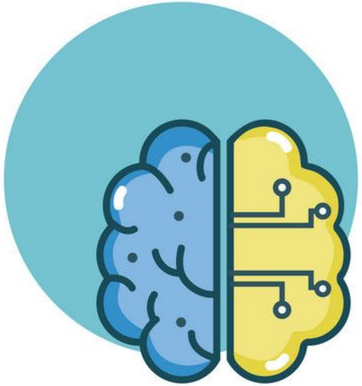
INTRODUCTION TO MACHINE LEARNING

BEYOND BINARY CLASSIFICATION

Elisa Ricci

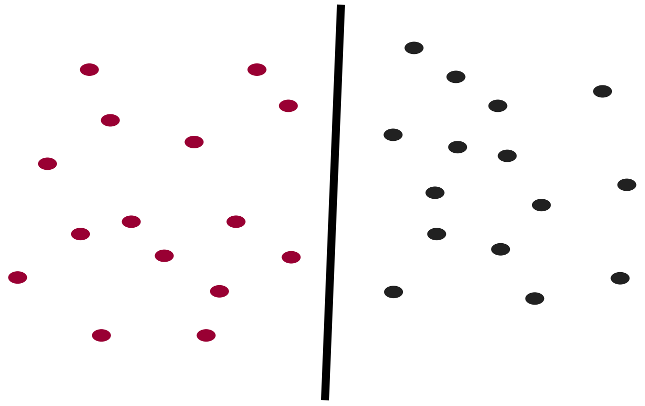


From Binary to Multiclass Classification

Linear model

A **linear model** is a model assumes that the data are linearly separable

Assume a specific hypothesis space, i.e. linear functions



Classifying with a linear model

We can classify with a linear model by checking the sign:

$f_1, f_2, \ldots, f_n$

classifier

$b + \sum_{i=1}^{n} w_i f_i > 0$ Positive example

$b + \sum_{i=1}^{n} w_i f_i < 0$ Negative example

Perceptron learning algorithm

repeat until convergence (or for some # of iterations):

for each training example (f_1, f_2, …, f_n, label):

$$prediction = b + \sum_{i=1}^{n} w_i f_i$$

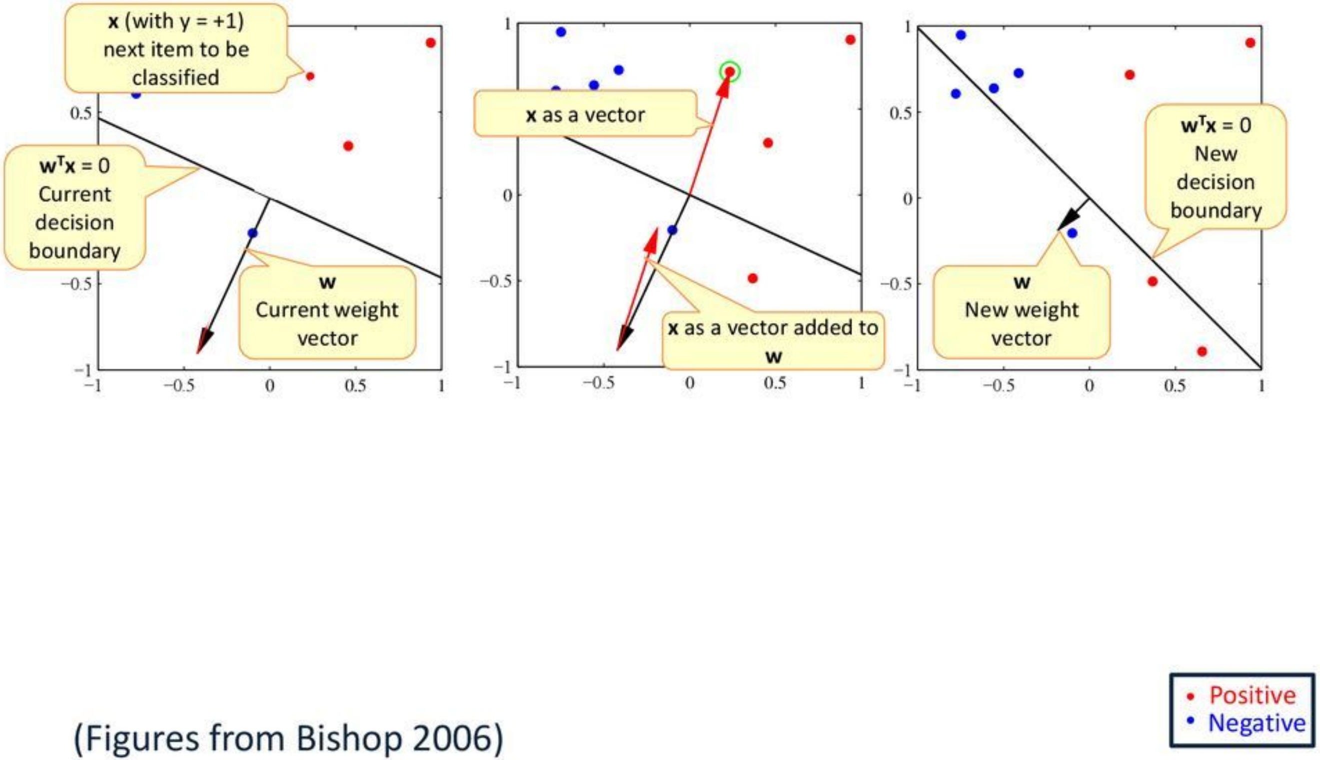
if *prediction is different from label*

for each w_i:

$w_i = w_i + f_i$*label

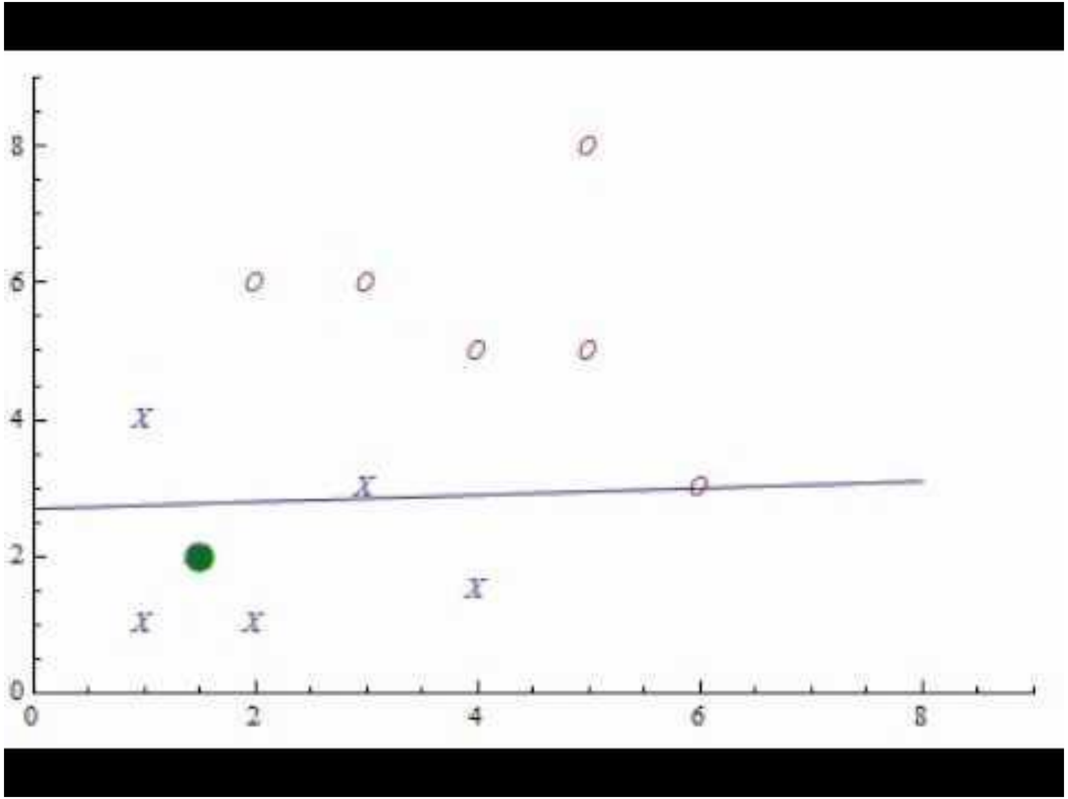
$b = b$ + label

Perceptron In Action



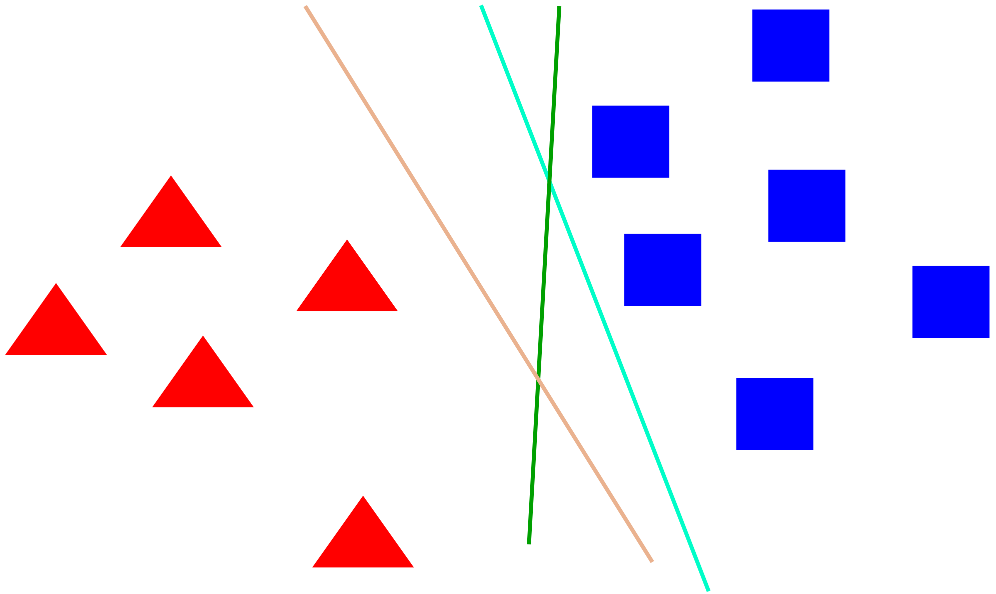
(Figures from Bishop 2006)

Perceptron In Action



https://www.youtube.com/watch?v=vGwemZhPlsA

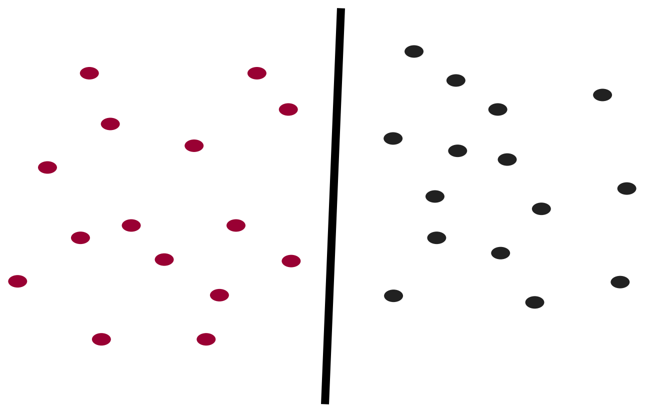
Which line will the perceptron find?



Only guaranteed to find ***some*** line that separates the data!

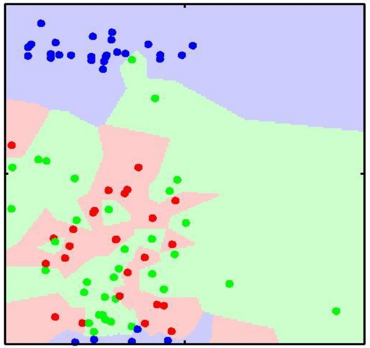
What is a Linear classifier for?

How flexible is it? Can we apply it to other problems?

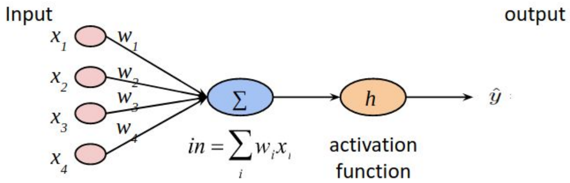


So far...

K-NN



PERCEPTRON



Binary classification

Formally...

Task: Binary Classification

Given:

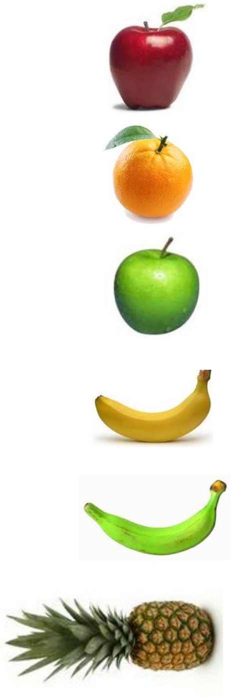
1. An input space $\mathcal{X}$
2. An unknown distribution $\mathcal{D}$ over $\mathcal{X} \times \{-1, +1\}$
3. A training set D sampled from $\mathcal{D}$

Compute: A function f minimizing: $\mathbb{E}_{(\boldsymbol{x},y)\sim\mathcal{D}}\big[f(\boldsymbol{x}) \neq y\big]$

http://ciml.info/dl/v0_99/ciml-v0_99-ch06.pdf

Multi-class classification

examples labels



apple

orange

apple

banana

banana

pineapple

Multiclass classification is a natural extension of binary classification.

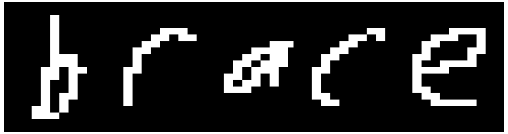
The goal is still to assign a **discrete label** to examples.

The difference is that you have $K > 2$ classes to choose from.

Real world multiclass classification

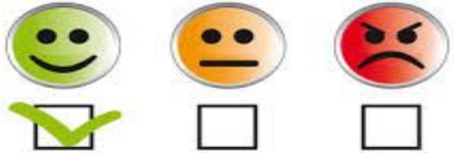
Most real-world applications involve multiclass predictions

document classification



handwriting recognition

face recognition



sentiment analysis

autonomous vehicles

emotion recognition

Multi-class classification

Formally...

TASK: MULTICLASS CLASSIFICATION

Given:

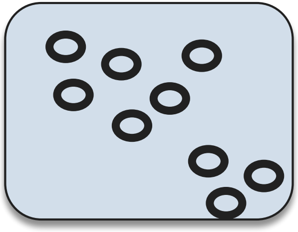
1. An input space $\mathcal{X}$ and number of classes K
2. An unknown distribution $\mathcal{D}$ over $\mathcal{X}\times[K]$
3. A training set D sampled from $\mathcal{D}$

Compute: A function f minimizing: $\mathbb{E}_{(x,y)\sim\mathcal{D}}\left[f(\boldsymbol{x}) \neq y\right]$

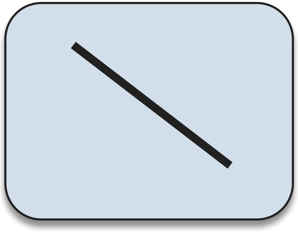
http://ciml.info/dl/v0_99/ciml-v0_99-ch06.pdf

Multiclass: current classifiers

Any of these work out of the box? With small modifications?



KNN

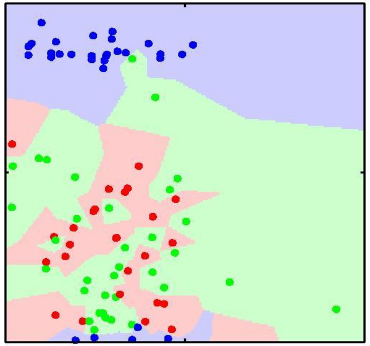


PERCEPTRON

k-Nearest Neighbor (k-NN)

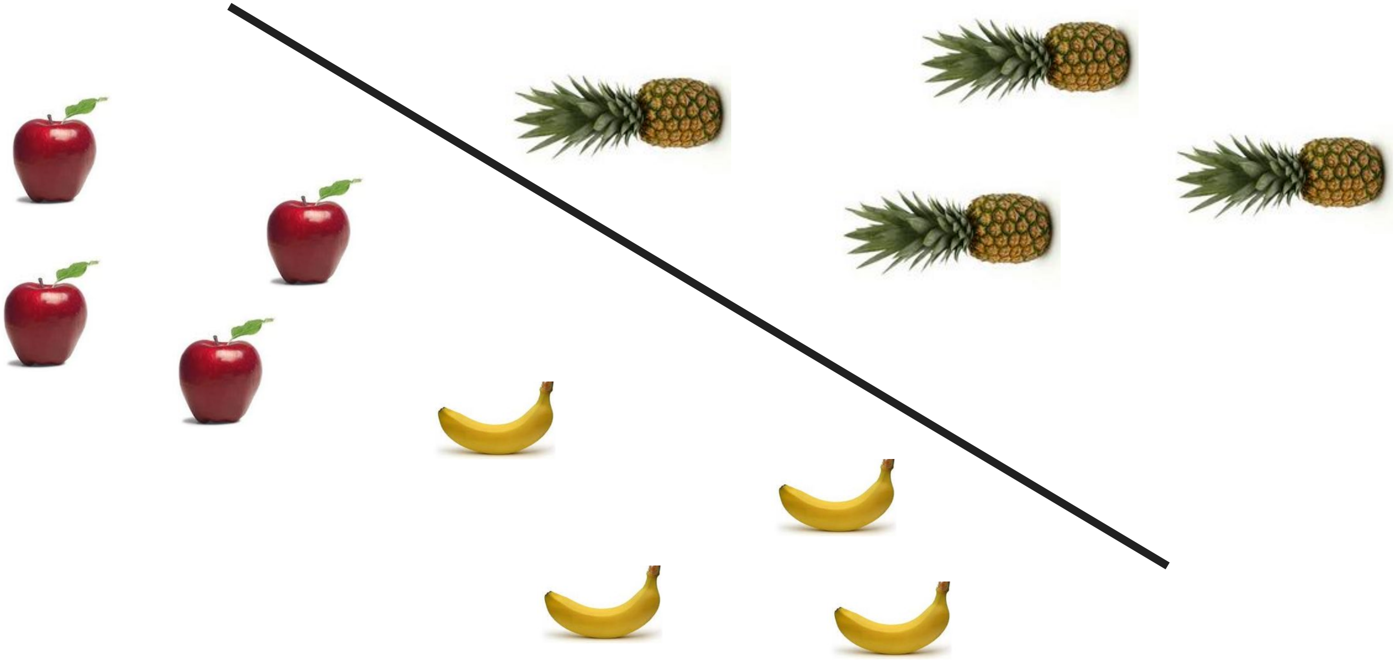
To classify an example ***d***:

- Find ***k*** nearest neighbors of ***d***
- Choose as the label the majority label within the ***k*** nearest neighbors



No algorithmic changes!

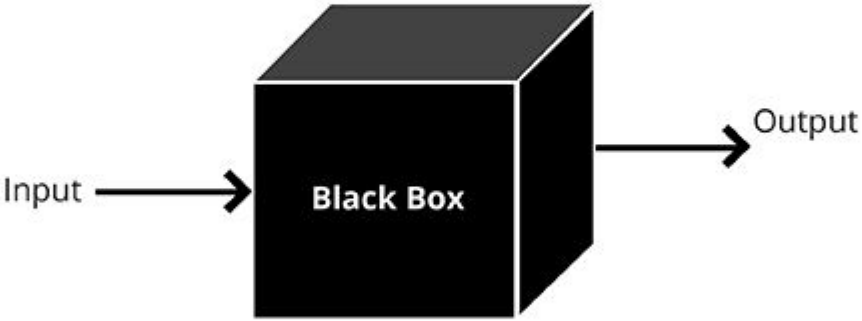
Perceptron learning



Hard to separate three classes with just one line

Black box approach to multiclass

I give you a binary classifier and you have to use it to solve the multiclass classification problem.



Black box approach to multiclass

Given a generic binary classifier, how can we use it to solve the new problem.

$$prediction = b + \sum_{i=1}^{n} w_i f_i$$

$f_1, f_2, \ldots, f_n$

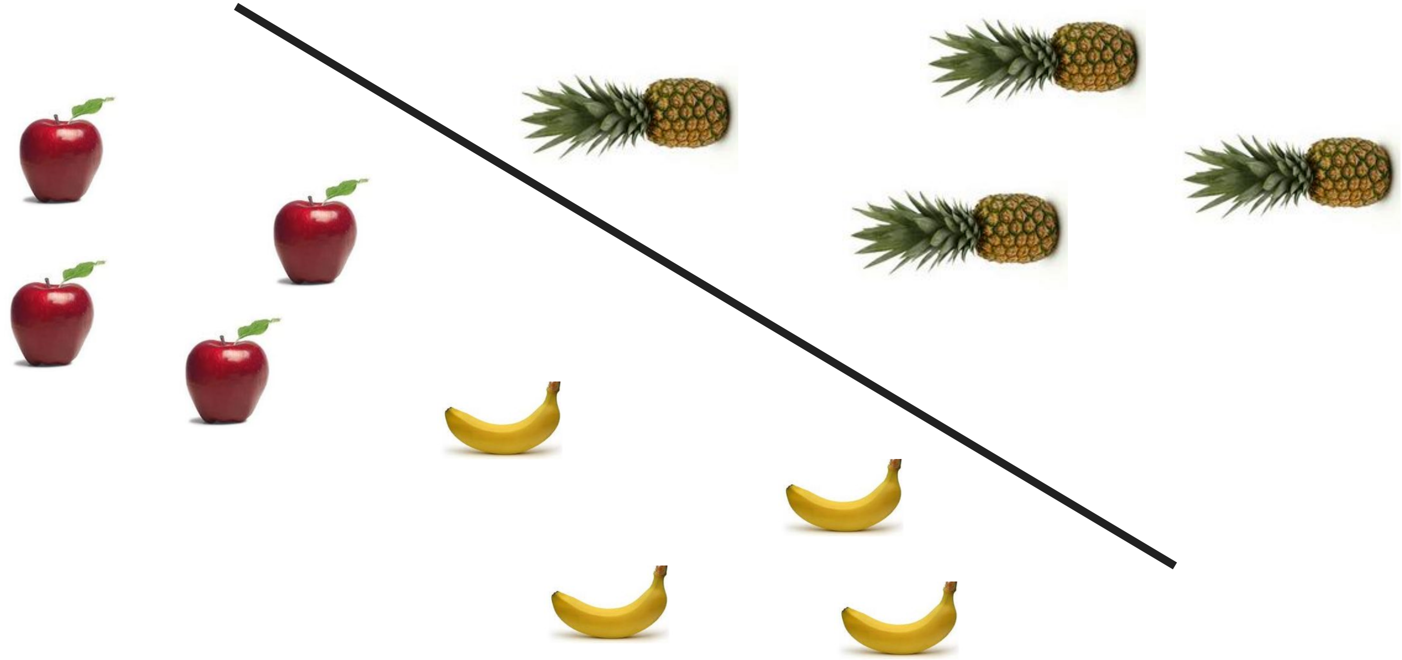
+1

-1

optionally: also output a confidence/score

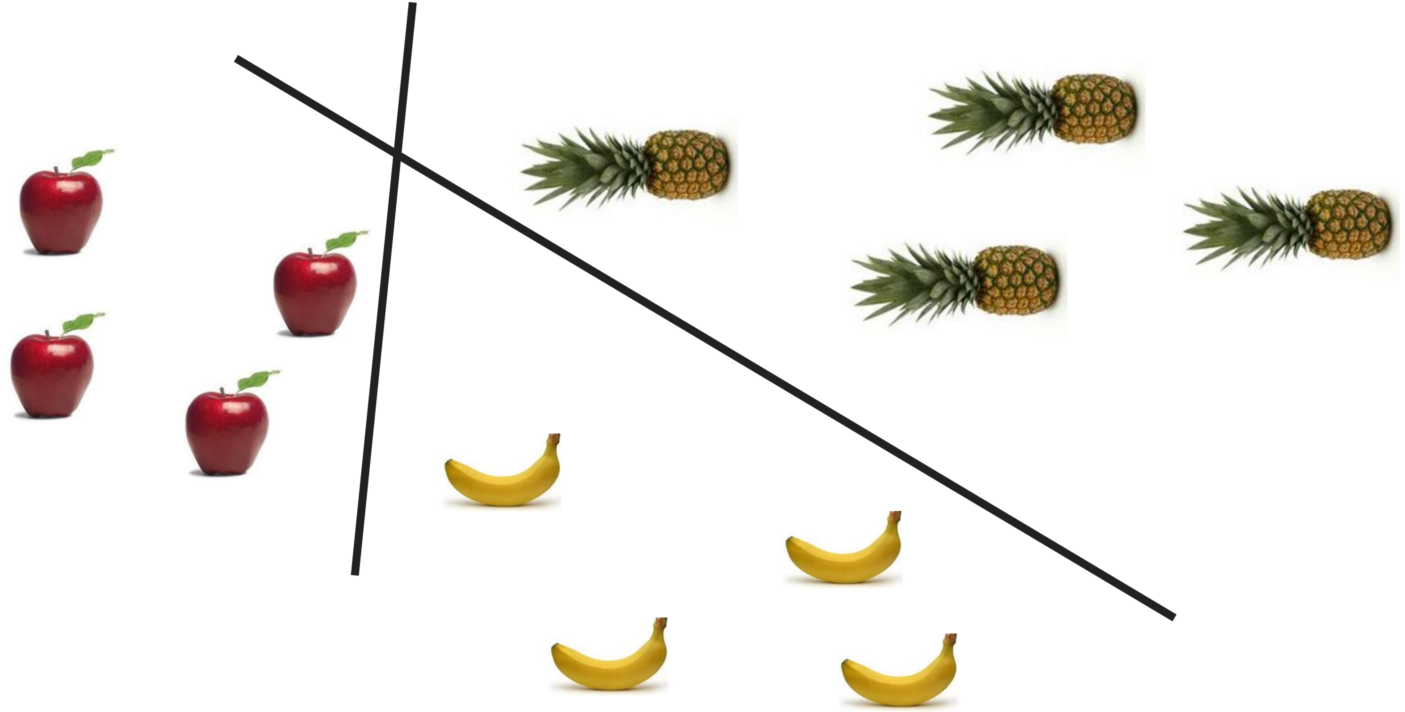
Can we solve our multiclass problem with this?

Perceptron learning

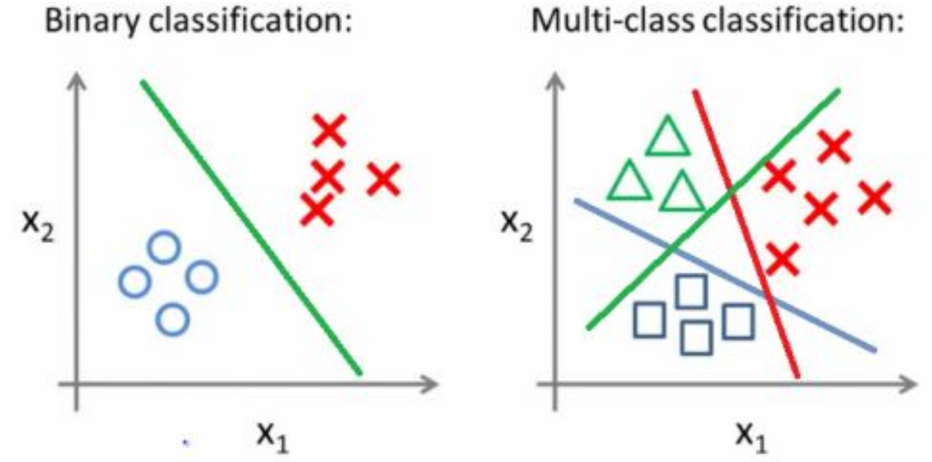


One line does not suffice but...

Perceptron learning



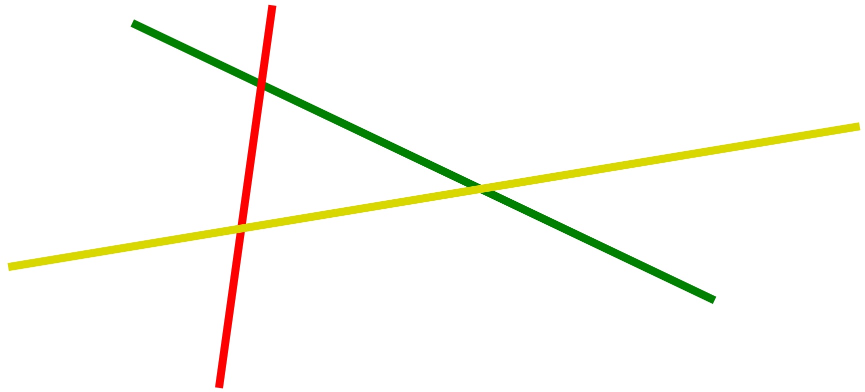
... we can combine more lines!!!



One vs All (OVA)
&
All vs All (AVA)

Approach 1: One vs all (OVA)

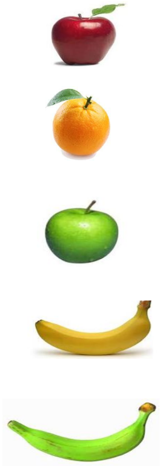
- Training: for each label *L* define a binary problem
  - all examples with label *L* are positive
  - all other examples are negative
- In practice, learn L different classification models



Approach 1: One vs. all (OVA)

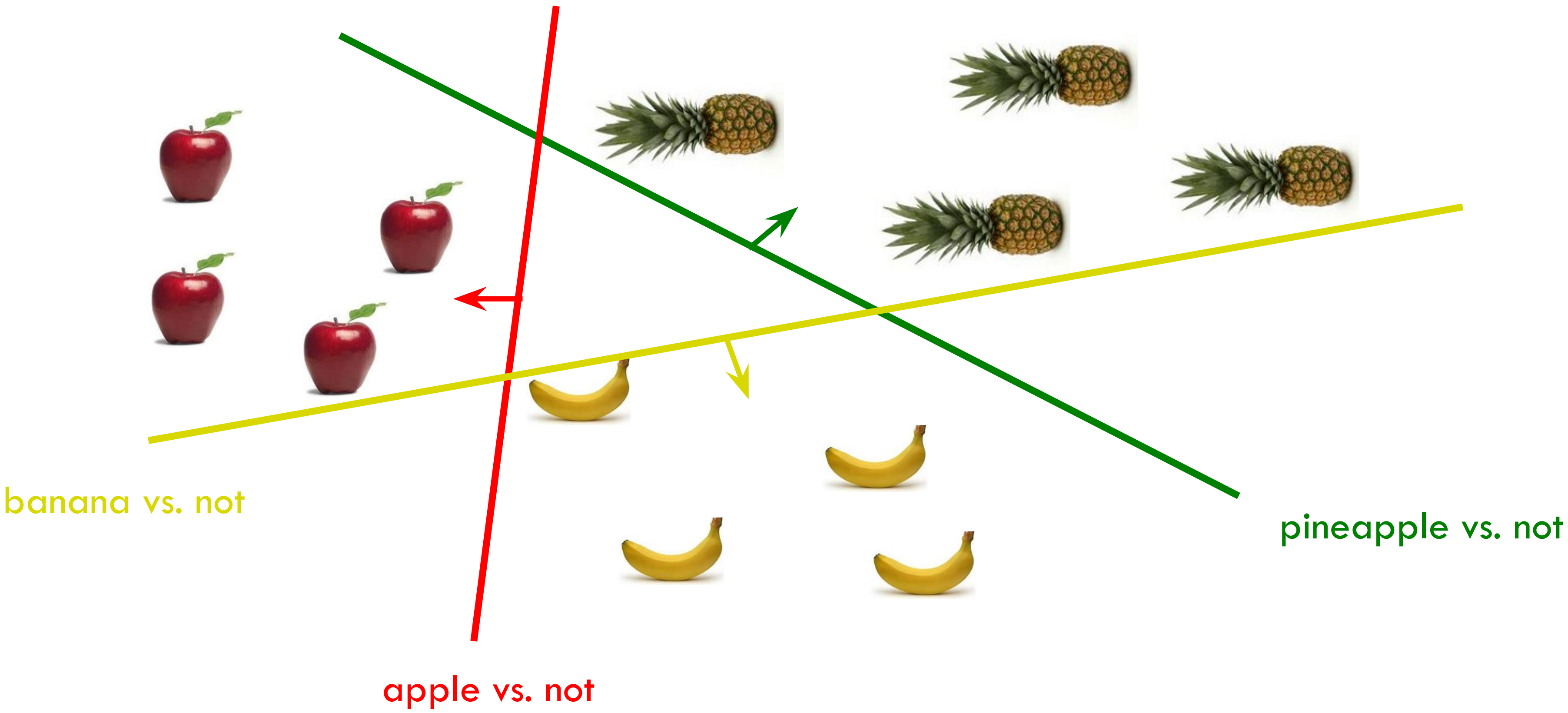
Training: for each label *L* define a binary problem

- all examples with label *L* are positive
- all other examples are negative

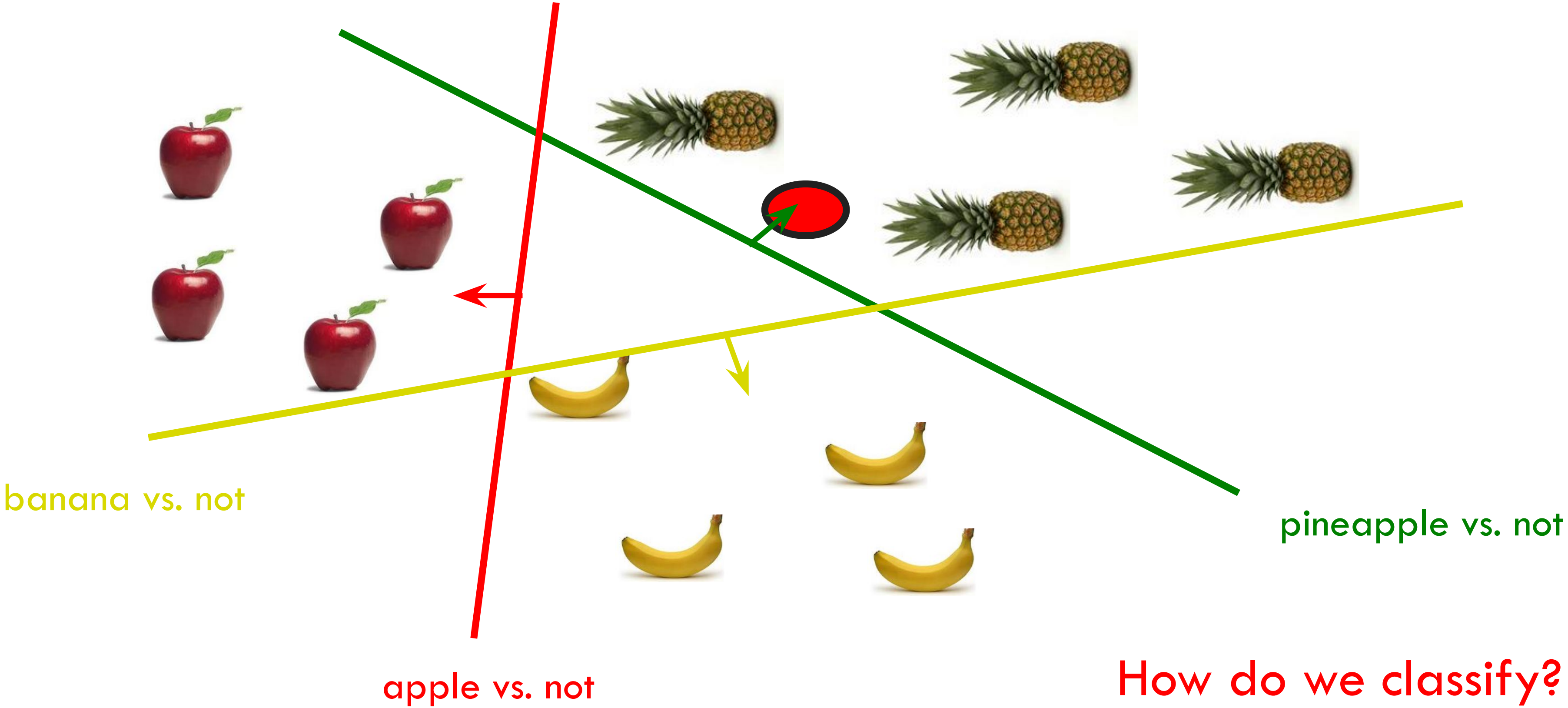


	apple vs. not	orange vs. not	banana vs. not
apple	+1	-1	-1
orange	-1	+1	-1
apple	+1	-1	-1
banana	-1	-1	+1
banana	-1	-1	+1

OVA: linear classifiers (e.g. perceptron)



OVA: LINEAR CLASSIFIERS (E.G. PERCEPTRON)



How do we classify?

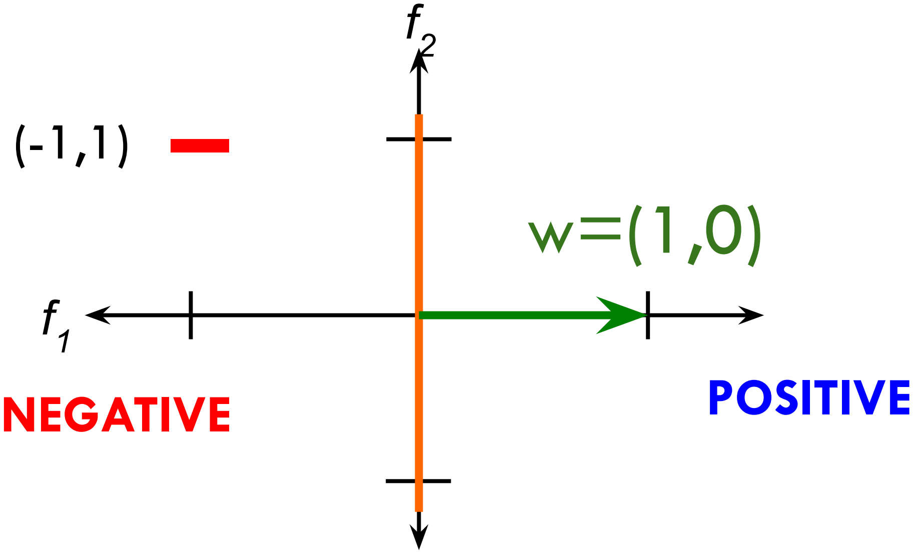
Recap: Learning a linear classifier

The classifier divide the plane in two half-planes:

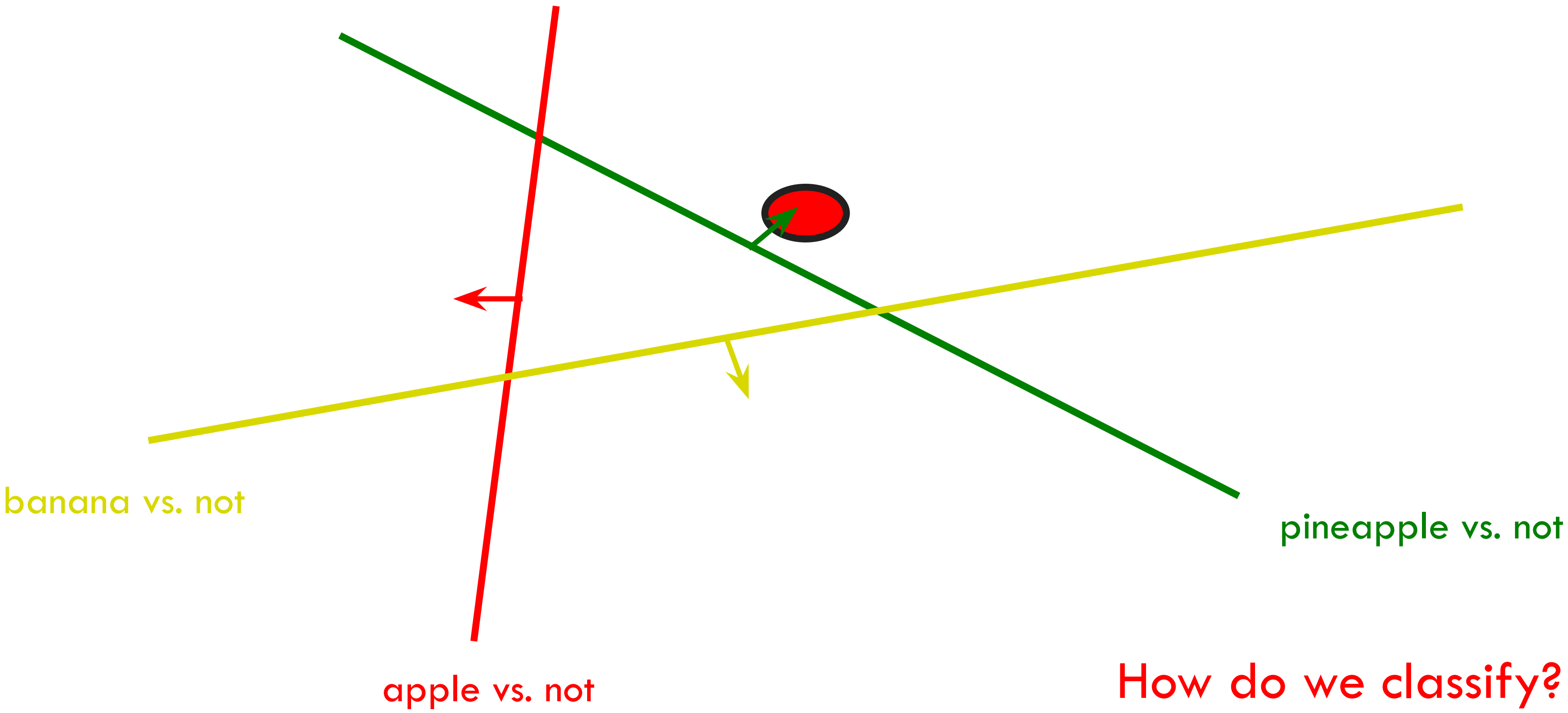
$$1 * f_1 + 0 * f_2 =$$

$$1 * -1 + 0 * 1 = -1$$

Negative!

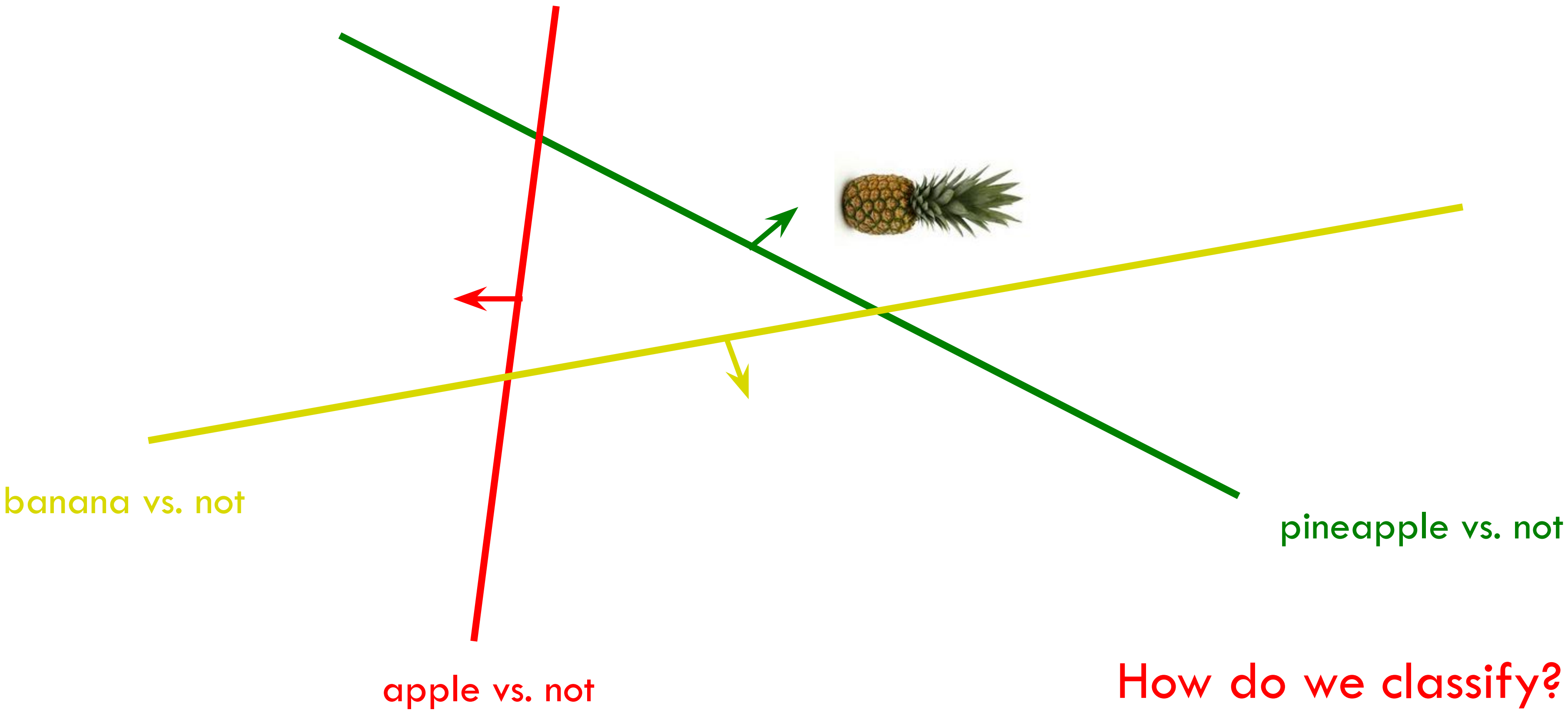


OVA: LINEAR CLASSIFIERS (E.G. PERCEPTRON)



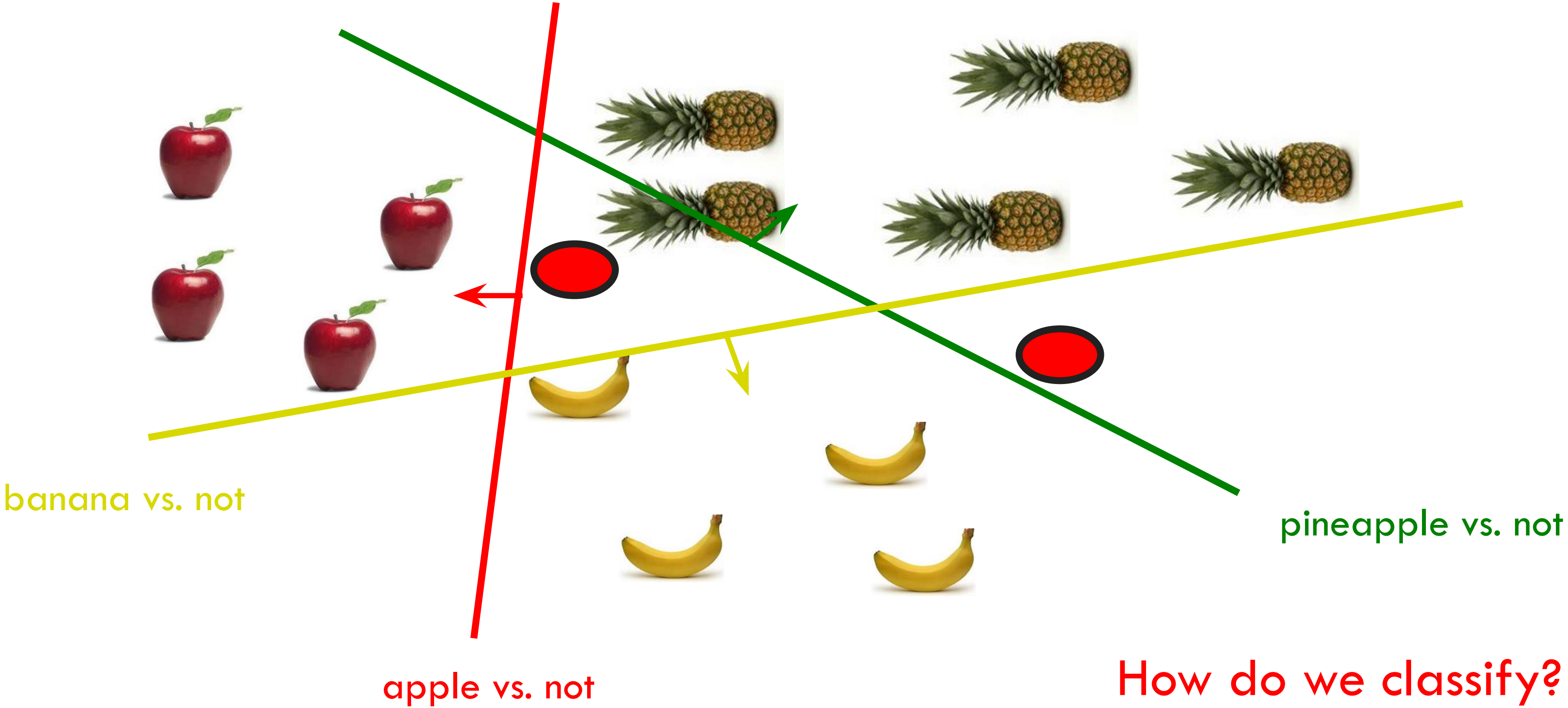
How do we classify?

OVA: linear classifiers (e.g. perceptron)



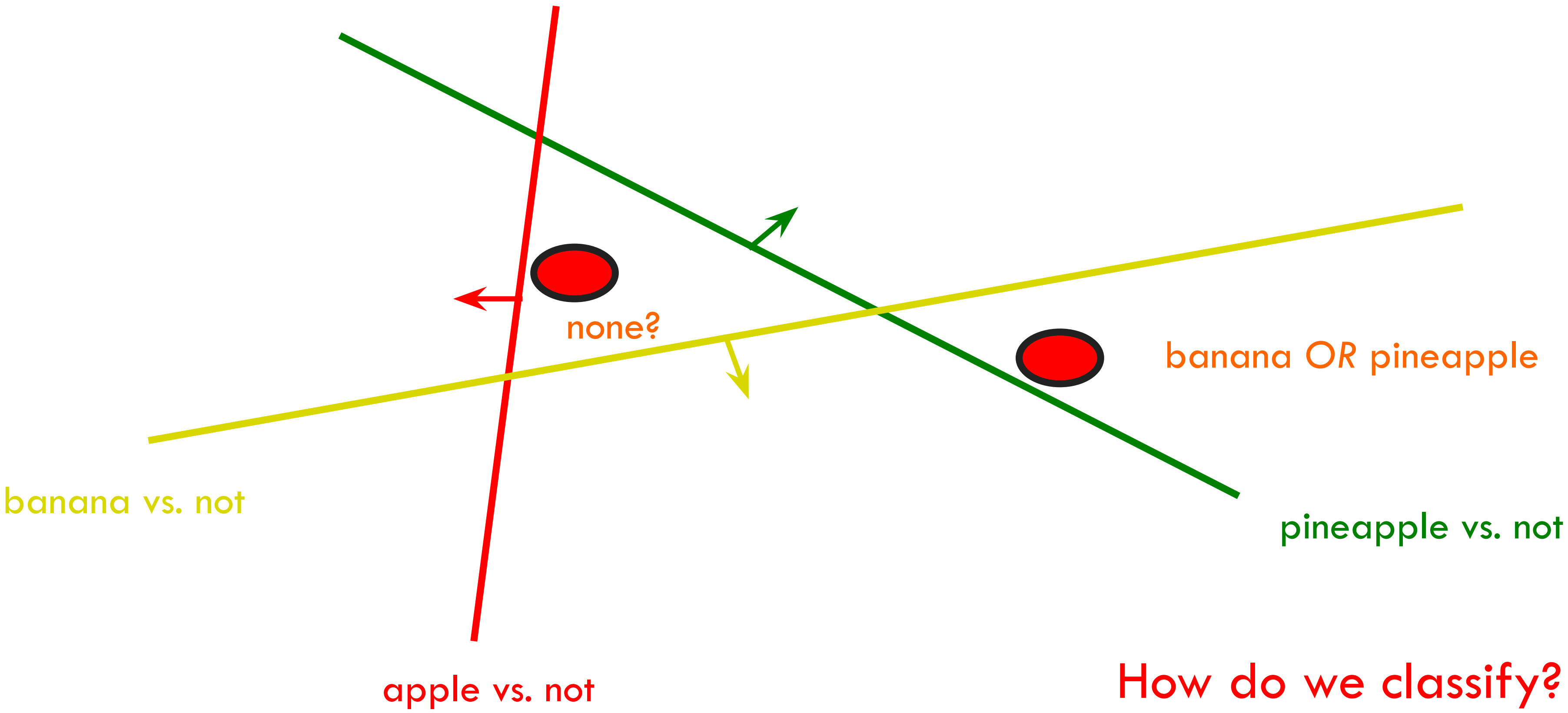
How do we classify?

OVA: LINEAR CLASSIFIERS (E.G. PERCEPTRON)



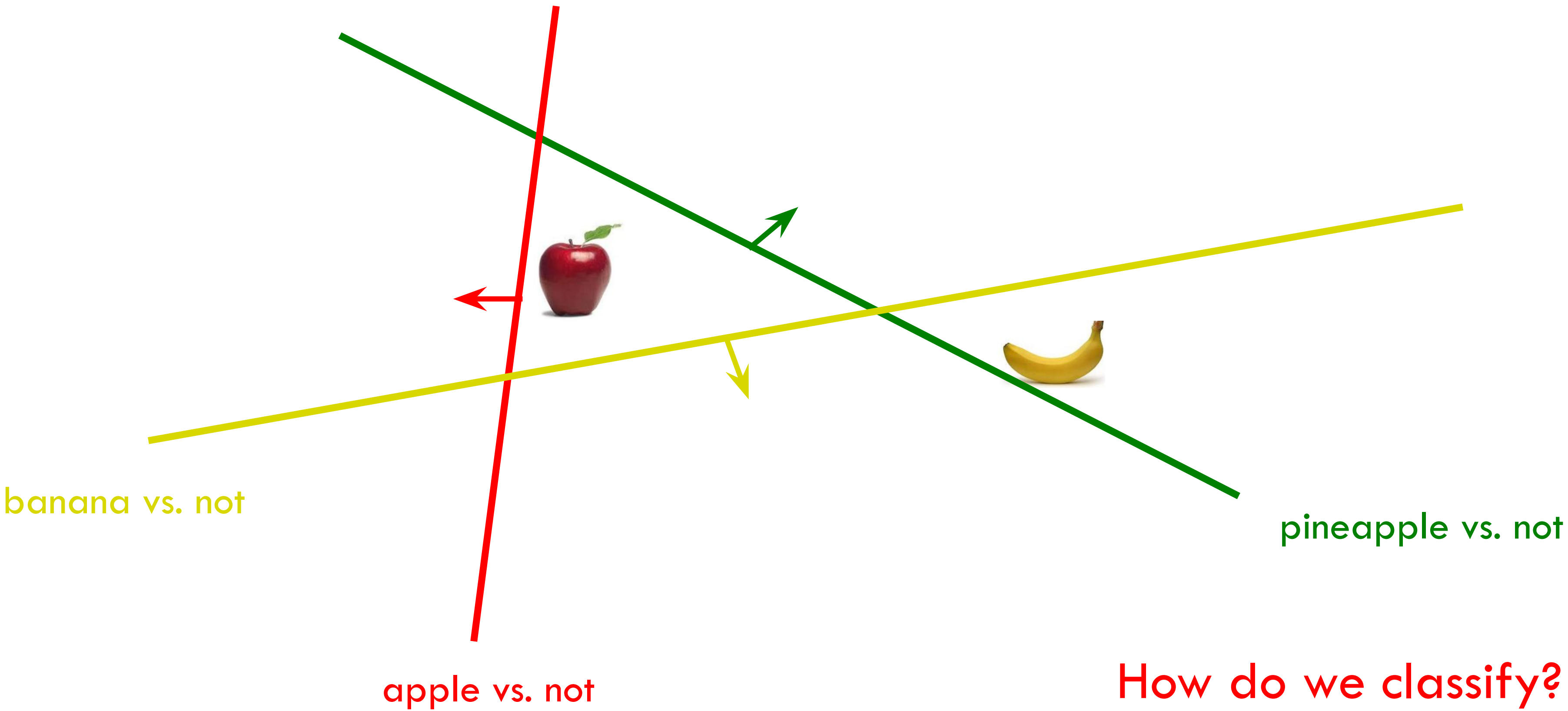
How do we classify?

OVA: linear classifiers (e.g. perceptron)



How do we classify?

OVA: LINEAR CLASSIFIERS (E.G. PERCEPTRON)



How do we classify?

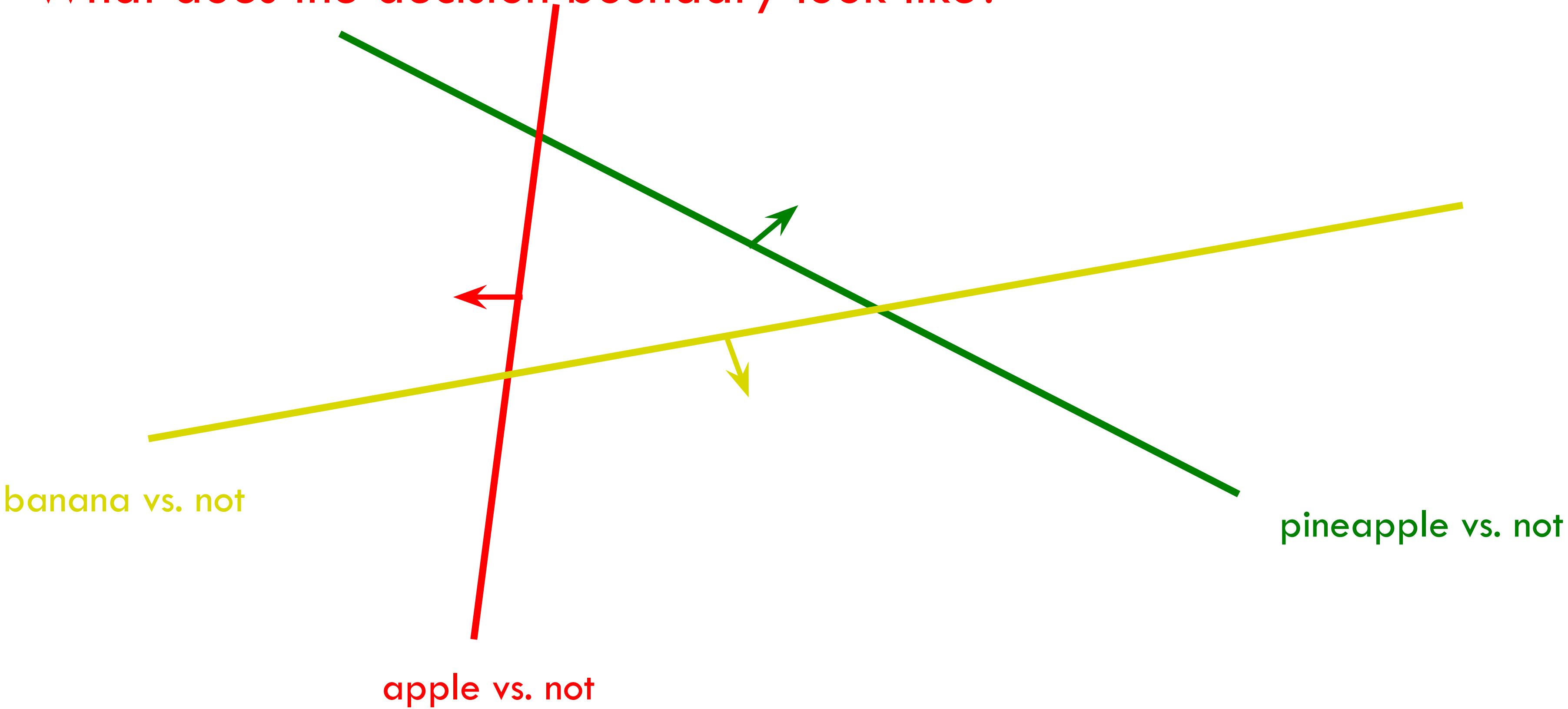
OVA: classify

How do we classify?

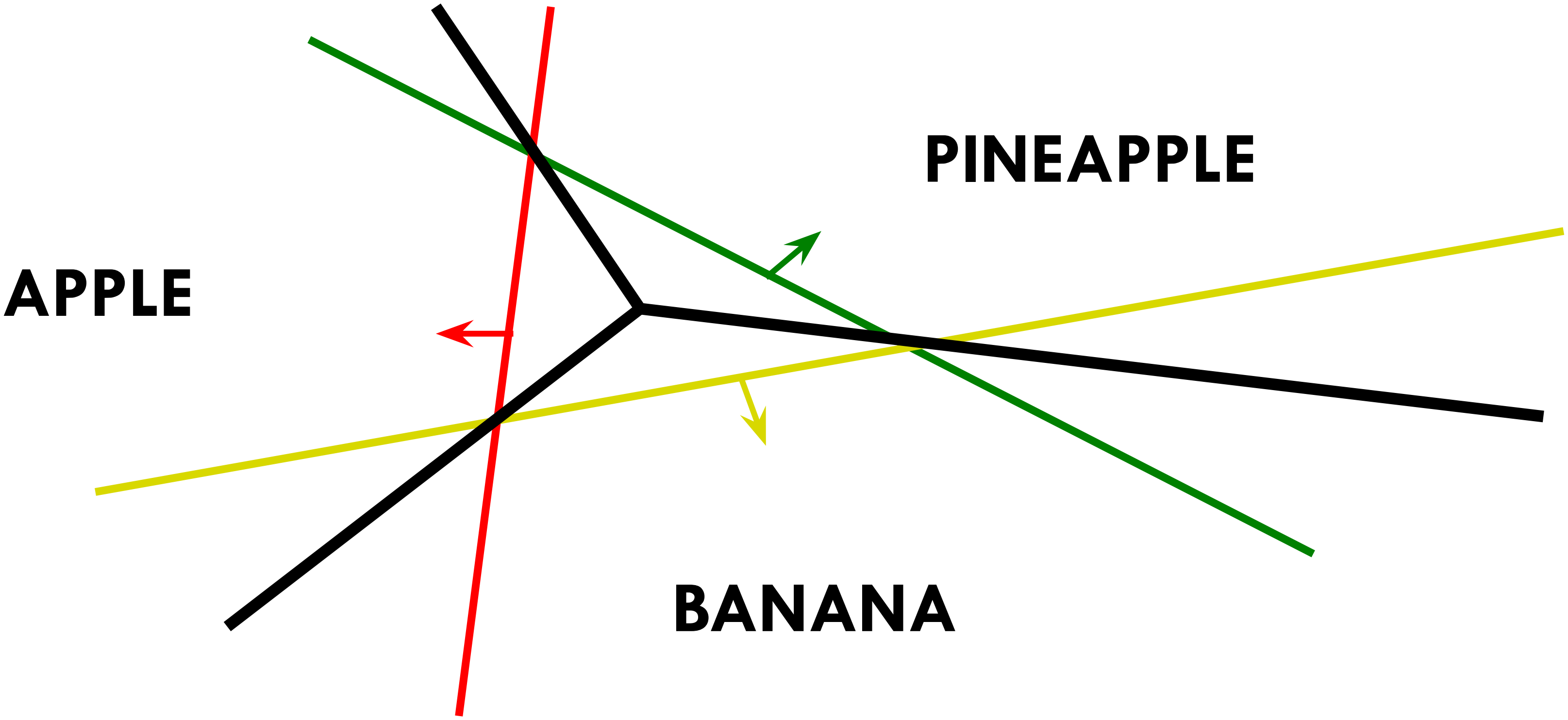
- If classifier does not provide confidence and there is ambiguity, pick one of the ones in conflict
- In general classifiers provide confidence.
- Then:
  - Pick the most confident positive
  - If none vote positive, pick *least* confident negative

OVA: linear classifiers (e.g. perceptron)

What does the decision boundary look like?



OVA: LINEAR CLASSIFIERS (E.G. PERCEPTRON)



OVA: CLASSIFY

How do we classify?

- If classifier does not provide confidence and there is ambiguity, pick one of the ones in conflict
- In general classifiers provide confidence.
- Then:
  - Pick the most confident positive
  - If none vote positive, pick *least* confident negative

How do we calculate this for the perceptron?

OVA: CLASSIFY

How do we classify?

- If classifier does not provide confidence and there is ambiguity, pick one of the ones in conflict
- In general classifiers provide confidence.
- Then:
  - Pick the most confident positive
  - If none vote positive, pick *least* confident negative

$$prediction = b + \sum_{i=1}^{n} w_i f_i$$

Distance from the hyperplane

OVA: summary

Algorithm 13 OneVersusAllTrain($\mathbf{D}^{multiclass}$, BinaryTrain)

1. **for** $i = 1$ **to** K **do**
2. $\quad \mathbf{D}^{bin} \leftarrow$ relabel $\mathbf{D}^{multiclass}$ so class i is positive and $\neg i$ is negative
3. $\quad f_i \leftarrow$ BinaryTrain($\mathbf{D}^{bin}$)
4. **end for**
5. **return** $f_1, \dots, f_K$

Algorithm 14 OneVersusAllTest($f_1, \dots, f_K, \hat{x}$)

1. $score \leftarrow \langle 0, 0, \dots, 0 \rangle$ // initialize K-many scores to zero
2. **for** $i = 1$ **to** K **do**
3. $\quad y \leftarrow f_i(\hat{x})$
4. $\quad score_i \leftarrow score_i + y$
5. **end for**
6. **return** $\operatorname{argmax}_k \, score_k$

http://ciml.info/dl/v0_99/ciml-v0_99-ch06.pdf

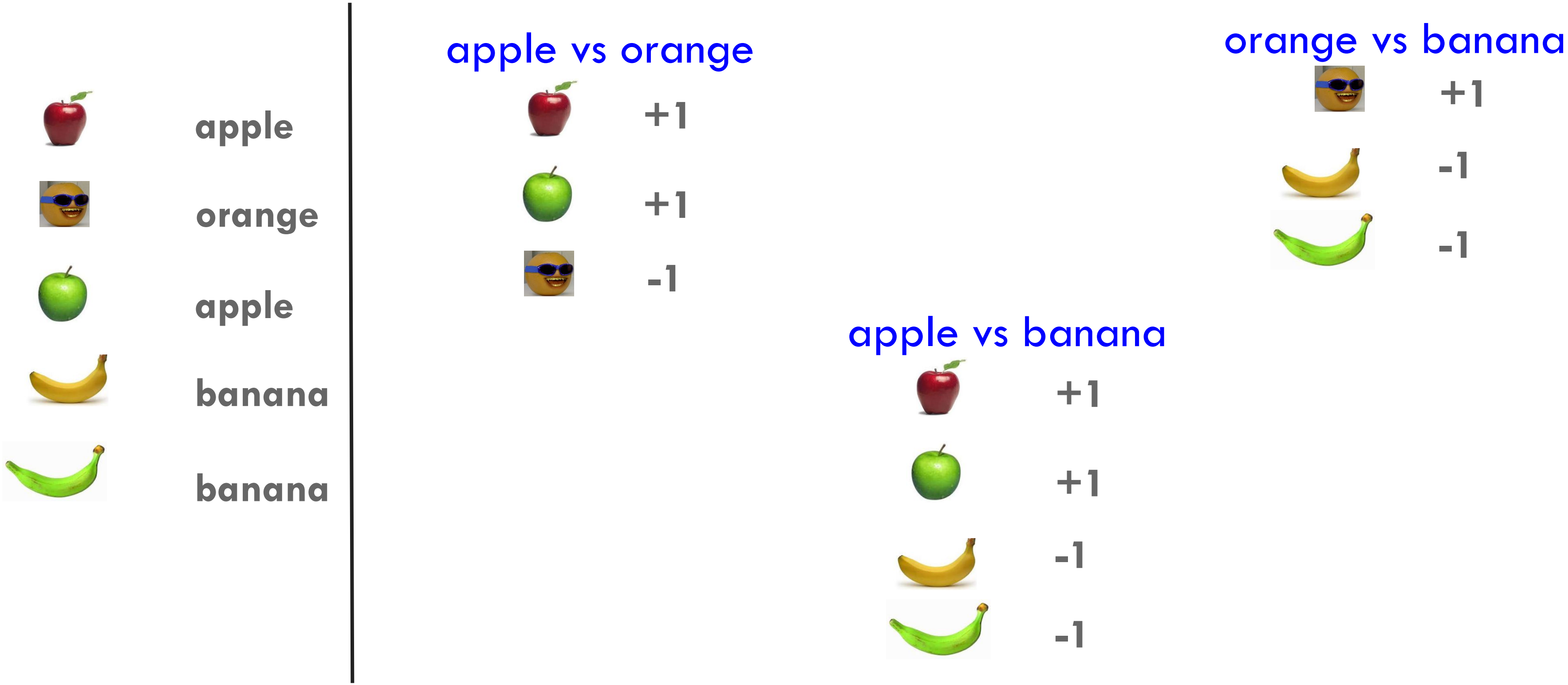
Approach 2: All vs. all (AVA)

- An alternative approach is handling the multiclass classification problem decomposing it into binary classification problems like in **sport tournaments**.
- You have K teams entering a tournament, but unfortunately the sport they are playing only allows two to compete at a time.
- You want to set up a way of pairing the teams and having them compete so that you can figure out which team is best.
- In our analogy the teams are the classes and you want to know which class is best.
- In practice, every team compete against every other team.
- The team that wins the majority of its matches is the best.

Approach 2: All vs. all (AVA)

- All versus All (or AVA) approach (sometimes called **all pairs**).
- We training $K(K-1)/2$ classifiers.
  - F_{ij}, $1 \leq i < j \leq K$, is the classifier that discriminates class i against class j.
- This classifier receives all the examples of class i as "positive" and all the examples of class j as "negative."
- When a test point arrives, we evaluate it on all the F_{ij} classifiers.
- Every time F_{ij} predicts positive, class i gets a vote; otherwise, class j gets a vote. After running all $K(K-1)/2$ classifiers, the class with the most votes wins.

Approach 2: All vs. all (AVA)



Approach 2: All vs. all (AVA)

apple vs orange

 +1

 +1

 -1

apple vs banana

 +1

 +1

 -1

 -1

orange vs banana

 +1

 -1

 -1

What class?

Approach 2: All vs. all (AVA)

apple vs orange

 +1

 +1

 -1

orange

apple vs banana

 +1

 +1

 -1

 -1

apple

orange vs banana

 +1

 -1

 -1

orange

What class?

AVA TRAINING

Training:

For each pair of labels, train a classifier to distinguish between them

for *i* = 1 to number of labels:

for *j* = i+1 to number of labels:

train a classifier F_{ij} to distinguish between *label*$_j$ and *label*$_i$:

- create a dataset with all examples *with label*$_j$ labeled positive and all examples with *label*$_i$ labeled negative

- train classifier F_{ij} on this subset of the data

AVA CLASSIFICATION

To classify example x, classify with each classifier F_{ij}

We have a few options to choose the final class:

- Take a majority vote
- Take a weighted vote based on confidence
  - $y = F_{ij}(\boldsymbol{x})$
  - $\text{score}_j \mathrel{+}= y$
  - $\text{score}_k \mathrel{-}= y$

AVA CLASSIFICATION

To classify example x, classify with each classifier F_{ij}

We have a few options to choose the final class:

- Take a majority vote
- Take a weighted vote based on confidence
  - $y = F_{ij}(\boldsymbol{x})$
  - $score_j \mathrel{+}= y$
  - $score_i \mathrel{-}= y$

If y is positive, classifier thought it was of type j:
- raise the score for j
- lower the score for i

if y is negative, classifier thought it was of type i:
- lower the score for j
- raise the score for i

AVA: summary

Algorithm 15 AllVersusAllTrain($\mathbf{D}^{multiclass}$, BinaryTrain)

$f_{ij} \leftarrow \emptyset, \forall 1 \leq i < j \leq K$
for i = 1 **to** K-1 **do**
  $\mathbf{D}^{pos} \leftarrow$ all $\boldsymbol{x} \in \mathbf{D}^{multiclass}$ labeled i
  for j = i+1 **to** K **do**
    $\mathbf{D}^{neg} \leftarrow$ all $\boldsymbol{x} \in \mathbf{D}^{multiclass}$ labeled j
    $\mathbf{D}^{bin} \leftarrow \{(\boldsymbol{x}, +1) : \boldsymbol{x} \in \mathbf{D}^{pos}\} \cup \{(\boldsymbol{x}, -1) : \boldsymbol{x} \in \mathbf{D}^{neg}\}$
    $f_{ij} \leftarrow$ BinaryTrain($\mathbf{D}^{bin}$)
  end for
end for
return all f_{ij}s

Algorithm 16 AllVersusAllTest(all f_{ij}, $\hat{\boldsymbol{x}}$)

$score \leftarrow \langle 0, 0, \dots, 0 \rangle$ // initialize K-many scores to zero
for i = 1 **to** K-1 **do**
  for j = i+1 **to** K **do**
    $y \leftarrow f_{ij}(\hat{\boldsymbol{x}})$
    $score_i \leftarrow score_i + y$
    $score_j \leftarrow score_j - y$
  end for
end for
return $\mathrm{argmax}_k\ score_k$

http://ciml.info/dl/v0_99/ciml-v0_99-ch06.pdf

OVA vs. AVA

Train/classify runtime?

Error Probability?

OVA vs. AVA

- Train time:
  - AVA learns more classifiers, however, they are trained on much smaller data this tends to make it faster if the labels are equally balanced
- Test time:
  - AVA has more classifiers, so often is slower
- Error:
  - AVA trains on more balanced data sets
  - AVA tests with more classifiers and therefore has more chances for errors

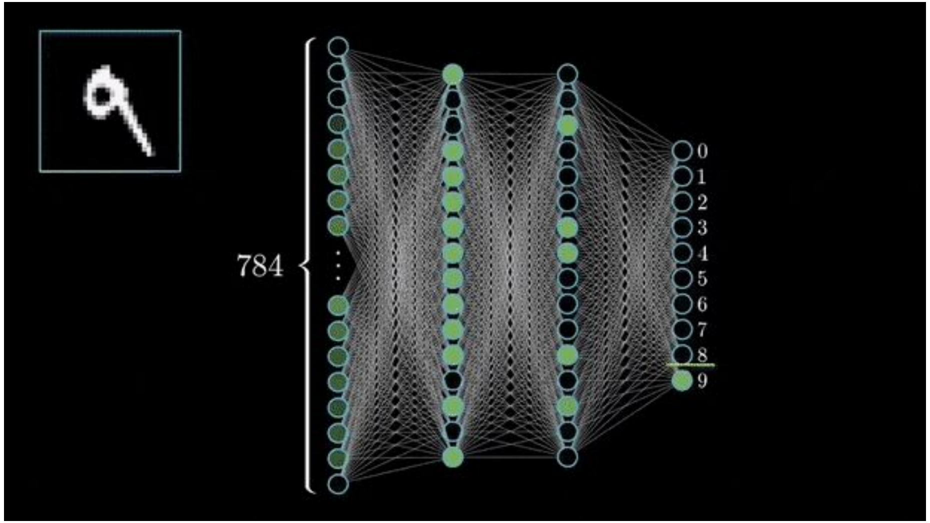
Multiclass summary

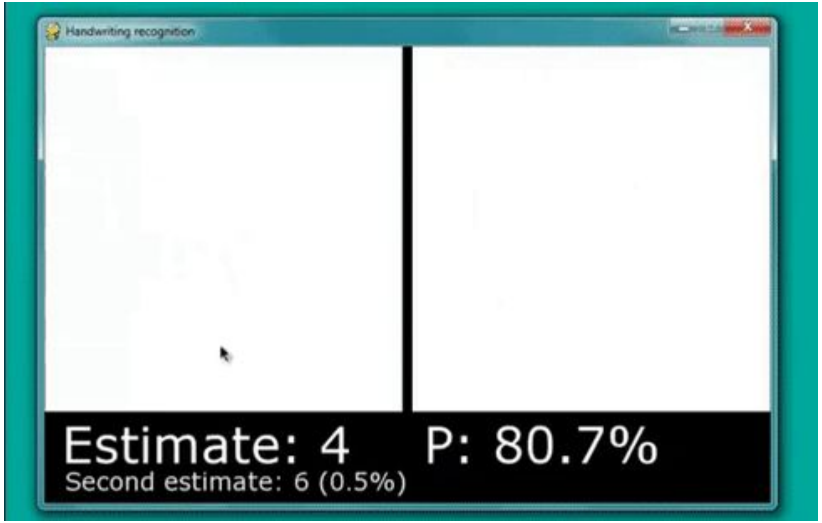
If using a binary classifier, the most common thing to do is OVA

Otherwise, use a classifier that allows for multiple labels:

- DT and k-NN work reasonably well
- Other more sophisticated methods work better (we will see them later in the course)

More in the Next Lectures





Evaluation

Class	1	2	3	4	Total
1	70	10	15	5	100
2	8	67	20	5	100
3	0	11	88	1	100
4	4	10	14	72	100

Multiclass evaluation

	label	prediction
	apple	orange
	orange	orange
	apple	apple
	banana	pineapple
	banana	banana
	pineapple	pineapple

How should we evaluate?

Multiclass evaluation

	label	prediction
	apple	orange
	orange	orange
	apple	apple
	banana	pineapple
	banana	banana
	pineapple	pineapple

How should we evaluate?

Accuracy: 4/6

Multiclass evaluation

	label	prediction
	apple	orange
	.....	
	apple	apple
	banana	pineapple
	banana	banana
	pineapple	pineapple

Problems?

Data Imbalance

Macroaveraging vs. microaveraging

Microaveraging: average over examples (this is the "normal" way of calculating)

Macroaveraging: calculate evaluation score (e.g. accuracy) for each label, then average over labels

Macroaveraging vs. microaveraging

Microaveraging: average over examples (this is the "normal" way of calculating)

Macroaveraging: calculate evaluation score (e.g. accuracy) for each label, then average over labels

Why?

- Puts more weight/emphasis on rarer labels
- Allows another dimension of analysis

Macroaveraging vs. microaveraging

	label	prediction
	apple	orange
	orange	orange
	apple	apple
	banana	pineapple
	banana	banana
	pineapple	pineapple

microaveraging: average over examples

macroaveraging: calculate evaluation score (e.g. accuracy) for each label, then average over labels

Macroaveraging vs. microaveraging

	label	prediction
	apple	orange
	orange	orange
	apple	apple
	banana	pineapple
	banana	banana
	pineapple	pineapple

microaveraging: 4/6

macroaveraging:

apple = 1/2

orange = 1/1

banana = 1/2

pineapple = 1/1

total = (1/2 + 1 + 1/2 + 1)/4

= 3/4

Confusion matrix

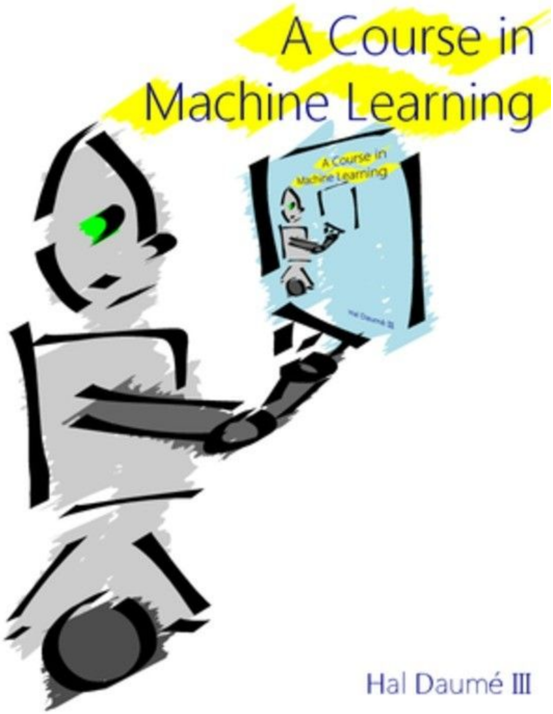
- Entry *(i, j)* represents the number of examples with label *i* that were predicted to have label *j*
- Often in percentage

	box	clap	wave	jog	Run	Walk
Box	100	0	0	0	0	0
Clap	0	94	6	0	0	0
Wave	0	1	99	0	0	0
Jog	0	0	0	91	7	2
Run	0	0	0	10	89	1
Walk	0	0	0	0	6	94

https://github.com/vkhoi/KTH-Action-Recognition

Useful Readings

Chapter 6



http://ciml.info/dl/v0_99/ciml-v0_99-ch06.pdf

Some slides are taken from David Kauchak

QUESTIONS?

Some slides are taken from David Kauchak