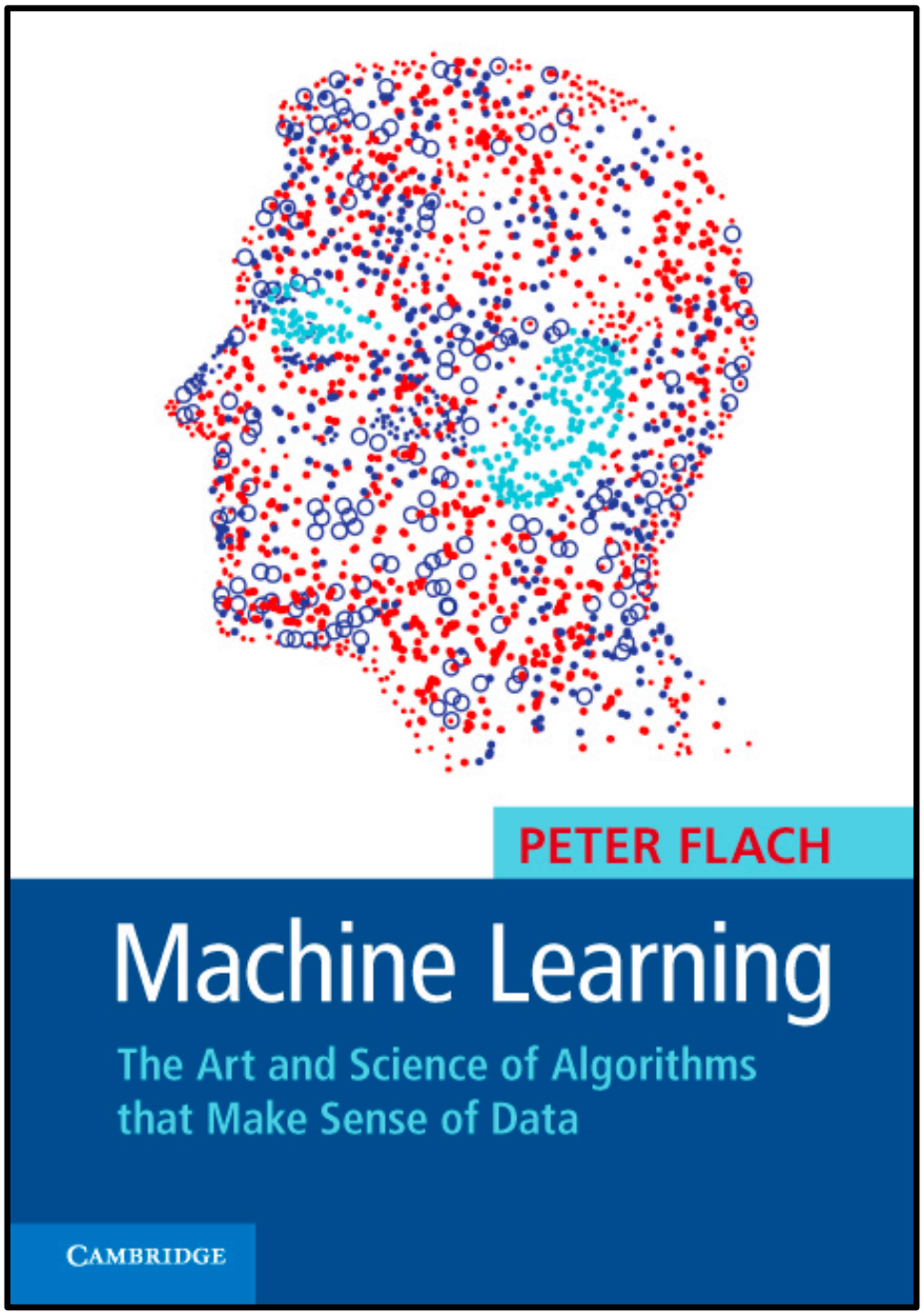
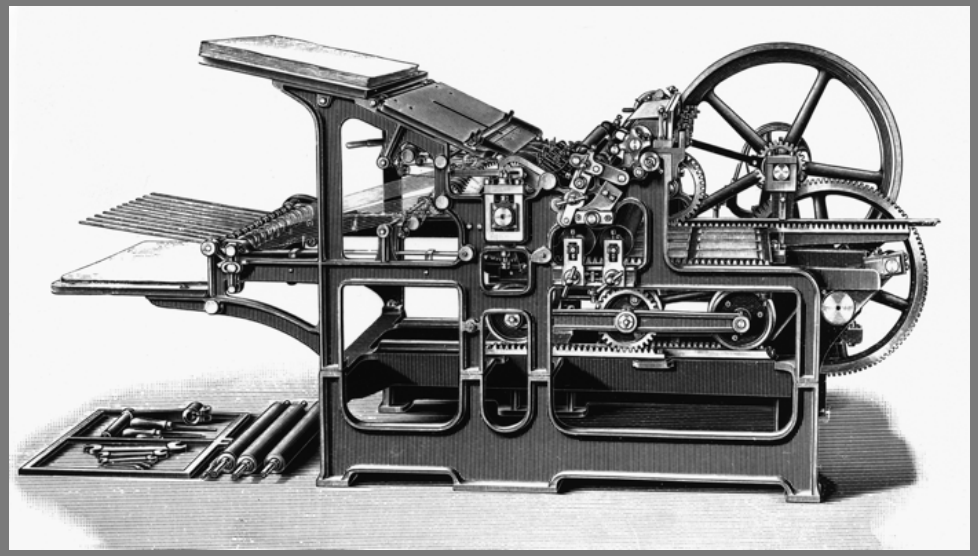
MACHINE LEARNING OVERVIEW


PETER FLACH
Machine Learning
The Art and Science of Algorithms
that Make Sense of Data
CAMBRIDGE

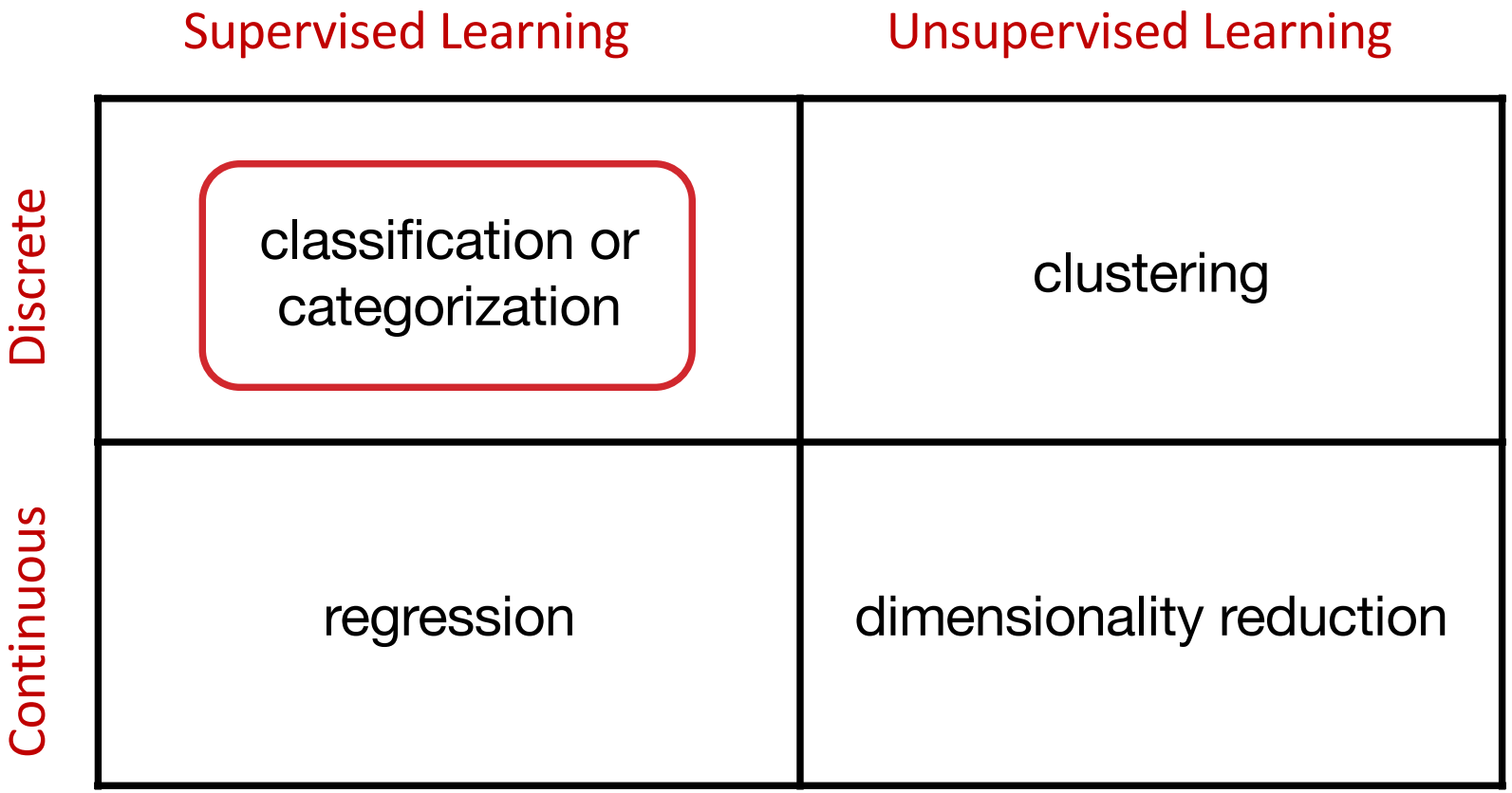
VS



MACHINE LEARNING PROBLEMS

	Supervised Learning	Unsupervised Learning
Discrete	classification or categorization	clustering
Continuous	regression	dimensionality reduction

MACHINE LEARNING PROBLEMS



WHAT IS A CLASSIFIER

Apply a prediction function to a feature representation of an image/data-set to get the desired output:

f() = “apple”

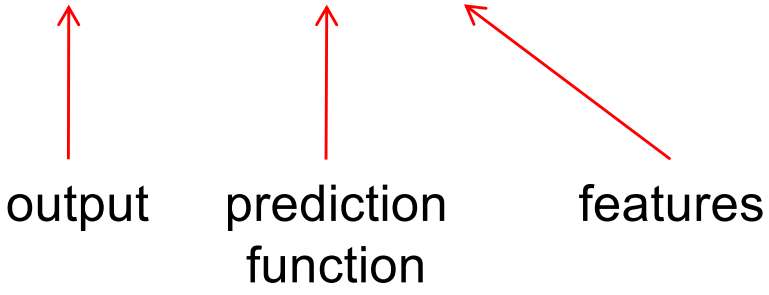
f() = “tomato”

f() = “cow”

Slide credit: L. Lazebnik

THE MACHINE LEARNING FRAMEWORK

$$y = f(x)$$

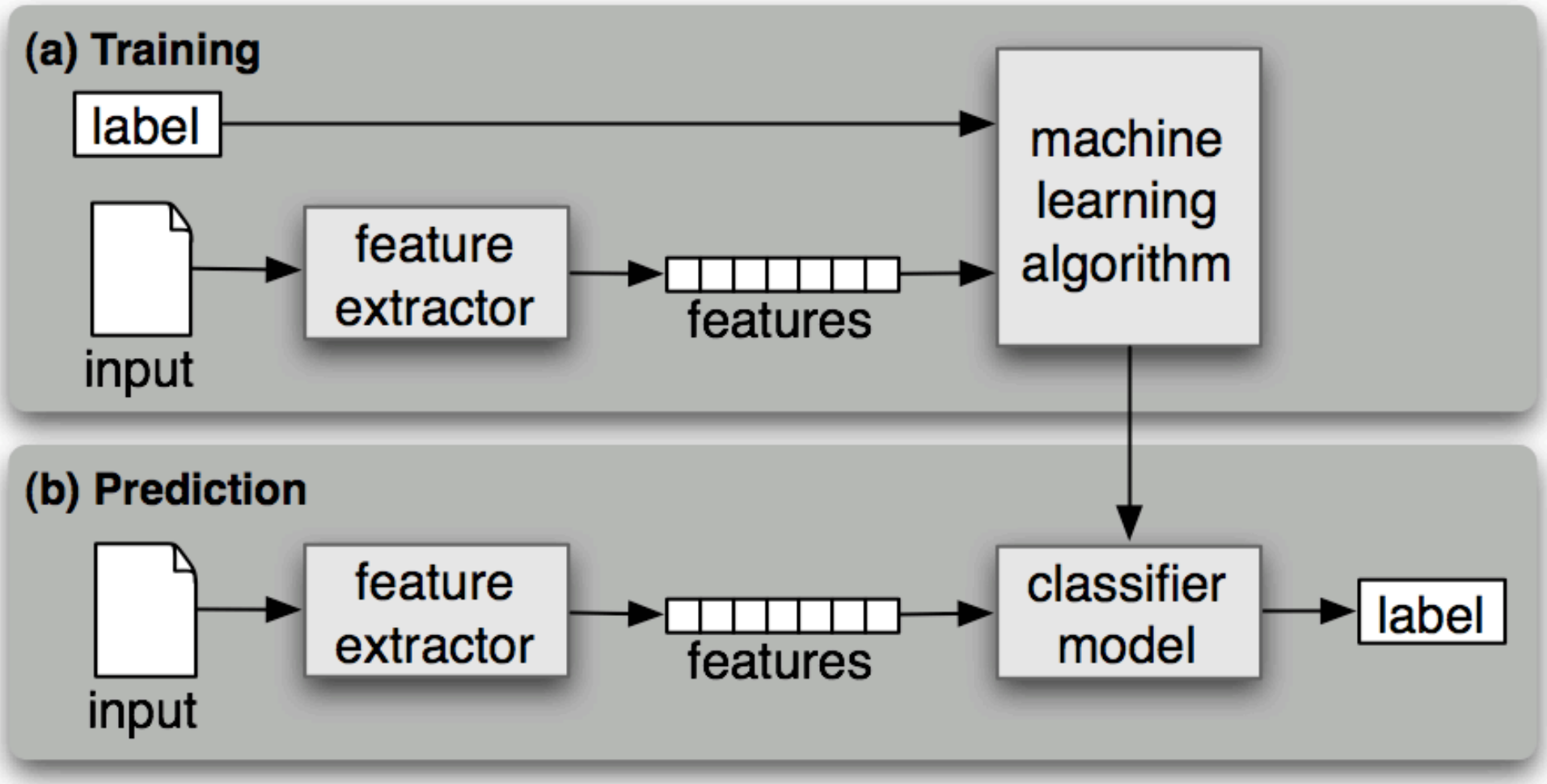


Training: given a *training set* of labeled examples $\{(x_1,y_1), \ldots, (x_N,y_N)\}$, estimate the prediction function f by minimizing the prediction error on the training set

Testing: apply f to a never before seen *test example* x and output the predicted value $y = f(x)$

Slide credit: L. Lazebnik

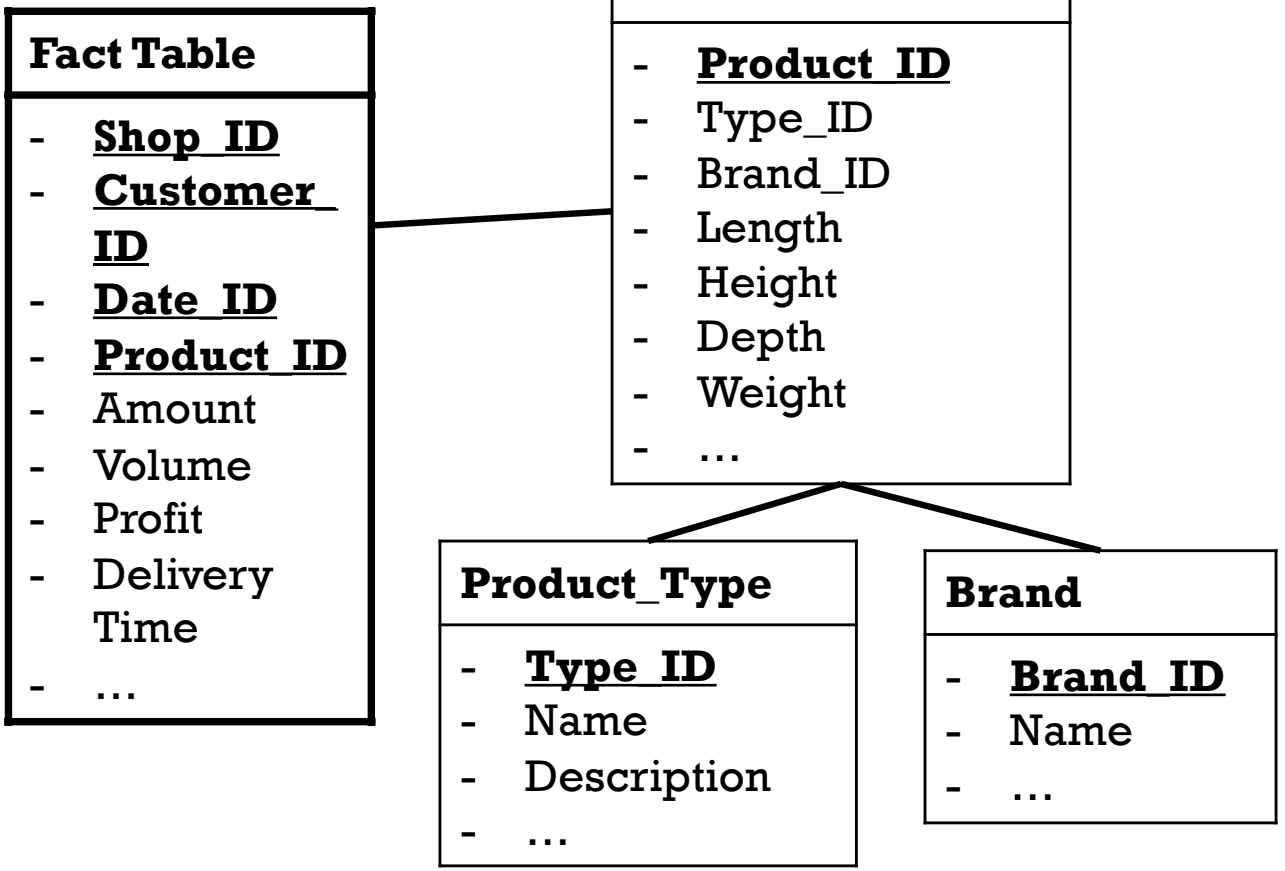
ML PIPELINE (SUPERVISED)



FEATURES

Fact Table

- **Shop_ID**
- **Customer_ID**
- **Date_ID**
- **Product_ID**
- Amount
- Volume
- Profit
- …

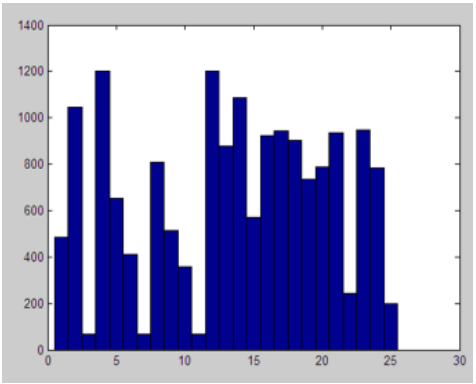


Custermer State	Product Type	Product Weight	Volume (L*H*D)	Month	Delivery Time

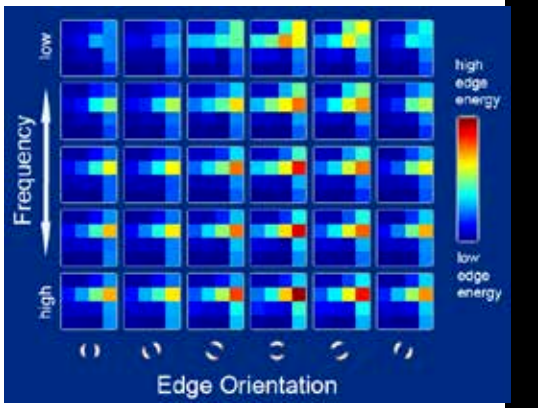
IMAGE FEATURES

Raw pixels

Histograms



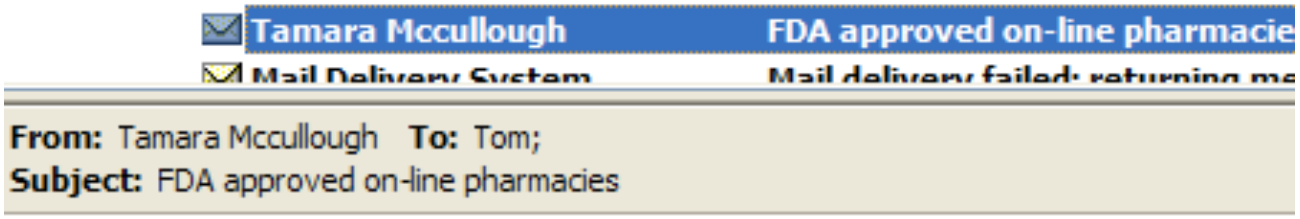
GIST descriptors

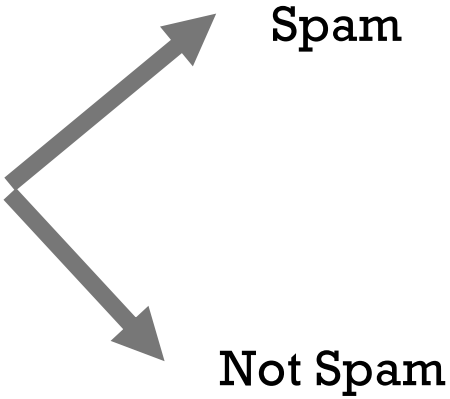


…

Slide credit: L. Lazebnik

TEXT FEATURES





Bag of Words

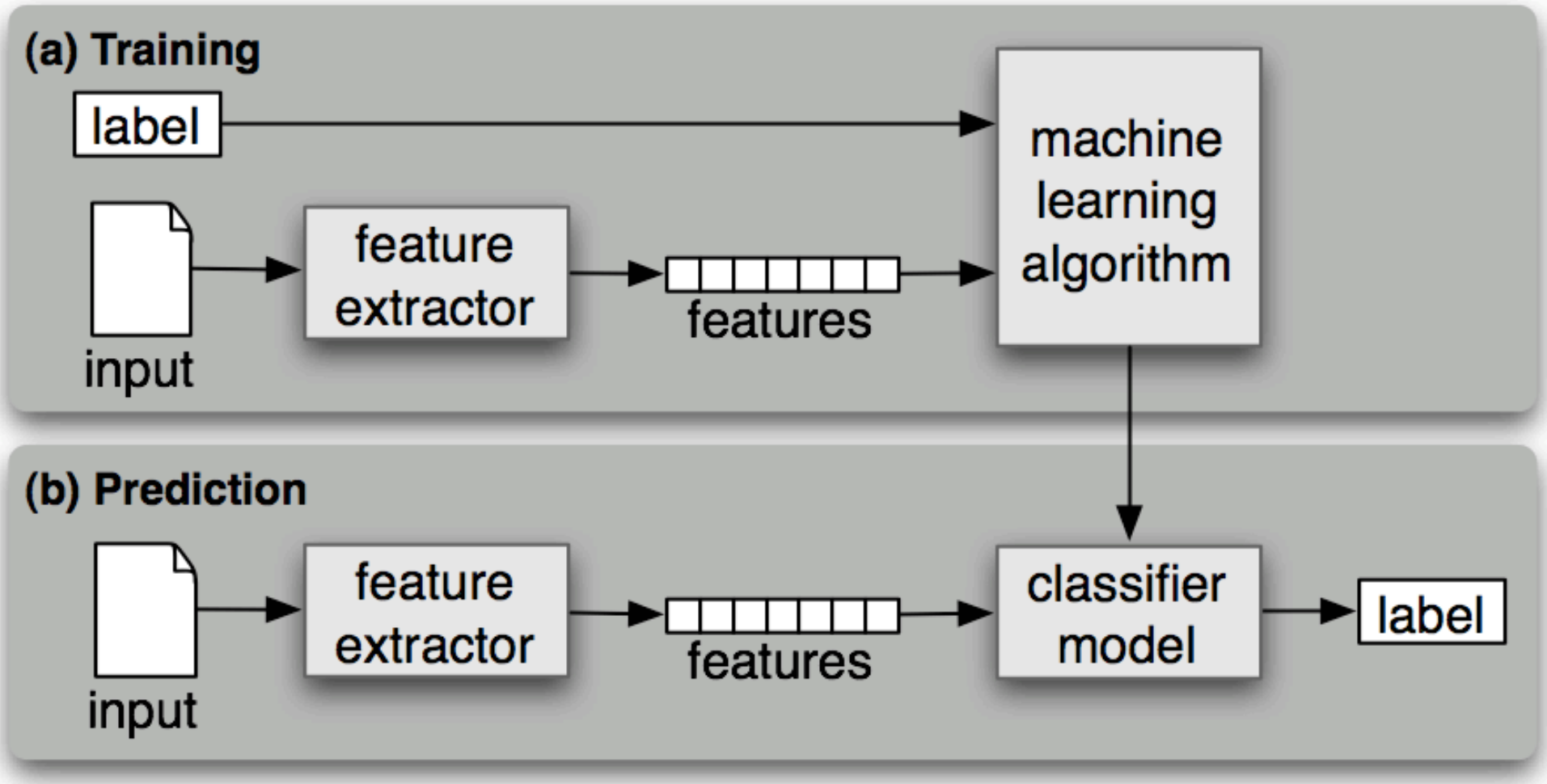
$$\begin{pmatrix} Viagra \\ Soft \\ Herbel \\ Pills \\ Are \\ \dots \end{pmatrix}$$

N-Grams

$$\begin{pmatrix} herbel\ pills \\ pills\ for \\ for\ Hair \\ Hair\ enlargement \\ enlargement\ Techniques \\ \dots \end{pmatrix}$$

FEATURE TO PREDICT UNEMPLOYMENT

ML PIPELINE (SUPERVISED)



CLASSIFIER OVERVIEW

Representation	Evaluation	Optimization
Instances	Accuracy/Error rate	Combinatorial optimization
K-nearest neighbor	Precision and recall	Greedy search
Support vector machines	Squared error	Beam search
Hyperplanes	Likelihood	Branch-and-bound
Naive Bayes	Posterior probability	Continuous optimization
Logistic regression	Information gain	Unconstrained
Decision trees	K-L divergence	Gradient descent
Sets of rules	Cost/Utility	Conjugate gradient
Propositional rules	Margin	Quasi-Newton methods
Logic programs		Constrained
Neural networks		Linear programming
Graphical models		Quadratic programming
Bayesian networks		
Conditional random fields		

MANY CLASSIFIERS TO CHOOSE FROM

K-nearest neighbor

Support Vector Machines

Decision Trees

Random Forrest

(Gradient) Boosted Decision Trees

Logistic Regression

Naïve Bayes

Bayesian network

RBMs

….

Which is the best one?

Slide credit: D. Hoiem

MANY CLASSIFIERS TO CHOOSE FROM

K-nearest neighbor

Support Vector Machines

Decision Trees

Random Forrest

(Gradient) Boosted Decision Trees

Logistic Regression

Naïve Bayes

Bayesian network

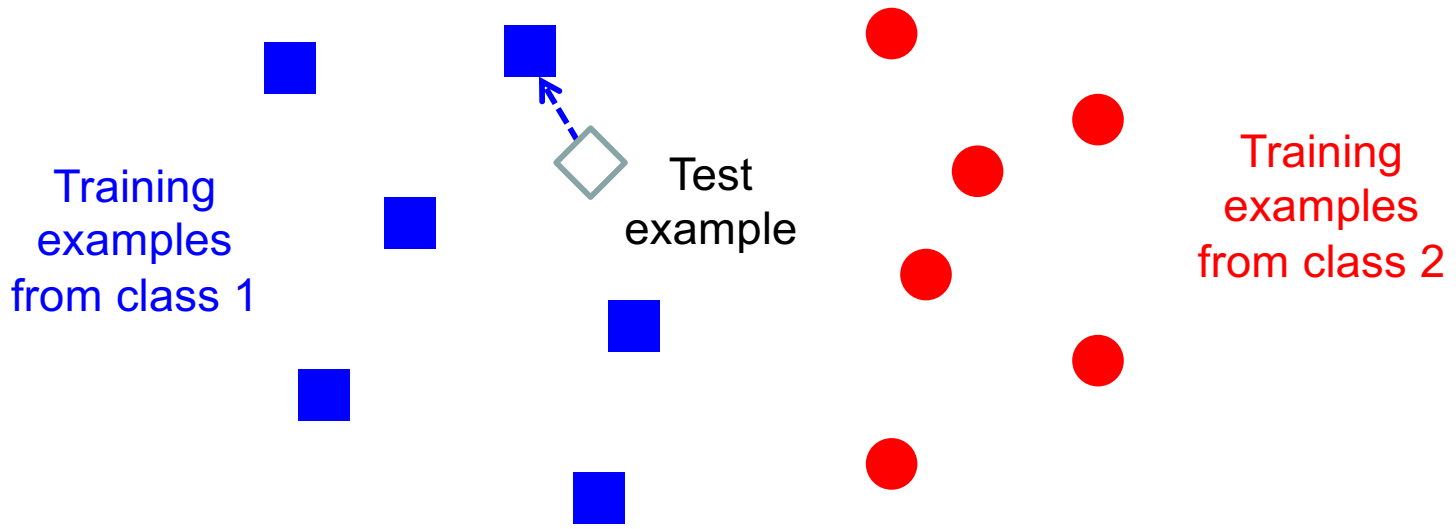
RBMs

….

Which is the best one?

Slide credit: D. Hoiem

CLASSIFIERS: NEAREST NEIGHBOR

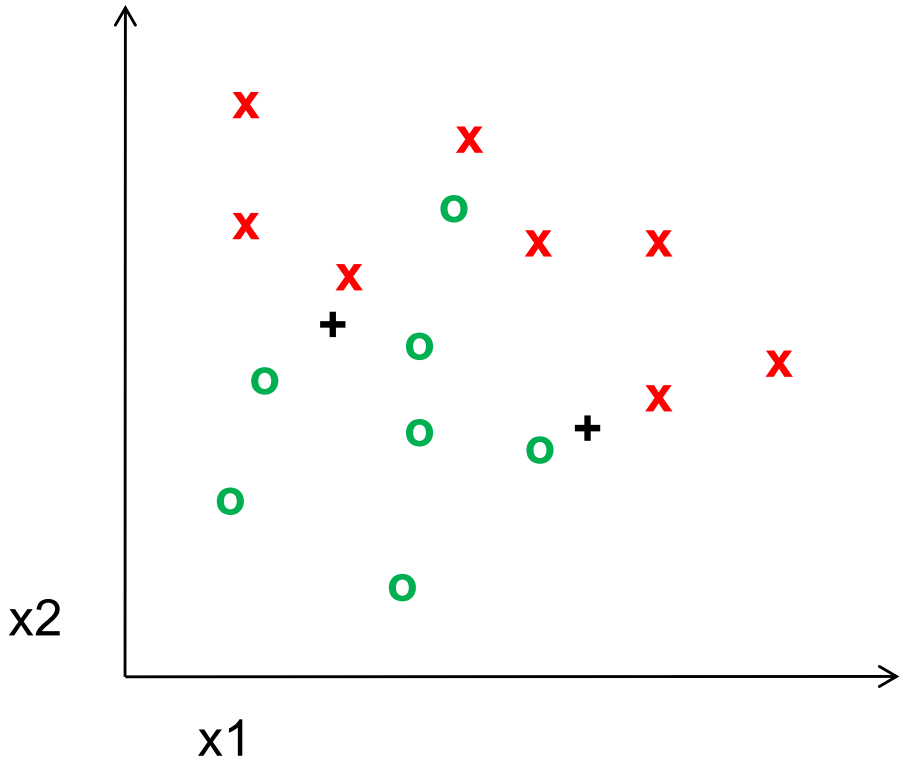


f(**x**) = label of the training example nearest to **x**

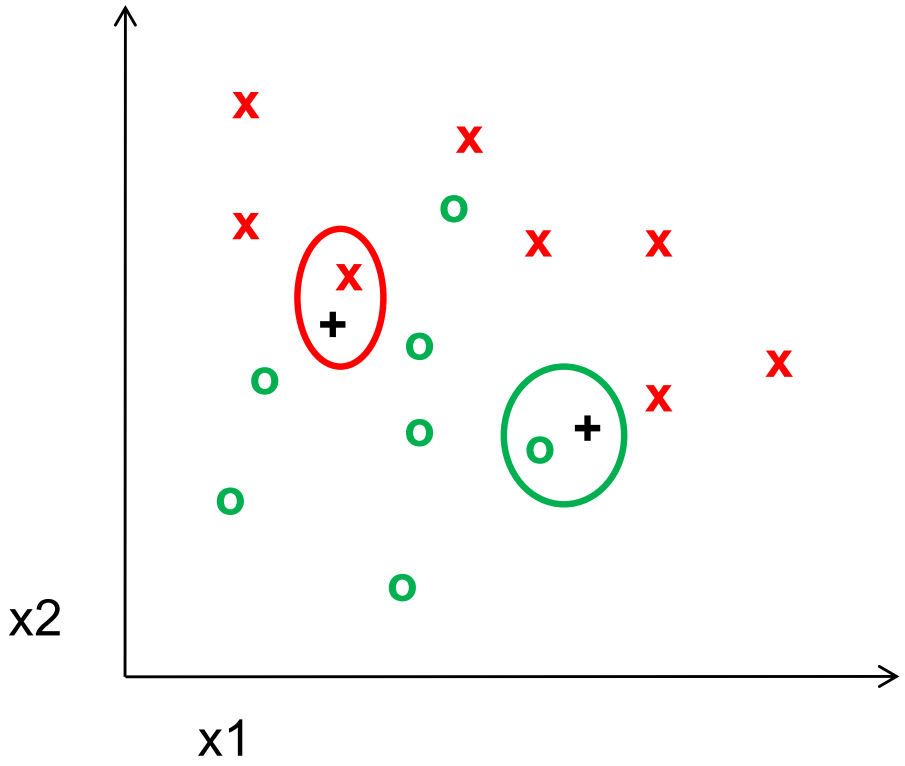
- All we need is a distance function for our inputs
- No training required!

Slide credit: L. Lazebnik

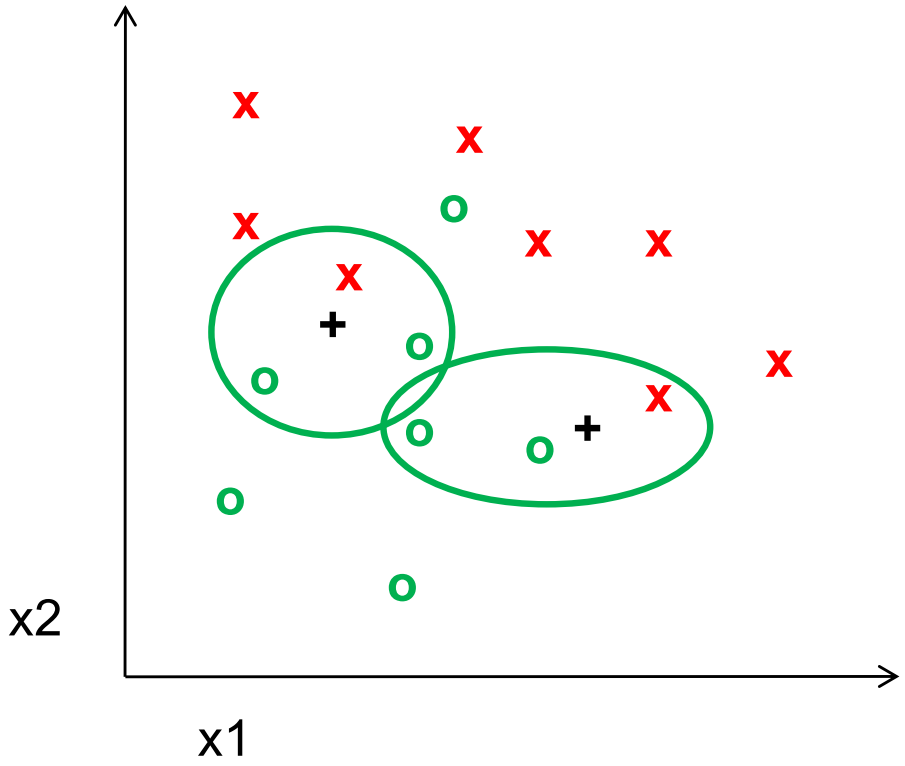
K-NEAREST NEIGHBOR



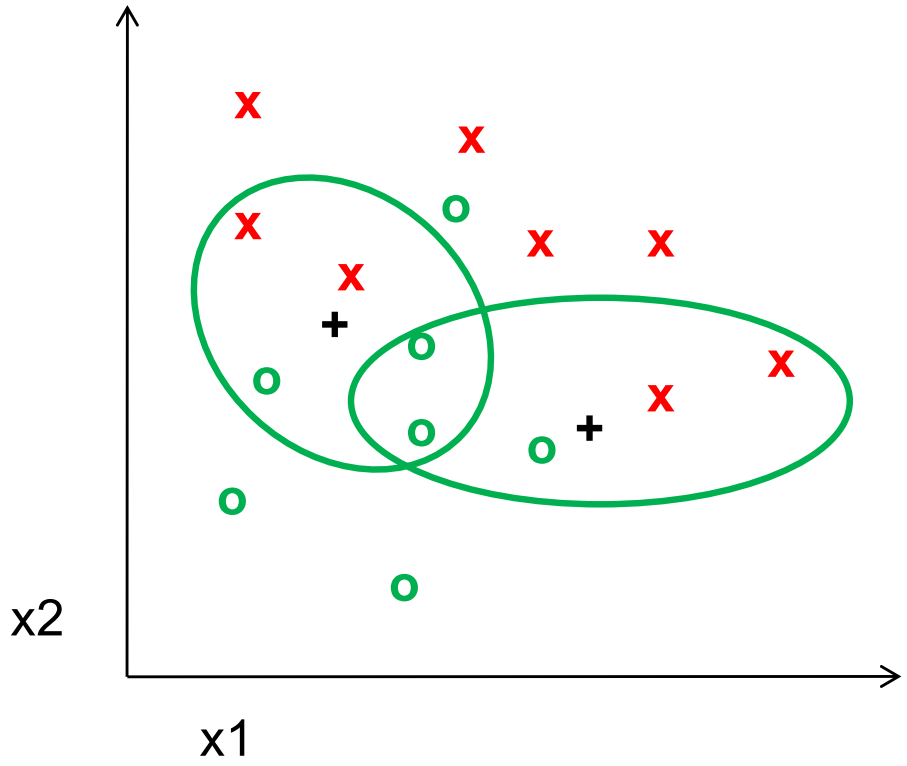
1-NEAREST NEIGHBOR



3-NEAREST NEIGHBOR

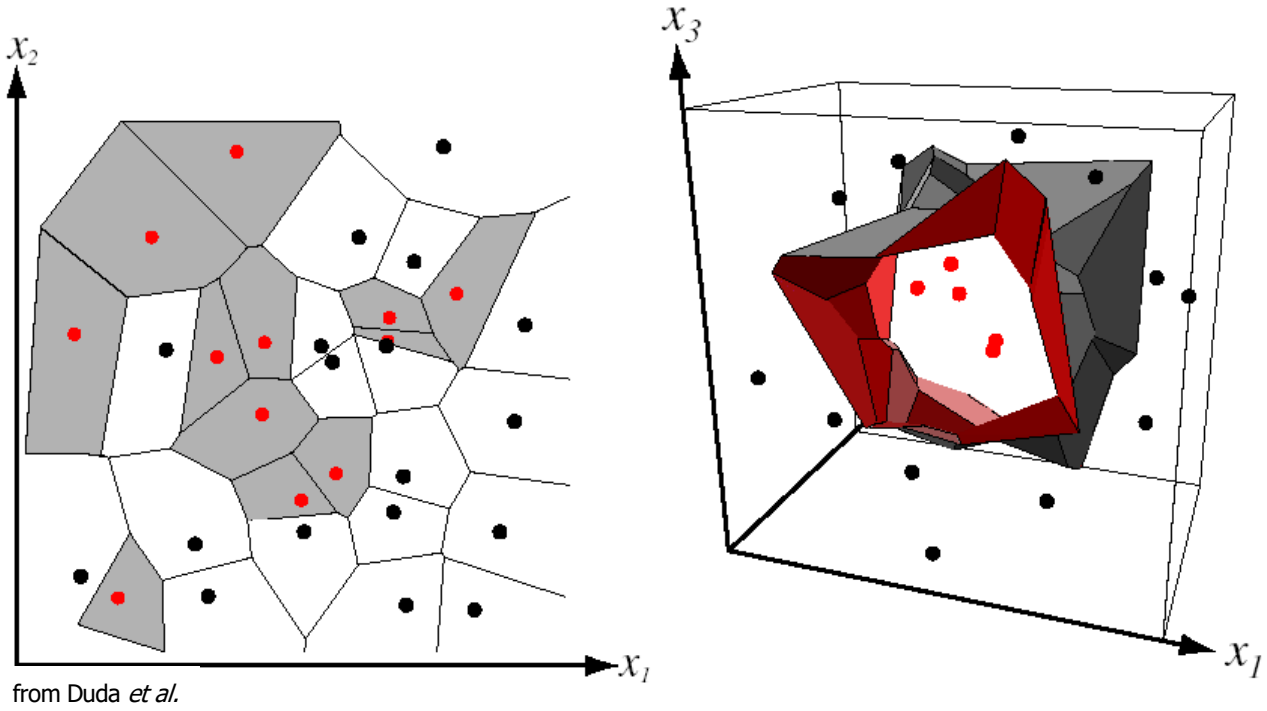


5-NEAREST NEIGHBOR



DECISION BOUNDARIES KNN

Assign label of nearest training data point to each test data point



from Duda *et al.*

Voronoi partitioning of feature space
for two-category 2D and 3D data

Source: D. Lowe

MANY CLASSIFIERS TO CHOOSE FROM

K-nearest neighbor

Support Vector Machines

Decision Trees

Random Forrest

(Gradient) Boosted Decision Trees

Logistic Regression

Naïve Bayes

Bayesian network

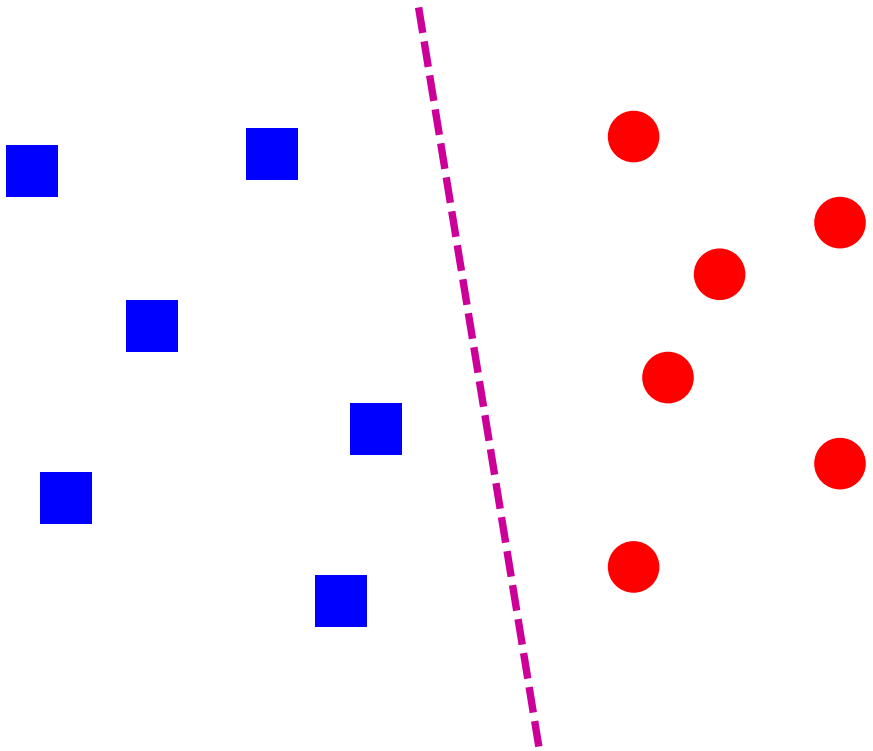
RBMs

….

Which is the best one?

Slide credit: D. Hoiem

CLASSIFIERS: LINEAR

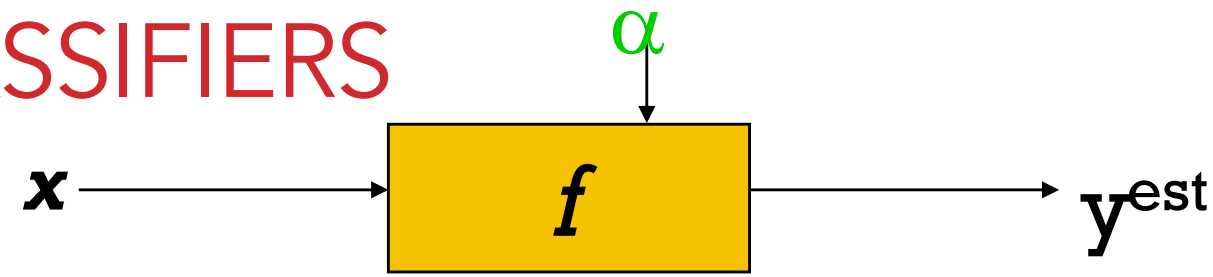


Find a *linear function* to separate the classes:

$$f(x) = \text{sgn}(w \cdot x + b)$$

Slide credit: L. Lazebnik

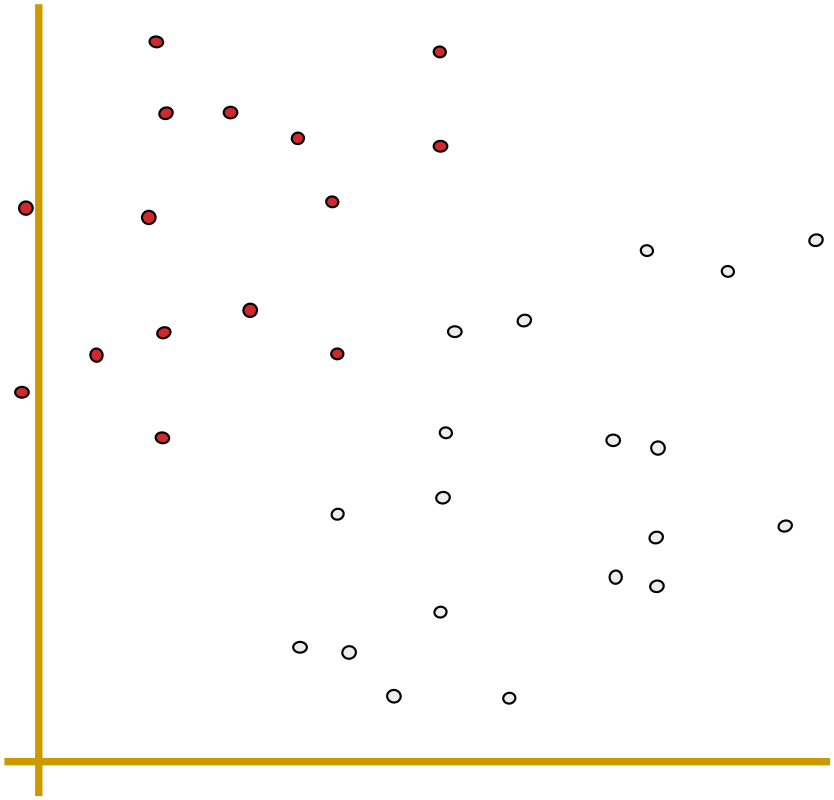
LINEAR CLASSIFIERS



$f(\mathbf{x}, \mathbf{w}, b) = sign(\mathbf{w}.\mathbf{x} - b)$

denotes +1

denotes -1

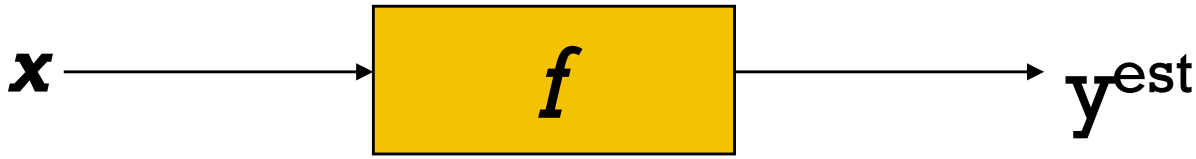


How would you classify this data?

Slides from Andrew W. Moore

LINEAR CLASSIFIERS

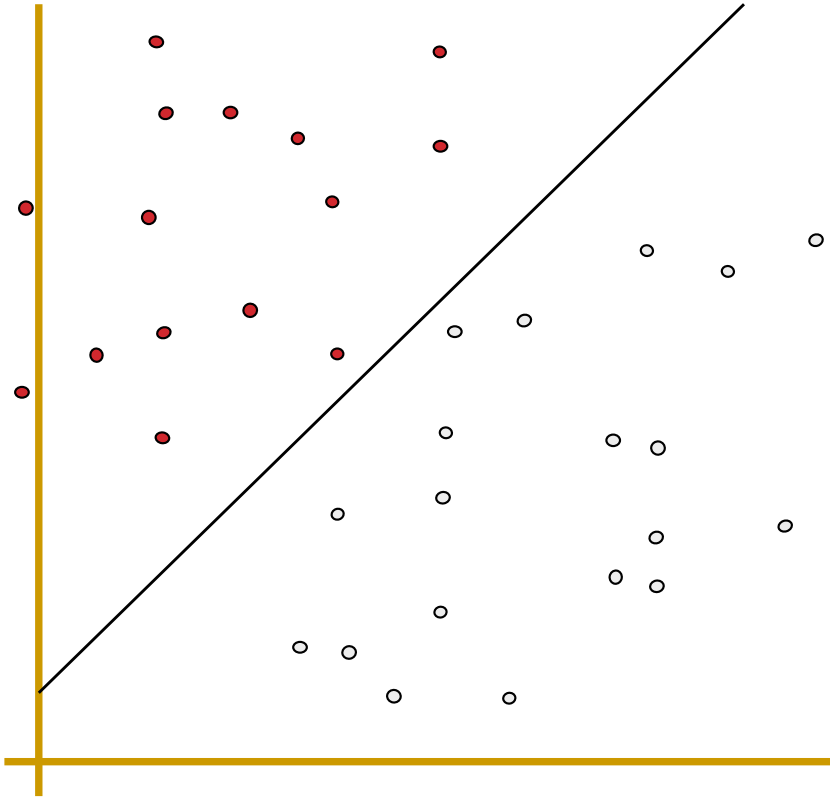
α



$f(x,w,b) = sign(w.x - b)$

denotes +1

denotes -1

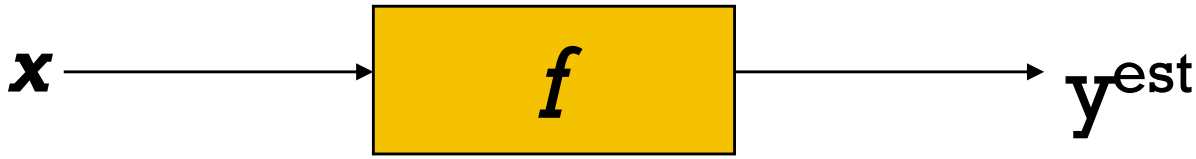


How would you classify this data?

Slides from Andrew W. Moore

LINEAR CLASSIFIERS

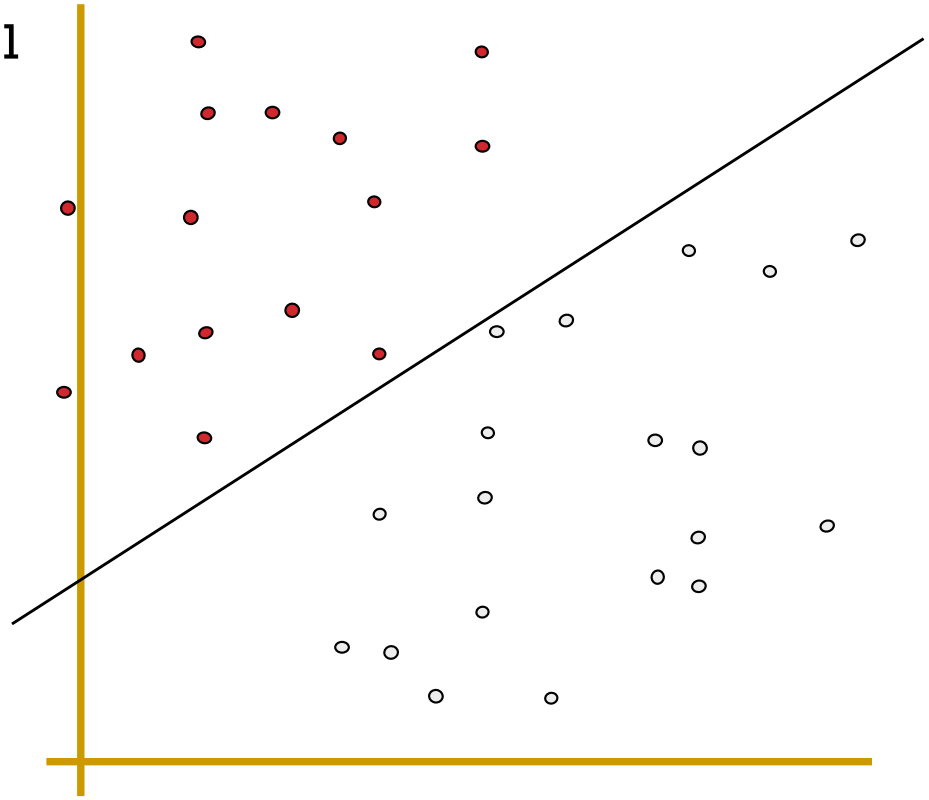
α



$f(x,w,b) = sign(w.x - b)$

denotes +1

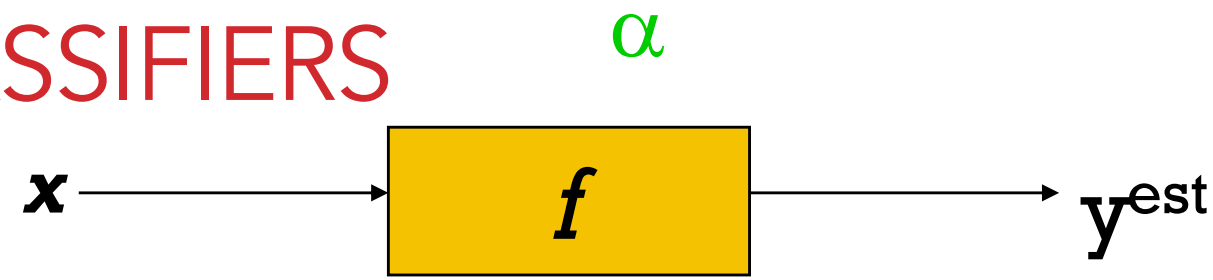
denotes -1



How would you classify this data?

Slides from Andrew W. Moore

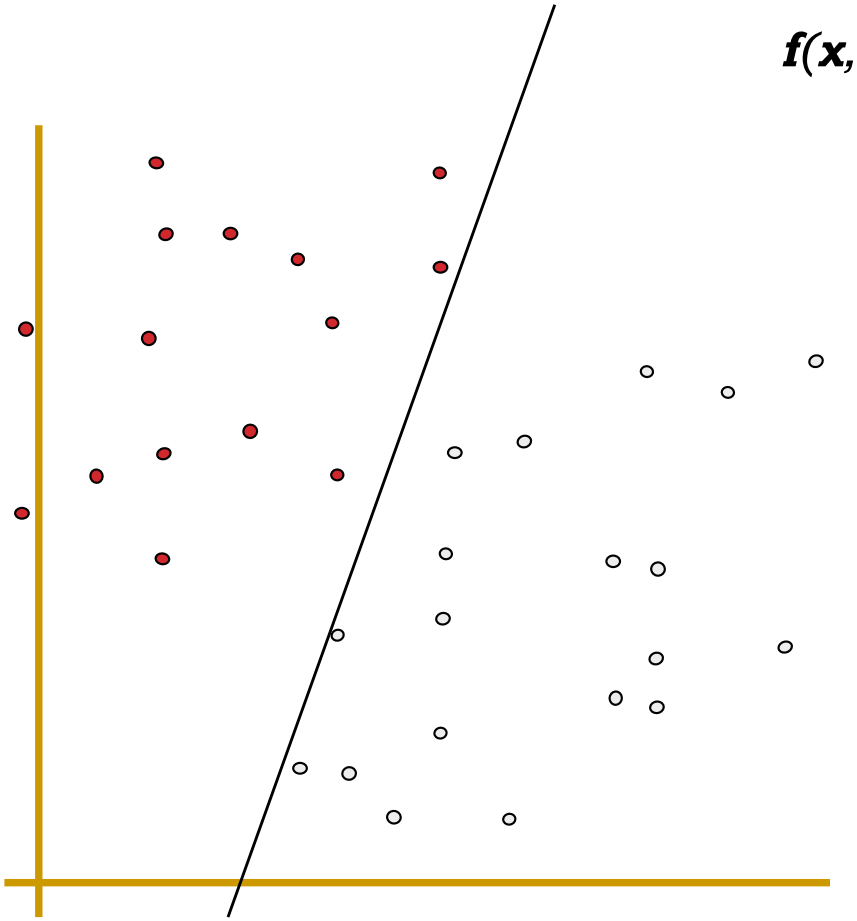
LINEAR CLASSIFIERS



$f(x,w,b) = sign(w.x - b)$

denotes +1

denotes -1

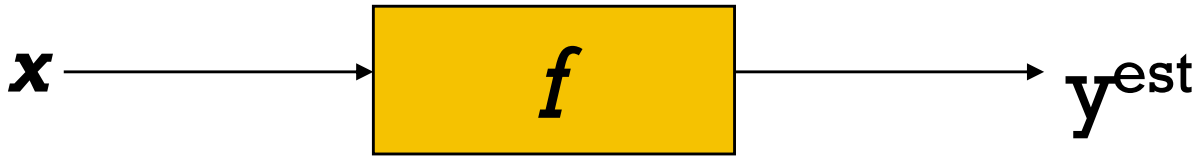


How would you classify this data?

Slides from Andrew W. Moore

LINEAR CLASSIFIERS

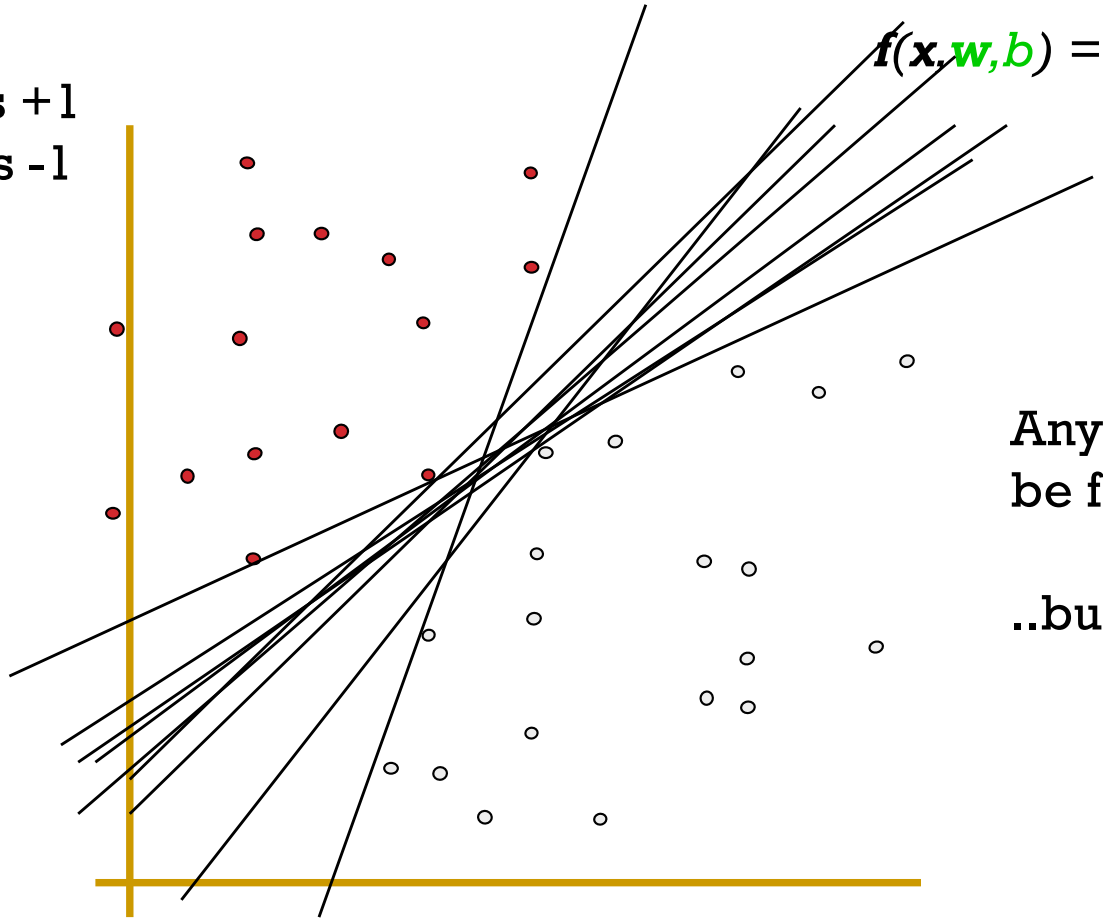
α



$f(x,w,b) = sign(w.x - b)$

denotes +1

denotes -1

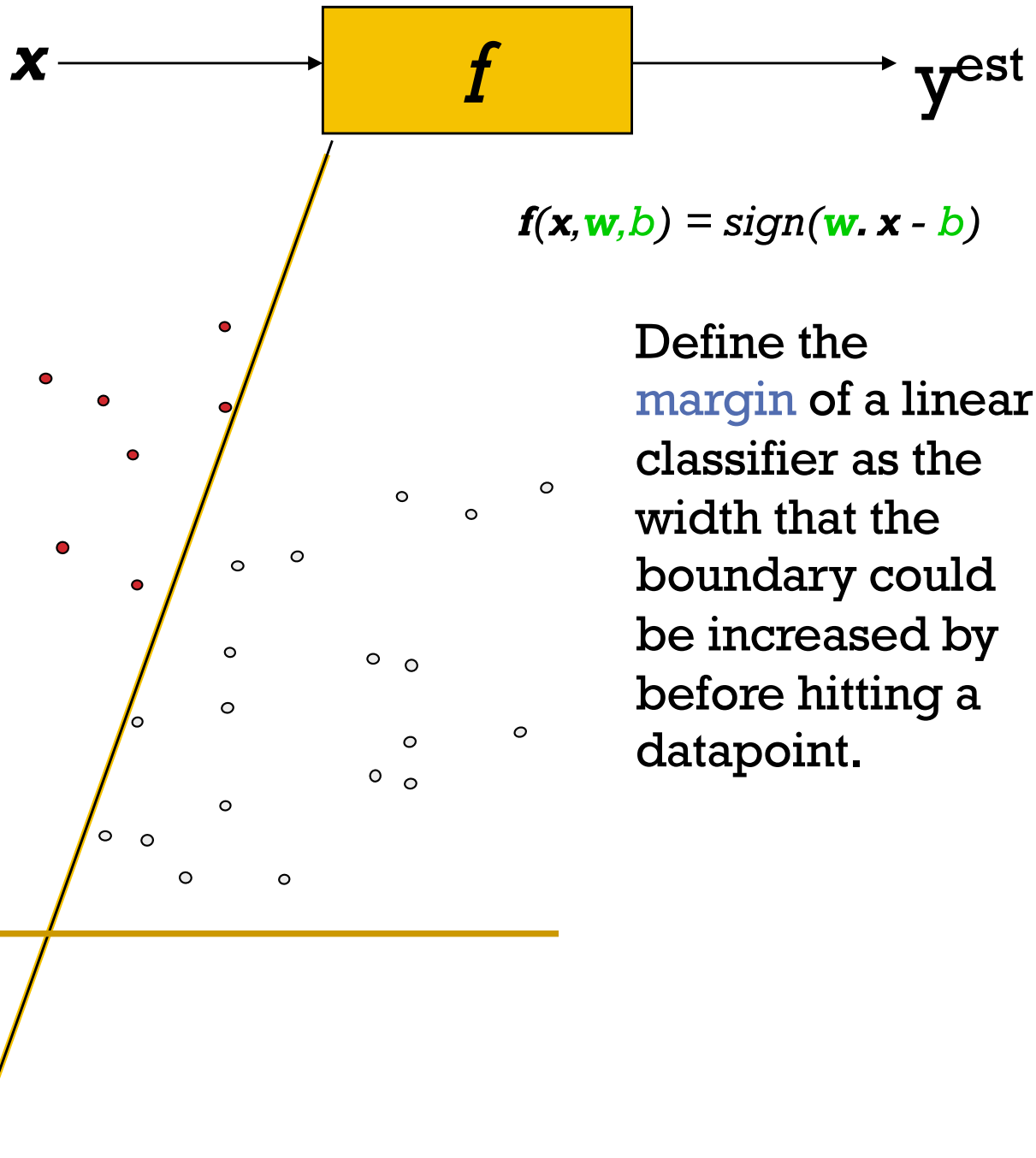


Any of these would be fine..

..but which is best?

Slides from Andrew W. Moore

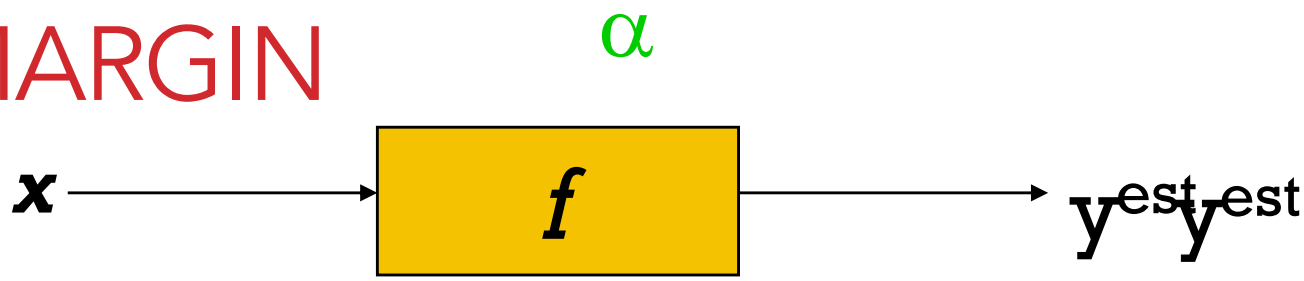
CLASSIFIER MARGIN

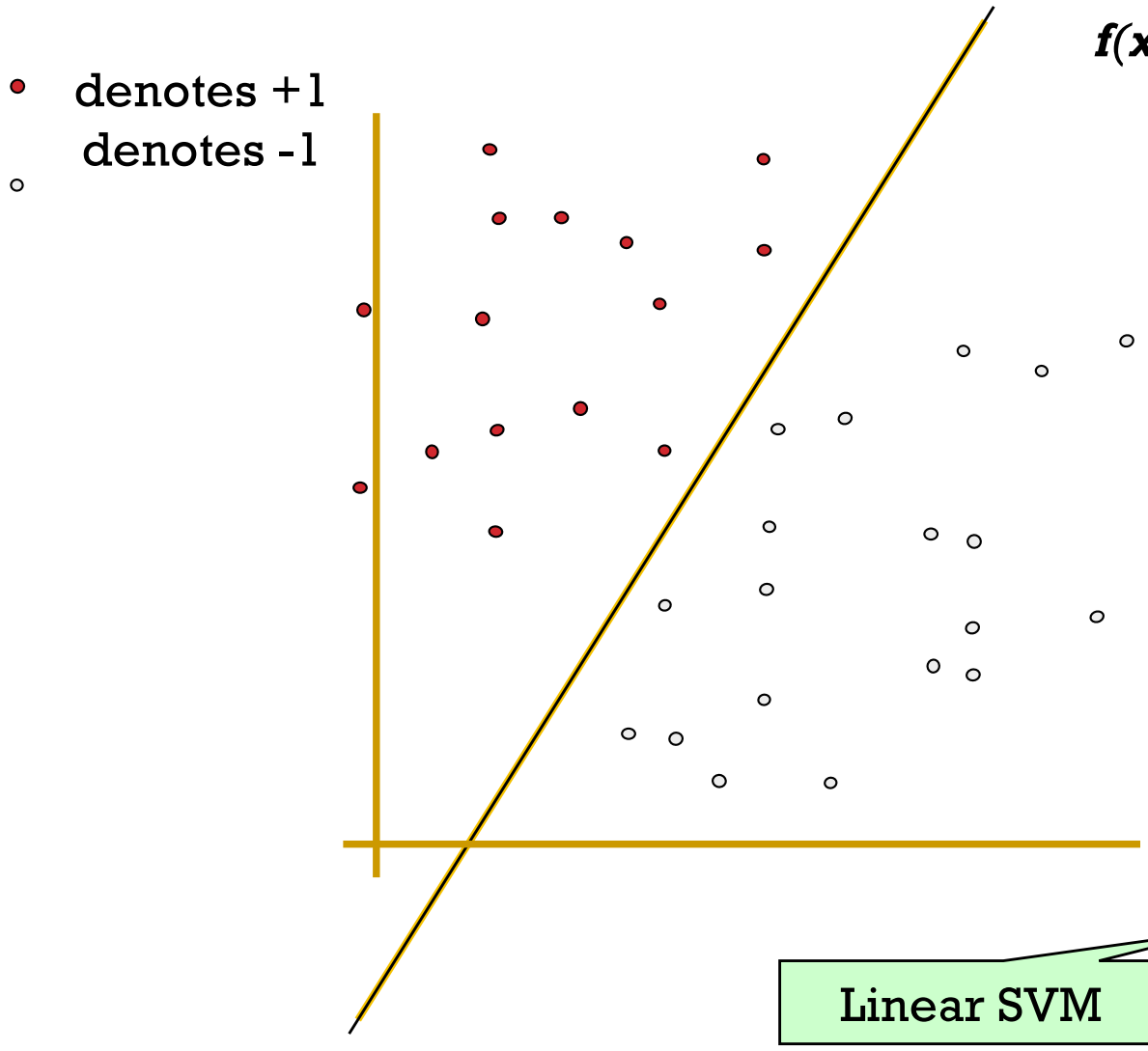


Define the margin of a linear classifier as the width that the boundary could be increased by before hitting a datapoint.

Slides from Andrew W. Moore

MAXIMUM MARGIN





$f(x,w,b) = sign(w.x - b)$

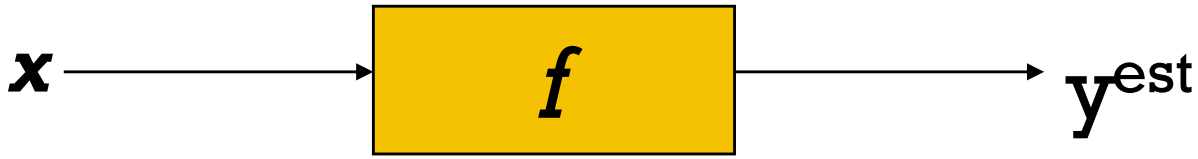
The maximum margin linear classifier is the linear classifier with the, um, maximum margin. This is the simplest kind of SVM (Called an LSVM)

Linear SVM

Slides from Andrew W. Moore

MAXIMUM MARGIN

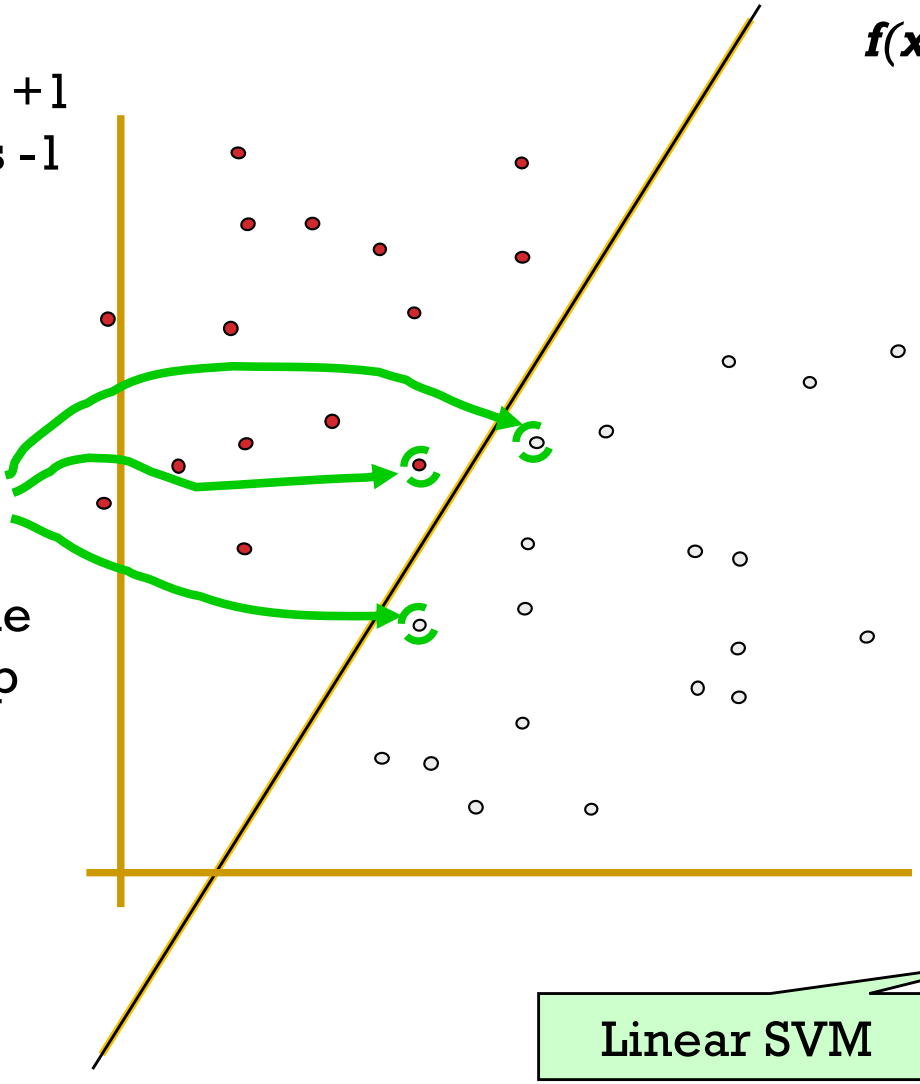
α



$f(x,w,b) = sign(w.x - b)$

denotes +1

denotes -1

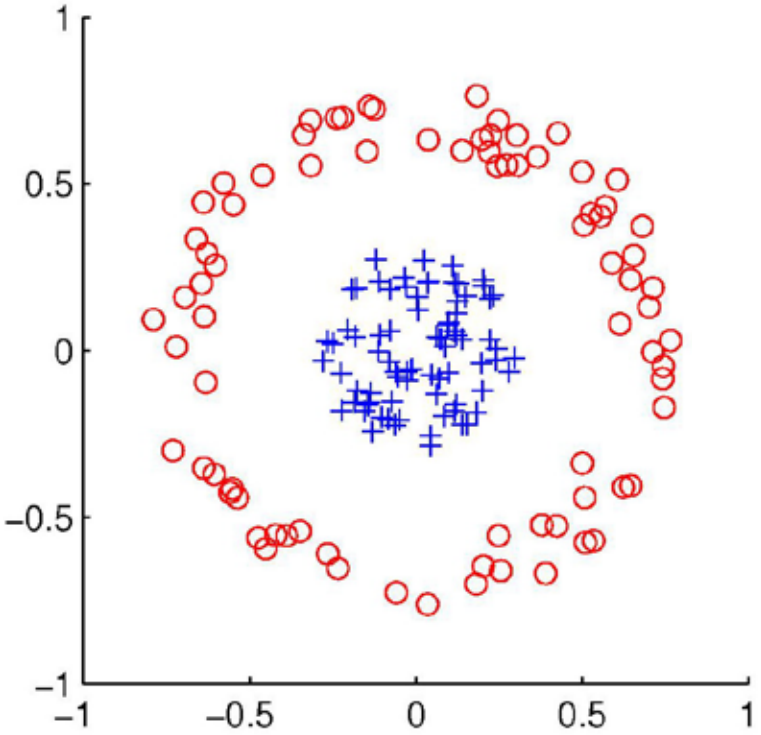


Support Vectors are those datapoints that the margin pushes up against

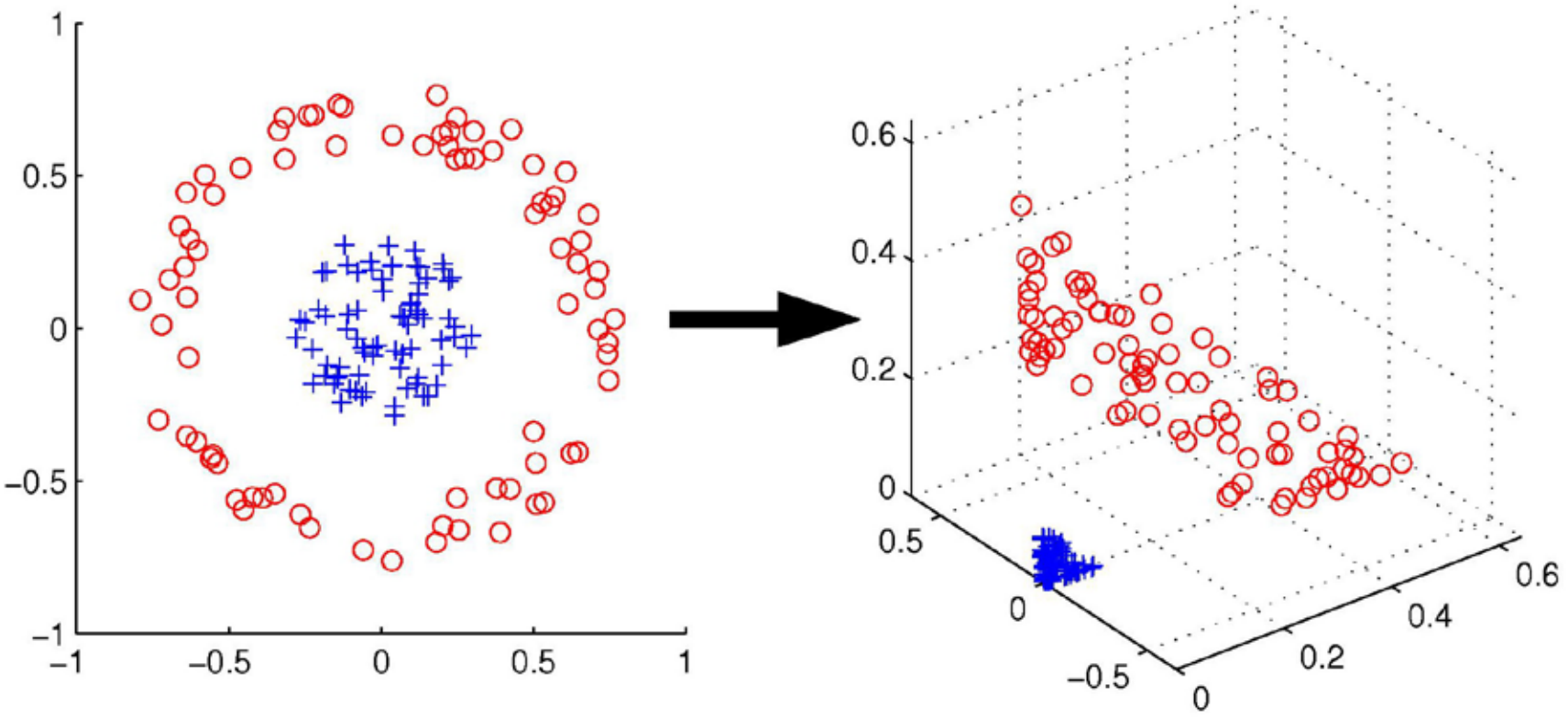
The maximum margin linear classifier is the linear classifier with the, um, maximum margin. This is the simplest kind of SVM (Called an LSVM)

Linear SVM

THE KERNEL TRICK

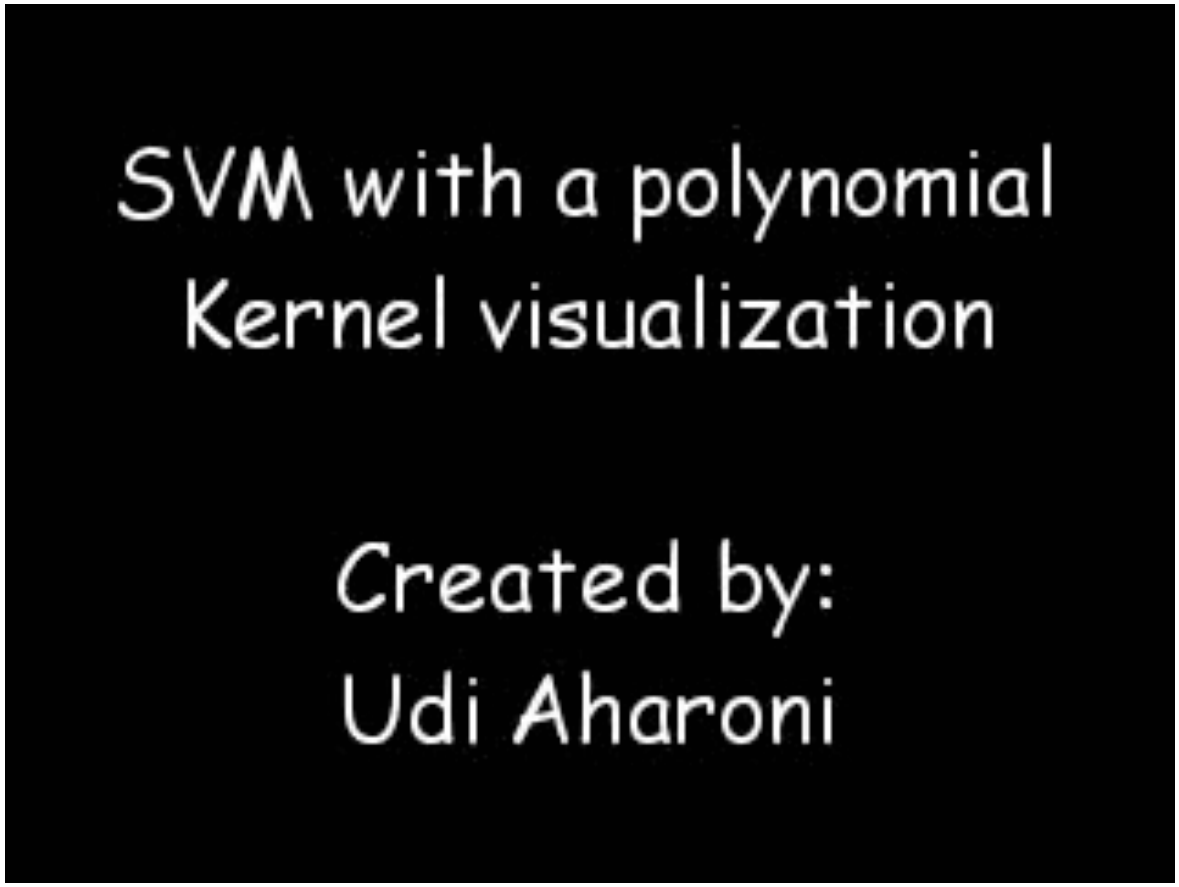


THE KERNEL TRICK



$$\phi: \quad \Re^2 \longrightarrow \Re^3$$
$$(x_1, x_2) \longmapsto (z_1, z_2, z_3) = (x_1^2, \sqrt{2}x_1x_2, x_2^2)$$

[http://www.cs.berkeley.edu/~jordan/courses/281B-spring04/lectures/lec3.pdf]



https://www.youtube.com/watch?v=3liCbRZPrZA

MANY CLASSIFIERS TO CHOOSE FROM

K-nearest neighbor

Support Vector Machines

Decision Trees

Random Forrest

(Gradient) Boosted Decision Trees

Logistic Regression

Naïve Bayes

Bayesian network

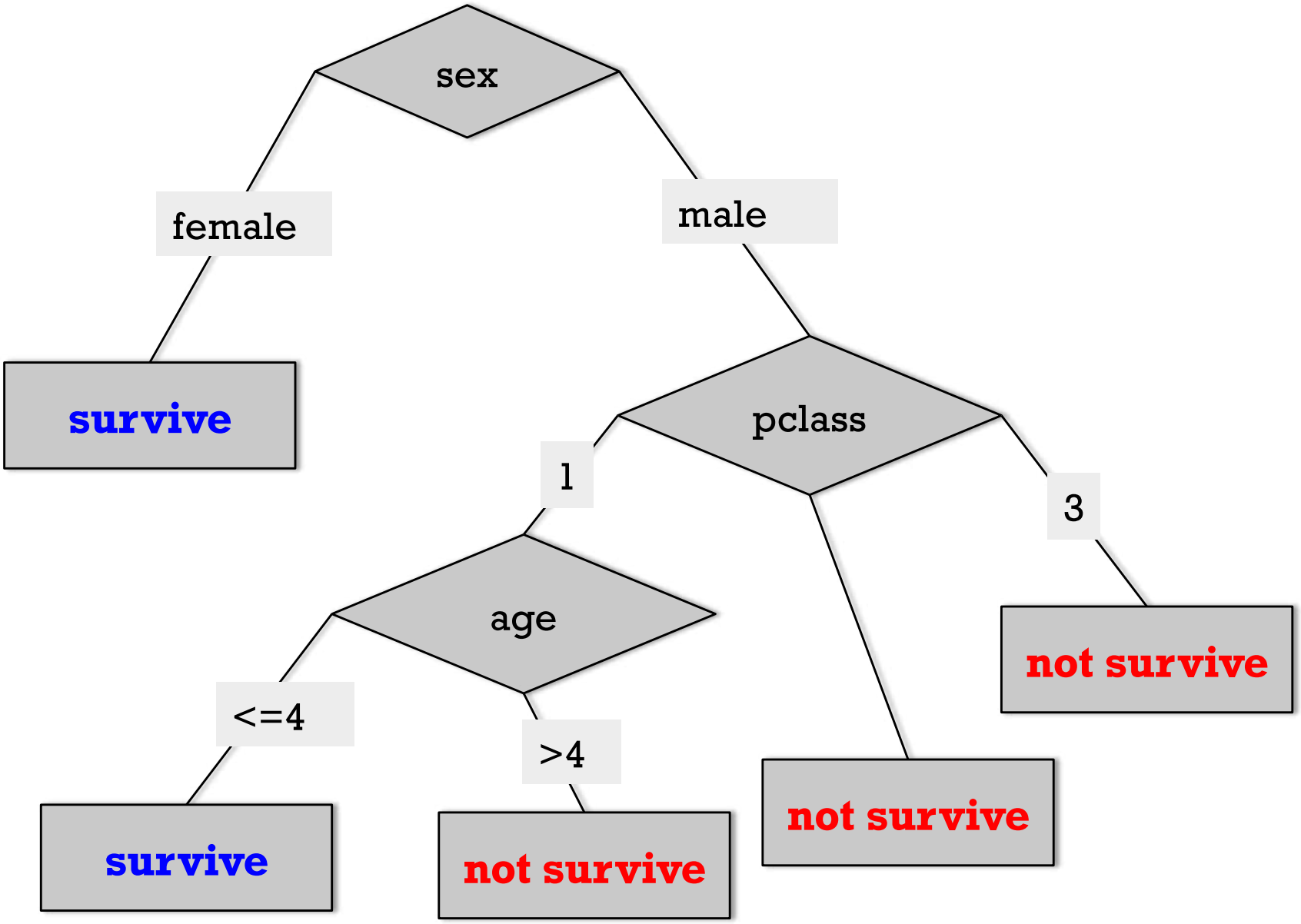
RBMs

….

Which is the best one?

Slide credit: D. Hoiem

DECISION TREES



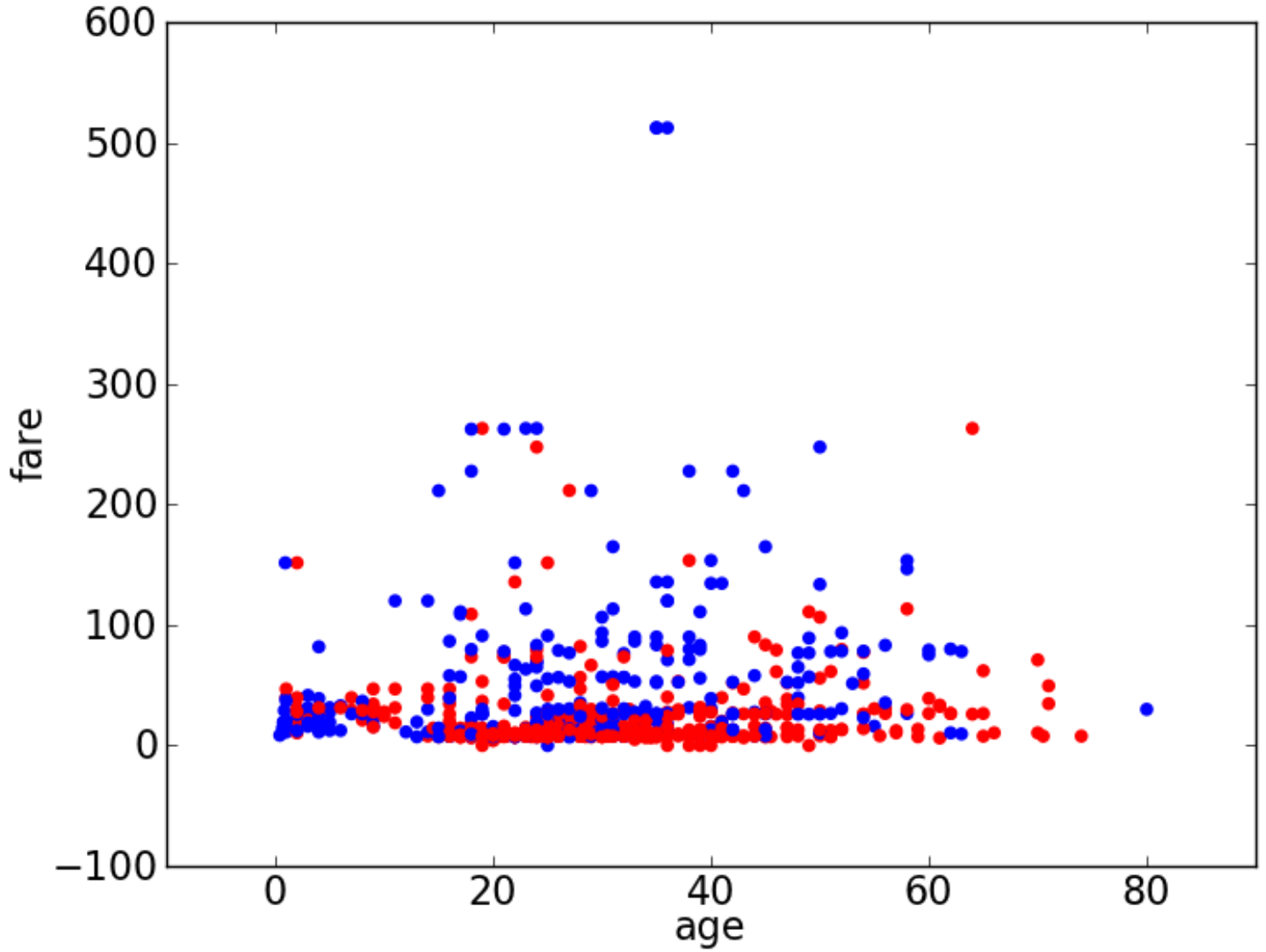
BUILDING A DECISION TREE (ID3 ALGORITHM)

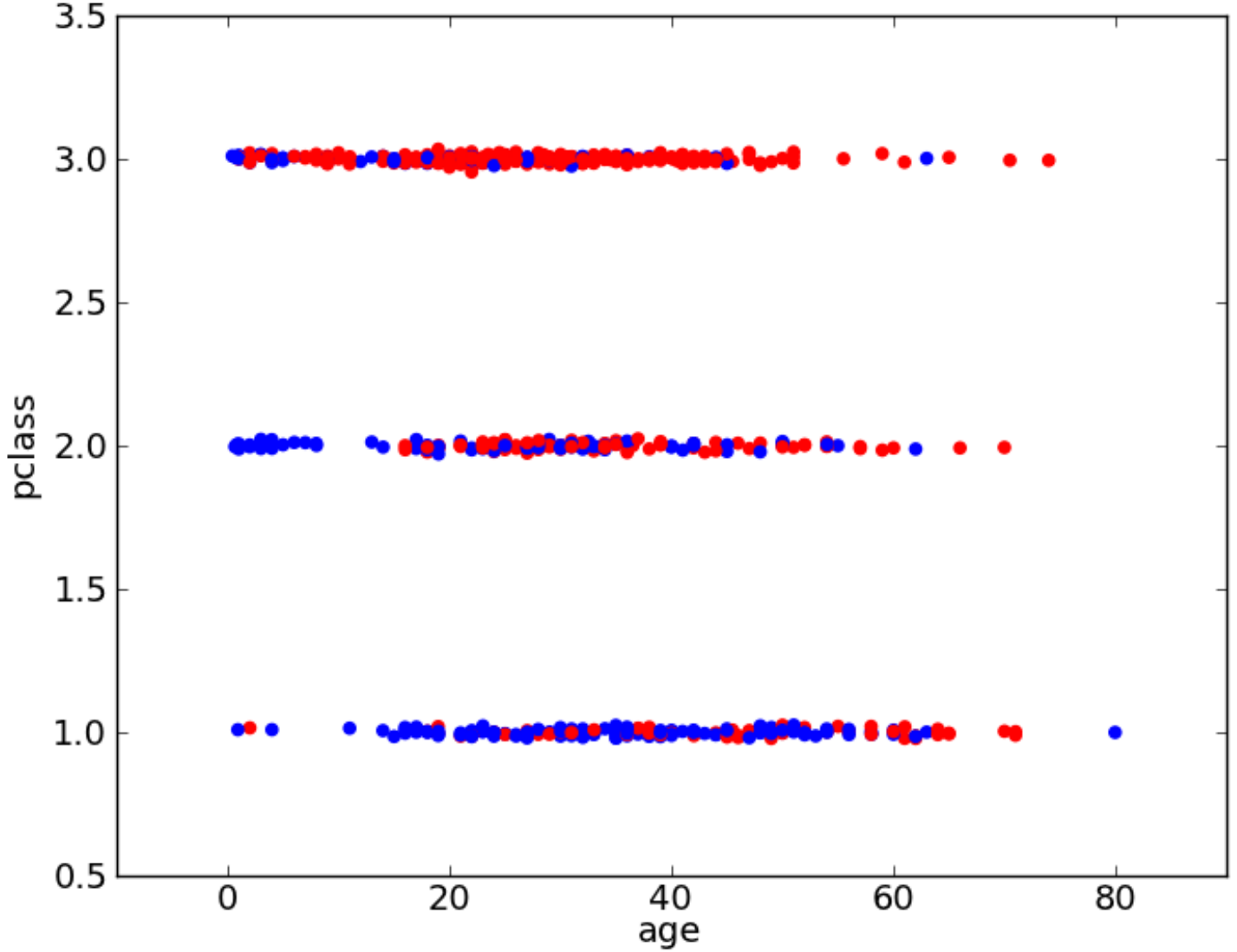
1. Assume attributes are discrete
    - Discretize continuous attributes
2. Choose the attribute with the highest Information Gain
3. Create branches for each value of attribute
4. Examples partitioned based on selected attributes
5. Repeat with remaining attributes
6. Stopping conditions
    - All examples assigned the same label
    - No examples left

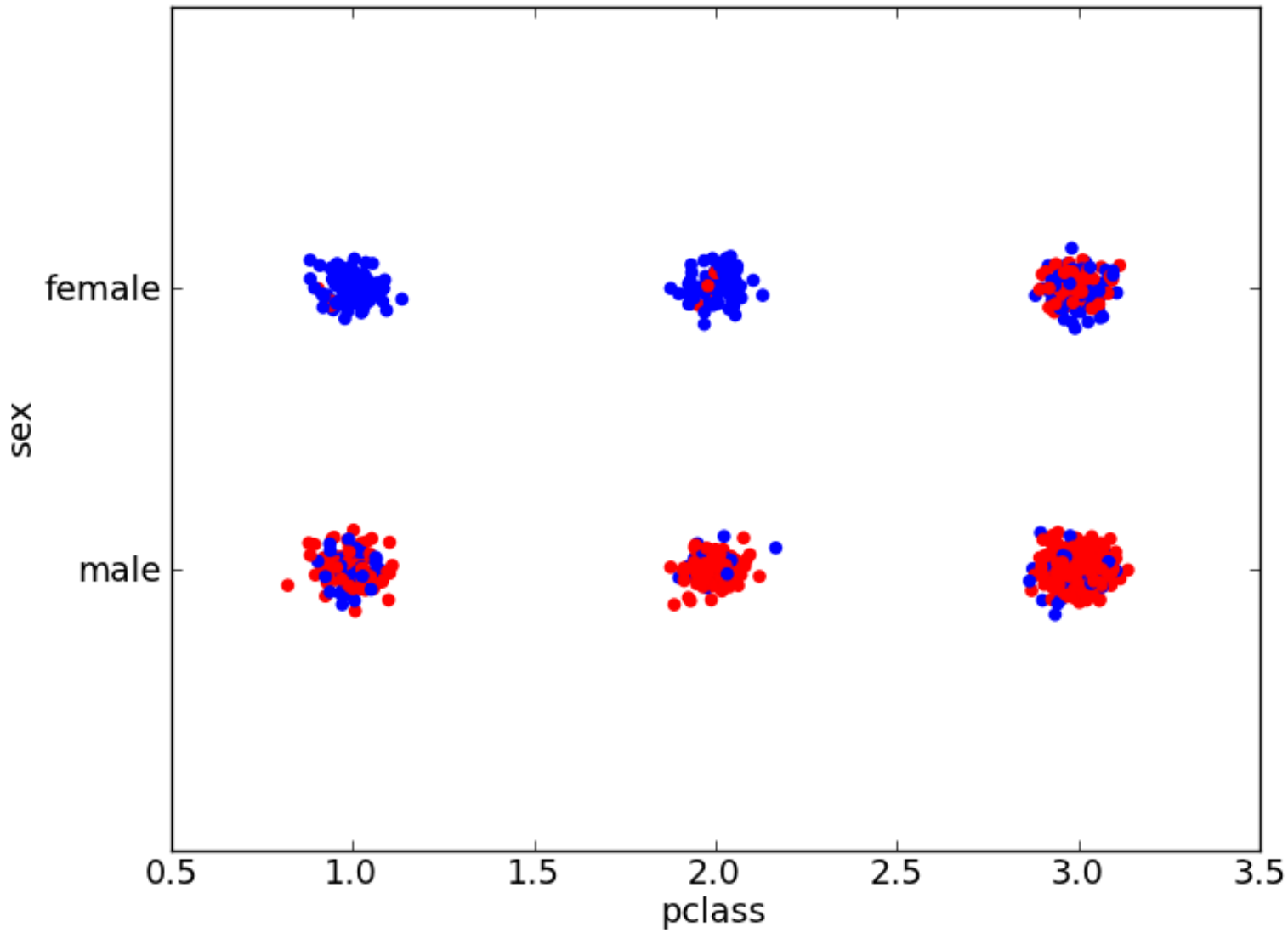
HIGHEST INFORMATION GAIN

Titanic Data

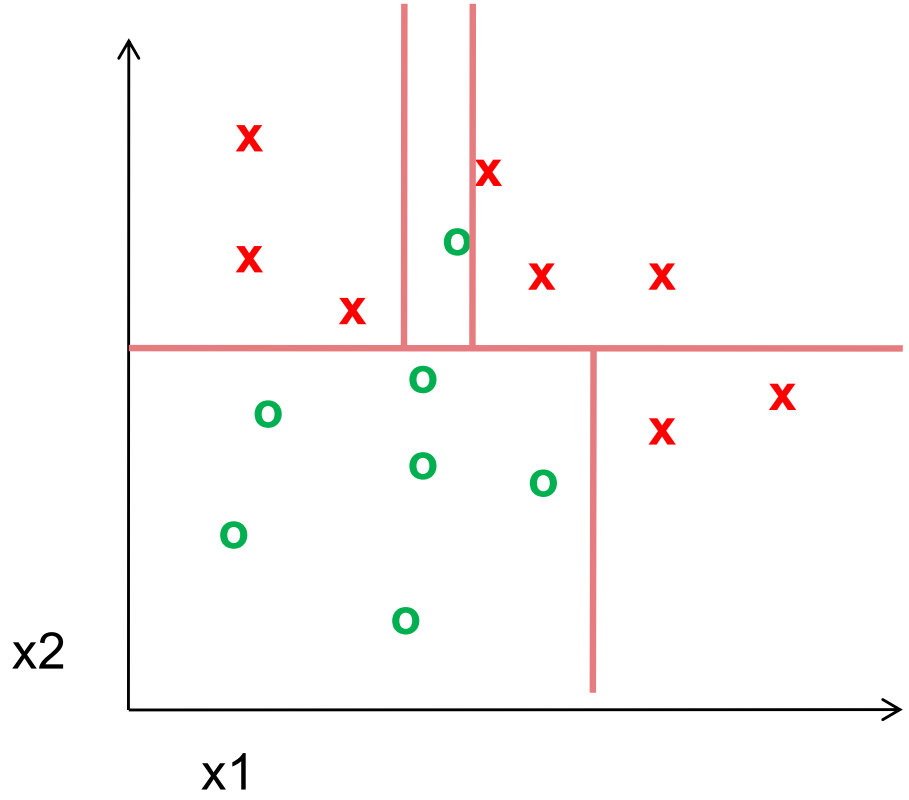
survived	pclass	sex	age	sibsp	parch	fare	cabin	embarked
0	3	male	22	1	0	7.25		S
1	1	female	38	1	0	71.2833	C85	C
1	3	female	26	0	0	7.925		S
1	1	female	35	1	0	53.1	C123	S
0	3	male	35	0	0	8.05		S
0	3	male		0	0	8.4583		Q
0	1	male	54	0	0	51.8625	E46	S
0	3	male	2	3	1	21.075		S
1	3	female	27	0	2	11.1333		S
1	2	female	14	1	0	30.0708		C
1	3	female	4	1	1	16.7	G6	S
1	1	female	58	0	0	26.55	C103	S
0	3	male	20	0	0	8.05		S


600
500
400
300
200
100
0
−100
fare
0
20
40
60
80
age


3.5
3.0
2.5
2.0
1.5
1.0
0.5
pclass
0
20
40
60
80
age


female
male
sex
0.5
1.0
1.5
2.0
2.5
3.0
3.5
pclass

DECISION BOUNDARIES: DECISION TREES



MANY CLASSIFIERS TO CHOOSE FROM

K-nearest neighbor

Support Vector Machines

Decision Trees

Random Forrest

(Gradient) Boosted Decision Trees

Logistic Regression

Naïve Bayes

Bayesian network

RBMs

….

Which is the best one?

Slide credit: D. Hoiem

Ensemble Methods

ENSEMBLE METHODS

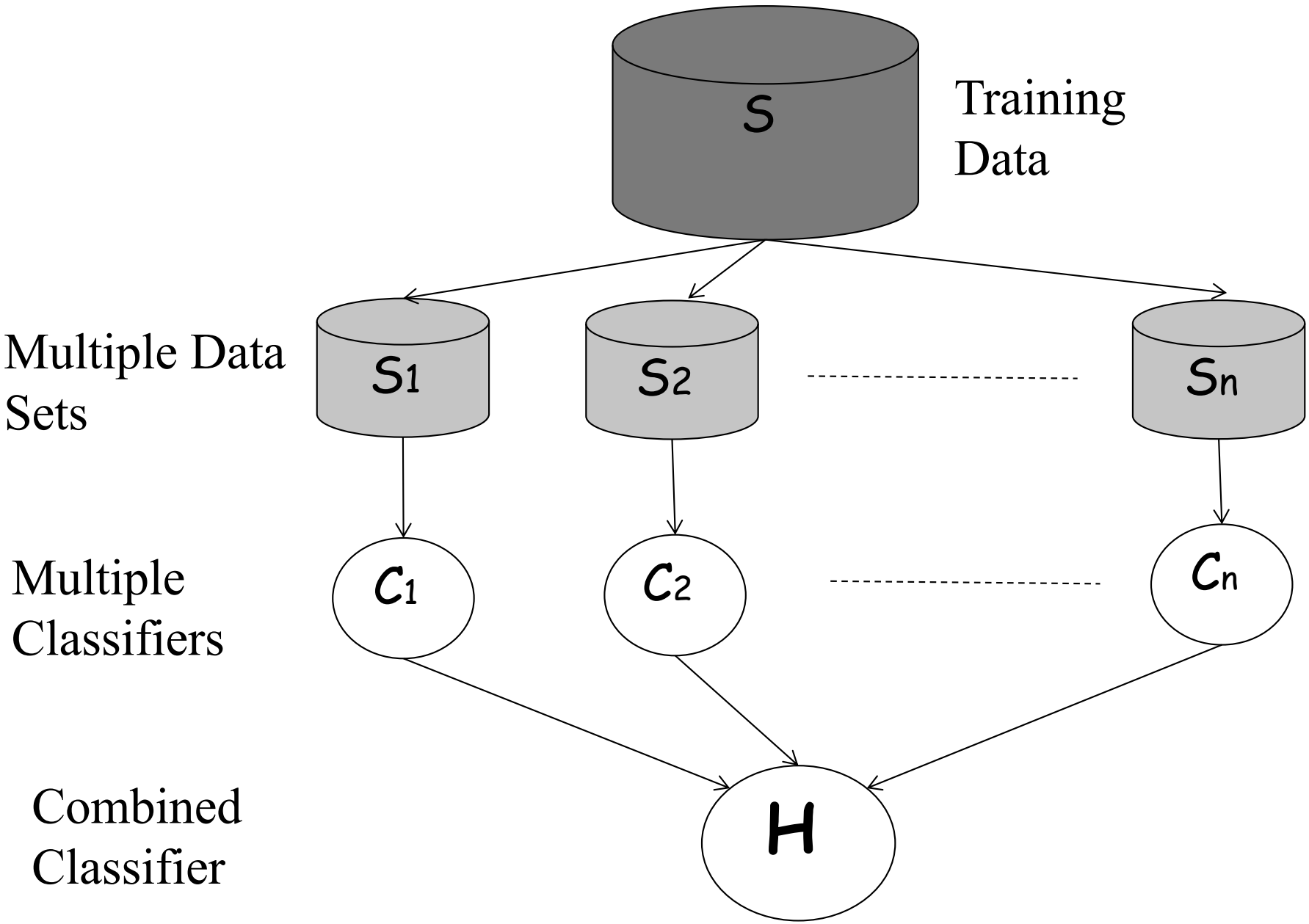
Bagging (Breiman 1994,…)

Random forests (Breiman 2001,…)

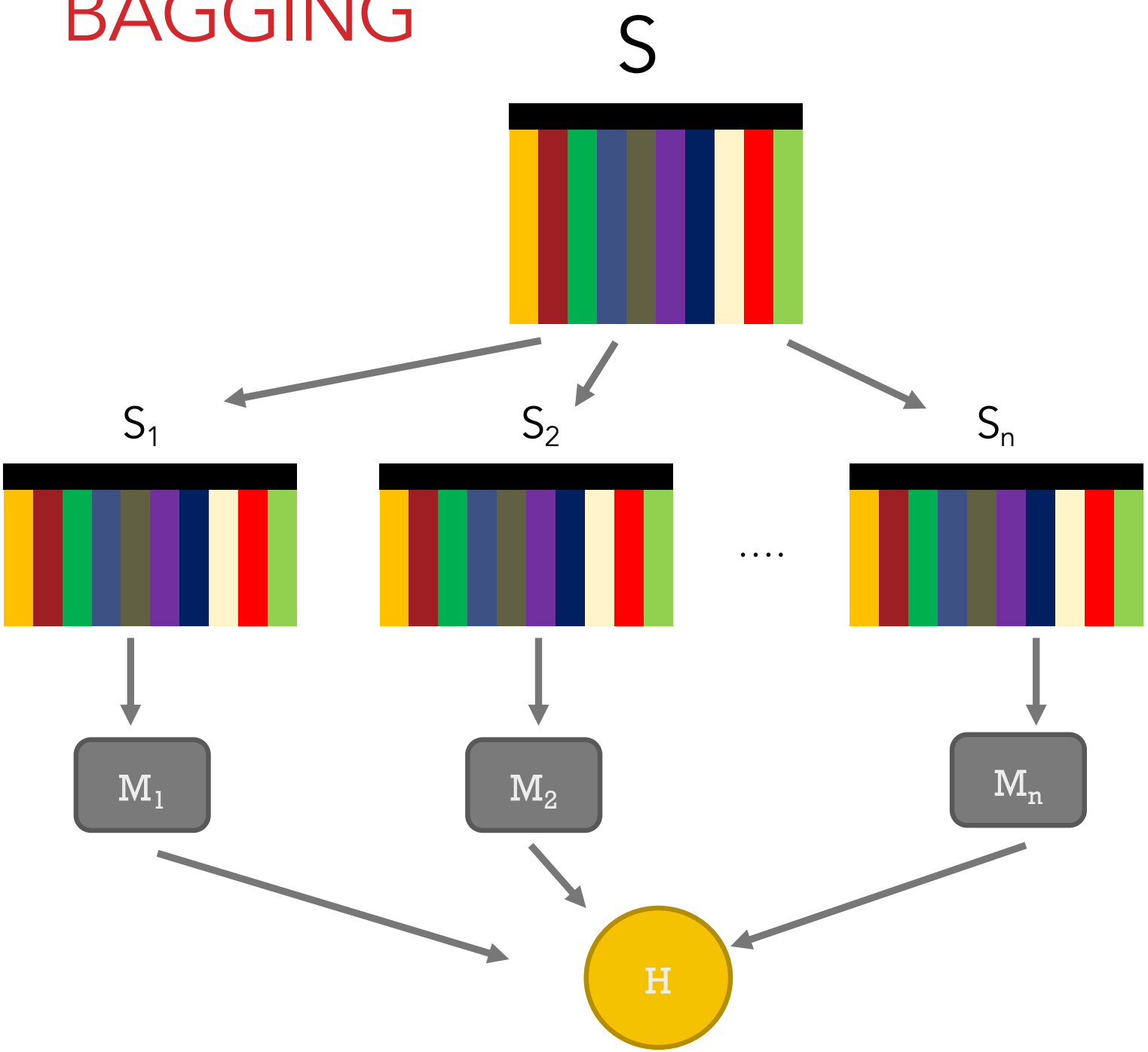
Boosting (Freund and Schapire 1995, Friedman et al. 1998,…)

Predict class label for unseen data by aggregating a set of predictions (classifiers learned from the training data).

GENERAL IDEA



BAGGING



Sample with replacement

Build classifier over sample

Use majority (or avg. for regression)

BAGGING

- Can help a lot if data is noisy.
- Bagging works because it reduces variance by voting/averaging
  - In some pathological hypothetical situations the overall error might increase
  - Usually, the more classifiers the better
- **If the learning algorithm is unstable, then Bagging almost always improves performance**
  - Learning algorithm is unstable: if small changes to the training set cause large changes in the learned classifier.
  - Some candidates: Decision tree, decision stump, regression tree, linear regression, SVMs

ENSEMBLE METHODS

Bagging (Breiman 1994,…)

Random forests (Breiman 2001,…)

Boosting (Freund and Schapire 1995, Friedman et al. 1998,…)

THE RANDOM FORESTS ALGORITHM

Given a training set S

For i = 1 to k do:

Build subset Si by sampling with replacement from S

Learn tree Ti from Si

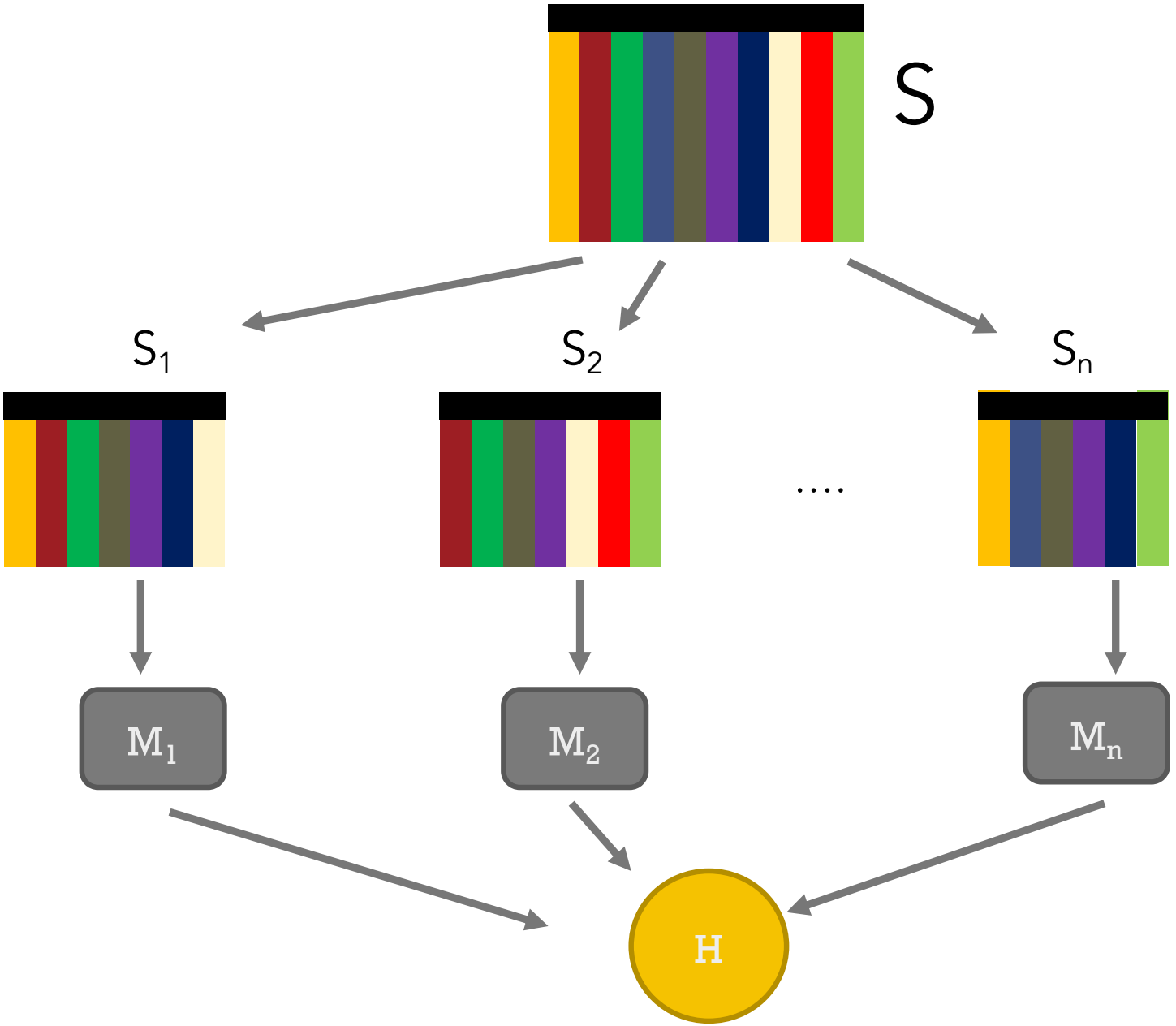
At each node:

Choose best split from random subset of F features

Each tree grows to the largest extend, and no pruning

Make predictions according to majority vote of the set of k trees.

RANDOM FORREST



Sample with Replacement & select random subset of features*

Build classifier over sample

Use majority Vote for classification (or avg. for regression)

* Normally done for each node of the decision tree – not once

ENSEMBLE METHODS

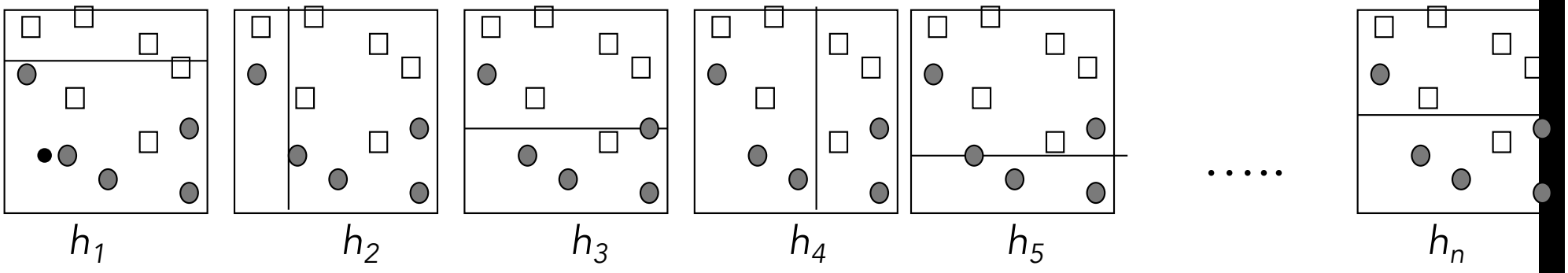
Bagging (Breiman 1994,…)

Random forests (Breiman 2001,…)

Boosting (Freund and Schapire 1995, Friedman et al. 1998,…)

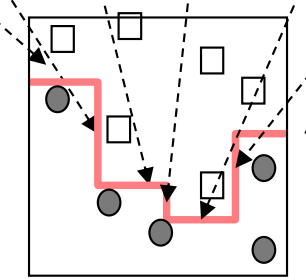
ADABOOST - CORE IDEA

Take a set of weak classifiers (normally they should do better than guessing)



Classification Result

θ_1 θ_2 θ_3 θ_4 θ_5 θ_n

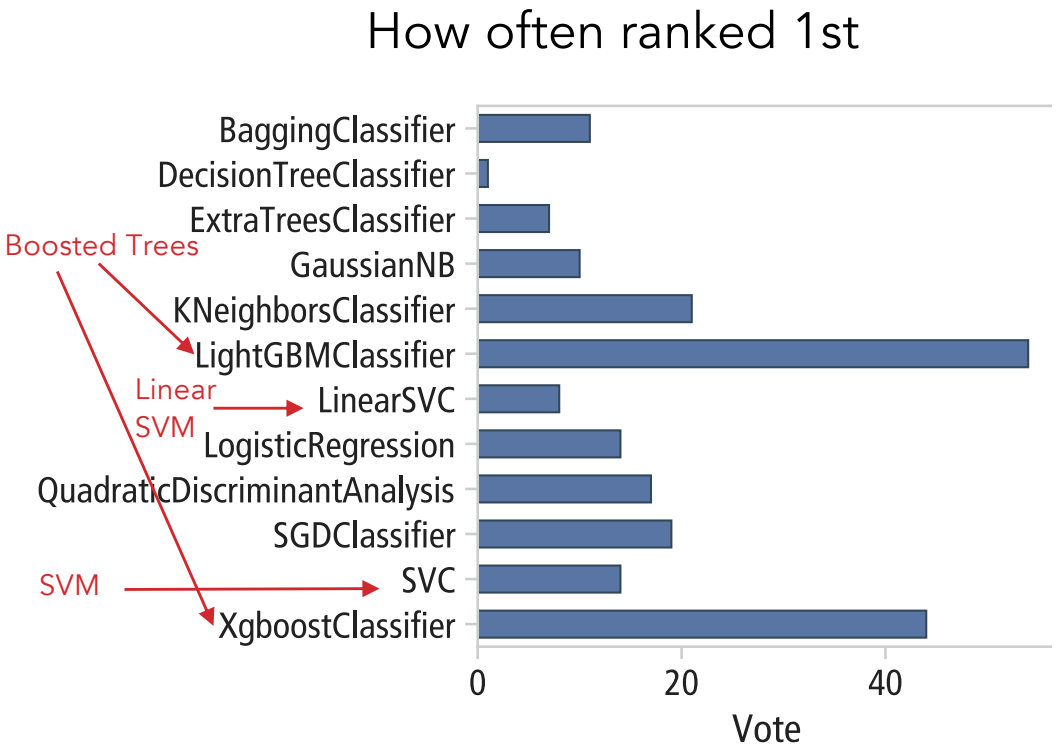


Weight the result of each classify with θ

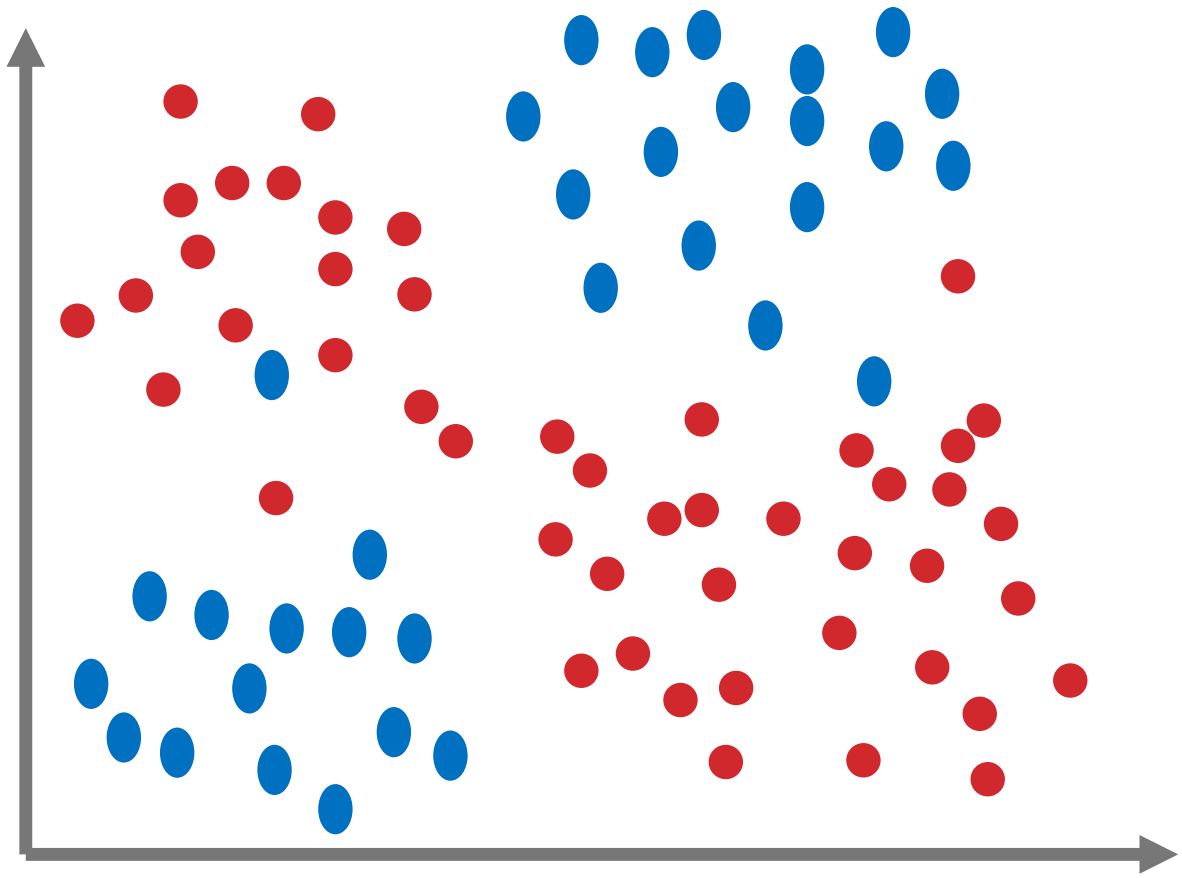
Combine to form the Final strong classifier

$$H(x) = sign\left(\sum_{i=1}^{n} \theta_i h_i(x)\right)$$

PERFORMANCE OF DIFFERENT ML MODEL FAMILIES



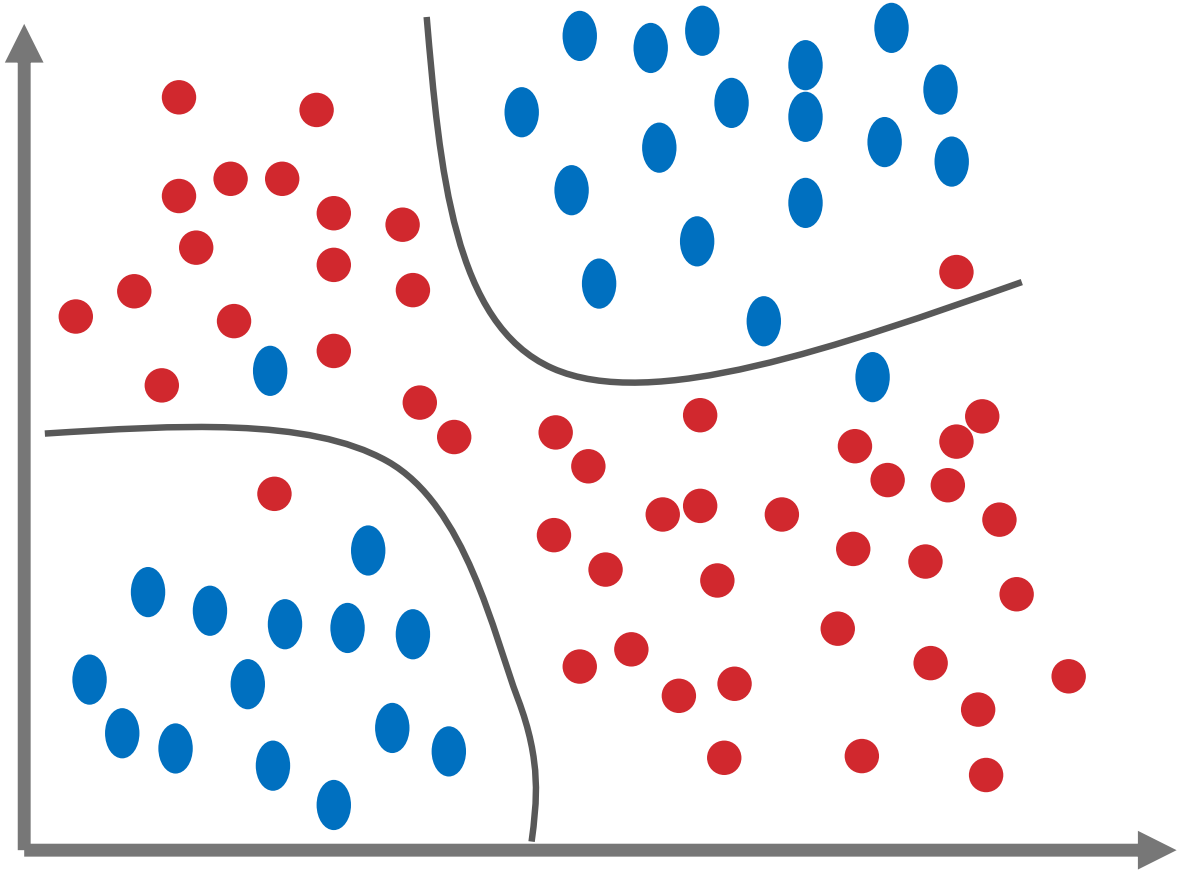
IN-CLASS TASK



How would you draw the expected decision boundary for

- Random Forrest
- SVM w/ kernel and regularization
- 1-KNN

CLICKER



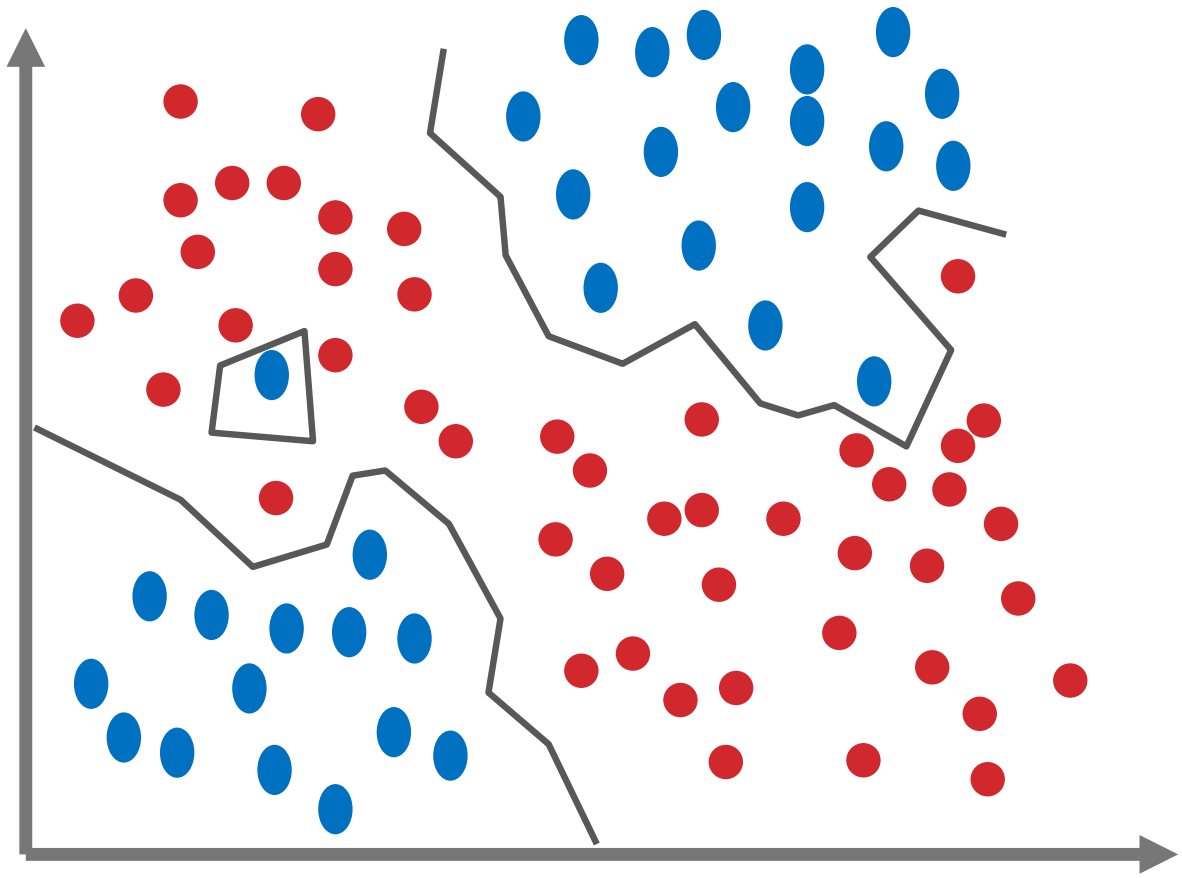
The decision boundary looks like the one of:

a) Random Forrest

b) SVM w/ kernel and regularization

c) 1-KNN

CLICKER



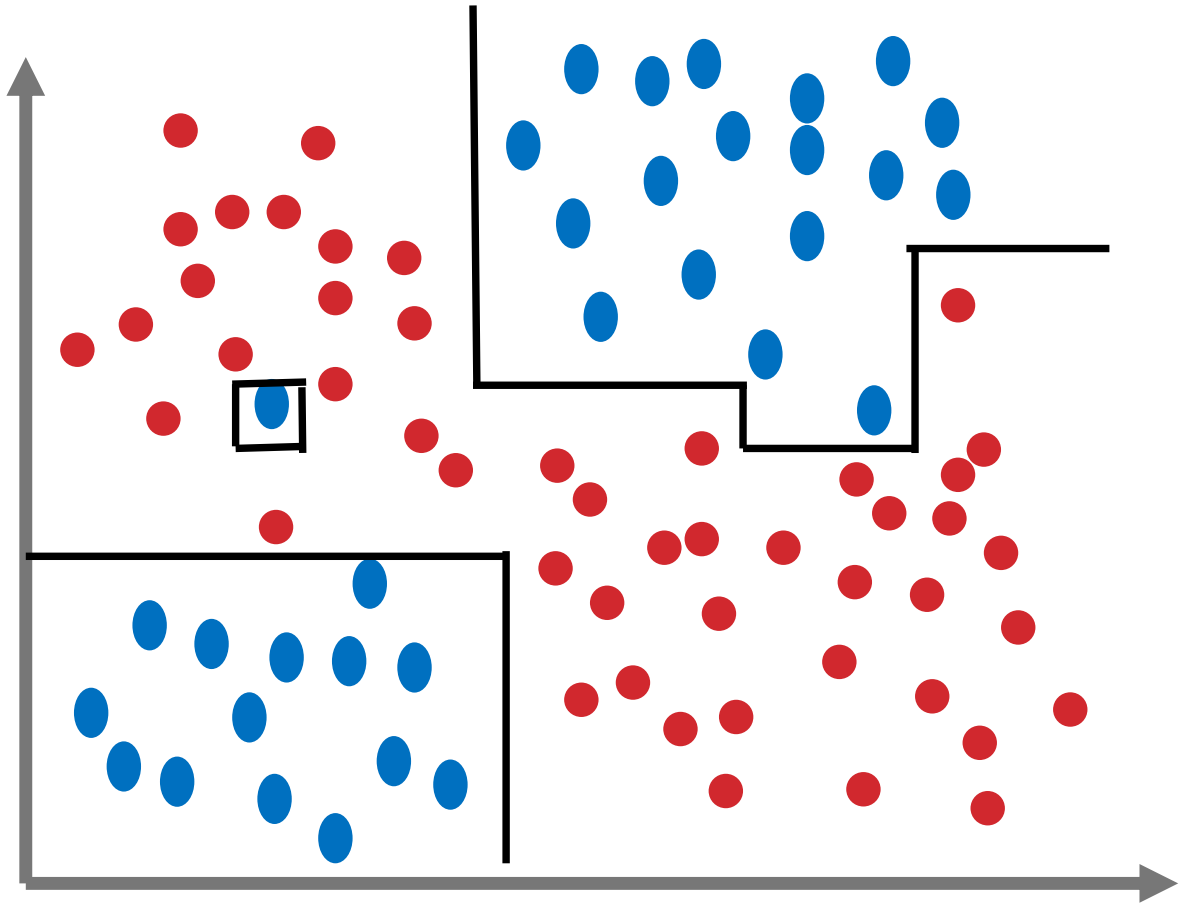
The decision boundary looks like the one of:

a) Random Forrest

b) SVM w/ kernel and regularization

c) 1-KNN

RANDOM FORREST



The decision boundary looks like the one of:

a) Random Forrest

b) SVM w/ kernel and regularization

c) 1-KNN

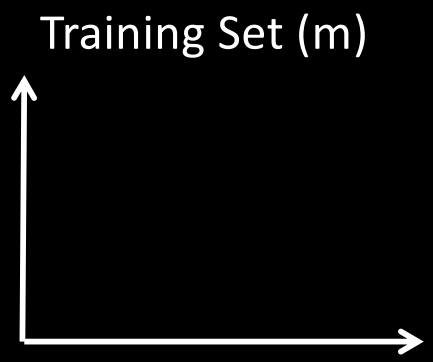
Machine Learning

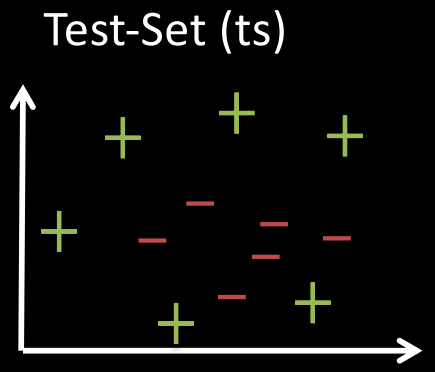
What if you model has an high error?

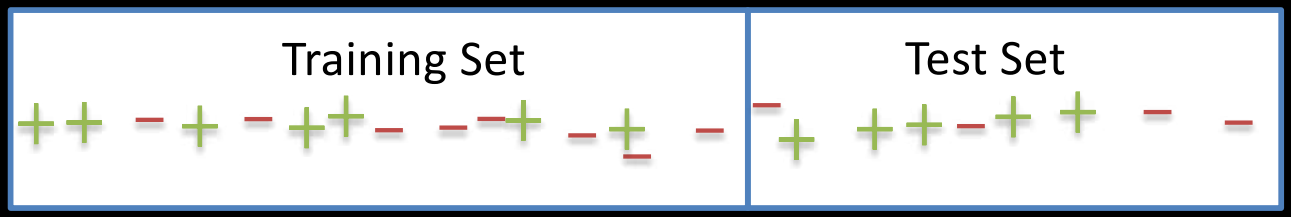
- Try getting more training examples
- Try smaller sets of features
- Try getting additional features
- Try creating features from existing features (kernels)
- Try decrease regularization
- Try increase regularization

Bias and Variance

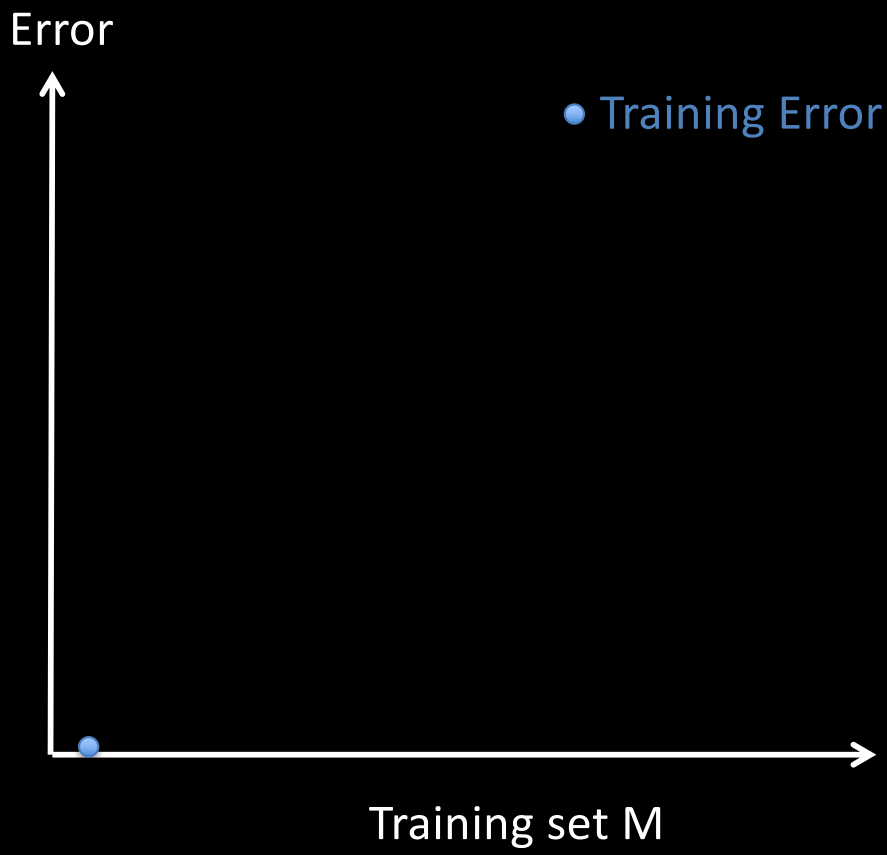
Bias and Variance

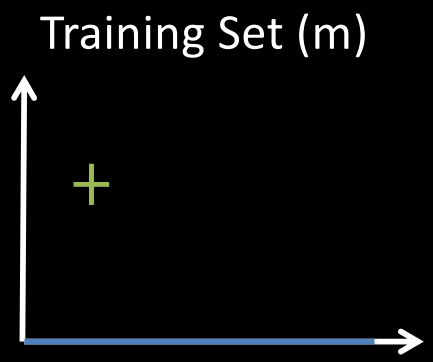




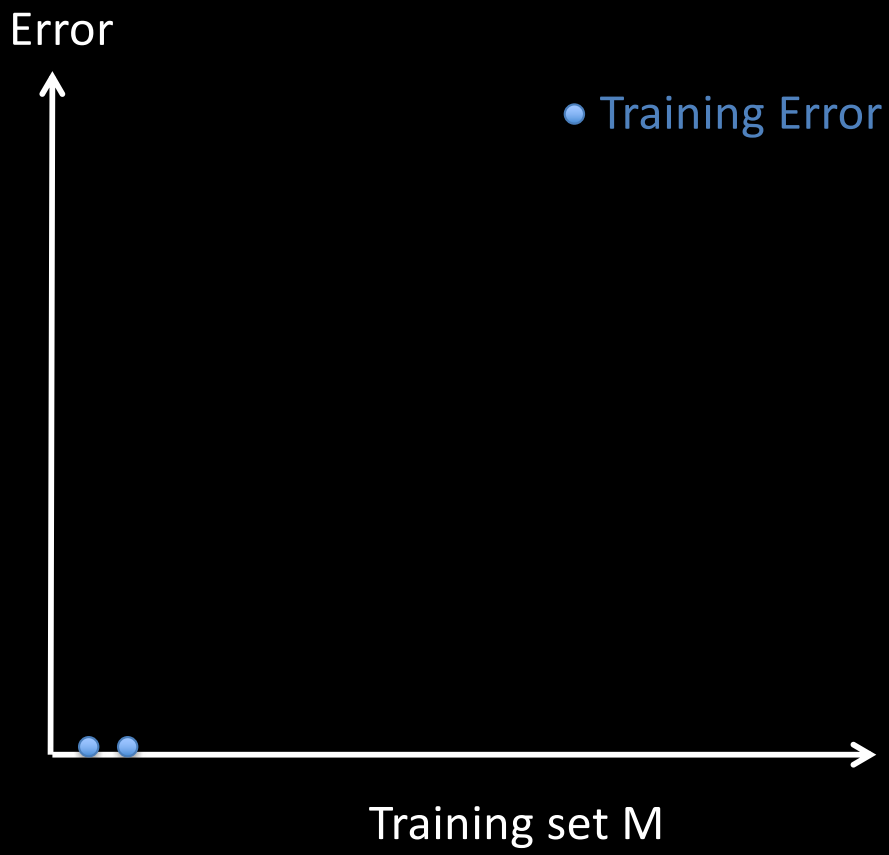


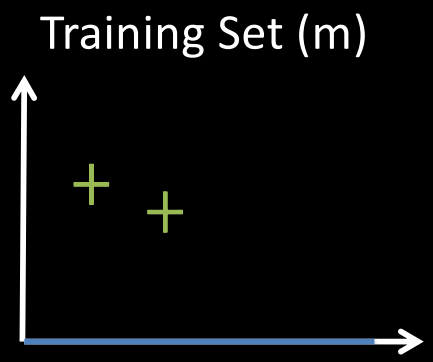
Bias and Variance



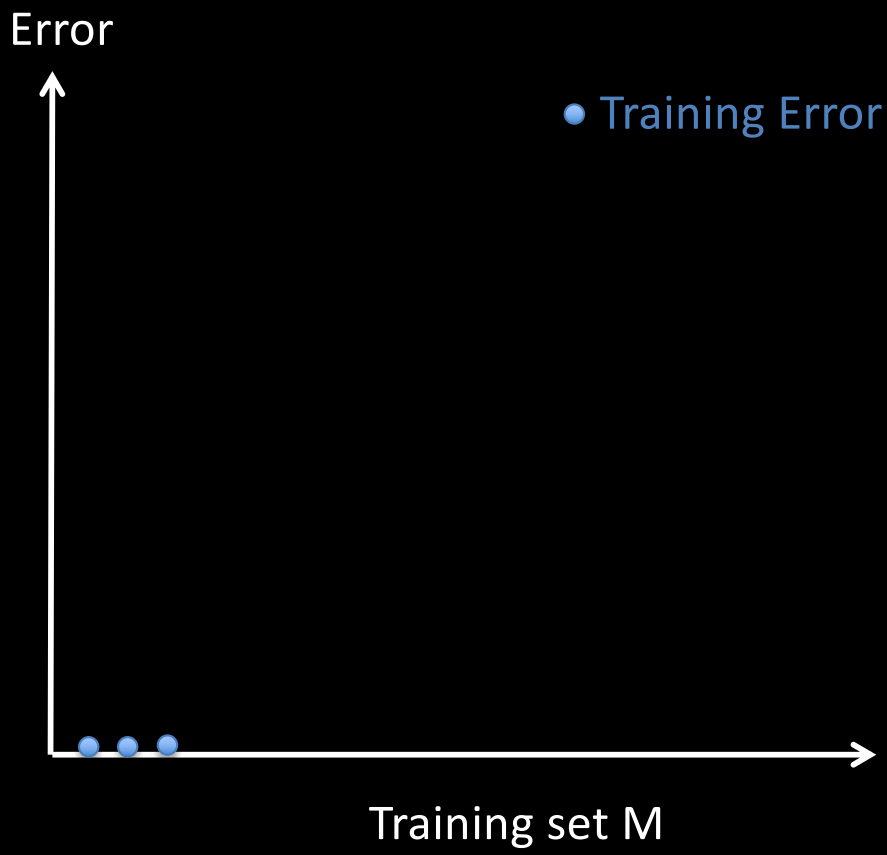


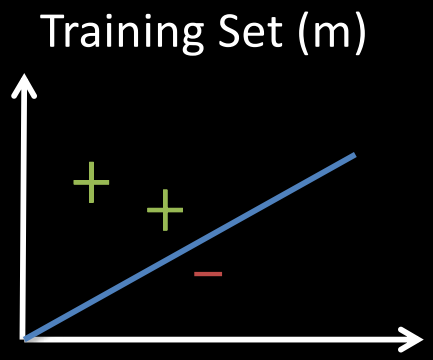
Bias and Variance



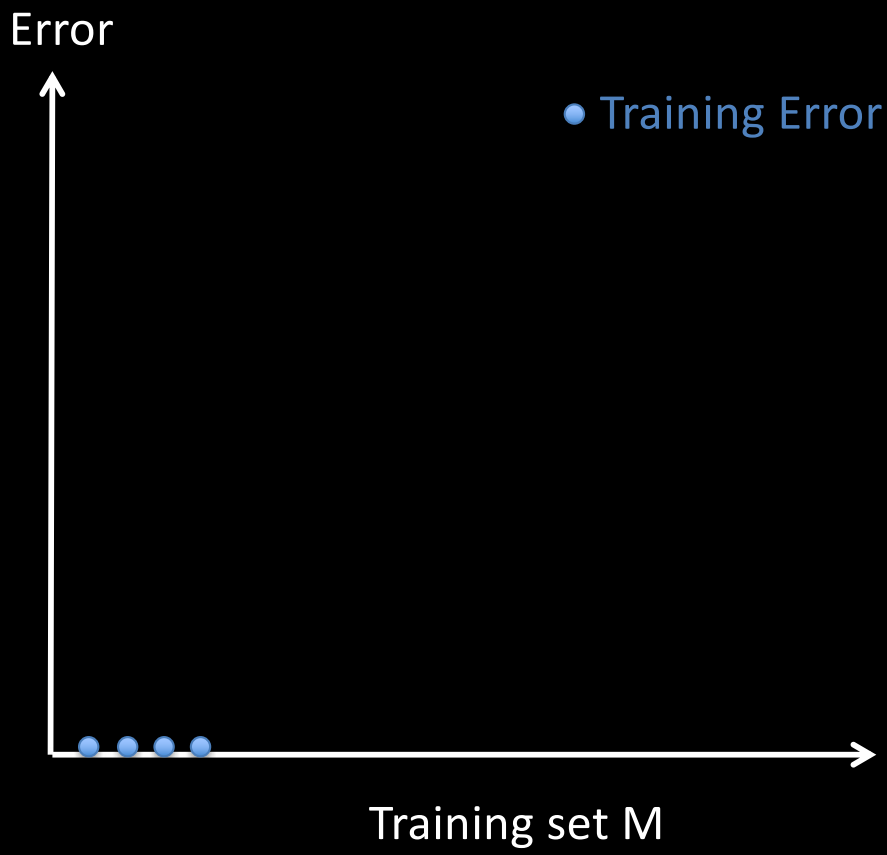


Bias and Variance

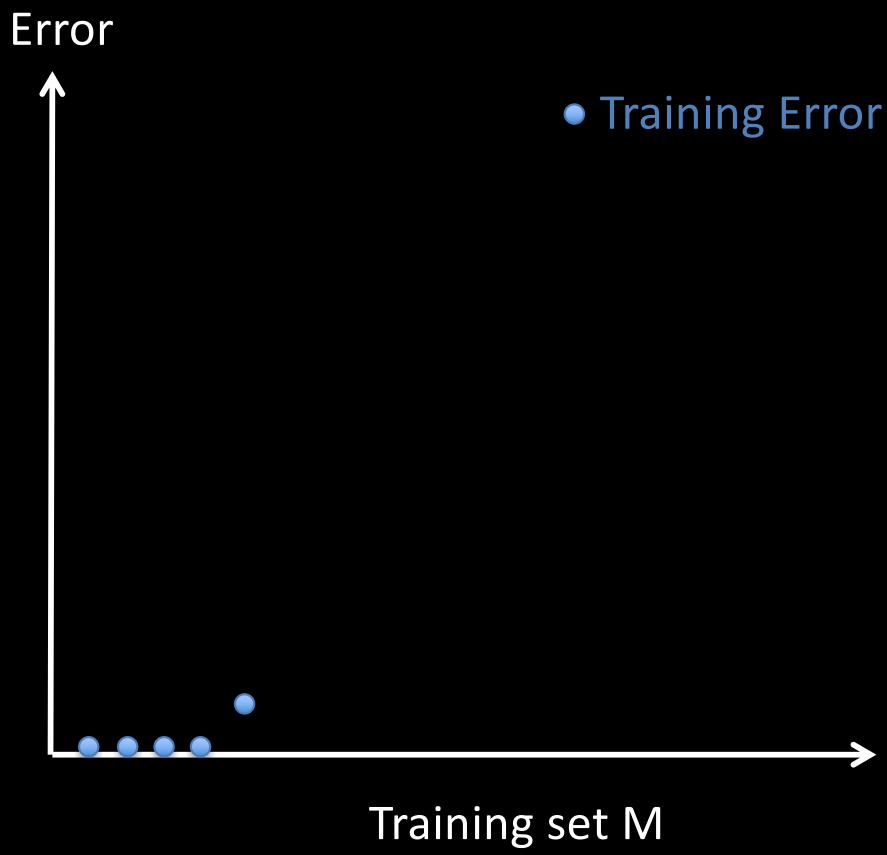




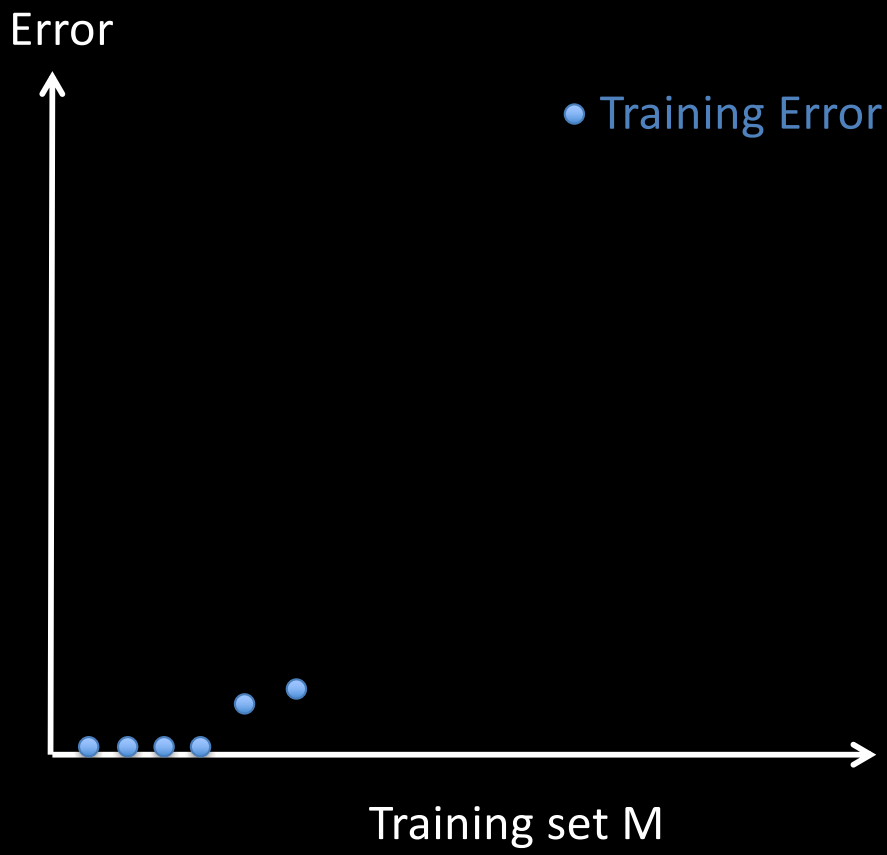
Bias and Variance

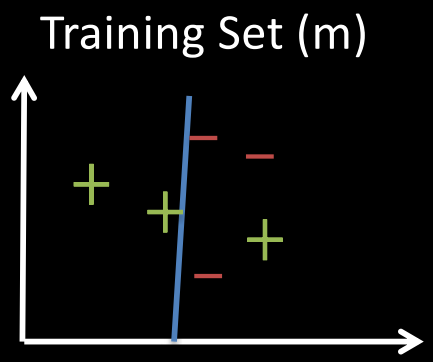


Bias and Variance

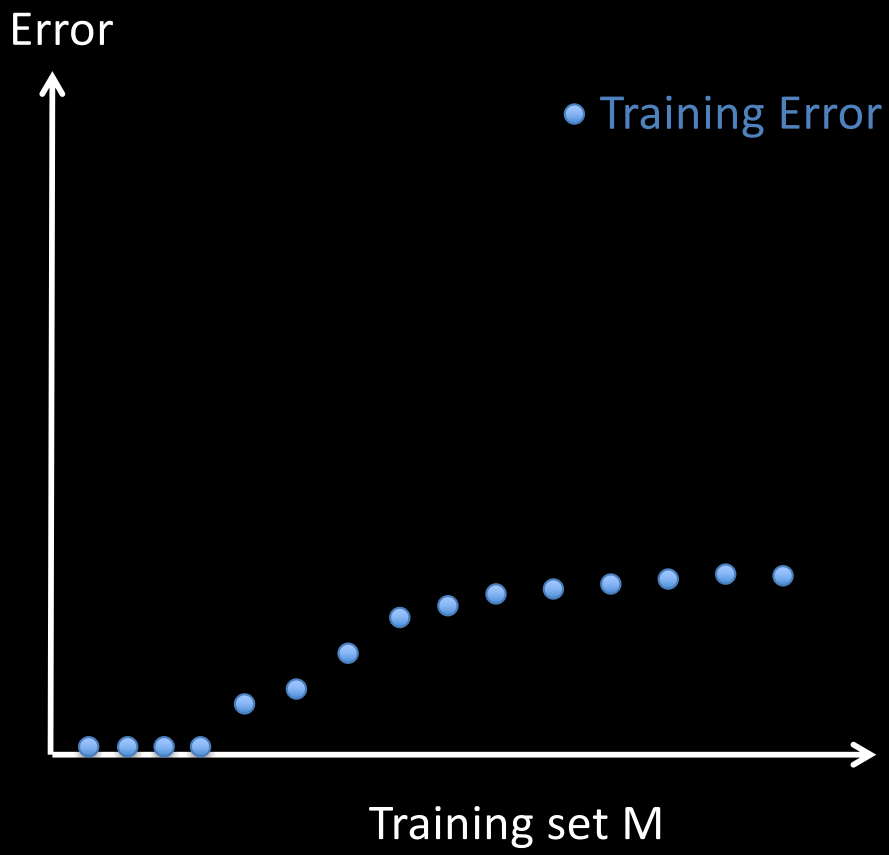


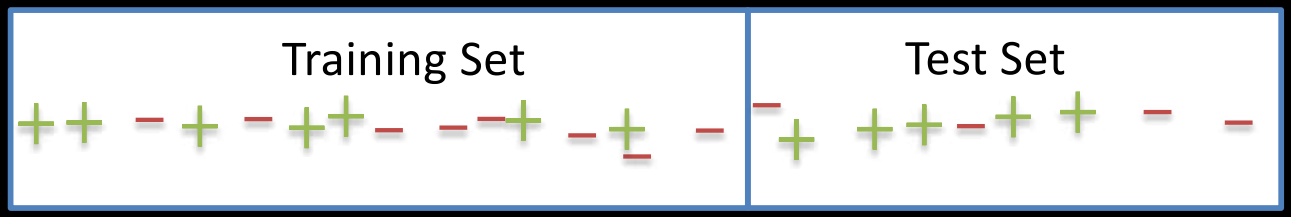
Bias and Variance



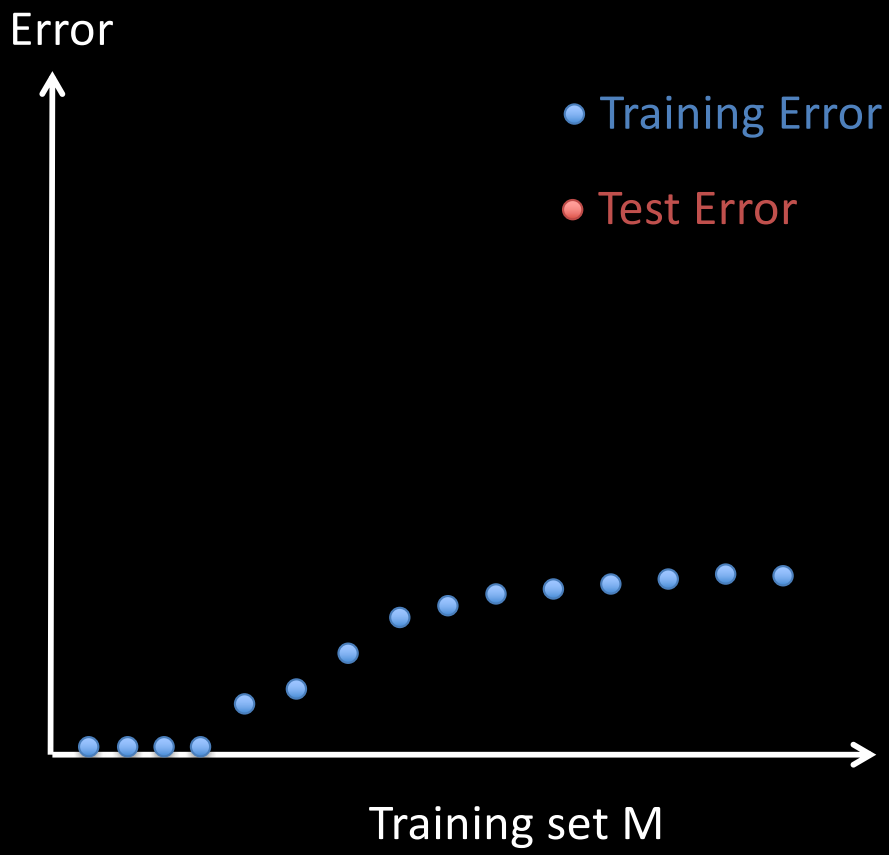


Bias and Variance

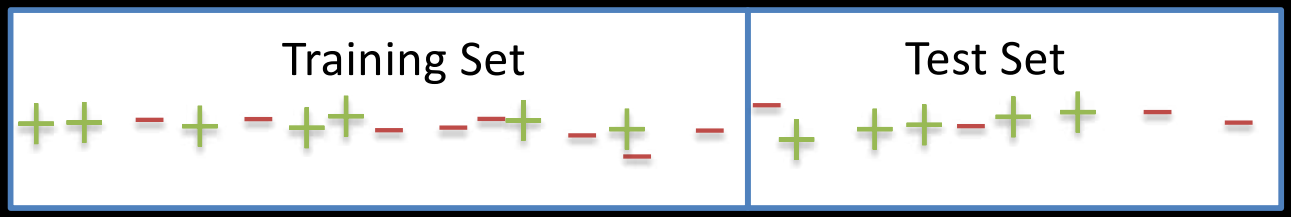




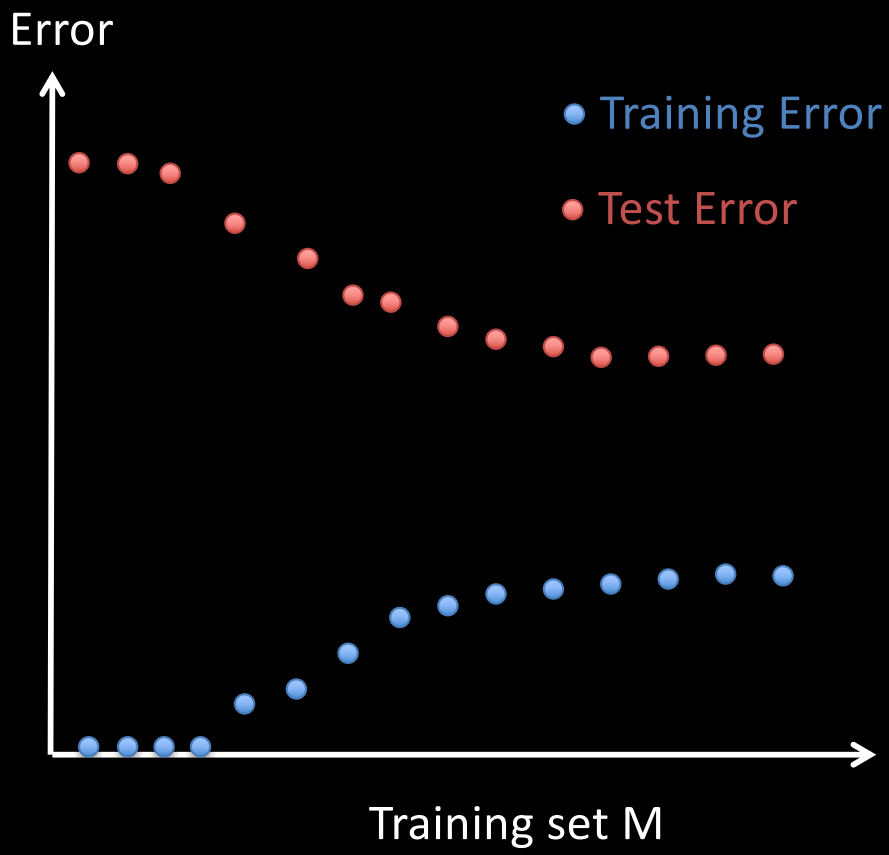
Bias and Variance

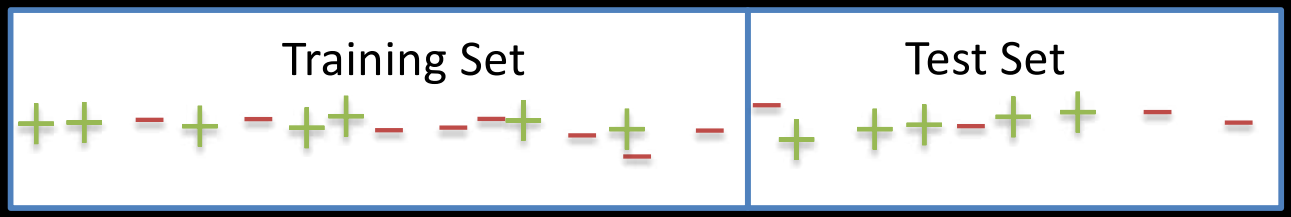


Clicker:
Test error
a) decreases with M
b) increases with M
c) stays constant

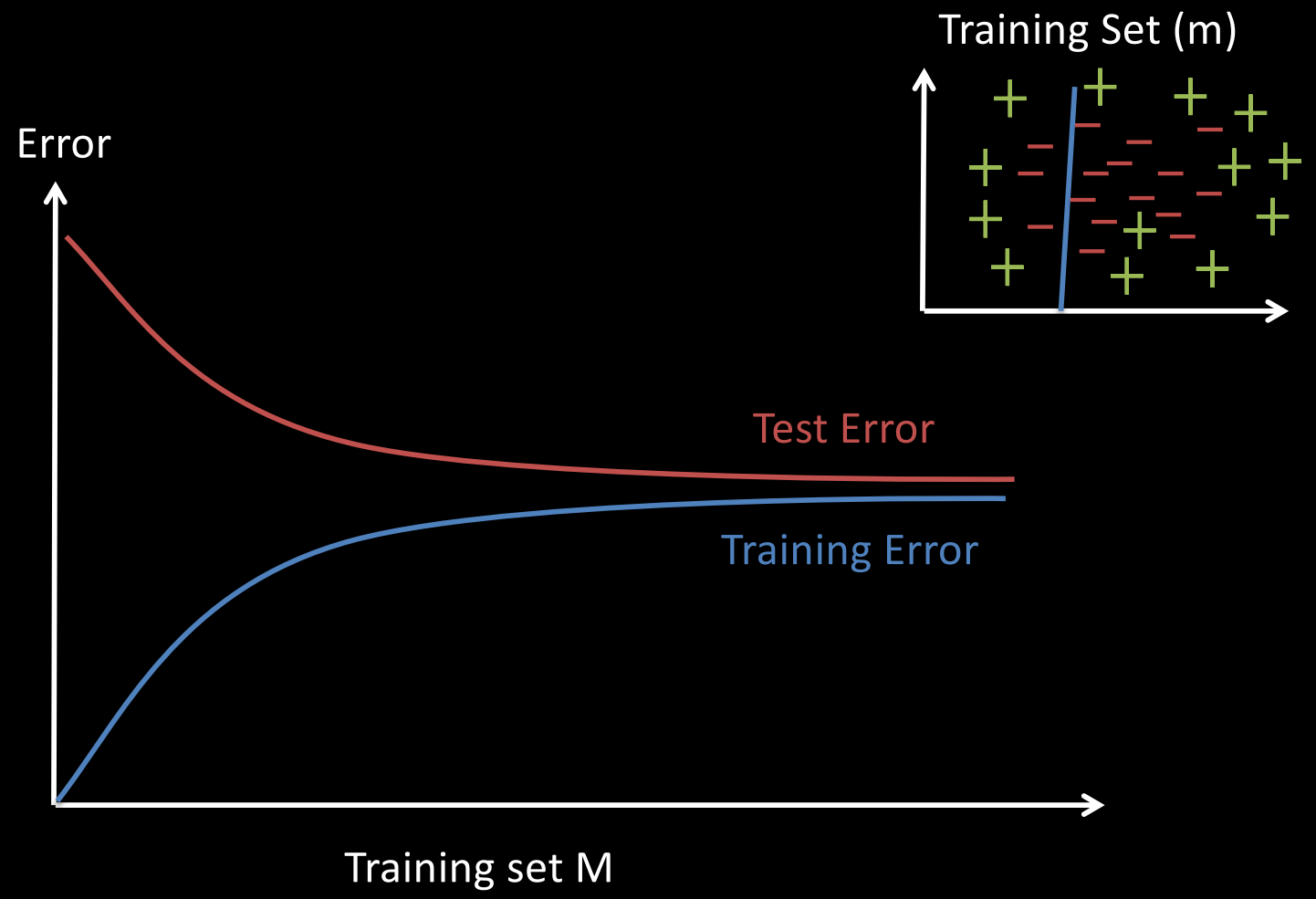


Bias and Variance

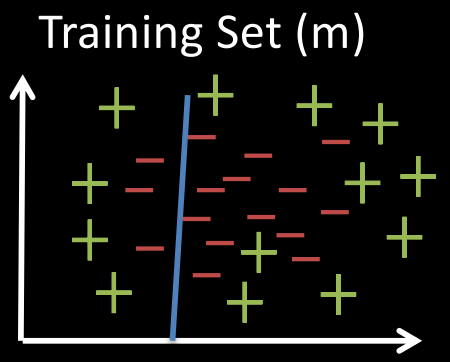




High Bias


Training Set (m)
Error
Test Error
Training Error
Training set M

High Bias

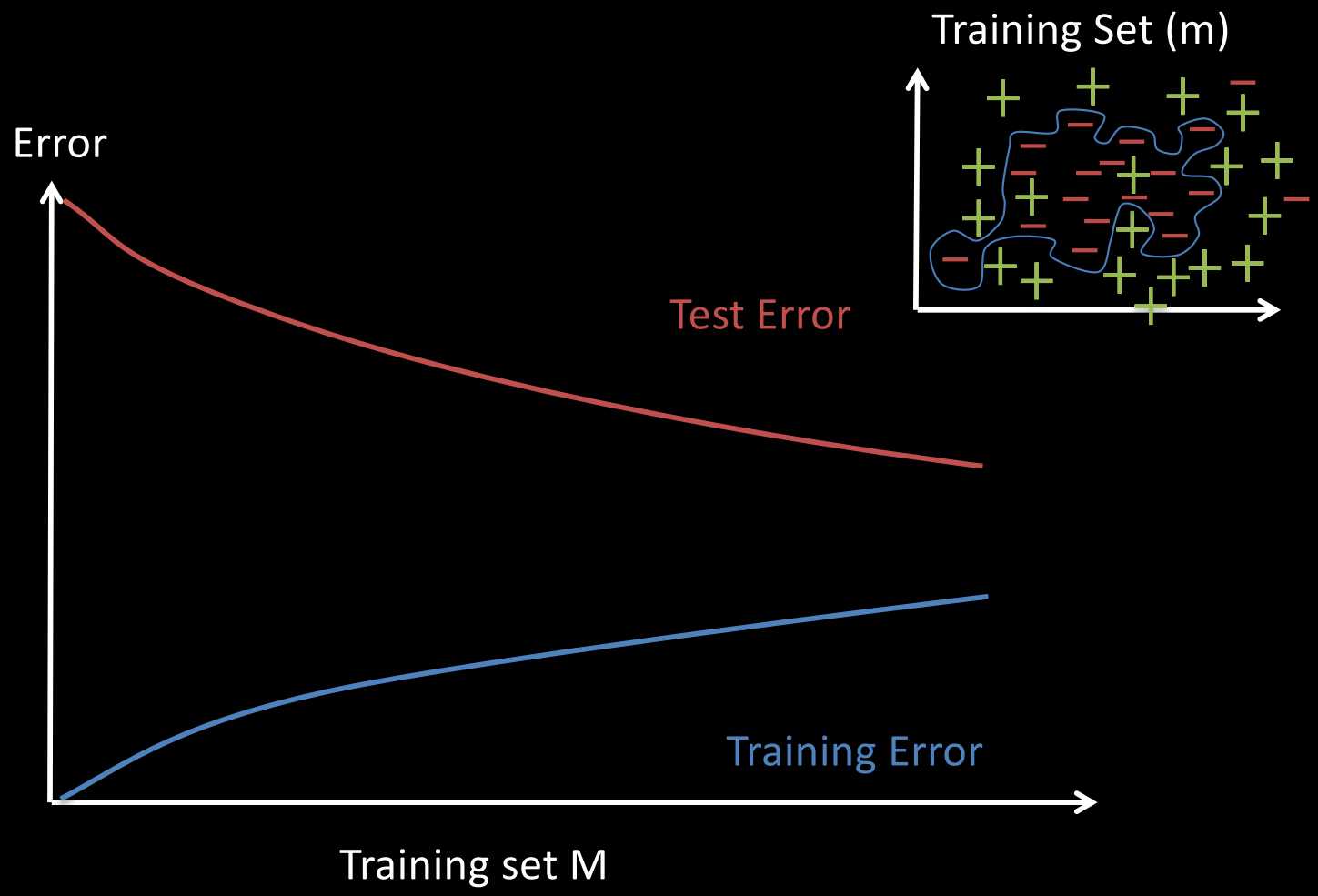


Clicker: If you have high-bias, does more data help?

a) No

b) Yes

High Variance



Clicker: If you have high-bias, does more data help?

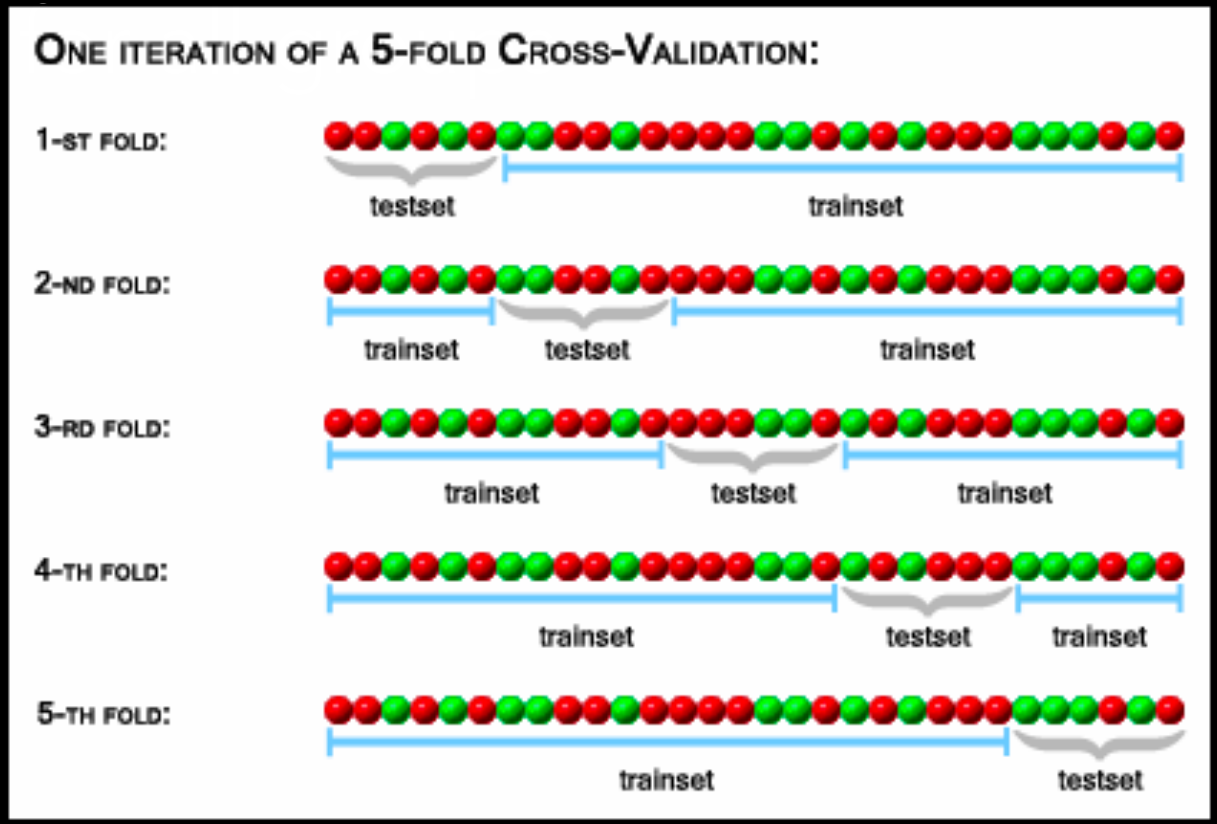
a) No

b) Yes

Cross-validation

k-fold: split the data into k groups, train on every group except for one, which you test on.

Repeat



Parameter Tuning

Grid Search