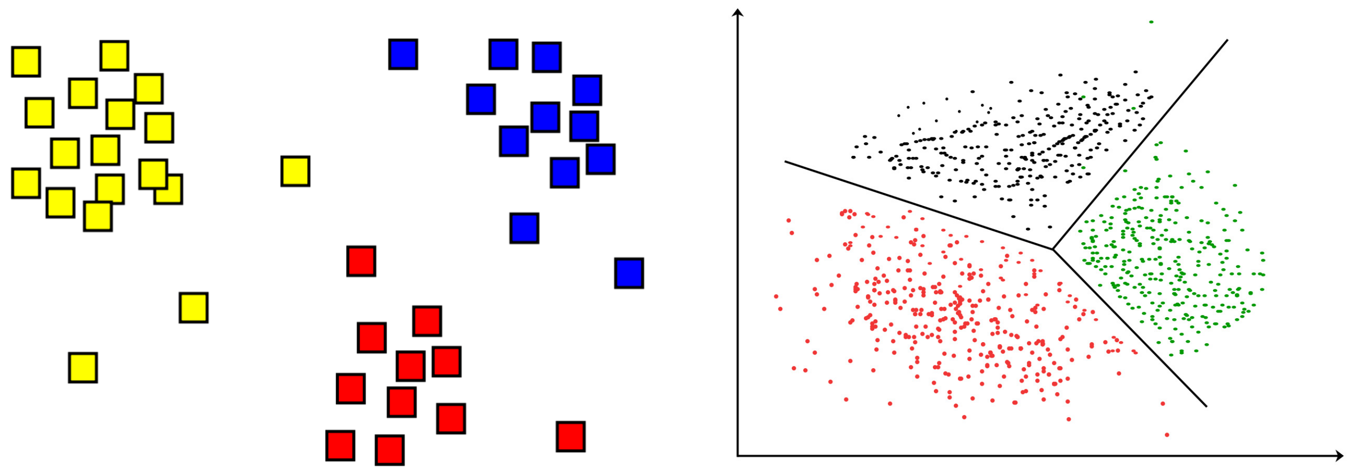
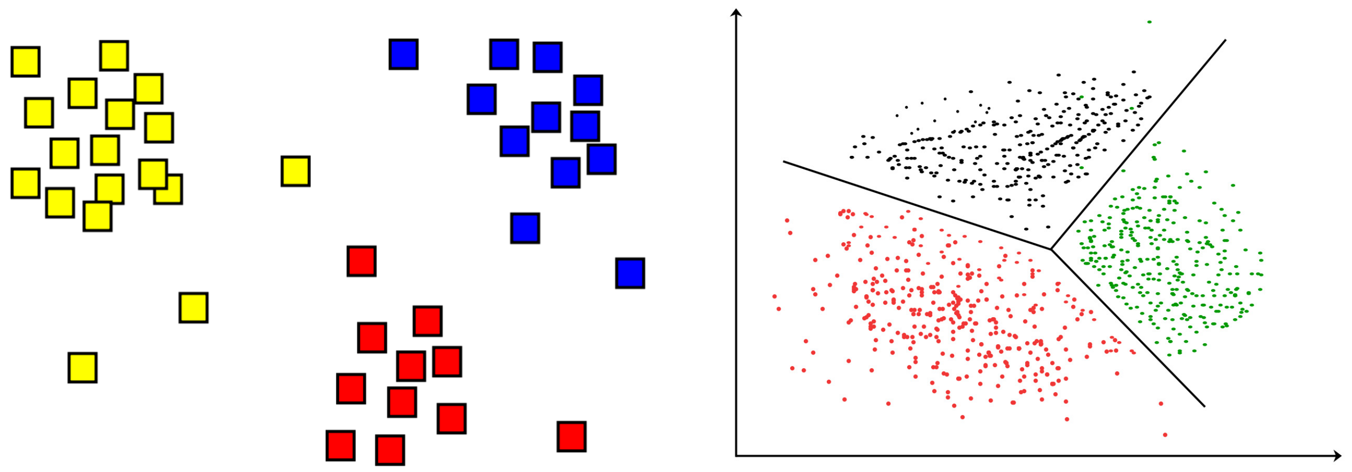
CPSC 425: Computer Vision

Lecture 21: Classification (cont)

Menu for Today (**March 21, 2019**)

Topics:

- Scene Classification
- Bag of Words Representation
- Decision Tree
- Boosting

Redings:

- **Today's** Lecture: Forsyth & Ponce (2nd ed.) 16.1.3, 16.1.4, 16.1.9
- **Next** Lecture: Forsyth & Ponce (2nd ed.) 17.1–17.2

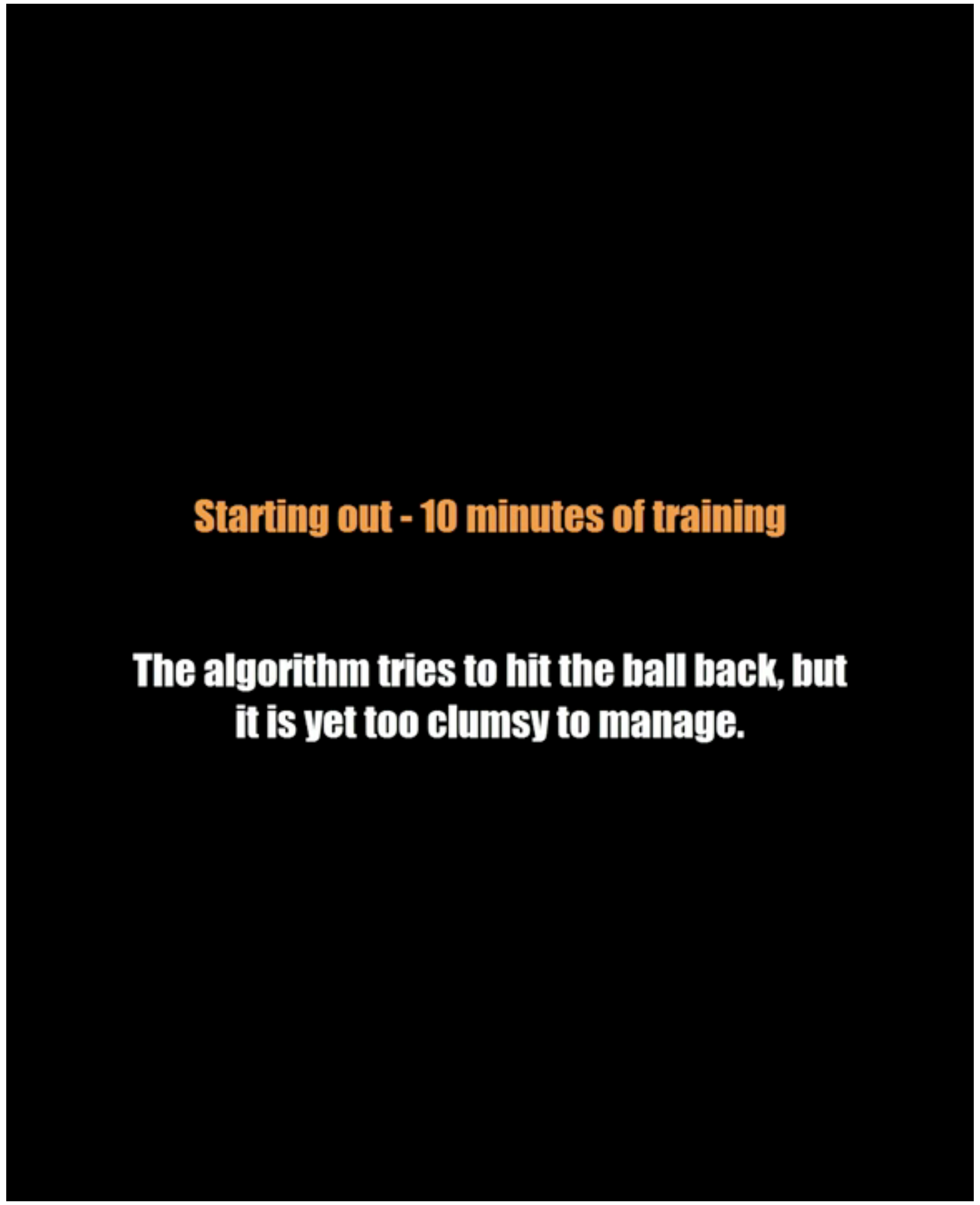
Reminders:

- **Assignment 5**: Scene Recognition with Bag of Words is **out**
- **Midterm solutions** are published on Piazza

Today’s “**fun**” Example: AlphaGo

Google DeepMind’s AlphaGo

Today's "**fun**" Example: AlphaGo



Lecture 20: Re-cap

— Collect a database of images with labels
— Use ML to train an image classifier
— Evaluate the classifier on test images

Label

Feature vector
computed from
the image

Example training set

Lecture 20: Re-cap Bayes Rule

Let c be the class label and let x be the measurement (i.e., evidence)

class–conditional probability
(a.k.a. likelihood)

prior probability

$$P(c|x) = \frac{P(x|c)p(c)}{P(x)}$$

posterior probability

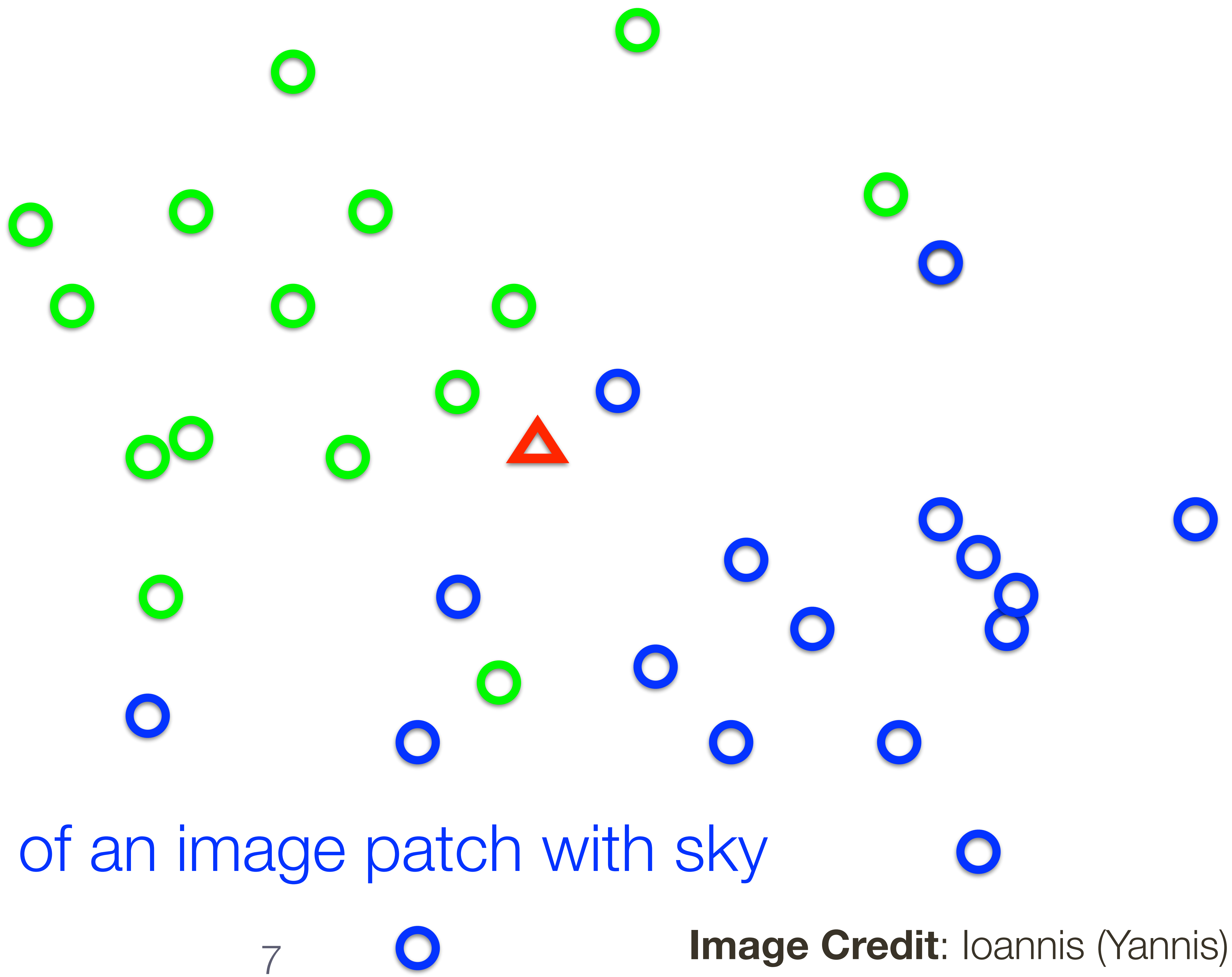
unconditional probability
(a.k.a. marginal likelihood)

Example: 2D Bayes Classifier

- 17 samples
- 15 samples

These could be (g,b) pixel value of an image patch with grass

Given a (g,b) pixel value from a new patch is it more likely to be be grass or sky?



These could be (g,b) pixel value of an image patch with sky

Image Credit: Ioannis (Yannis) Gkioulekas (CMU)

Example: 2D Bayes Classifier

- 17 samples
- 15 samples

$$p(blue) = \frac{17}{17+15}$$

$$p(green) = \frac{15}{17+15}$$

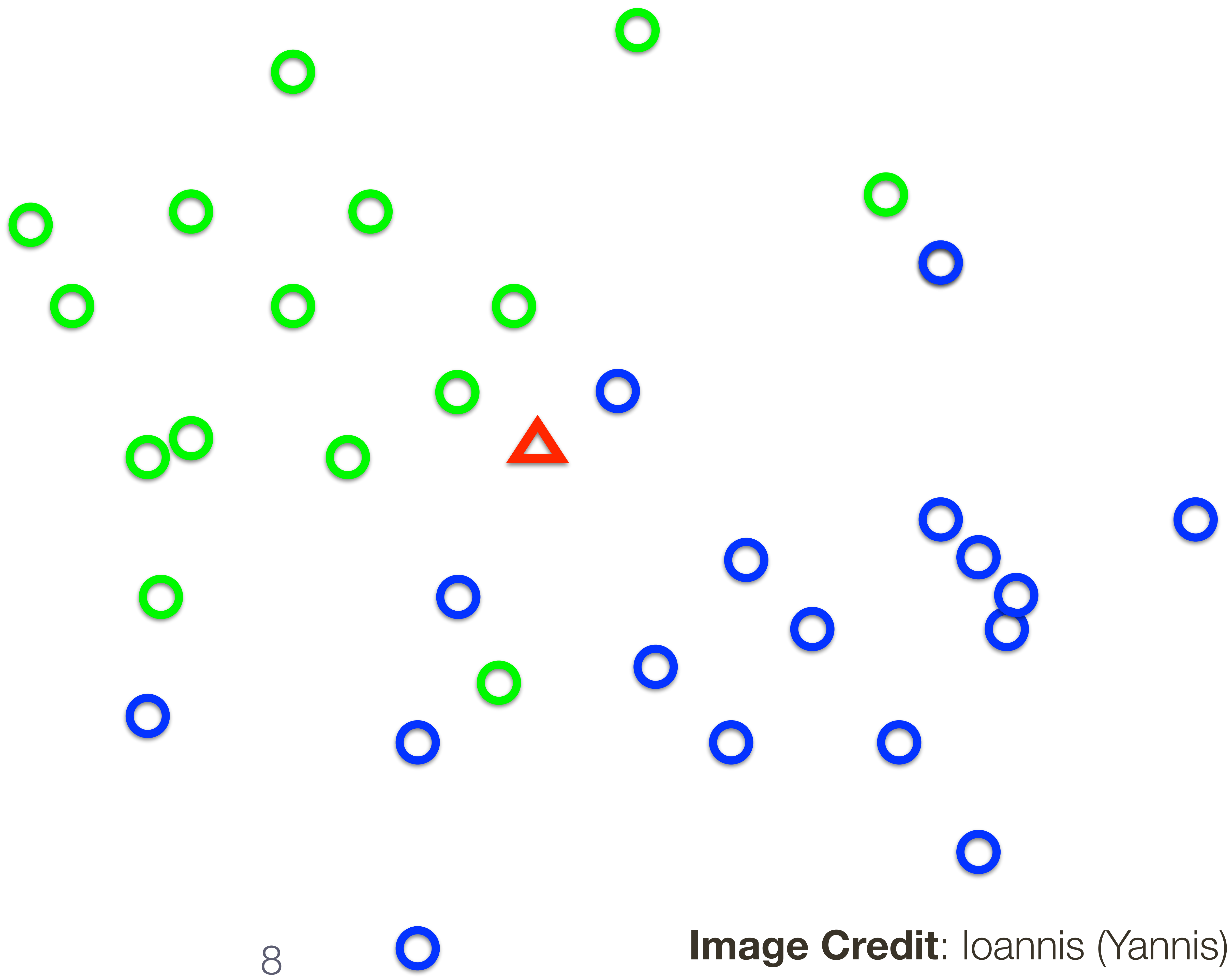


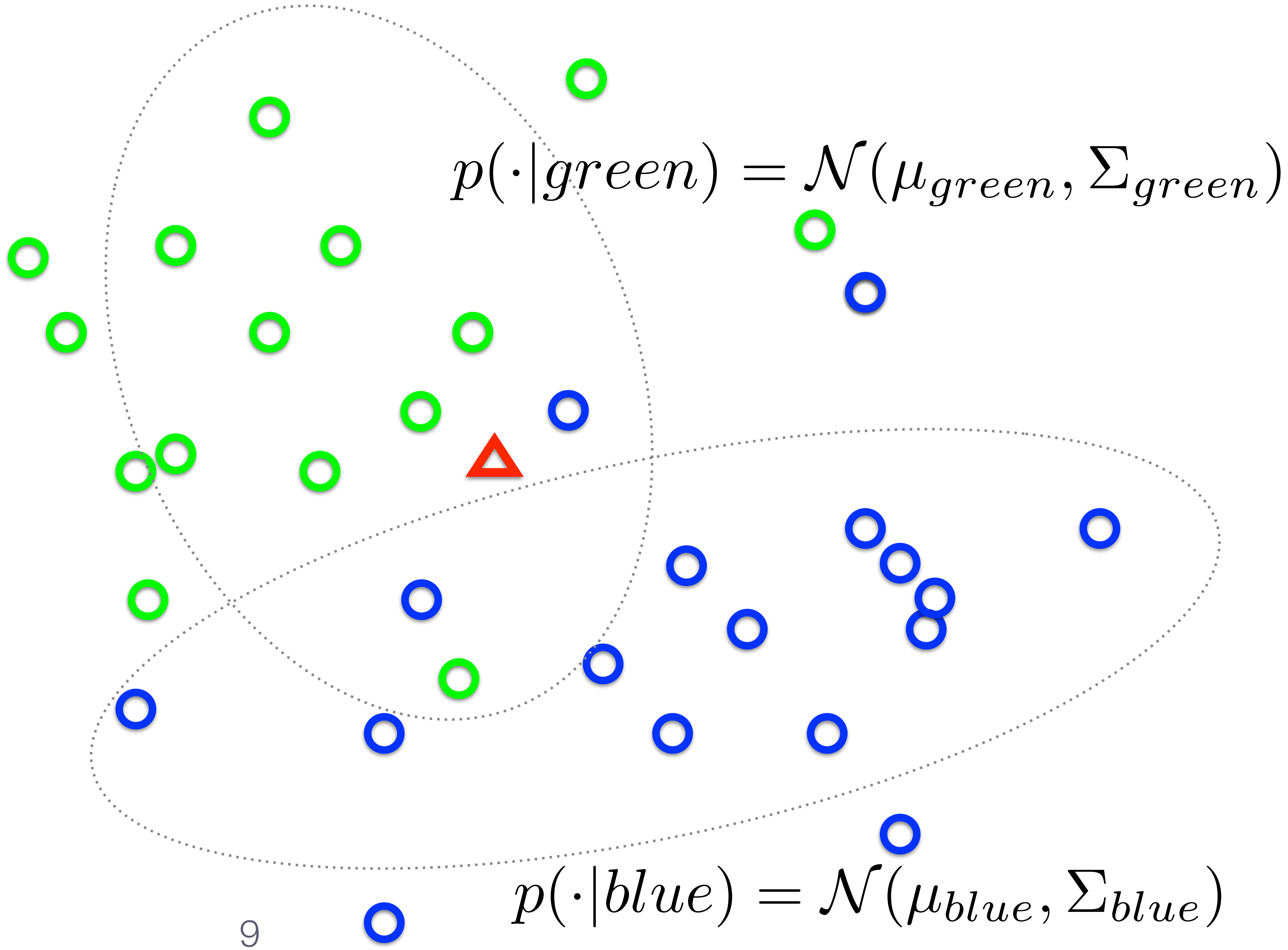
Image Credit: Ioannis (Yannis) Gkioulekas (CMU)

Example: 2D Bayes Classifier

- 17 samples (blue)
- 15 samples (green)

$$p(blue) = \frac{17}{17+15}$$

$$p(green) = \frac{15}{17+15}$$



Example: 2D Bayes Classifier

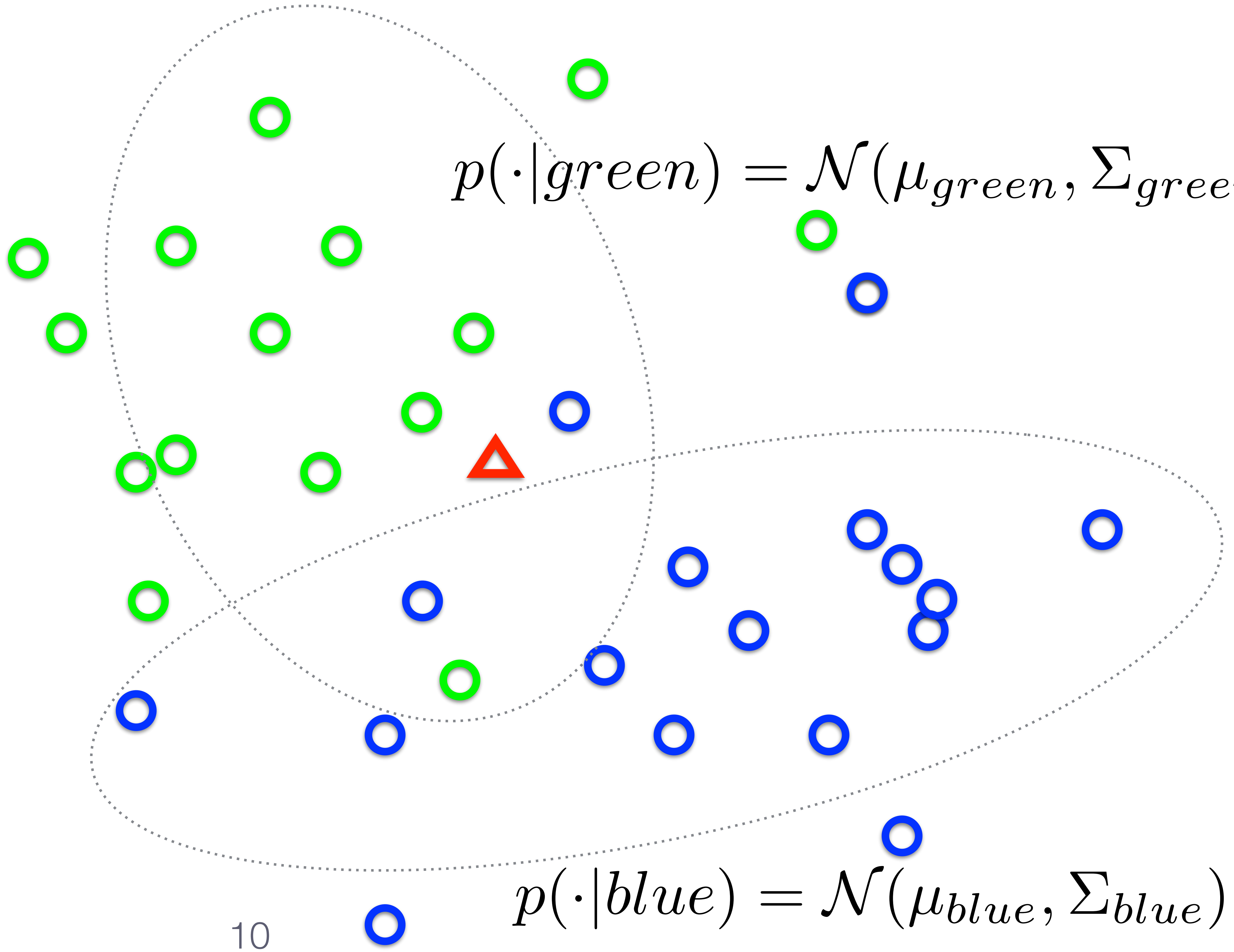
$$p(green|\triangle) \propto \mathcal{N}(\triangle; \mu_{green}, \Sigma_{green})p(green)$$

$$p(blue|\triangle) \propto \mathcal{N}(\triangle; \mu_{blue}, \Sigma_{blue})p(blue)$$

- 17 samples
- 15 samples

$$p(blue) = \frac{17}{17+15}$$

$$p(green) = \frac{15}{17+15}$$



Loss Functions and Classifiers

Loss

- Some errors may be more expensive than others

  Example: A fatal disease that is easily cured by a cheap medicine with no side-effects. Here, false positives in diagnosis are better than false negatives

- We discuss two class classification:
  $L(1 \rightarrow 2)$ is the loss caused by calling 1 a 2

Total risk of using classifier ***s*** is

$$R(s) = \Pr\{1 \rightarrow 2 \mid \text{using } \boldsymbol{s}\}\, L(1 \rightarrow 2) + \Pr\{2 \rightarrow 1 \mid \text{using } \boldsymbol{s}\}\, L(2 \rightarrow 1)$$

Two Class Classification

Generally, we should classify as 1 if the expected loss of classifying as 1 is less than for 2

Classify $\boldsymbol{x}$ as

$$1 \quad \text{if } p(1|\boldsymbol{x})\, L(1 \rightarrow 2) > p(2|\boldsymbol{x})\, L(2 \rightarrow 1)$$

$$2 \quad \text{if } p(1|\boldsymbol{x})\, L(1 \rightarrow 2) < p(2|\boldsymbol{x})\, L(2 \rightarrow 1)$$

Decision boundary: points where the loss is the same for either class.

Training Error, **Testing** Error, and Overfitting

Training error is the error a classifier makes on the training set

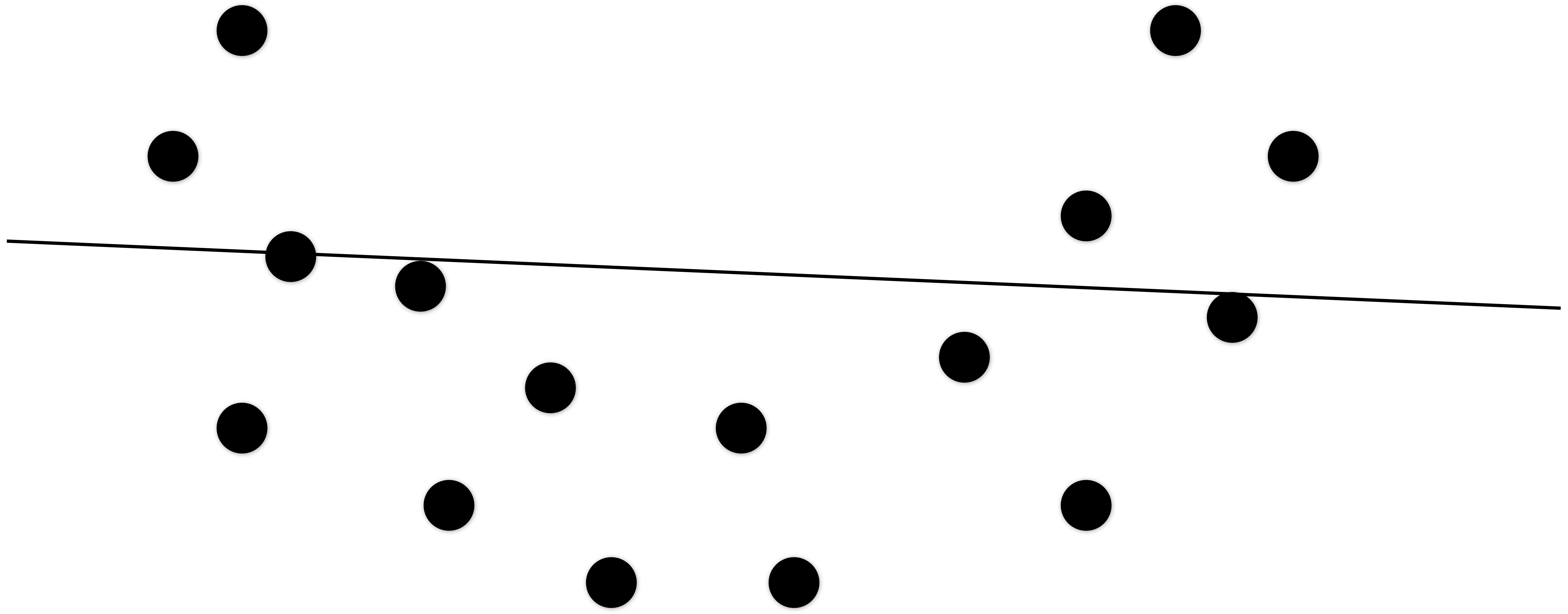
We want to minimize the **testing error** – the error the classifier makes on an unseen testing set

Classifiers that have small training error may not necessarily have small testing error

The phenomenon that causes testing error to be worse than training error is called **overfitting**

Training Error, Testing Error, and **Overfitting**

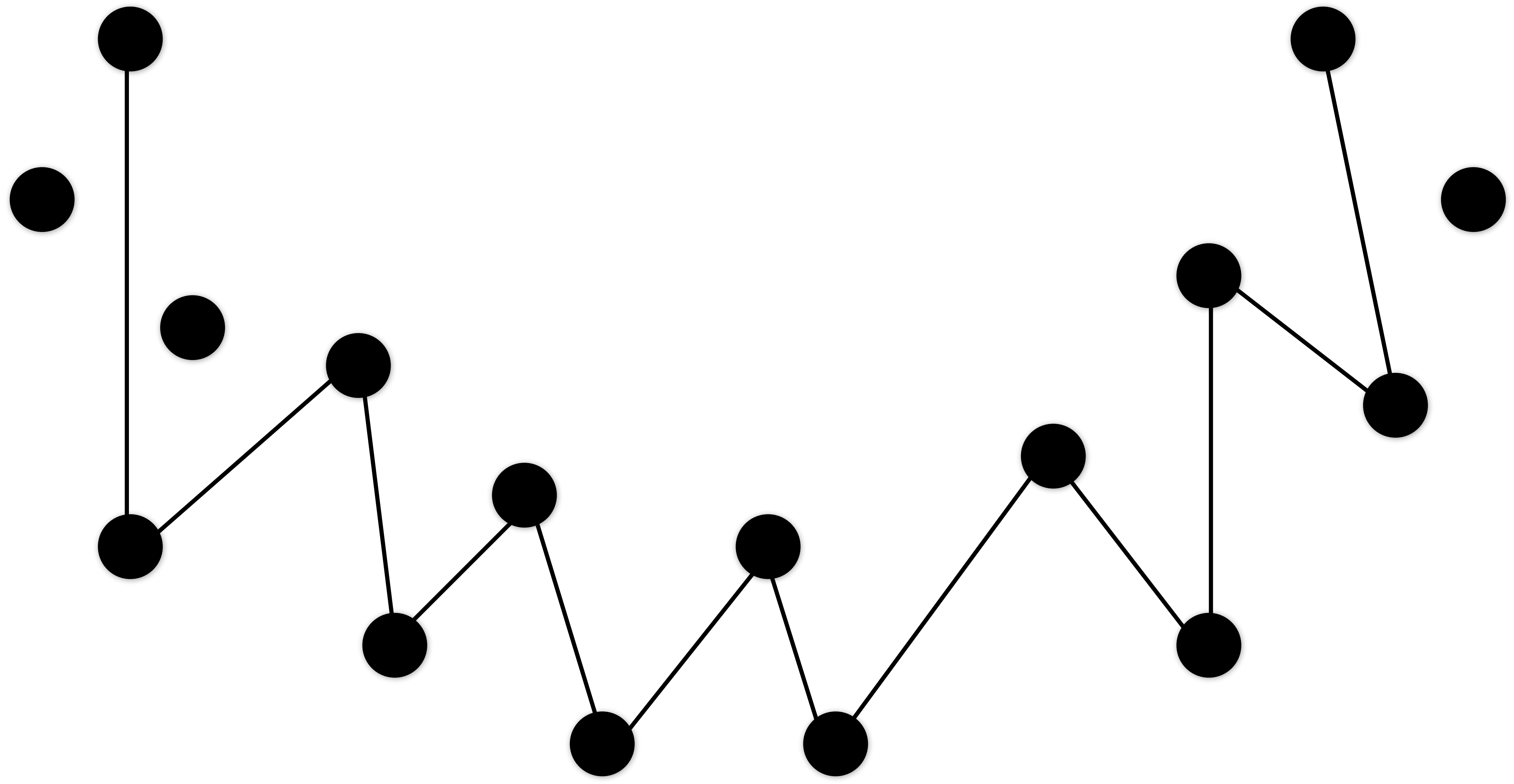
Underfitting: model is too simple to represent all the relevant class characteristics



Training Error, Testing Error, and **Overfitting**

Underfitting: model is too simple to represent all the relevant class characteristics

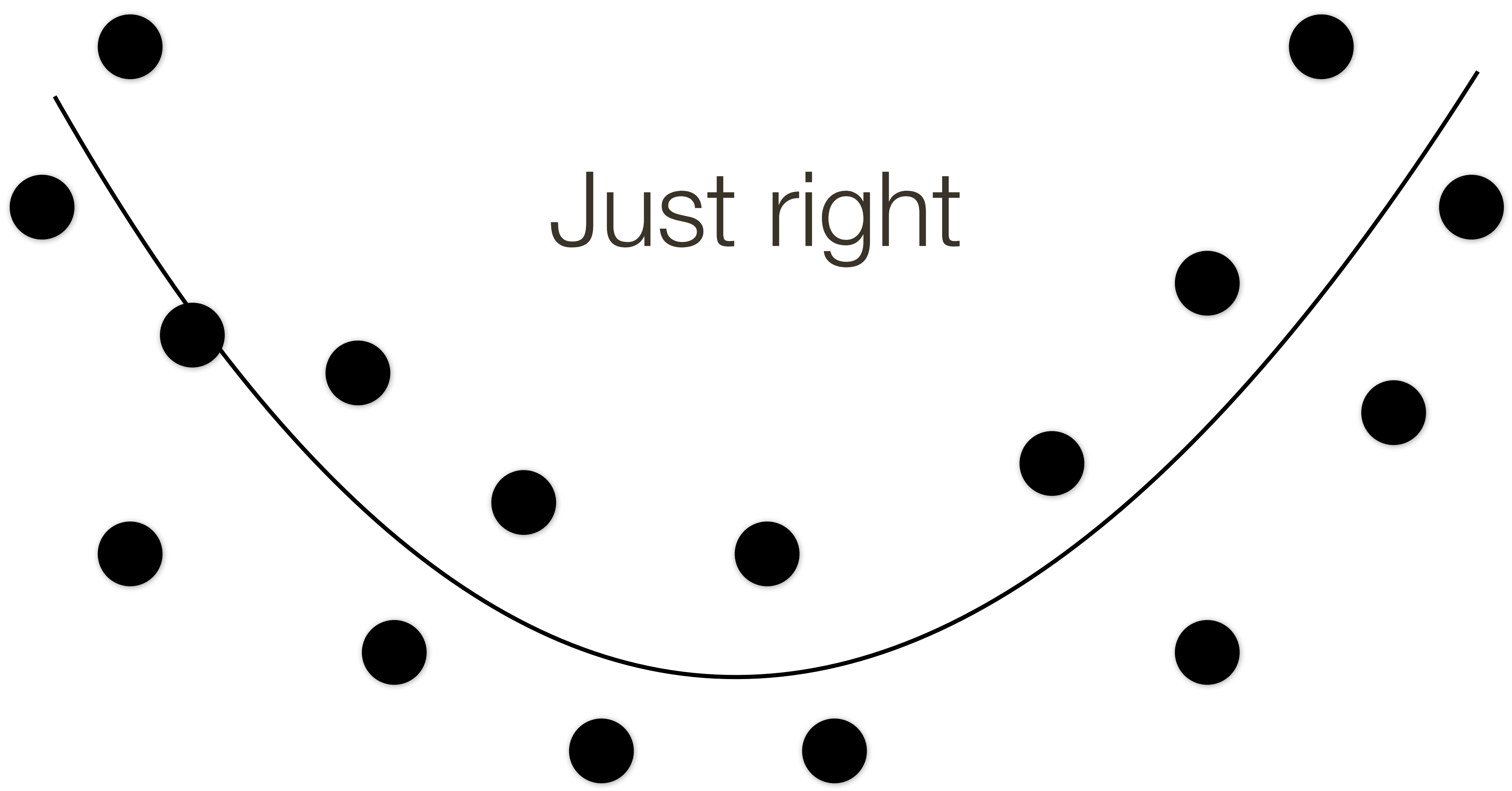
Overfitting: model is too complex and fits irrelevant characteristics (noise) in the data



Training Error, Testing Error, and **Overfitting**

Underfitting: model is too simple to represent all the relevant class characteristics

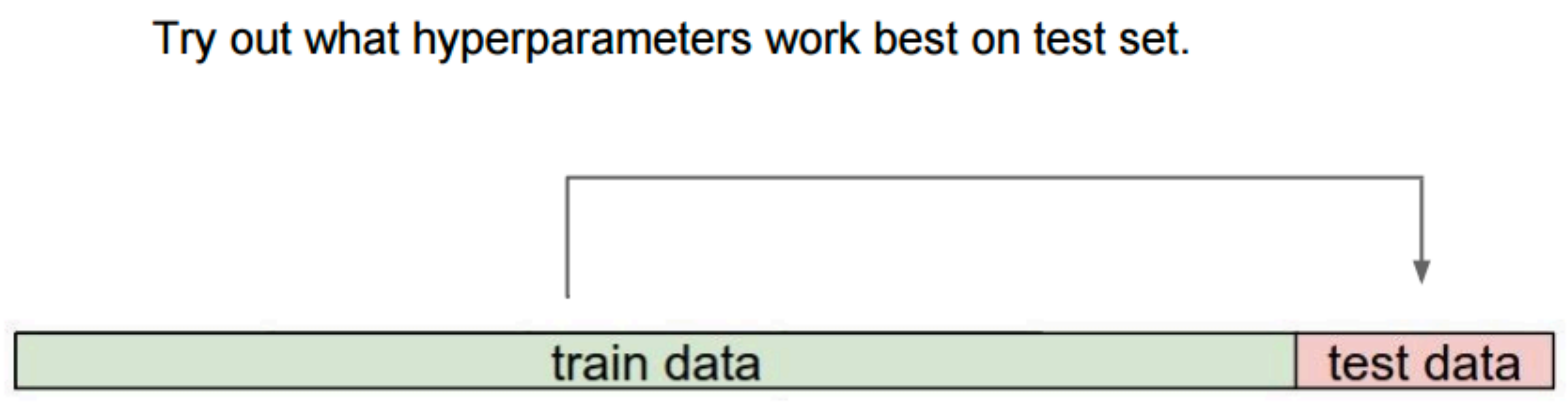
Overfitting: model is too complex and fits irrelevant characteristics (noise) in the data



Cross-Validation

We cannot reliably estimate the error rate of the classifier using the training set

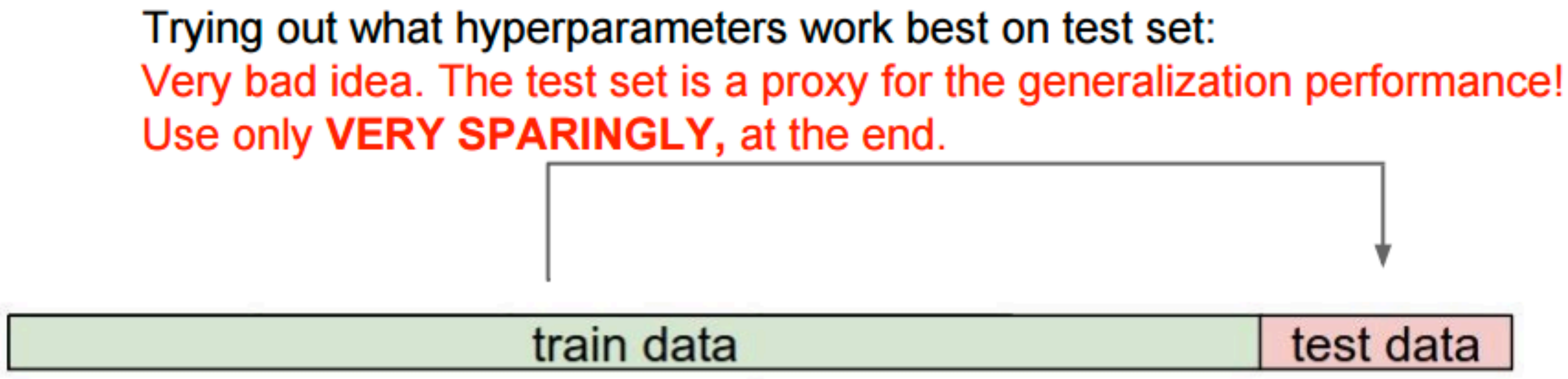
An alternative is to split some training data to form a **validation** set, then train the classifier on the rest of the data and evaluate on the validation set



Cross-Validation

We cannot reliably estimate the error rate of the classifier using the training set

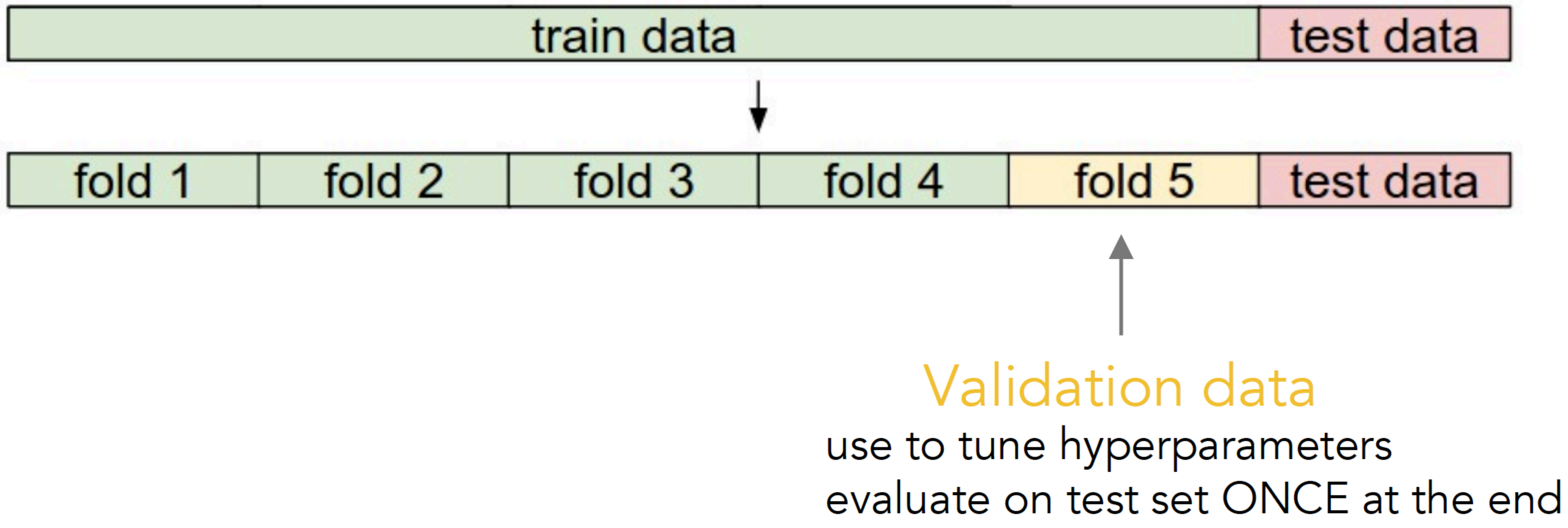
An alternative is to split some training data to form a **validation** set, then train the classifier on the rest of the data and evaluate on the validation set



Cross-Validation

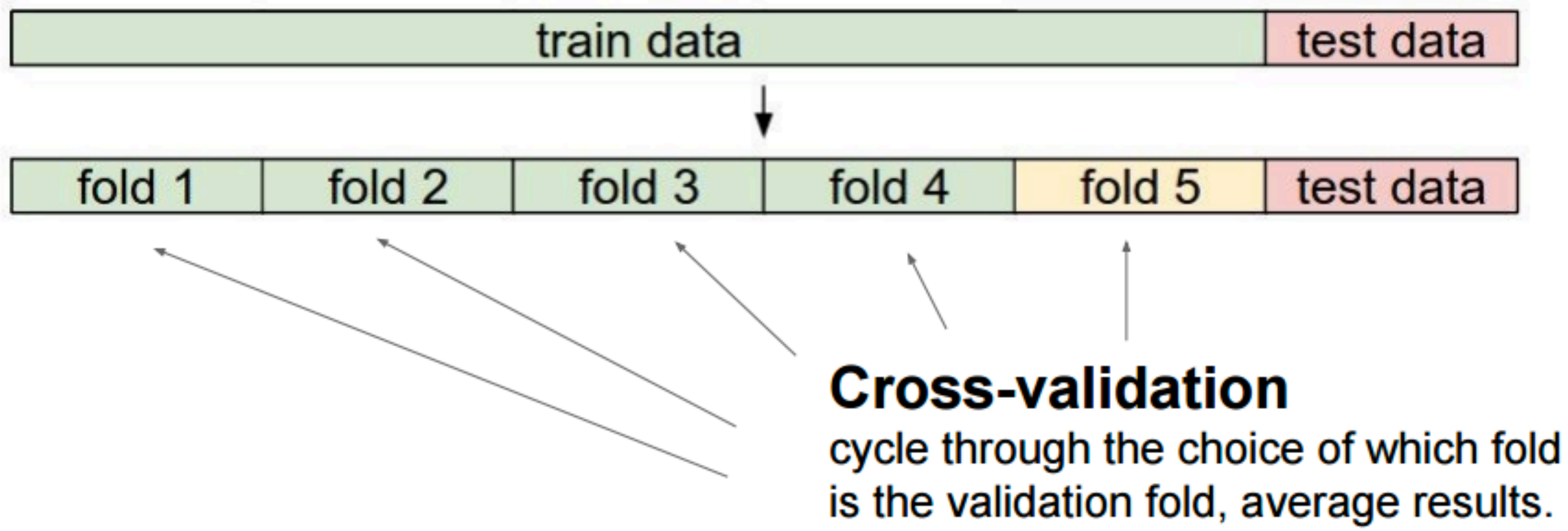
We cannot reliably estimate the error rate of the classifier using the training set

An alternative is to split some training data to form a **validation** set, then train the classifier on the rest of the data and evaluate on the validation set



Cross-Validation

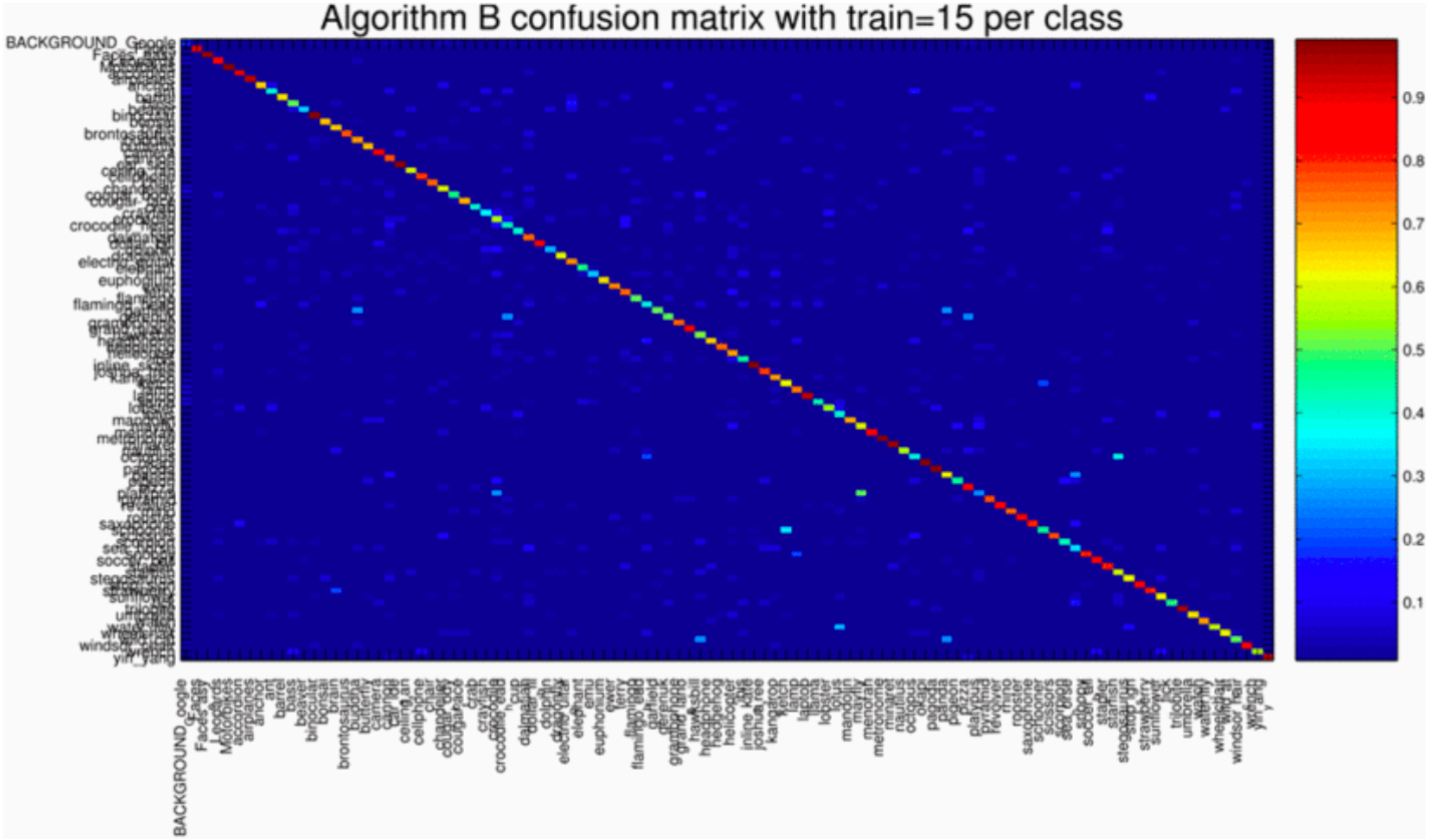
Cross-validation involves performing multiple splits and averaging the error over all splits



Confusion Matrix

When evaluating a multi-class classifier, it may be useful to know how often certain classes are often misclassified as others.

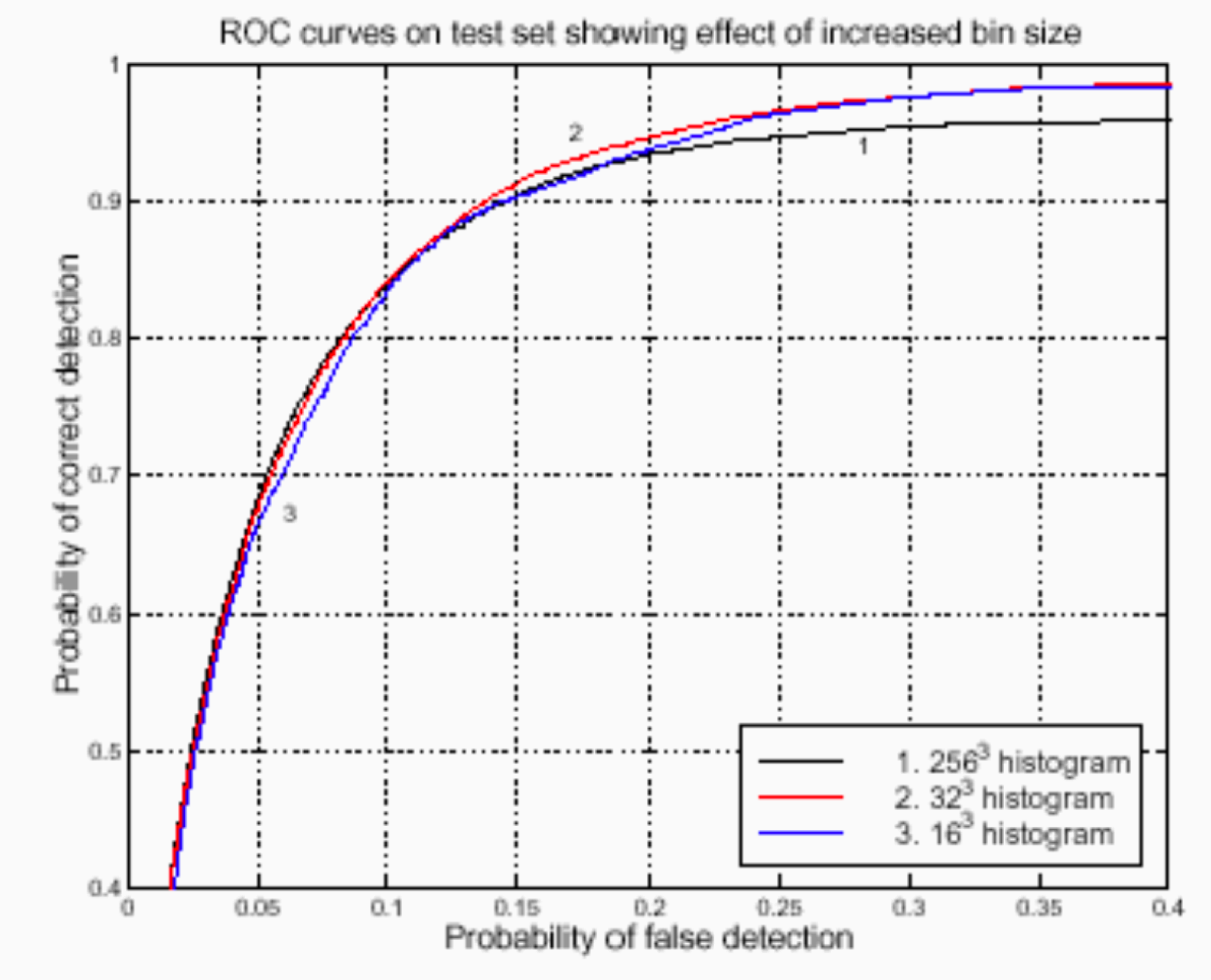
A **confusion matrix** is a table whose (i,j)th entry is the frequency (or proportion) an item of true class i was labelled as j by the classifier.



Forsyth & Ponce (2nd ed.) Figure 15.3. Original credit: H. Zhang et al., 2006.

Receiver Operating Characteristics (ROC)

ROC curves plot trade-off between false positives and false negatives



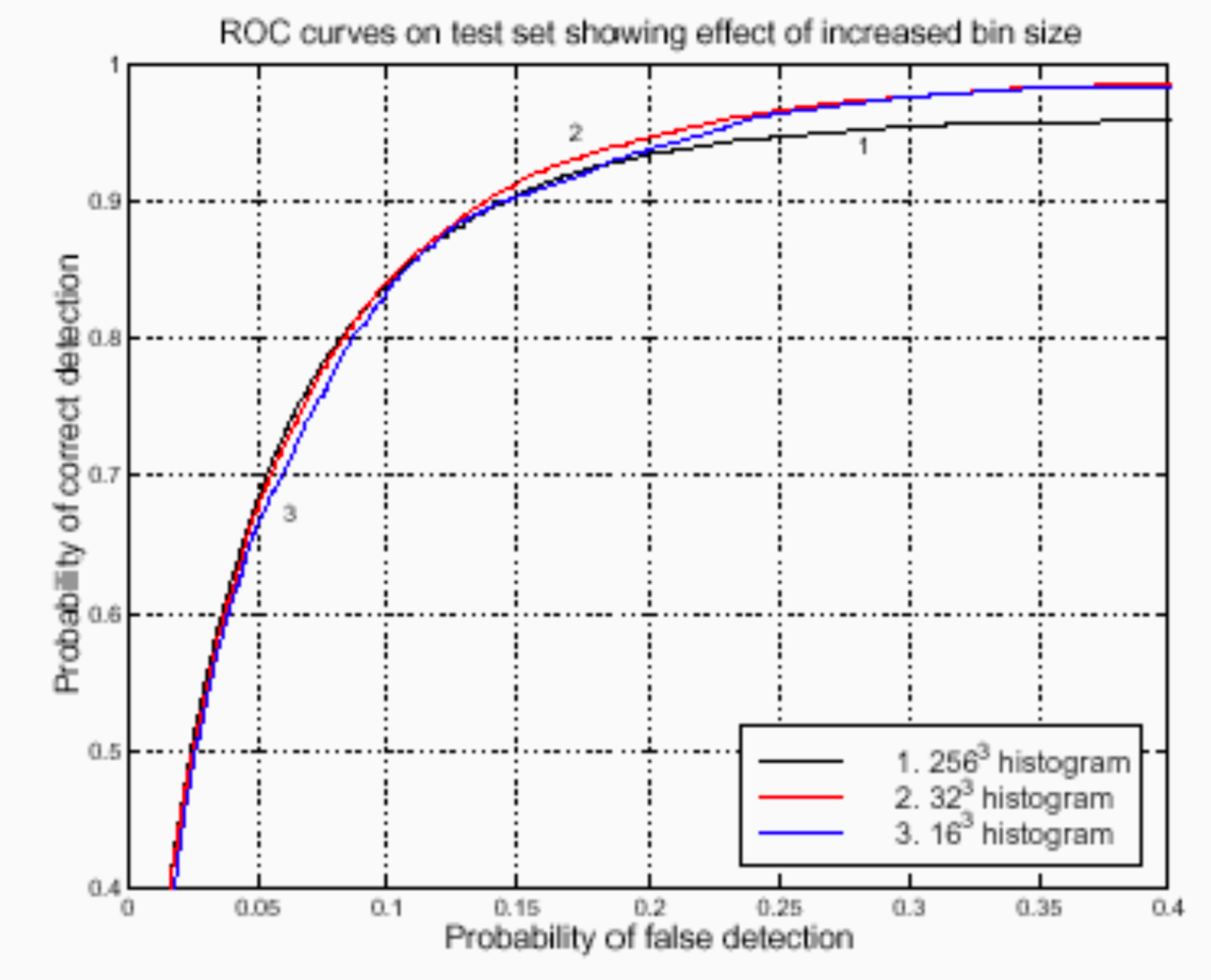
Forsyth & Ponce (2nd ed.) Figure 15.4

Figure from M. J. Jones and J. Rehg, “Statistical color models with application to skin detection,” Proc. CVPR, 1999, IEEE

Receiver Operating Characteristics (ROC)

What is a ROC curve for a perfect classifier?

ROC curves plot trade-off between false positives and false negatives



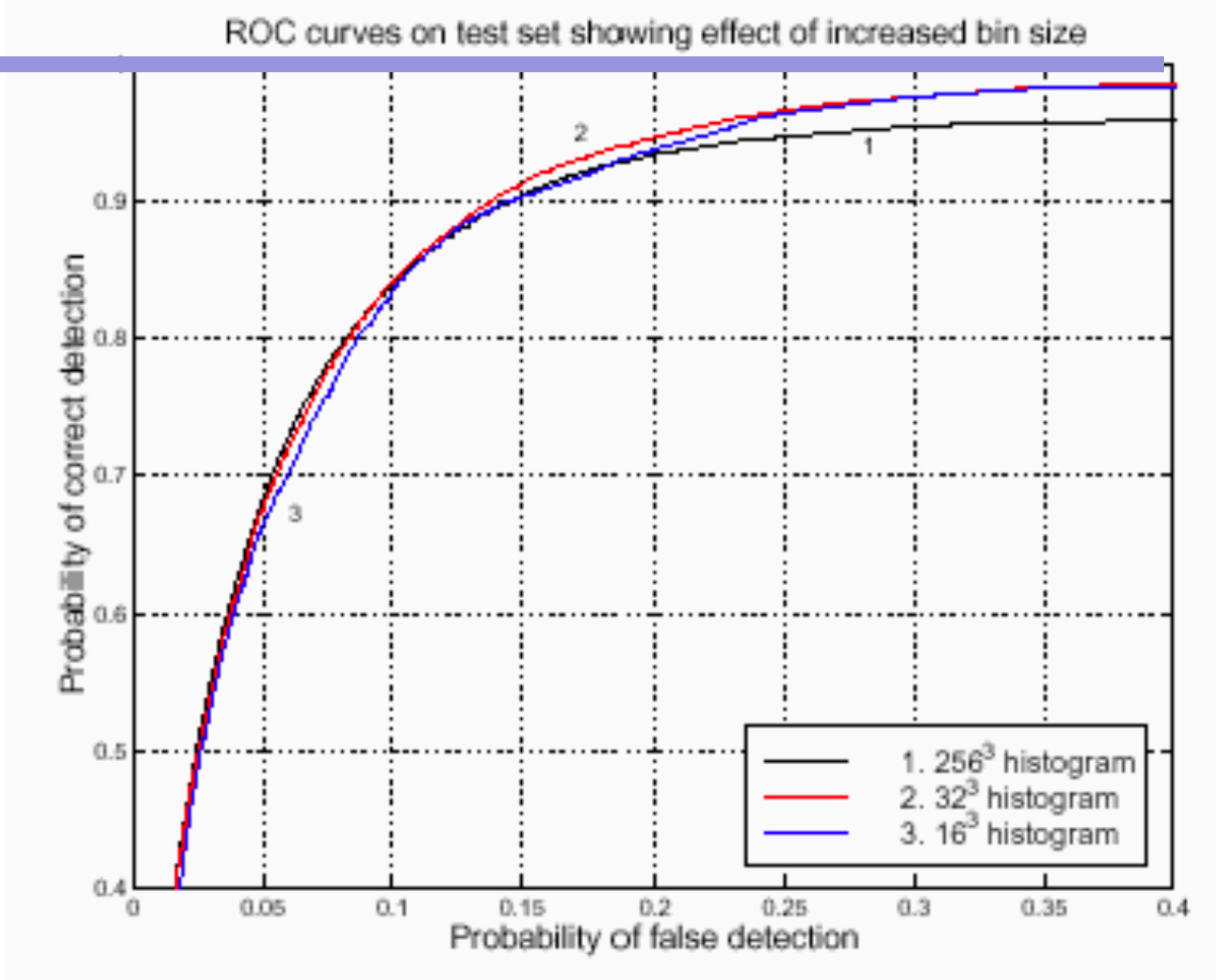
Forsyth & Ponce (2nd ed.) Figure 15.4

Figure from M. J. Jones and J. Rehg, “Statistical color models with application to skin detection,” Proc. CVPR, 1999, IEEE

Receiver Operating Characteristics (ROC)

What is a ROC curve for a perfect classifier?

ROC curves plot trade-off between false positives and false negatives



Forsyth & Ponce (2nd ed.) Figure 15.4

Figure from M. J. Jones and J. Rehg, “Statistical color models with application to skin detection,” Proc. CVPR, 1999, IEEE

Classifier Strategies

Classification strategies fall under two broad types: **parametric** and **non-parametric**.

Classifier Strategies

Classification strategies fall under two broad types: **parametric** and **non-parametric**.

Parametric classifiers are **model driven**. The parameters of the model are learned from training examples. New data points are classified by the learned model.
— fast, compact
— flexibility and accuracy depend on model assumptions

Classifier Strategies

Classification strategies fall under two broad types: **parametric** and **non-parametric**.

Parametric classifiers are **model driven**. The parameters of the model are learned from training examples. New data points are classified by the learned model.
— fast, compact
— flexibility and accuracy depend on model assumptions

Non-parametric classifiers are **data driven**. New data points are classified by comparing to the training examples directly. "The data is the model".
— slow
— highly flexible decision boundaries

Nearest Neighbor Classifier

Given a new data point, assign the label of nearest training example in feature space.

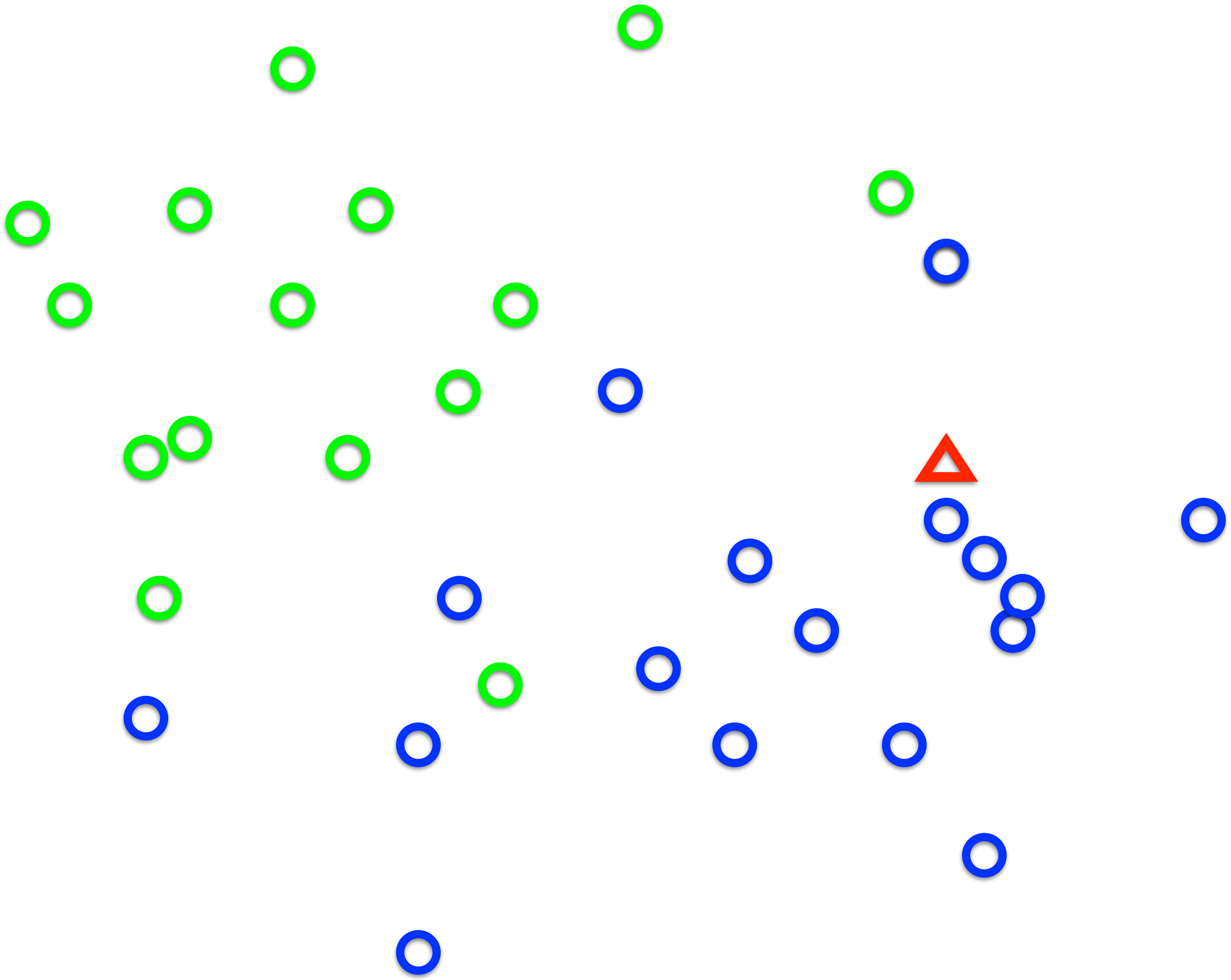


Image Credit: Ioannis (Yannis) Gkioulekas (CMU)

Nearest Neighbor Classifier

Given a new data point, assign the label of nearest training example in feature space.

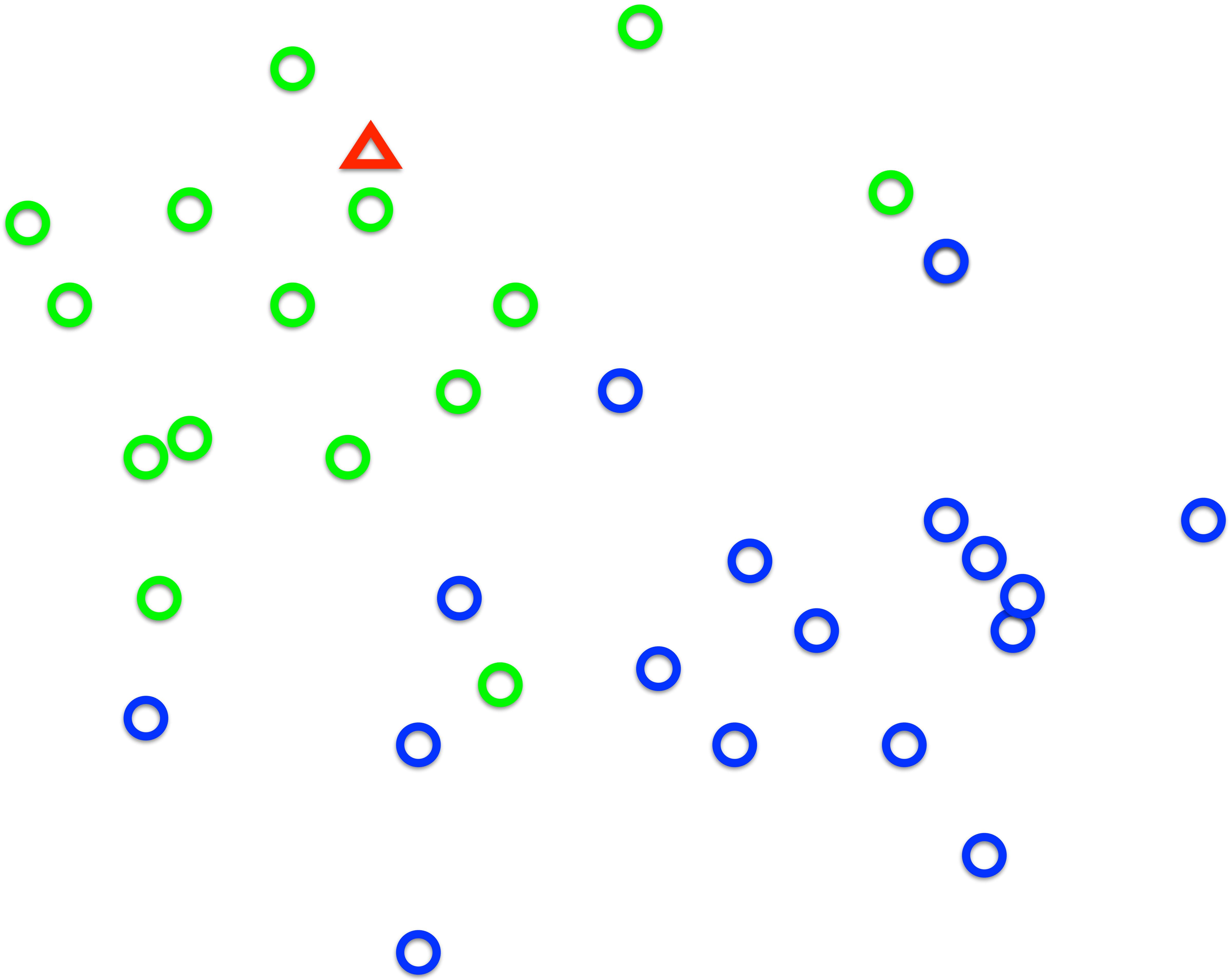


Image Credit: Ioannis (Yannis) Gkioulekas (CMU)

k-**Nearest Neighbor** (kNN) Classifier

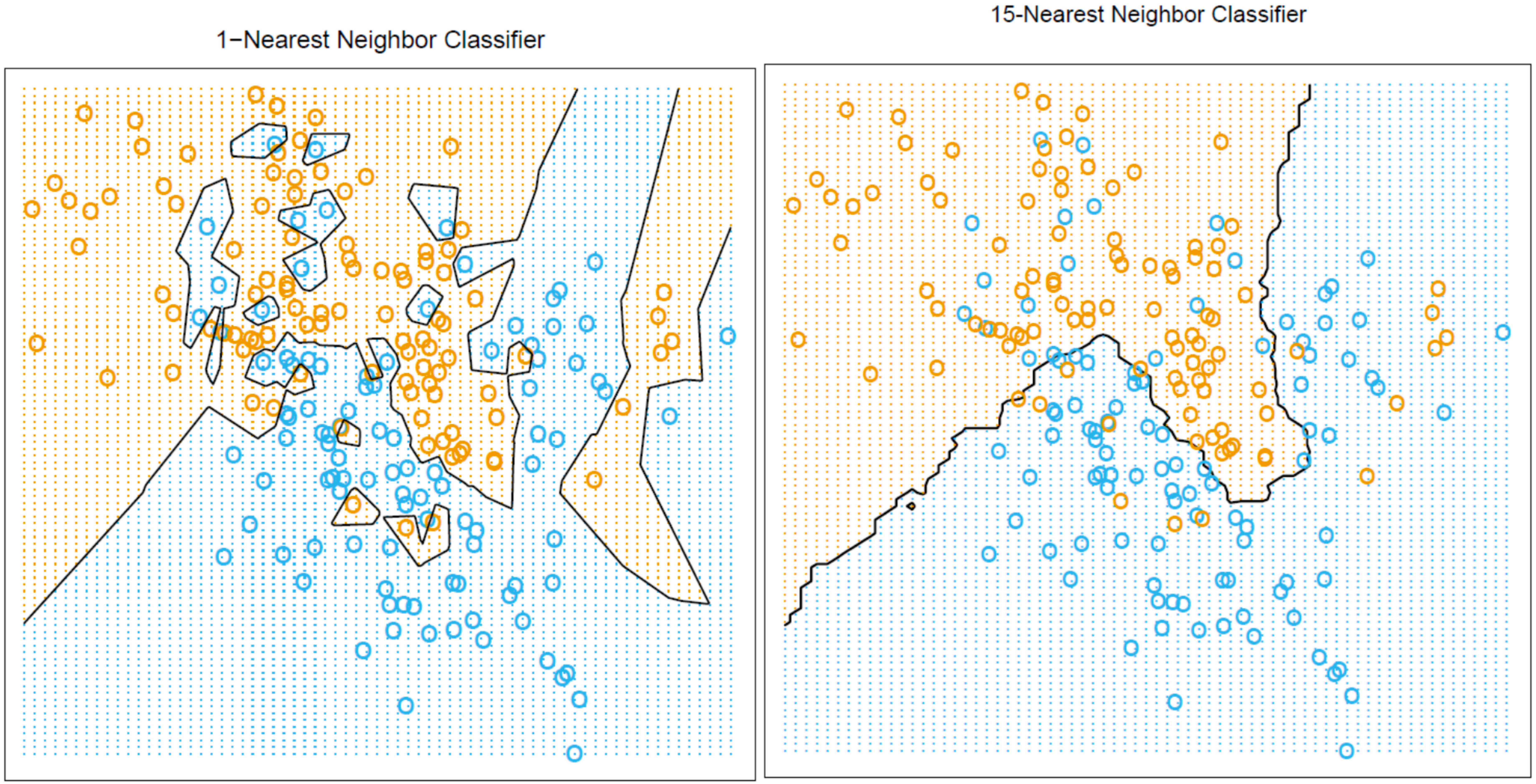
We can gain some robustness to noise by voting over multiple neighbours.

Given a new data point, find the k nearest training examples. Assign the label by majority vote.

Simple method that works well if the distance measure correctly weights the various dimensions

For large data sets, as k increases kNN approaches optimality in terms of minimizing probability of error

k-**Nearest Neighbor** (kNN) Classifier



kNN decision boundaries respond to local clusters where one class dominates

Figure credit: Hastie, Tibshirani & Friedman (2nd ed.)

Support Vector Machines (SVM)

Idea: Try to obtain the decision boundary directly

The decision boundary is parameterized as a **separating hyperplane** in feature space.
— e.g. a separating line in 2D

We choose the hyperplane that is as far as possible from each class - that maximizes the distance to the closest point from either class.

Linear Classifier

Defines a score function:

image features

$$f(\mathbf{x}_i, \mathbf{W}, \mathbf{b}) = \mathbf{W}\mathbf{x}_i + \mathbf{b}$$

weights (parameters)

bias vector

Image Credit: Ioannis (Yannis) Gkioulekas (CMU)

Linear Classifier

Example with an image with 4 pixels, and 3 classes (cat/dog/ship)

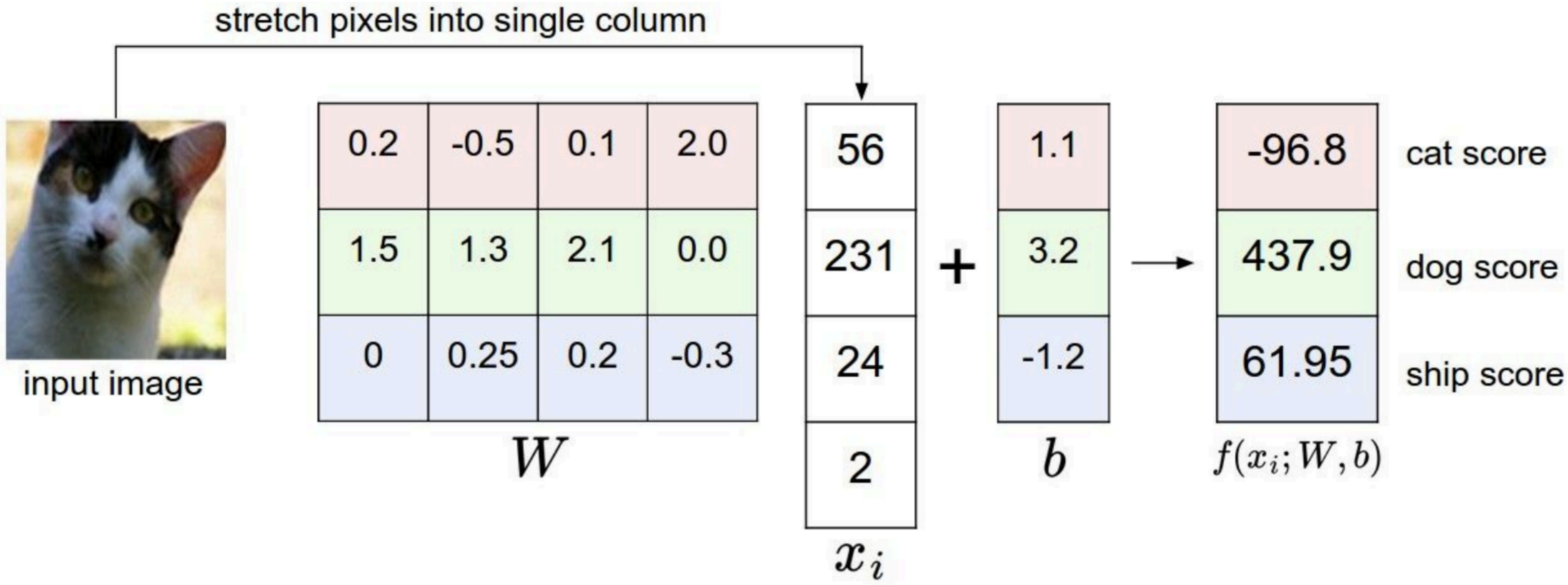
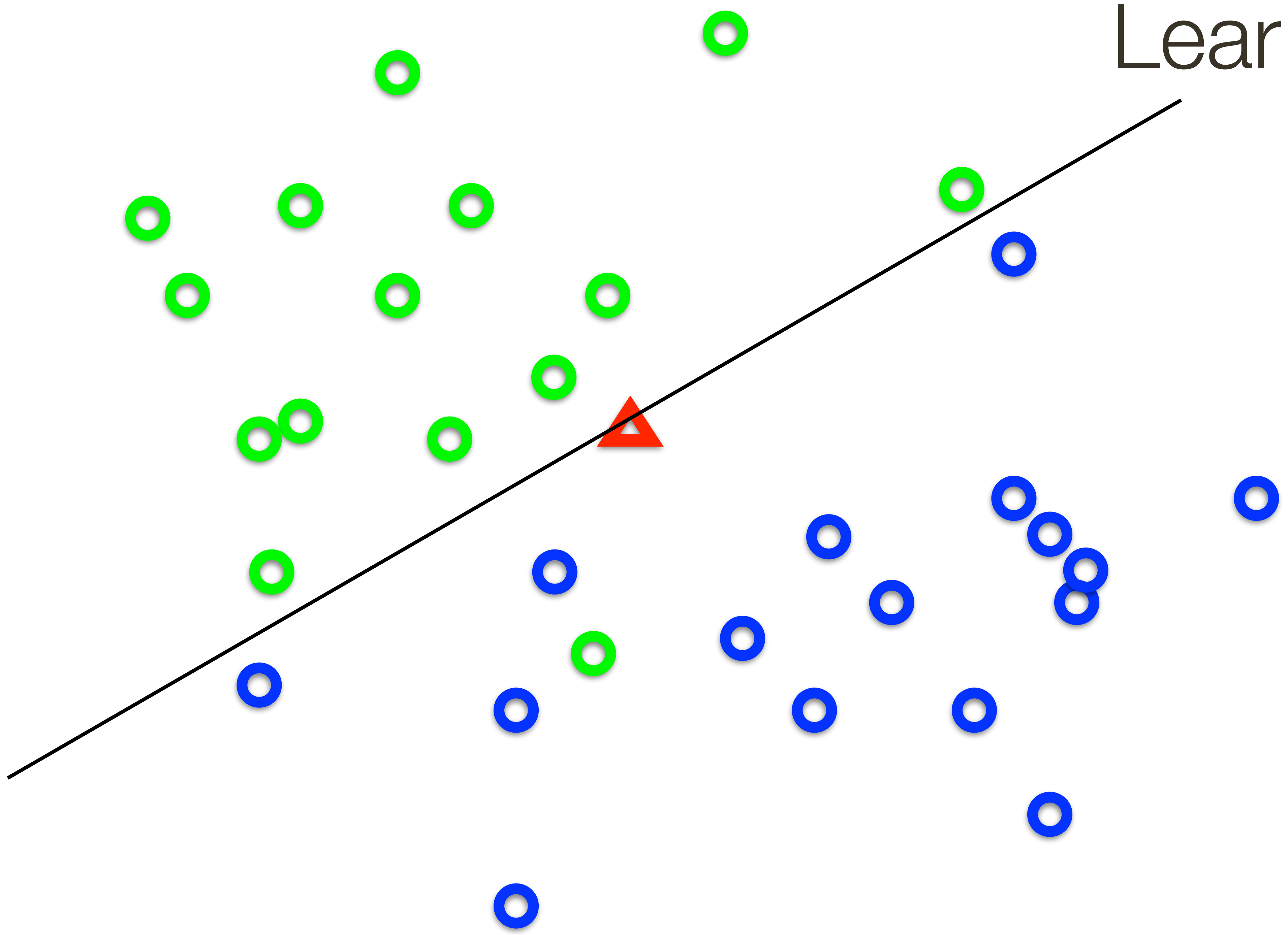


Image Credit: Ioannis (Yannis) Gkioulekas (CMU)

Support Vector Machines (SVM)



Learn the decision boundary

Image Credit: Ioannis (Yannis) Gkioulekas (CMU)

Support Vector Machines (SVM)

What's the best **w** ?

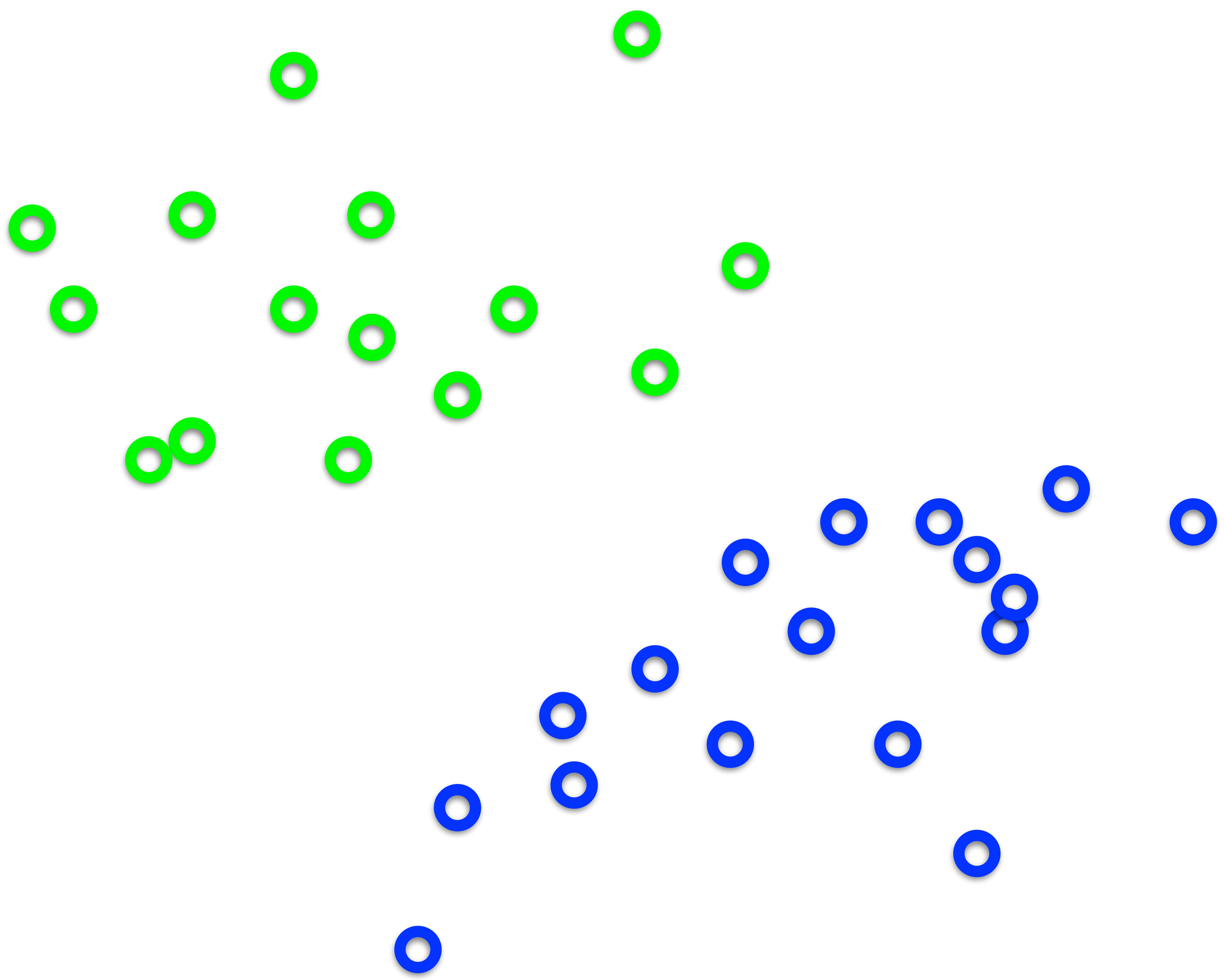


Image Credit: Ioannis (Yannis) Gkioulekas (CMU)

Support Vector Machines (SVM)

What's the best **w** ?

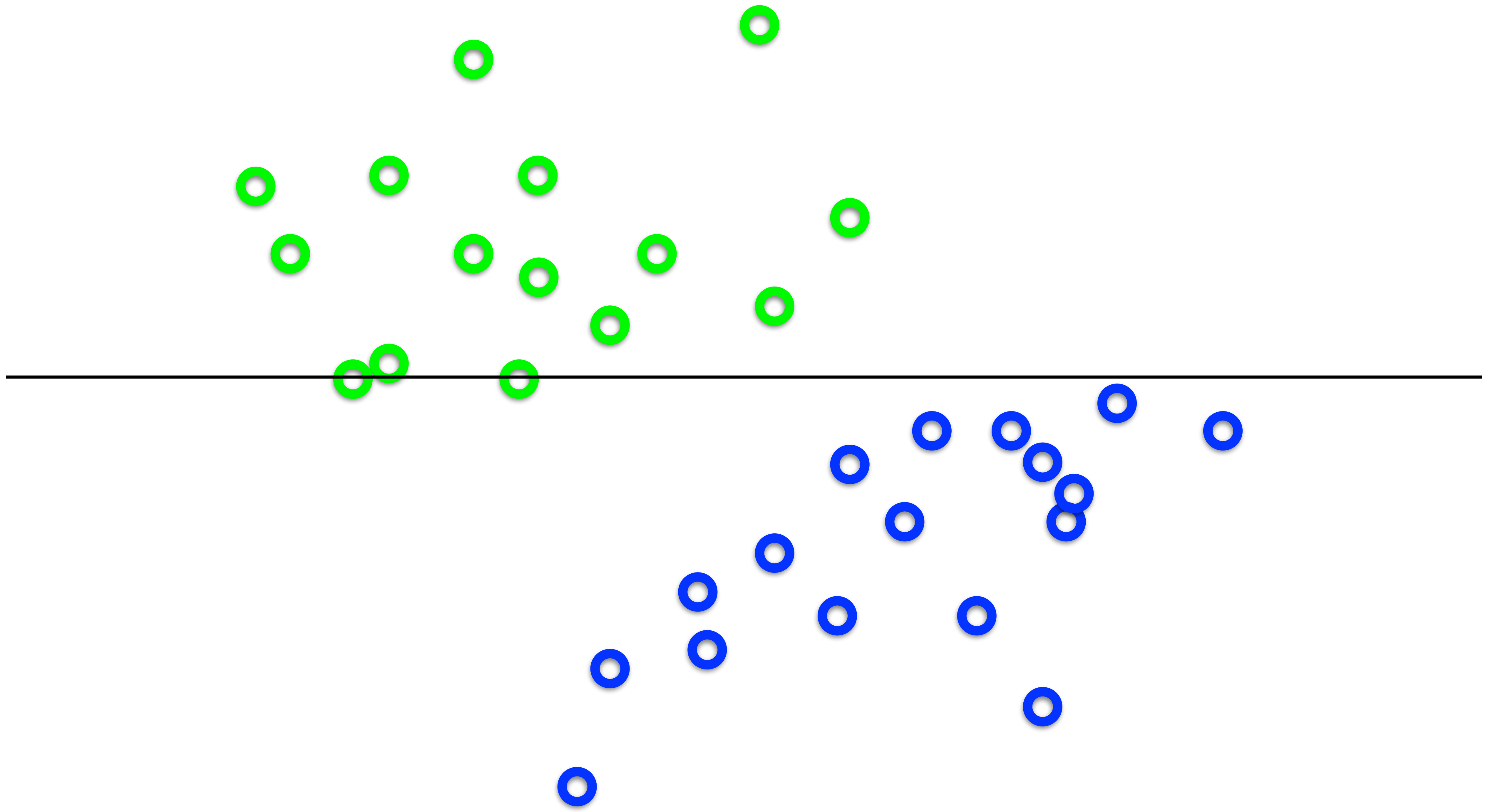


Image Credit: Ioannis (Yannis) Gkioulekas (CMU)

Support Vector Machines (SVM)

What's the best **w** ?

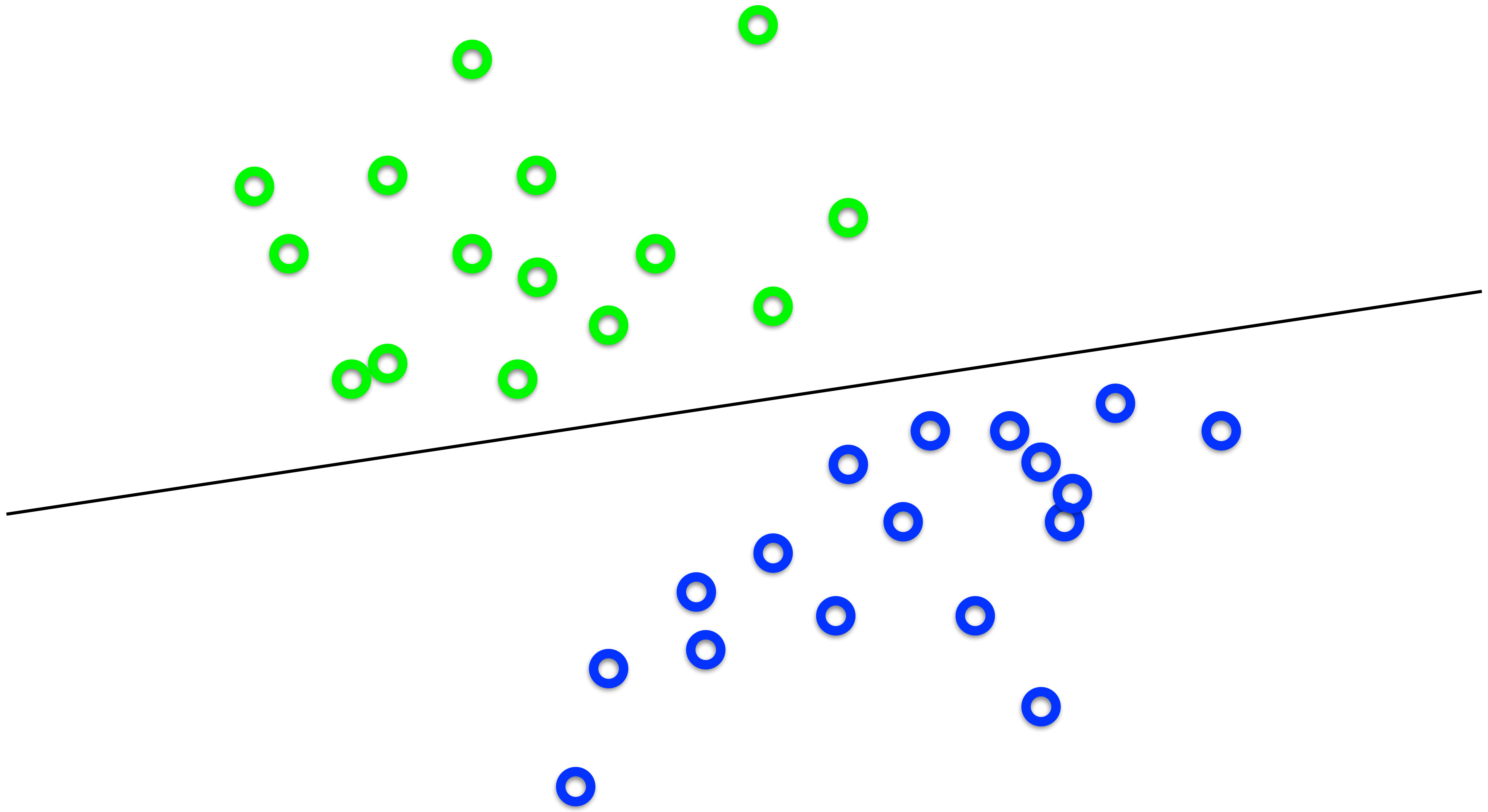


Image Credit: Ioannis (Yannis) Gkioulekas (CMU)

Support Vector Machines (SVM)

What's the best **w** ?

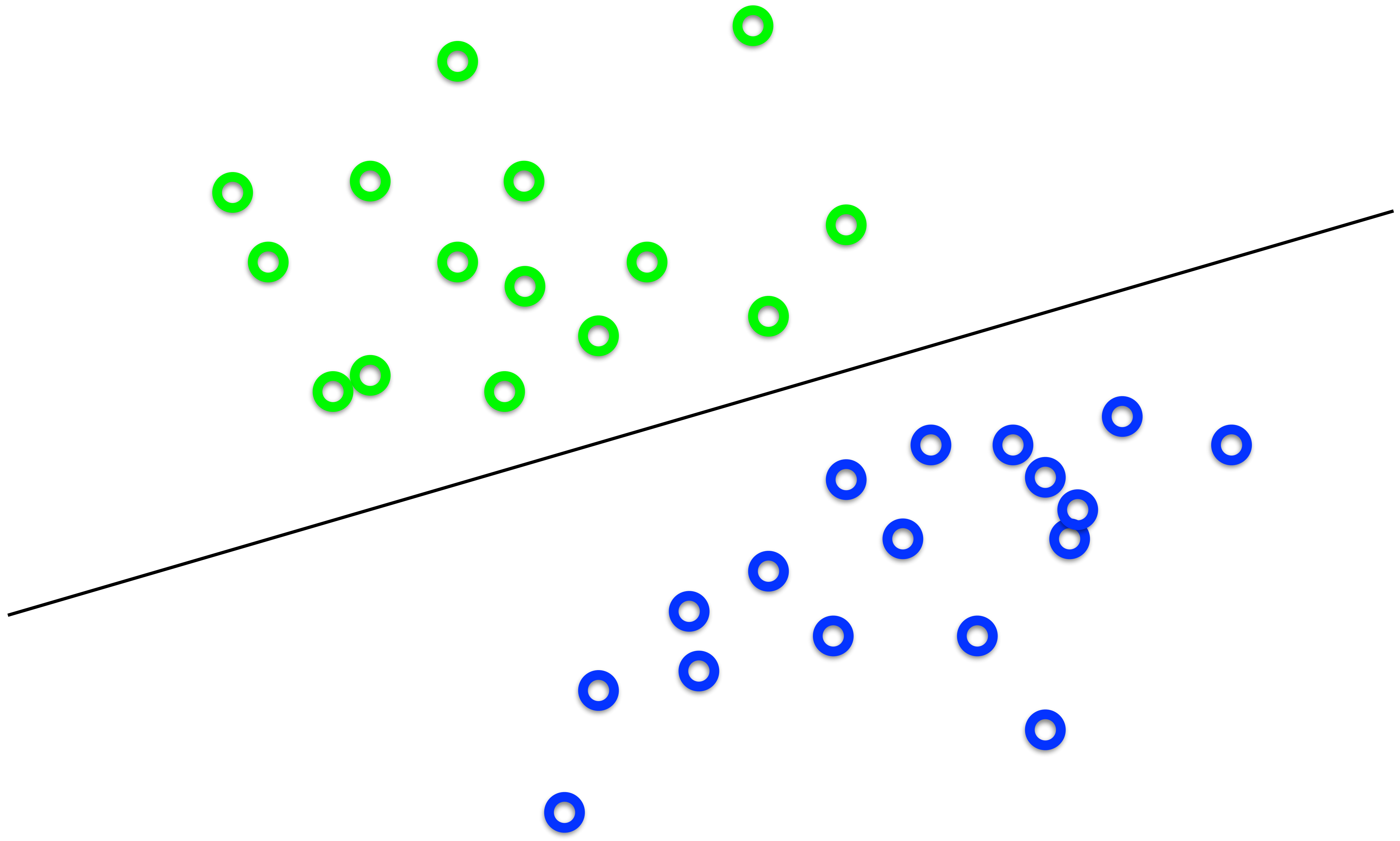


Image Credit: Ioannis (Yannis) Gkioulekas (CMU)

Support Vector Machines (SVM)

What's the best **w** ?

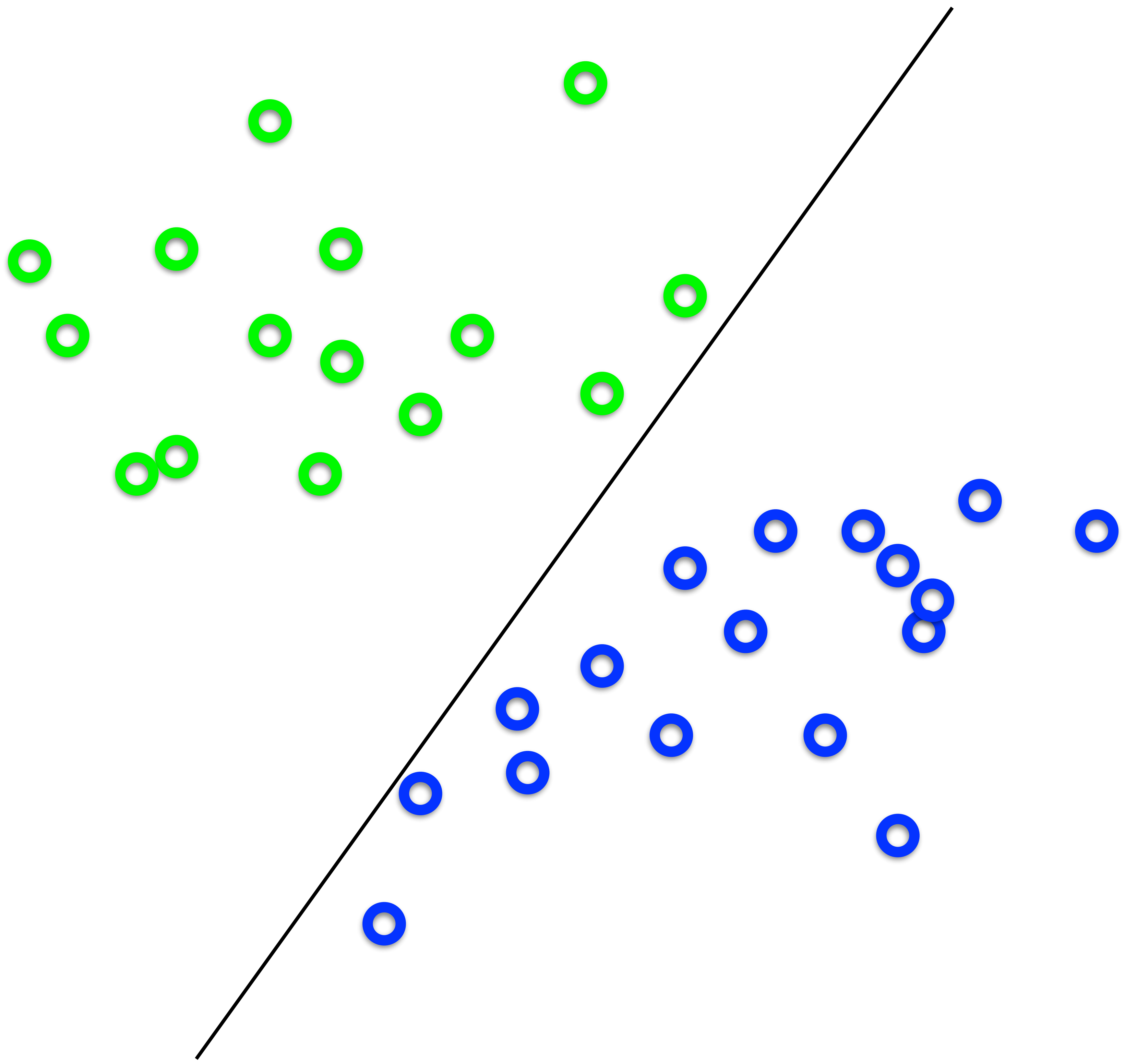
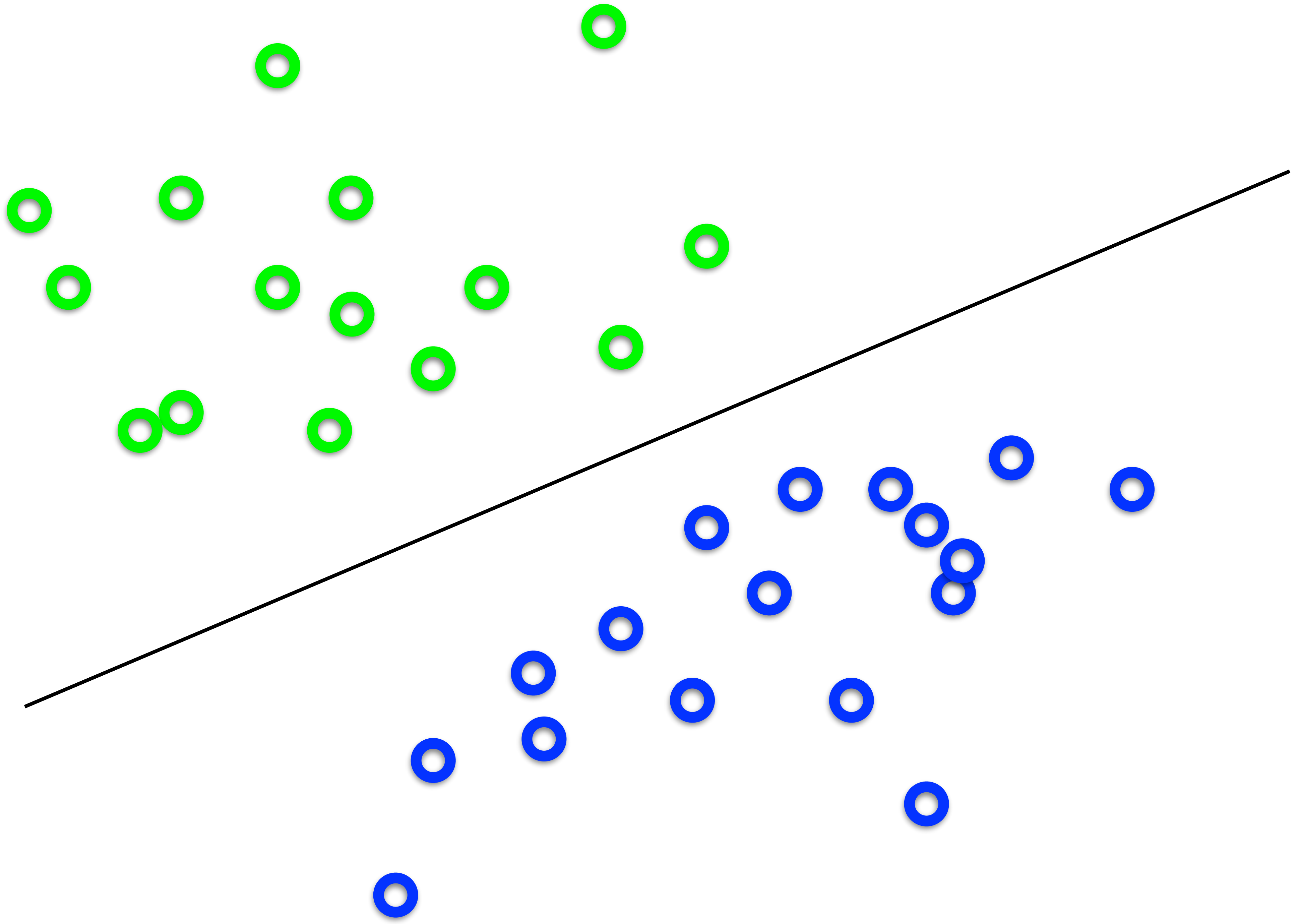


Image Credit: Ioannis (Yannis) Gkioulekas (CMU)

Support Vector Machines (SVM)

What's the best **w** ?

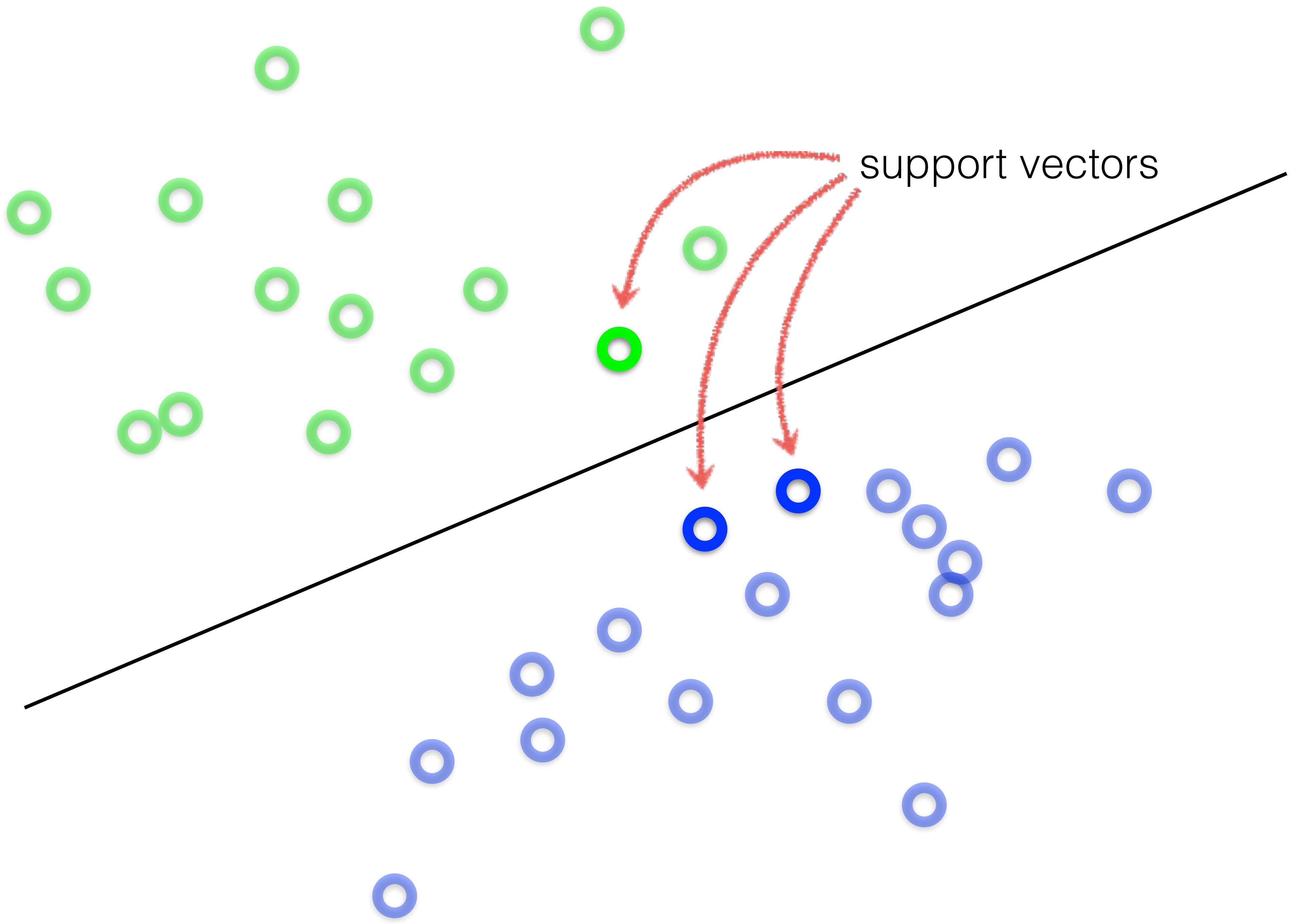


Intuitively, the line that is the farthest from all interior points

Image Credit: Ioannis (Yannis) Gkioulekas (CMU)

Support Vector Machines (SVM)

What's the best **w** ?

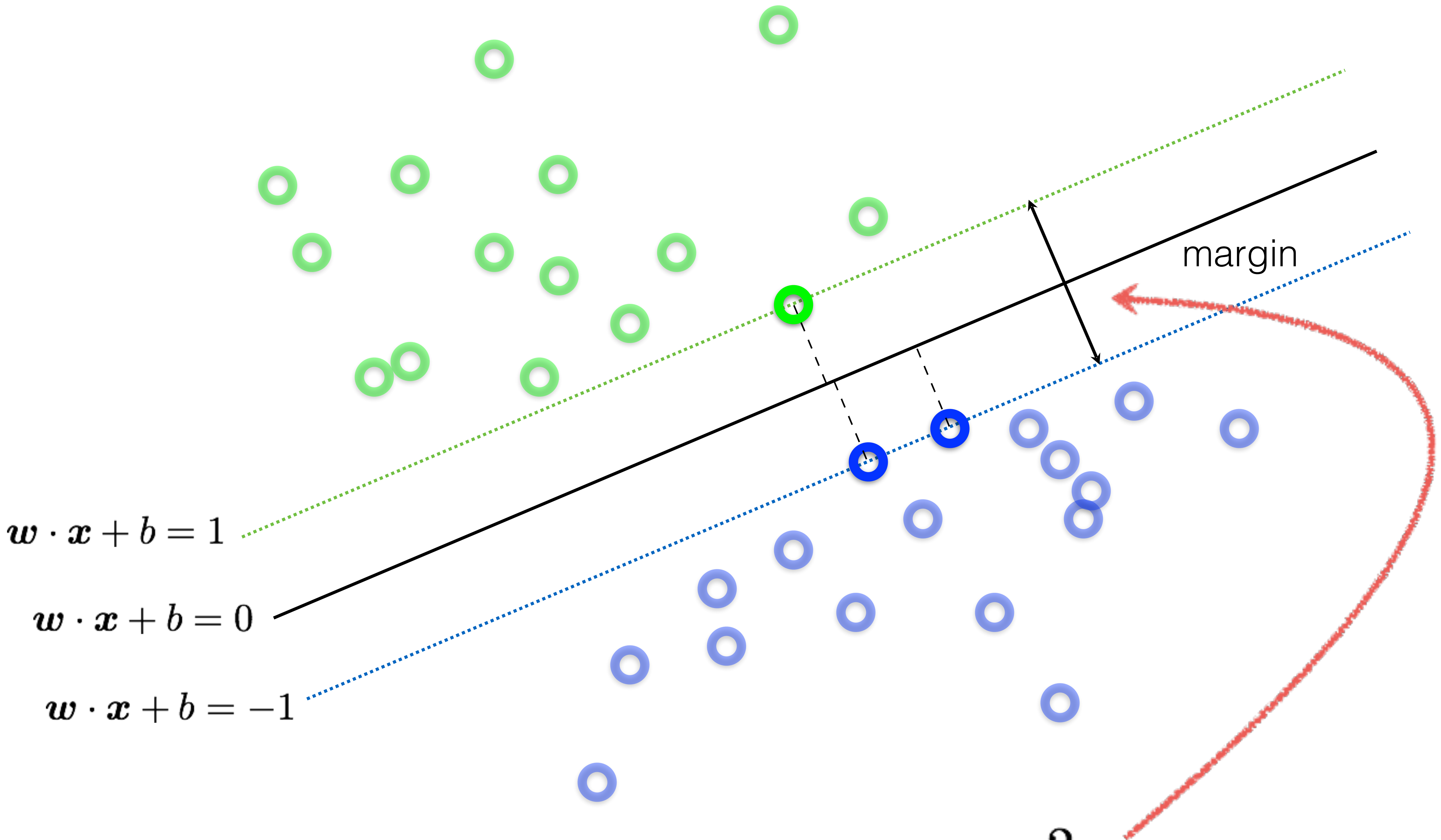


Want a hyperplane that is far away from 'inner points'

Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

Support Vector Machines (SVM)

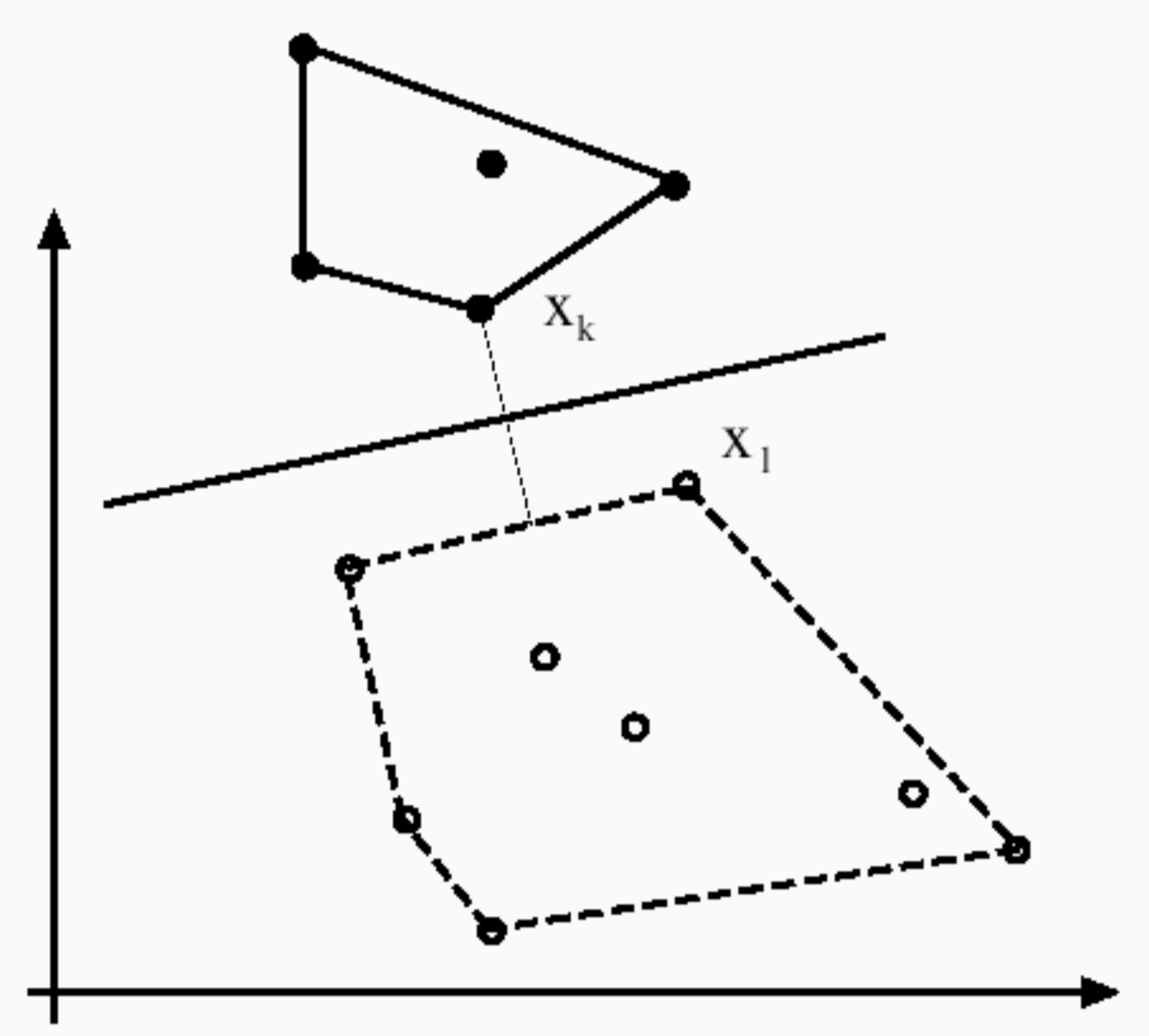
Find hyperplane **w** such that …



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

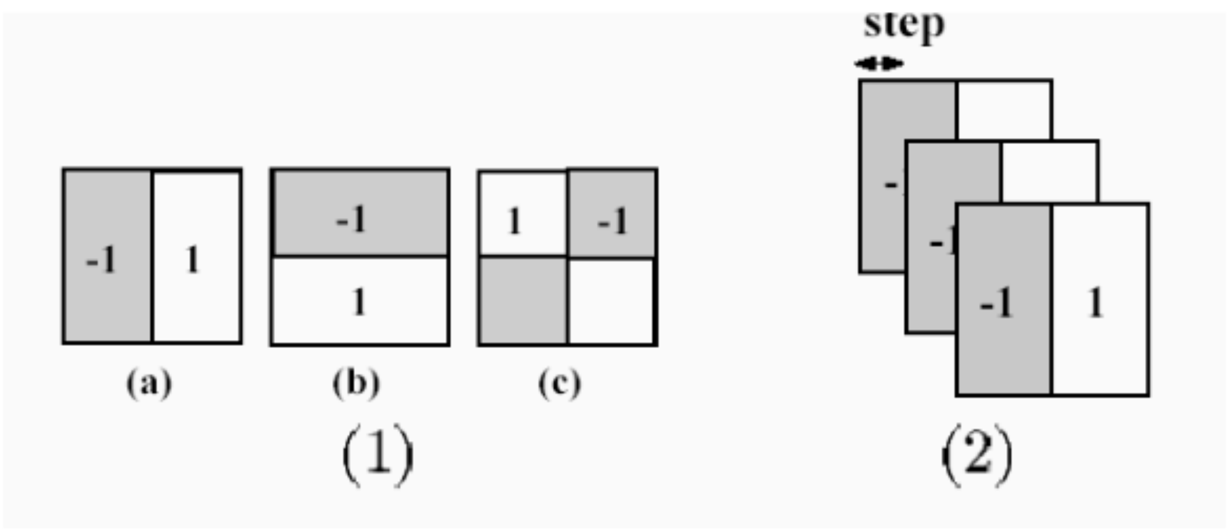
Slide Credit: Ioannis (Yannis) Gkioulekas (CMU)

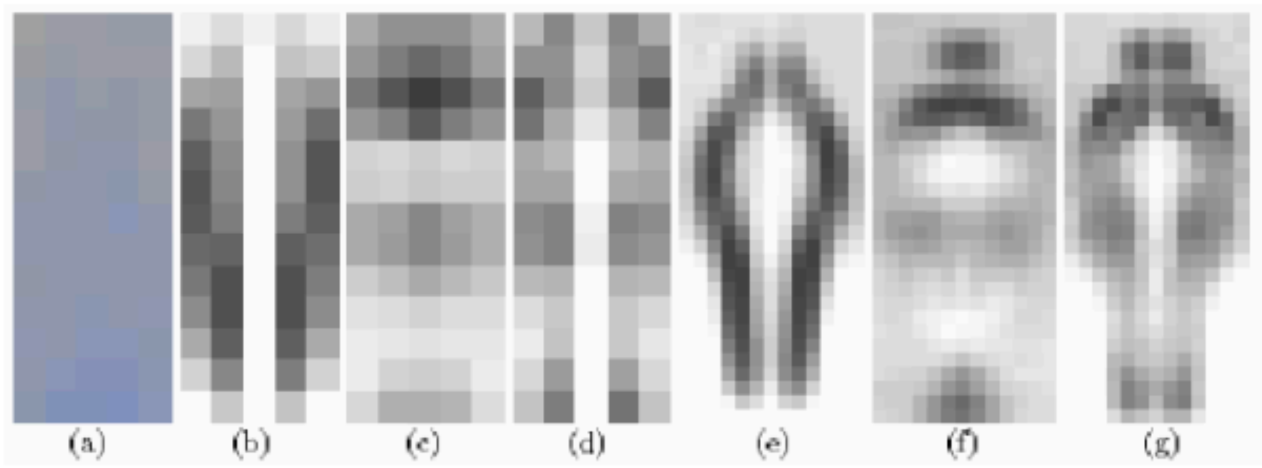
Support Vector Machines (SVM)

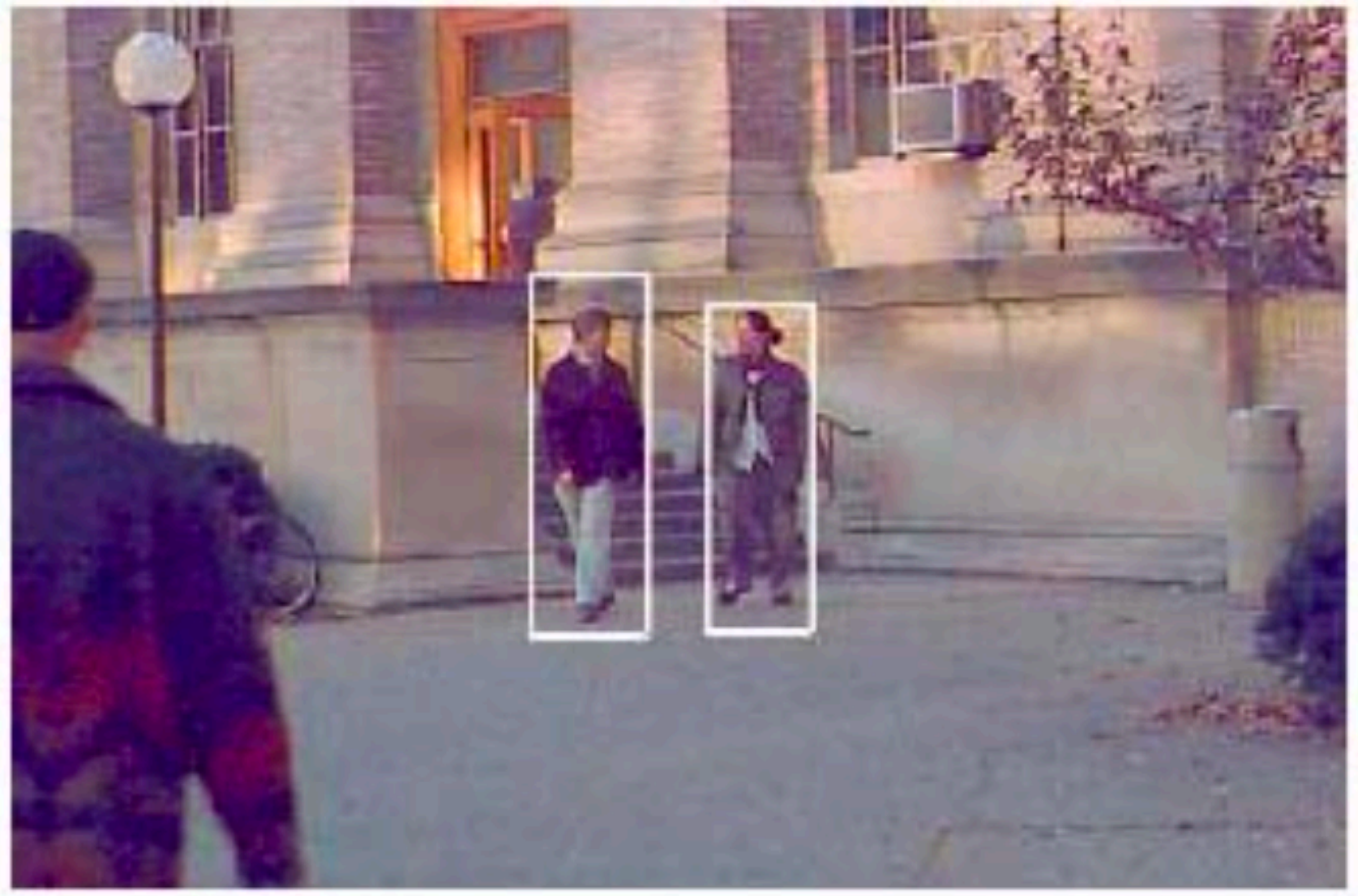


Forsyth & Ponce (2nd ed.) Figure 15.6

Example: Pedestrian Detection with SVM







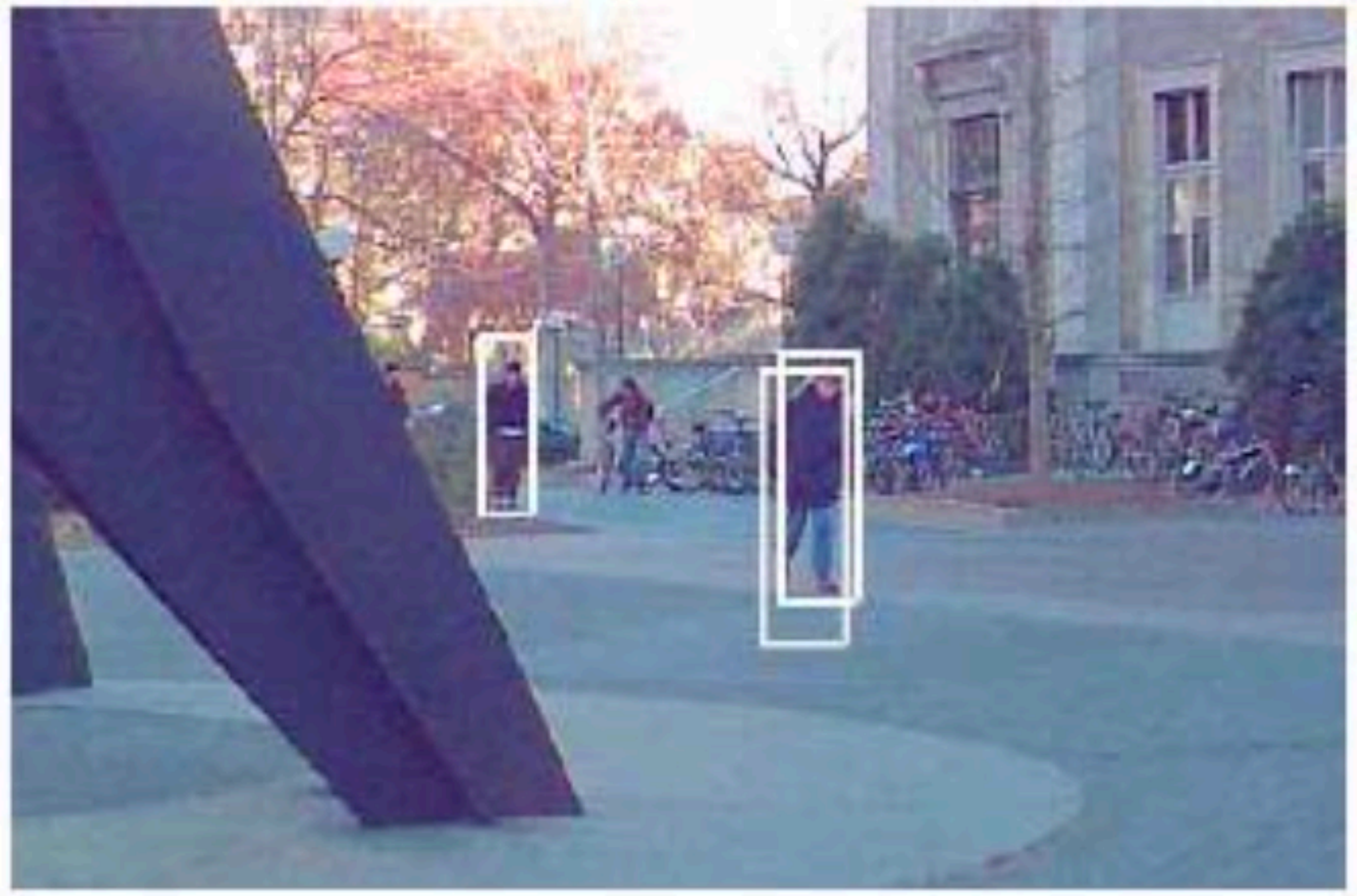


Figure credit: Papageorgiou, Oren, and Poggio, 1998

Summary

A **classifier** accepts as input a set of features and outputs (predicts) a class label

Classifiers need to take into account “loss" associated with each kind of classification error

A Receiver Operating Characteristic (ROC) curve plots the trade-off between false negatives and false positives

Parametric classifiers are model driven. The parameters of the model are learned from training examples
— e.g. support vector machine, decision tree

Non-parametric classifiers are data driven. New data points are classified by comparing to the training examples directly
— e.g. k-nearest neighbour