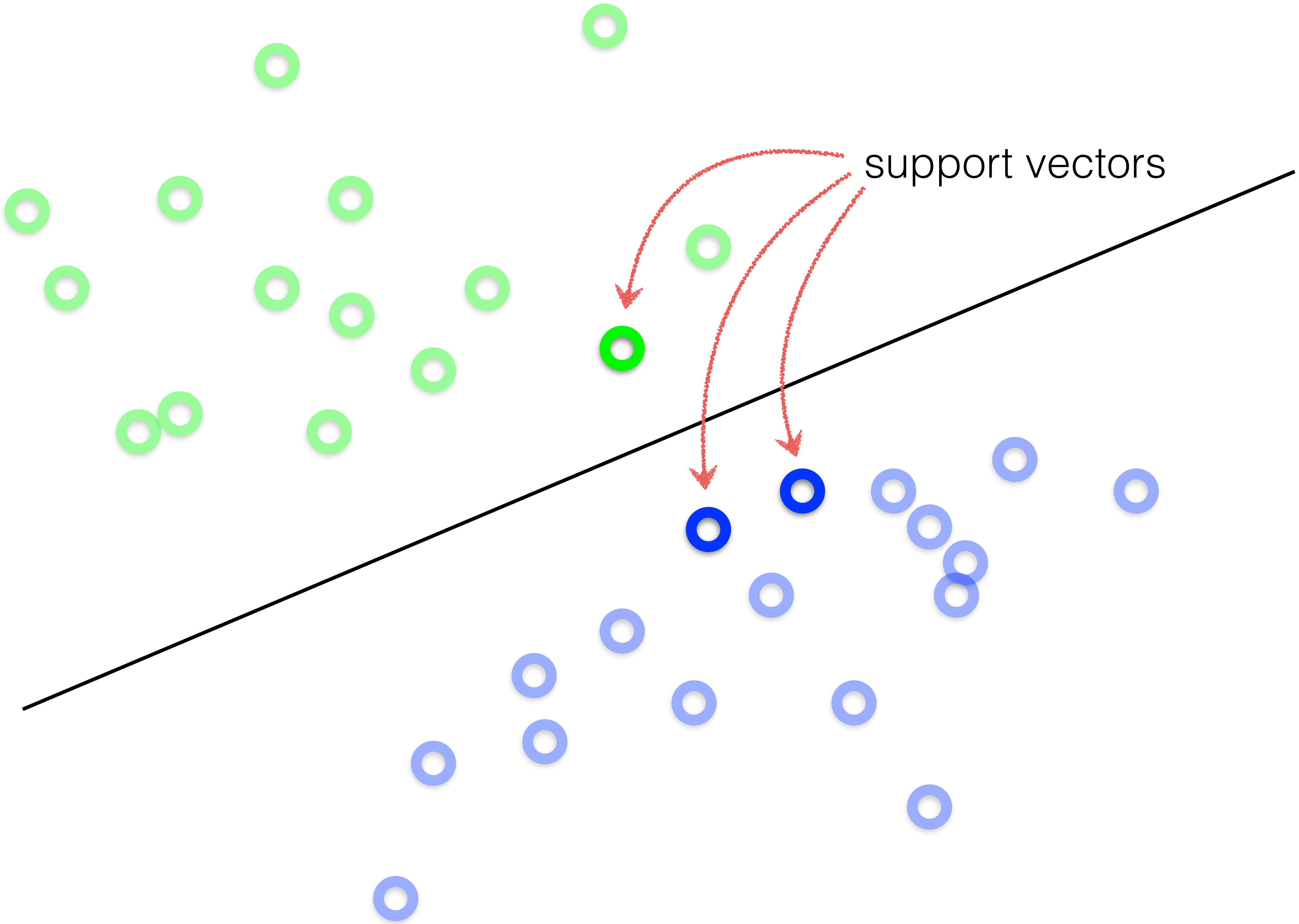
What's the best **w**?



Maximum Margin solution:

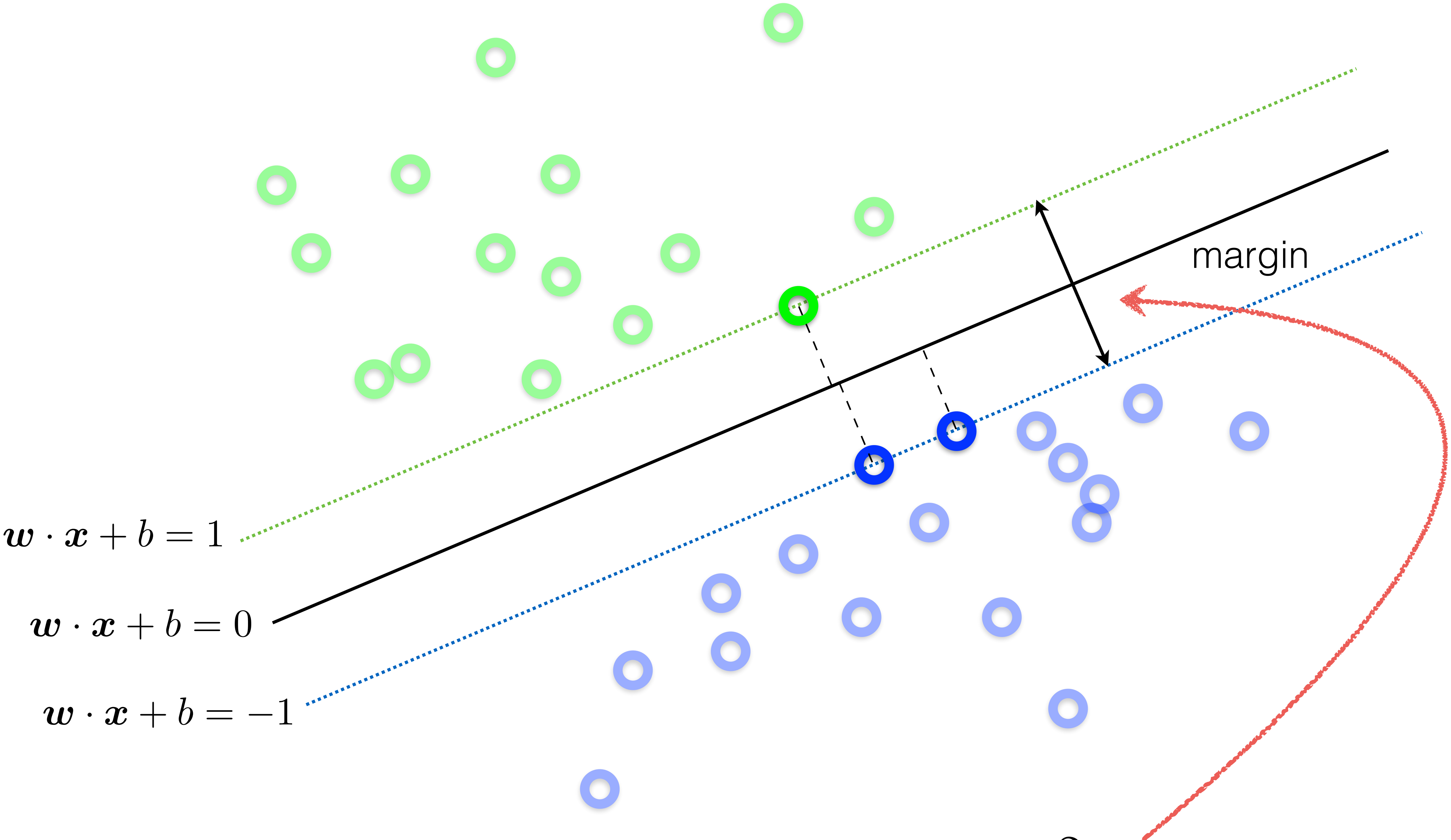
most stable to perturbations of data

What's the best **w**?



Want a hyperplane that is far away from 'inner points'

Find hyperplane **w** such that ...



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

Can be formulated as a maximization problem

$$\max_{\boldsymbol{w}} \frac{2}{\|\boldsymbol{w}\|}$$

$$\text{subject to } \boldsymbol{w} \cdot \boldsymbol{x}_i + b \begin{array}{l} \geq +1 \ \text{ if } \ y_i = +1 \\ \leq -1 \ \text{ if } \ y_i = -1 \end{array} \text{ for } \ i = 1, \ldots, N$$

What does this constraint mean?

label of the data point

Why is it +1 and -1?

Can be formulated as a maximization problem

$$\max_{\boldsymbol{w}} \frac{2}{\|\boldsymbol{w}\|}$$

$$\text{subject to } \boldsymbol{w} \cdot \boldsymbol{x}_i + b \begin{array}{l} \geq +1 \;\text{ if }\; y_i = +1 \\ \leq -1 \;\text{ if }\; y_i = -1 \end{array} \text{ for } \; i = 1, \ldots, N$$

Equivalently,

Where did the 2 go?

$$\min_{\boldsymbol{w}} \|\boldsymbol{w}\|$$

$$\text{subject to } \; y_i(\boldsymbol{w} \cdot \boldsymbol{x}_i + b) \geq 1 \;\text{ for }\; i = 1, \ldots, N$$

What happened to the labels?

‘Primal formulation’ of a linear SVM

$$\min_{\boldsymbol{w}} \|\boldsymbol{w}\|$$

Objective Function

$$\text{subject to } y_i(\boldsymbol{w} \cdot \boldsymbol{x}_i + b) \geq 1 \text{ for } \; i = 1, \ldots, N$$

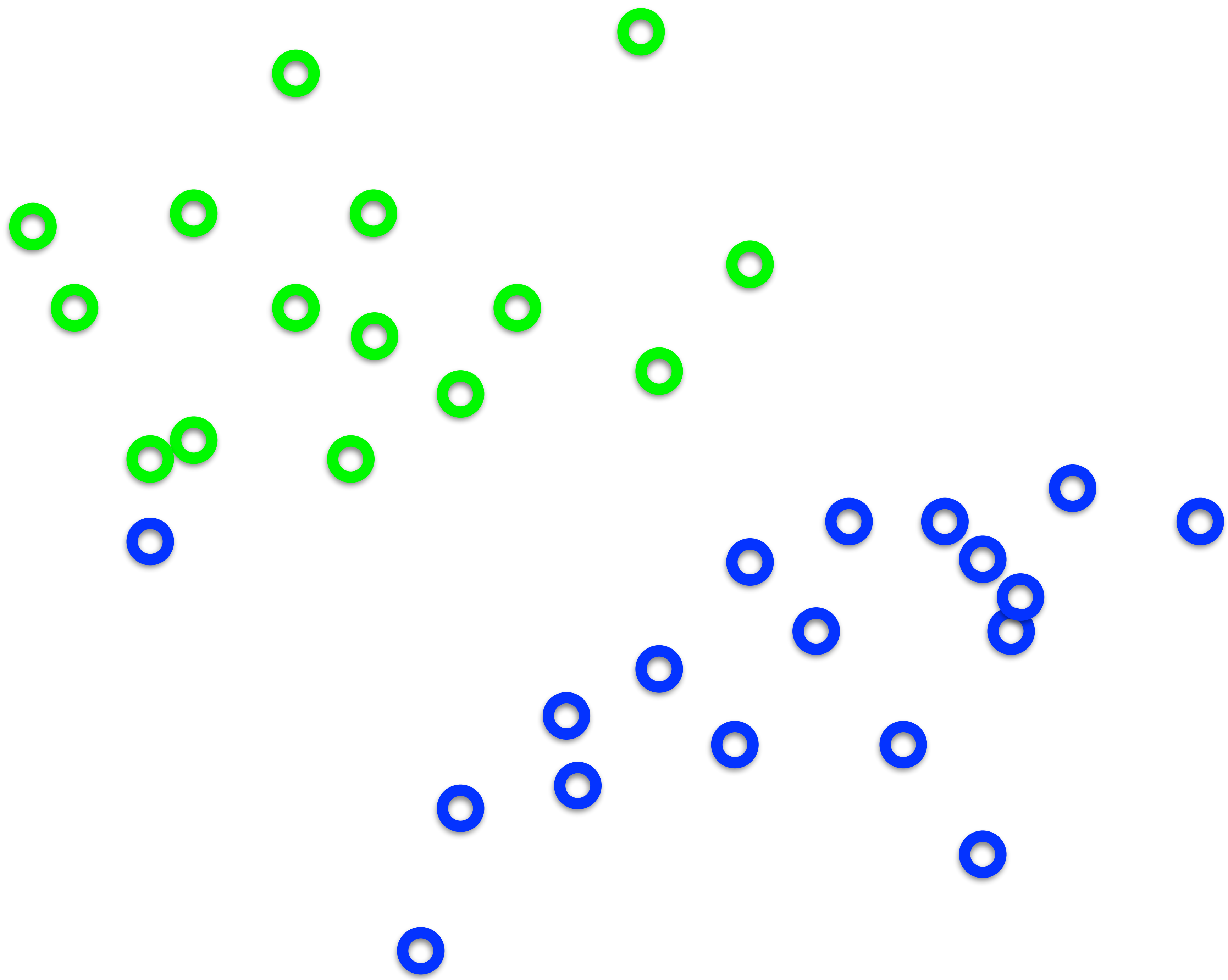
Constraints

This is a convex quadratic programming (QP) problem

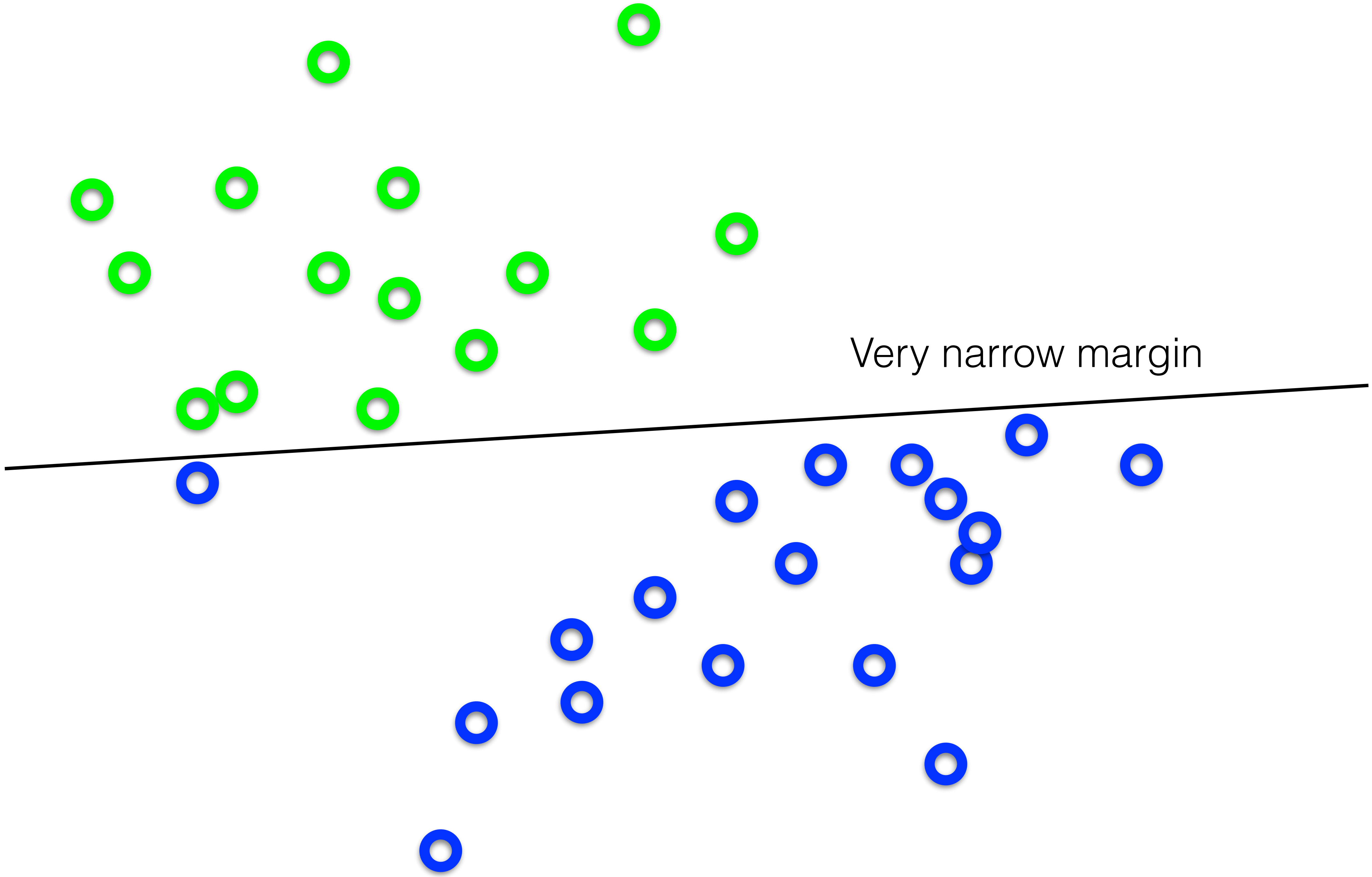
(a unique solution exists)

‘soft’ margin

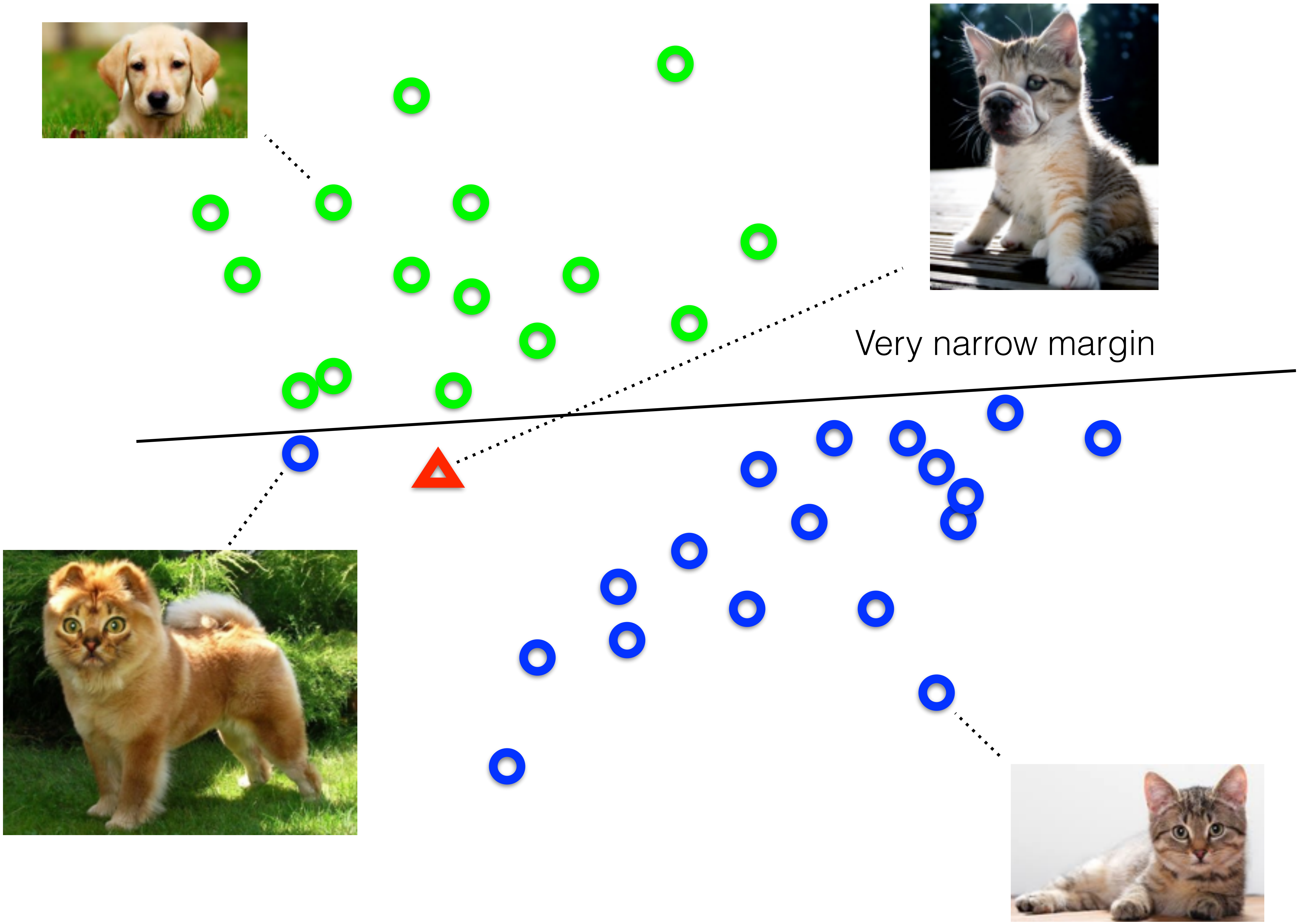
What's the best **w**?



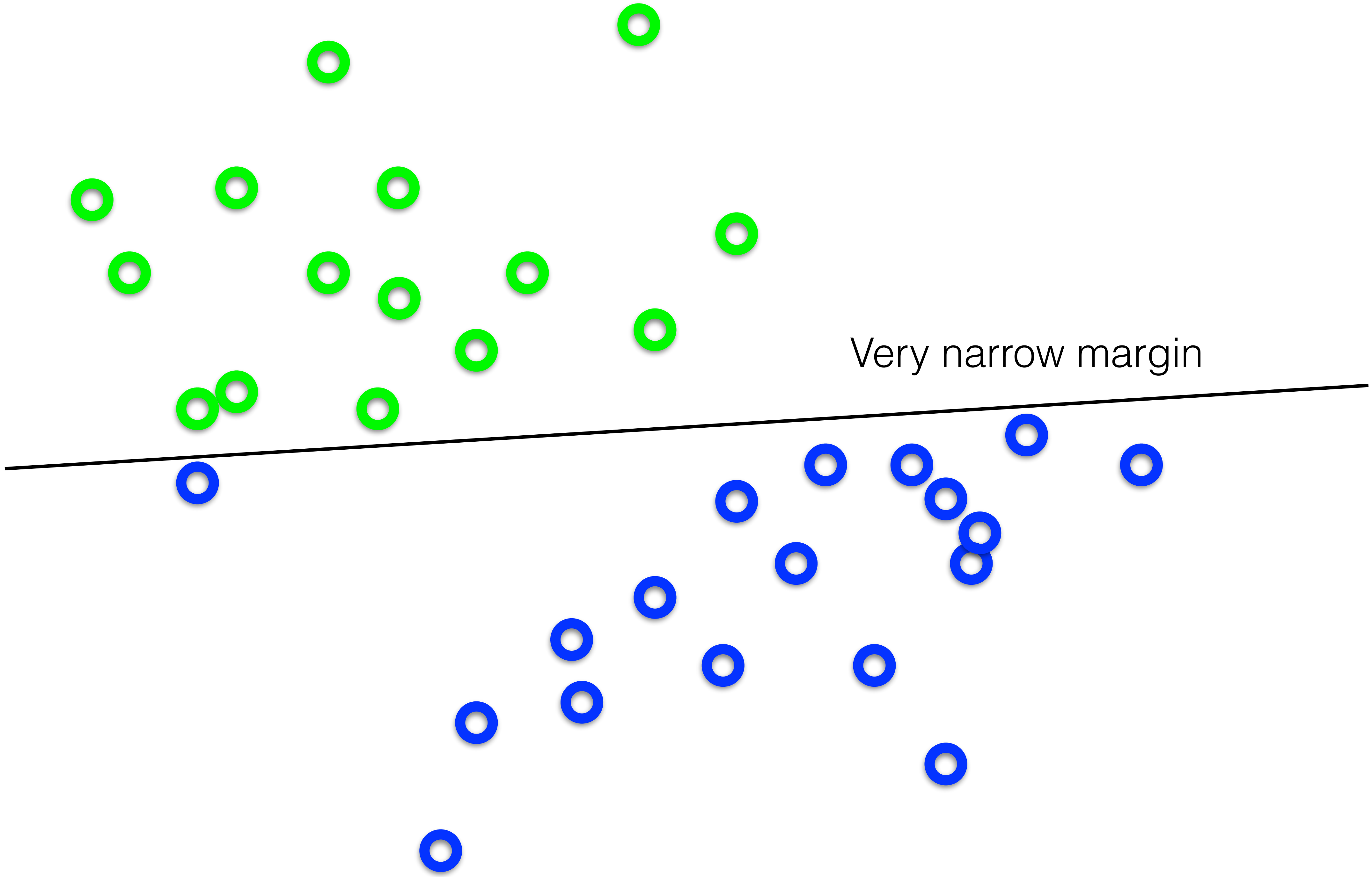
What's the best **w**?



Separating cats and dogs

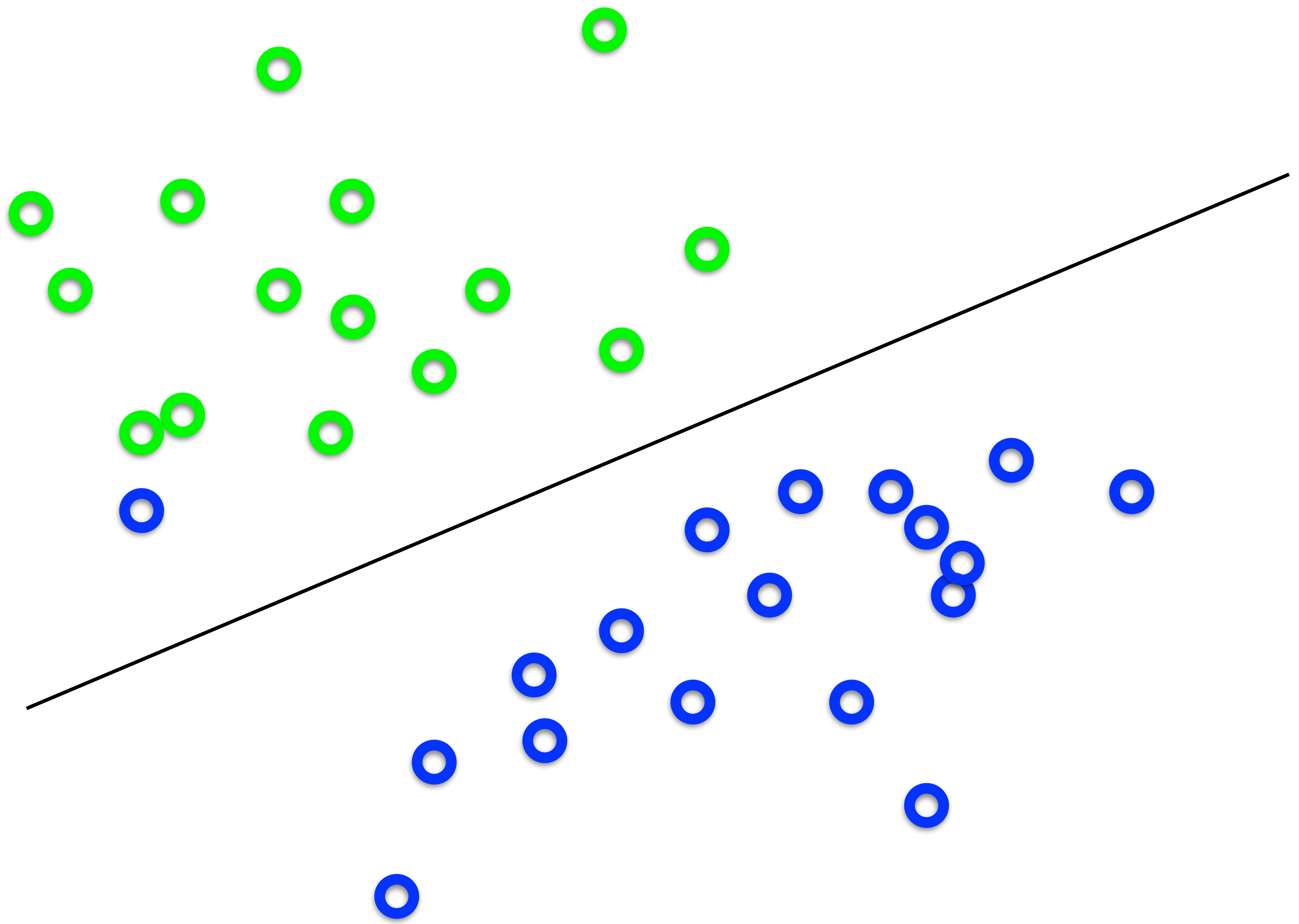


What's the best **w**?



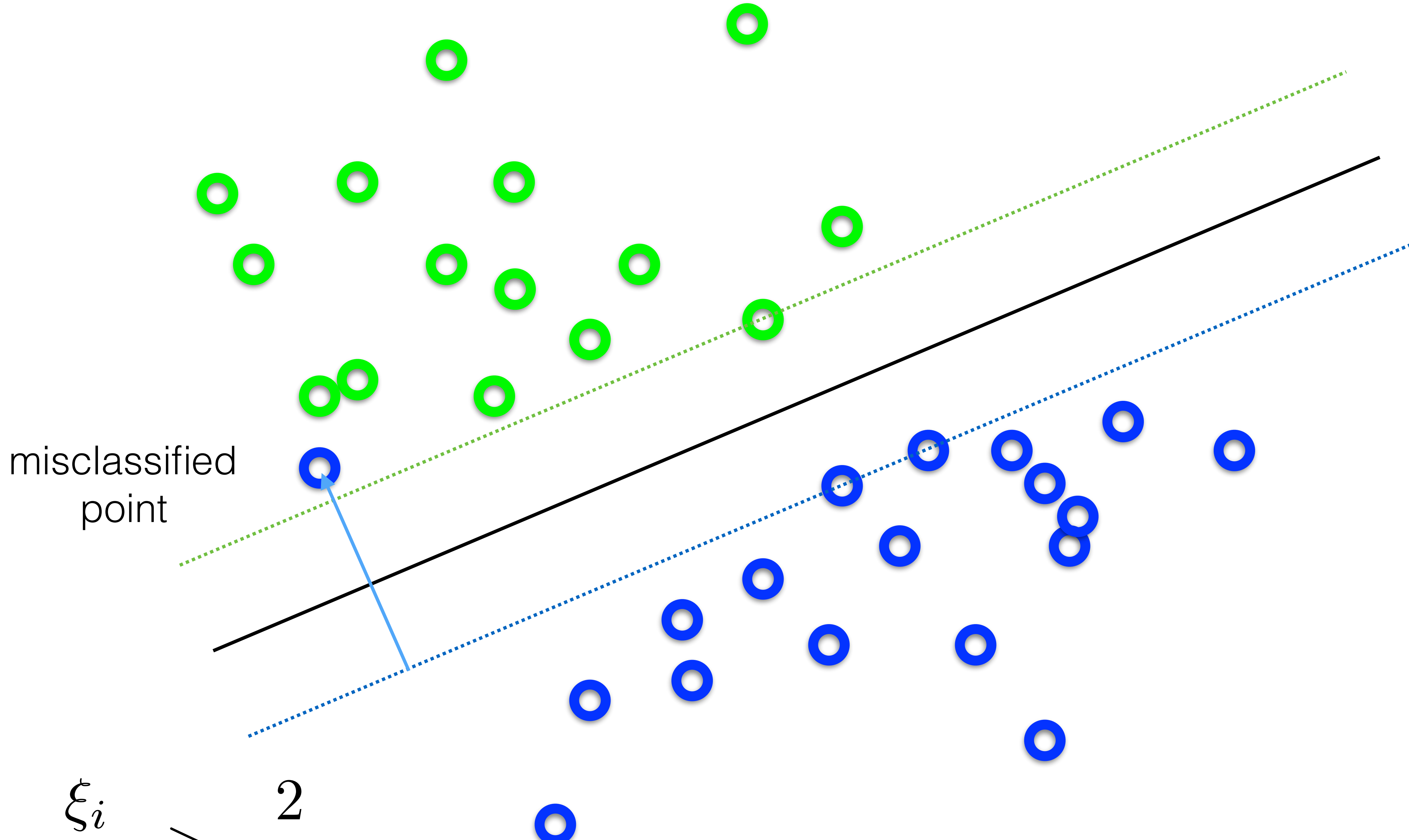
Intuitively, we should allow for some misclassification if we can get more robust classification

What's the best **w**?



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

Adding slack variables $\xi_i \geq 0$



$$\frac{\xi_i}{\|\boldsymbol{w}\|} > \frac{2}{\|\boldsymbol{w}\|}$$

‘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

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

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

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

'soft' margin

objective

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

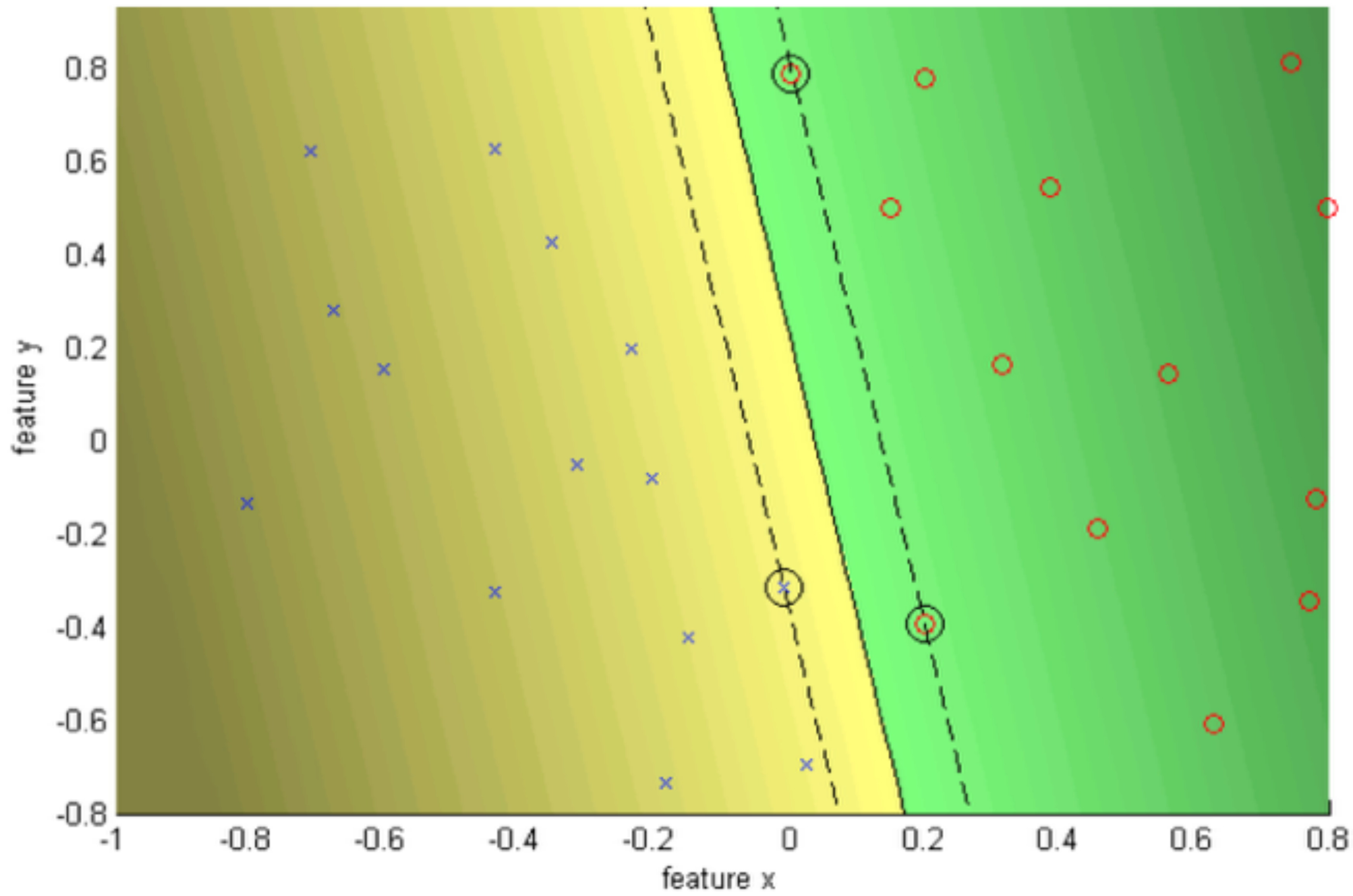
subject to

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

for $i = 1, \ldots, N$

- 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)

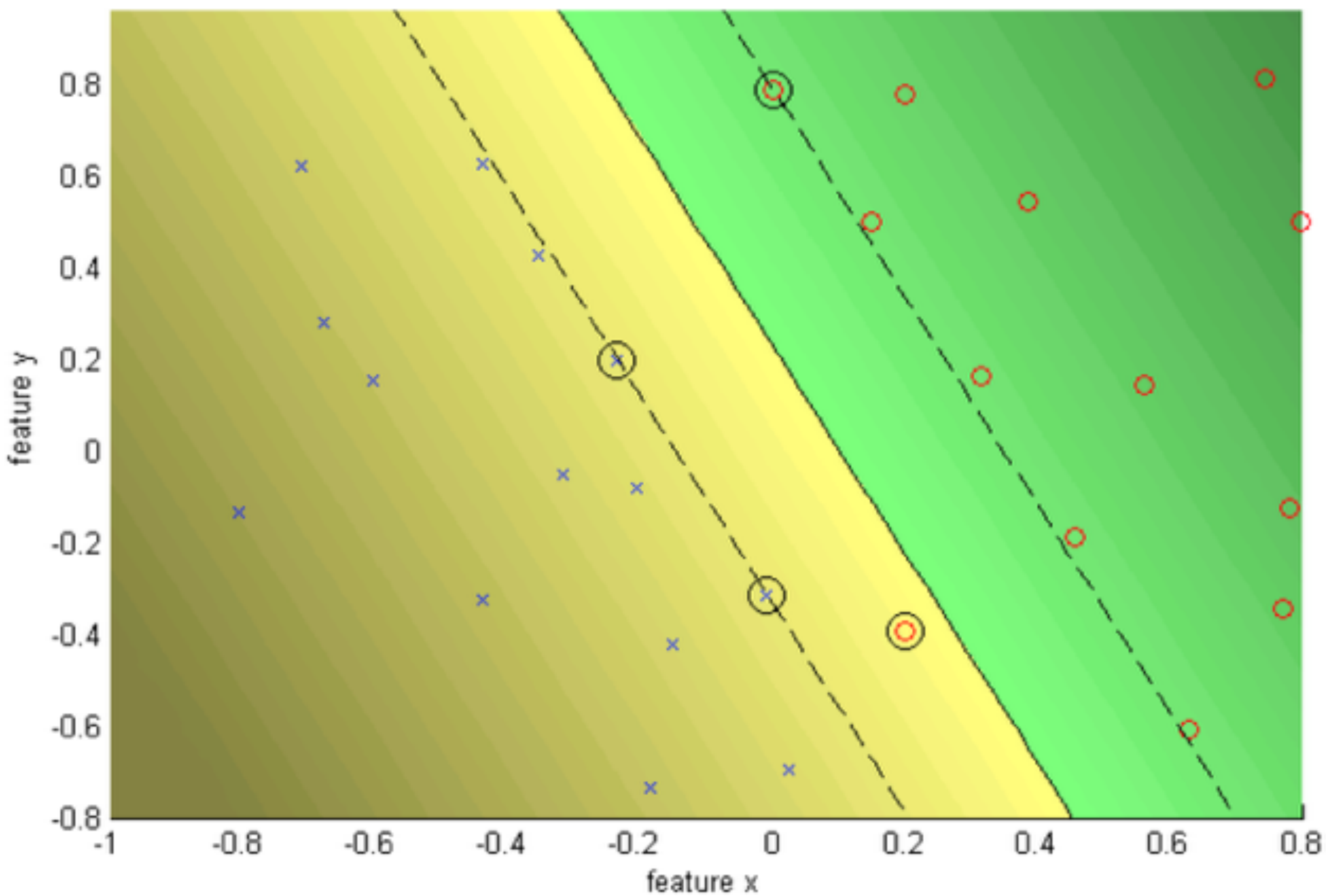
C = Infinity   hard margin



Comment Window

SVM (L1) by Sequential Minimal Optimizer
Kernel: linear (-), C: Inf
Kernel evaluations: 971
Number of Support Vectors: 3
Margin: 0.0966
Training error: 0.00%

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%