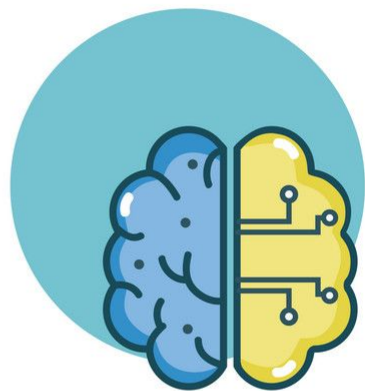


INTRODUCTION TO MACHINE LEARNING

BEYOND BINARY CLASSIFICATION



Elisa Ricci





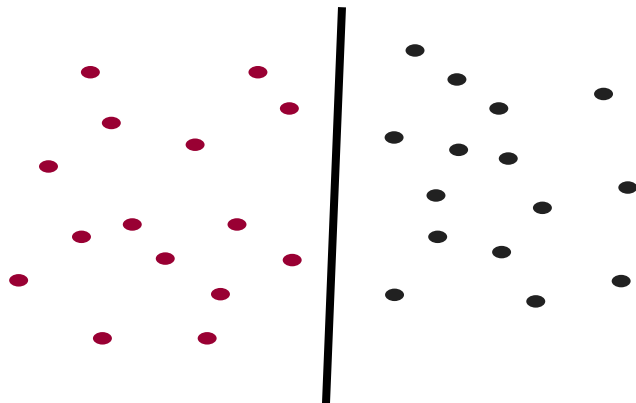
FROM BINARY TO MULTICLASS CLASSIFICATION

LINEAR MODEL

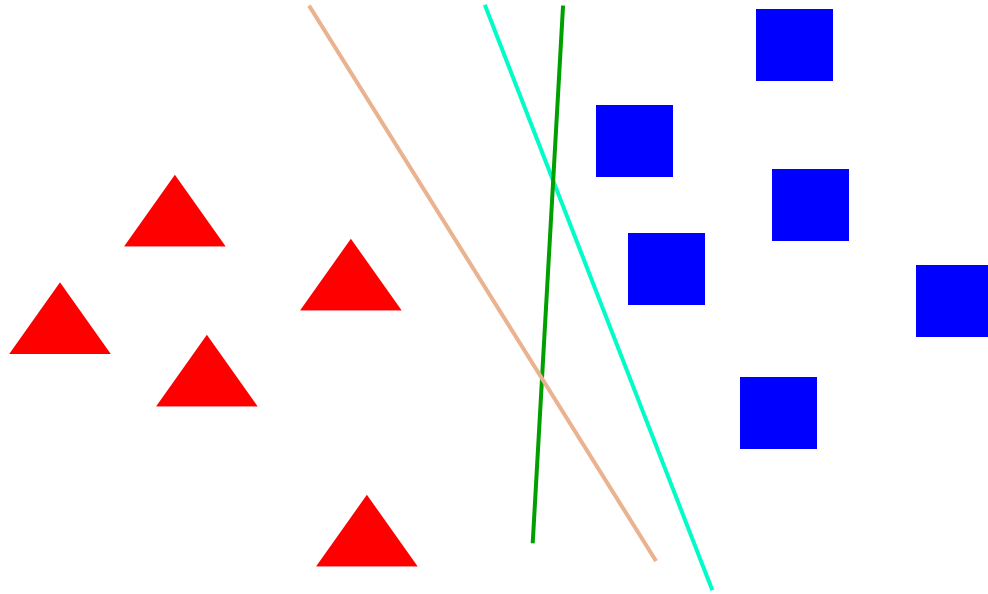
A **linear model** is a model assumes that the data are linearly separable

in questo caso restringo lo spazio delle ipotesi ad una linea retta

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



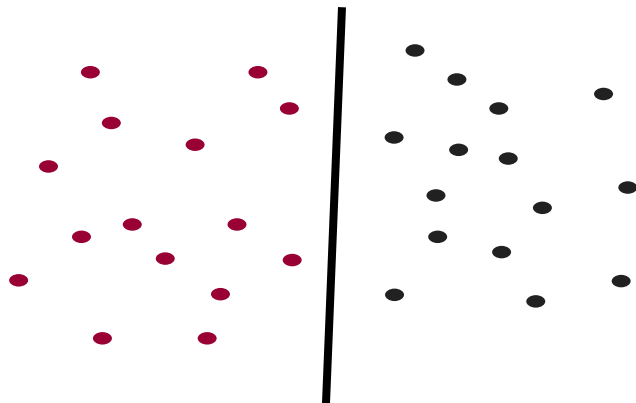
WHICH LINE WILL THE PERCEPTRON FIND?



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

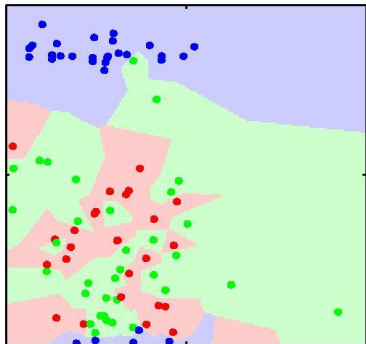
WHAT IS A LINEAR CLASSIFIER FOR?

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

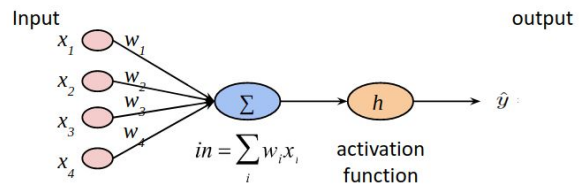


SO FAR...

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}

input space

output space

cartesian product

data generating distribution

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

indicator function che conta quante volte
la predicon differisce dal label

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

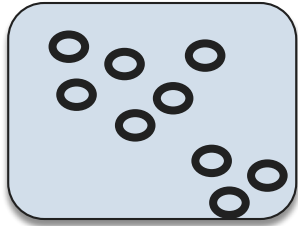
Given:

1. An input space \mathcal{X} and number of classes K set of #k labels
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: $\mathbb{E}_{(x,y) \sim \mathcal{D}} [f(x) \neq y]$

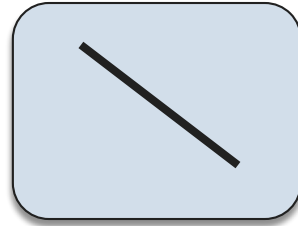
MULTICLASS: CURRENT CLASSIFIERS

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



KNN

funziona out of the box



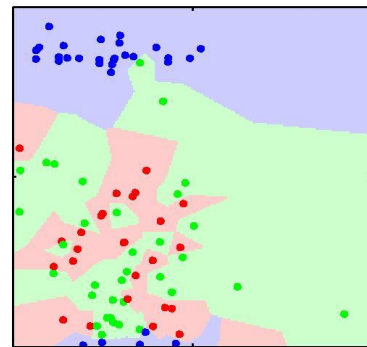
PERCEPTRON

con modifiche

K-NEAREST NEIGHBOR (K-NN)

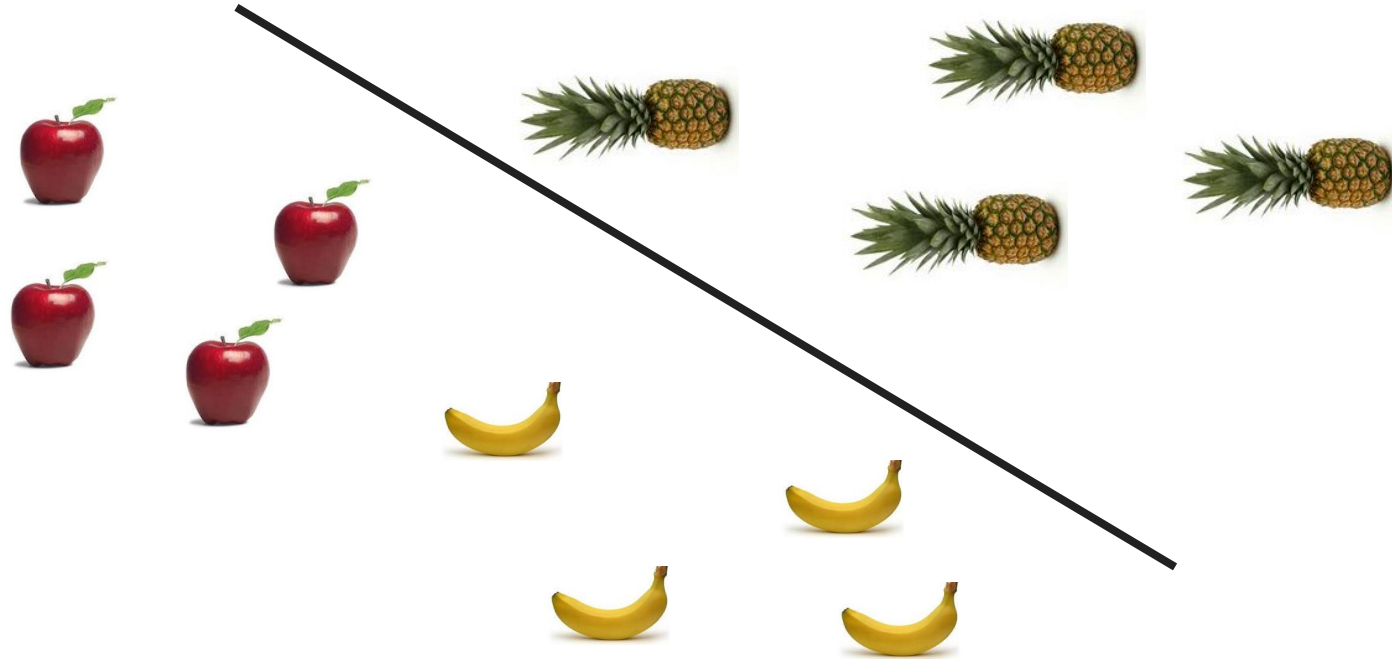
To classify an example d :

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



No algorithmic changes!

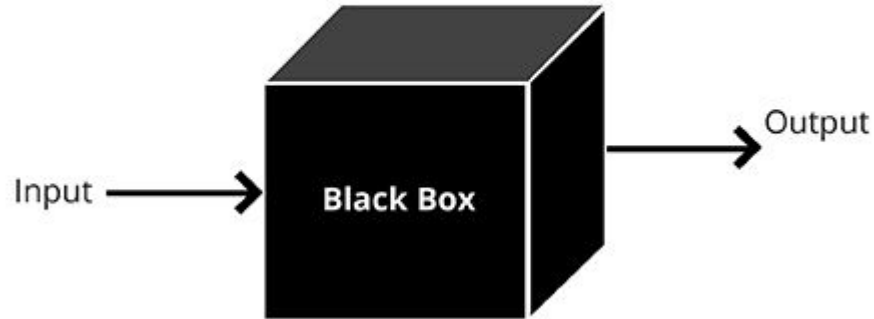
PERCEPTRON LEARNING



Hard to separate three classes with just one line

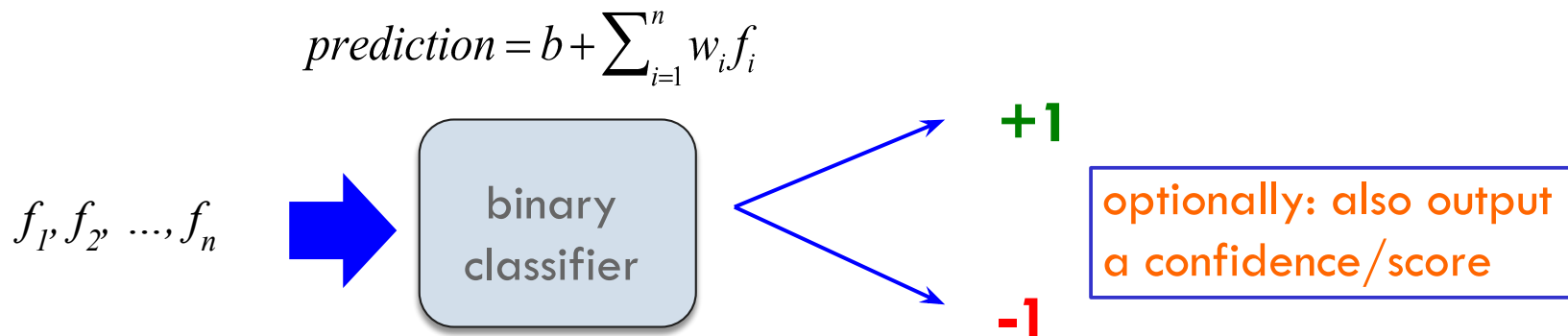
BLACK BOX APPROACH TO MULTICLASS

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



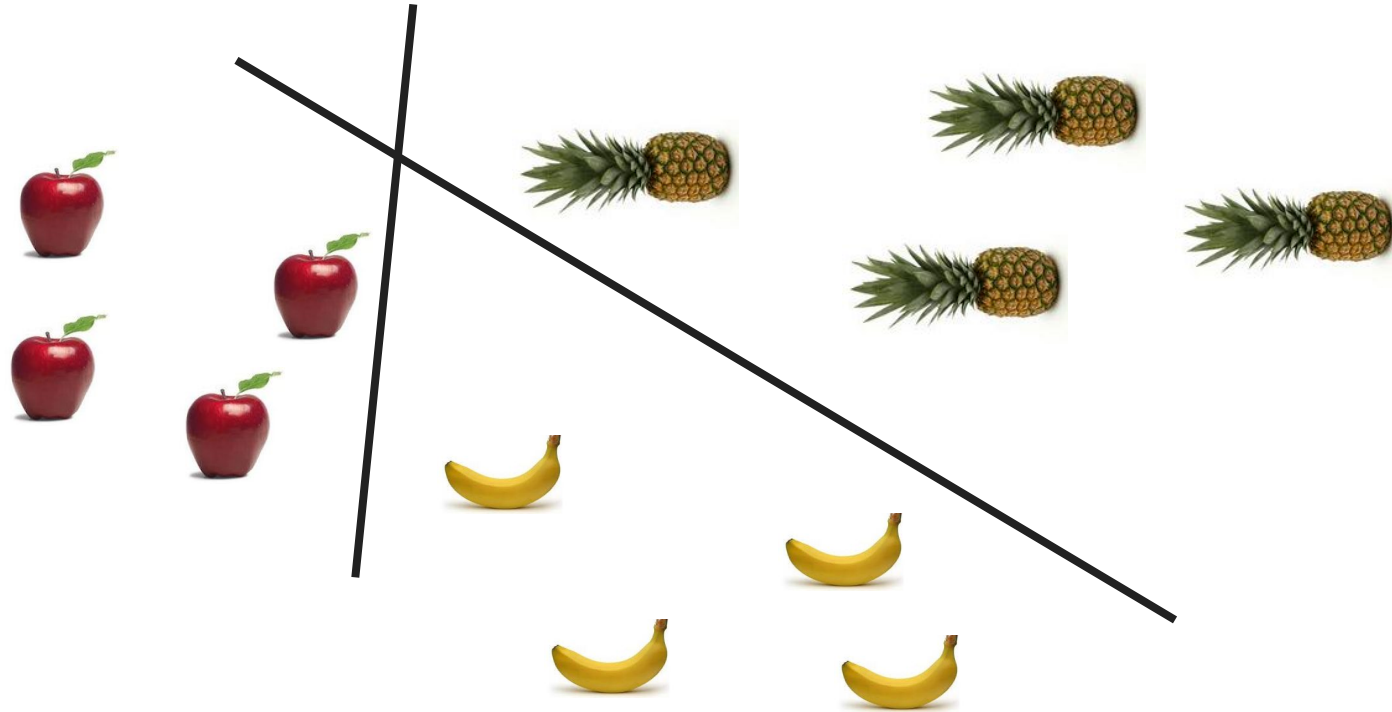
BLACK BOX APPROACH TO MULTICLASS

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



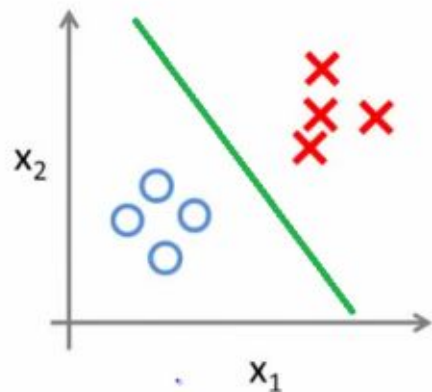
Can we solve our multiclass problem with this?

PERCEPTRON LEARNING

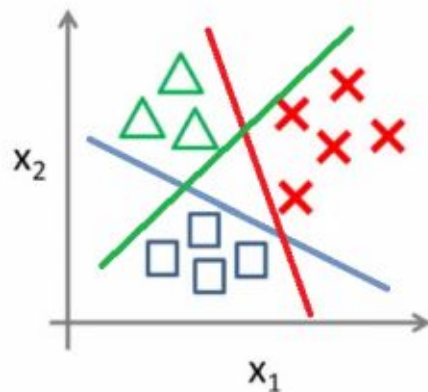


... we can combine more lines!!!

Binary classification:



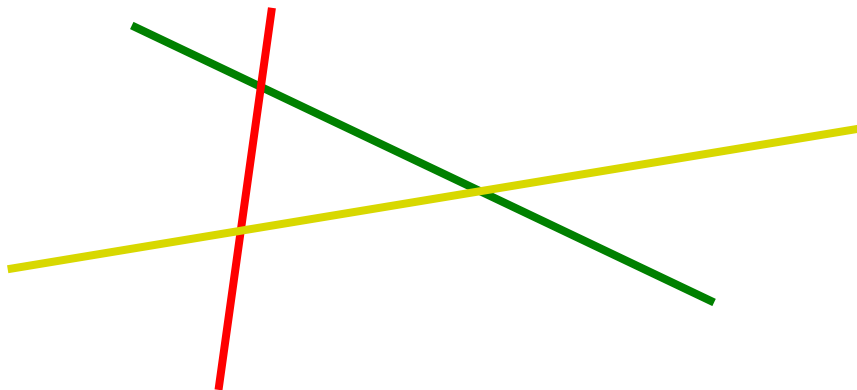
Multi-class classification:



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



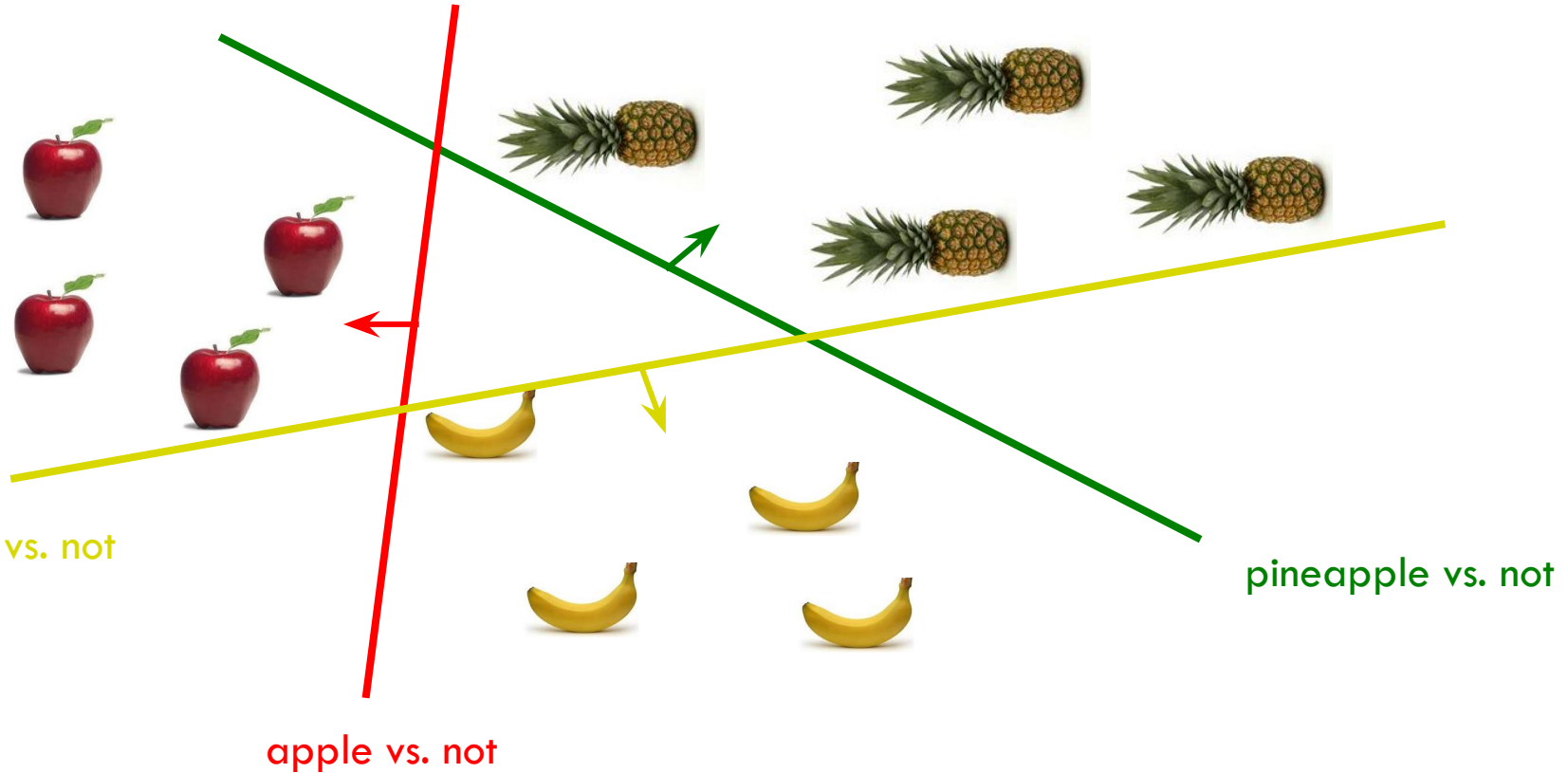
APPROACH 1: ONE VS. ALL (OVA)

Training: for each label L define a binary problem

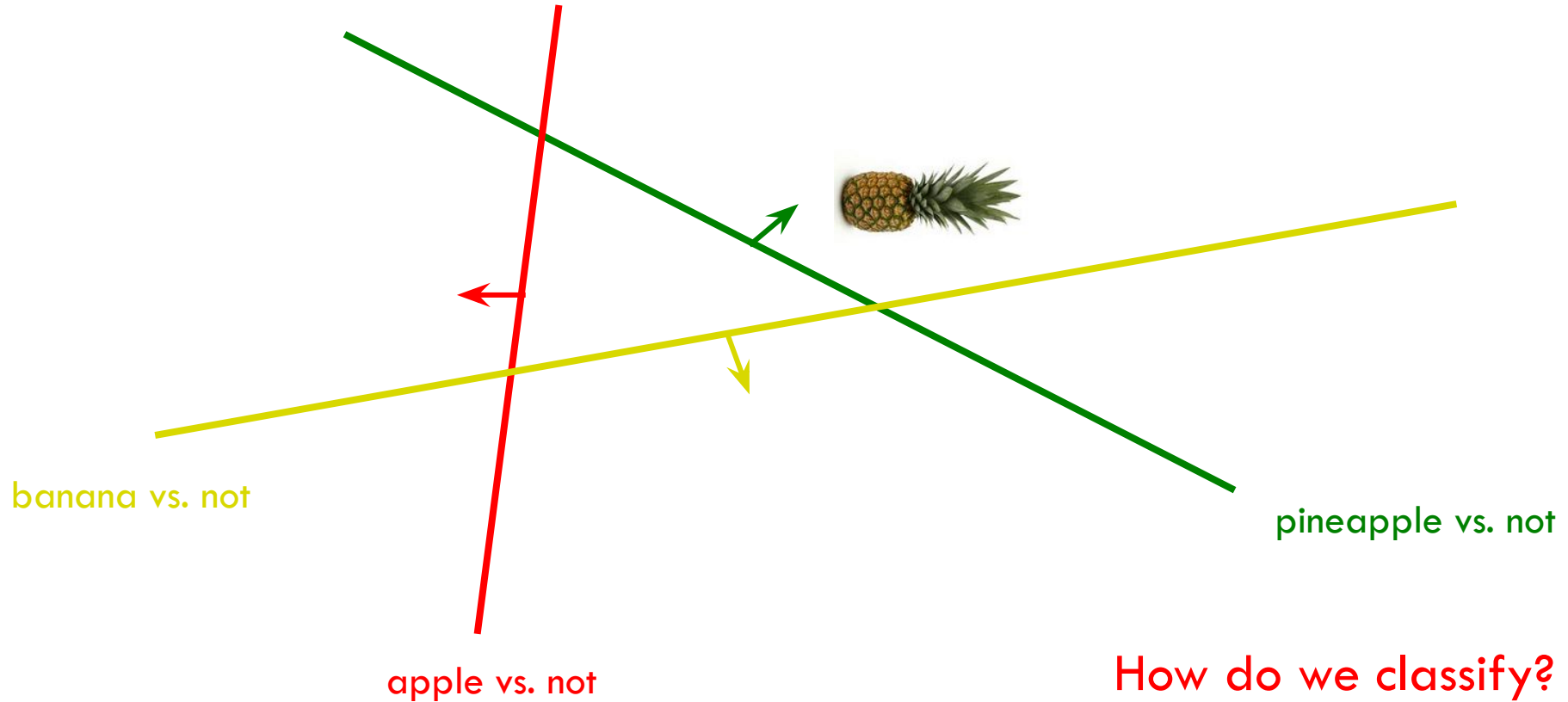
- all examples with label L are positive
- all other examples are negative

			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

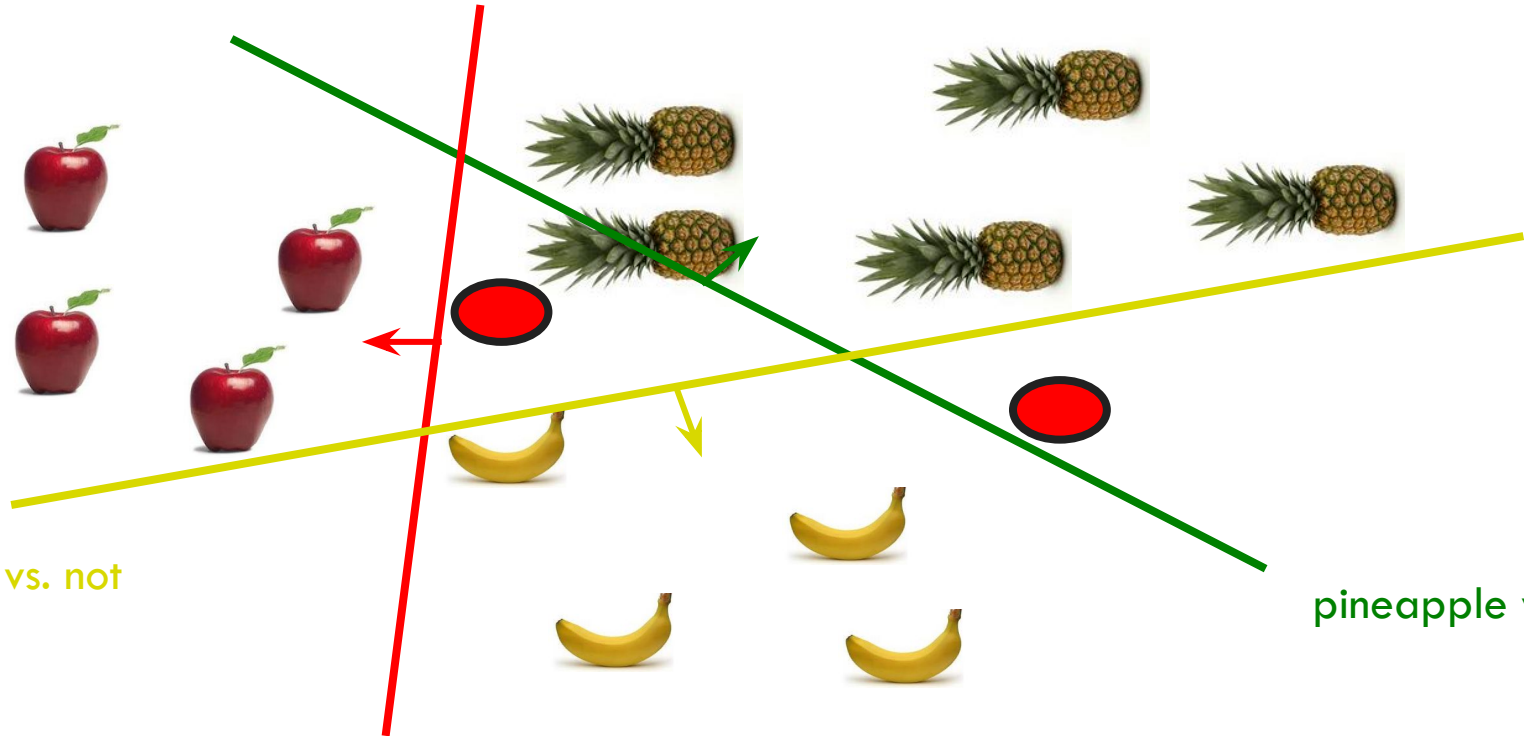
OVA: LINEAR CLASSIFIERS (E.G. PERCEPTRON)



OVA: LINEAR CLASSIFIERS (E.G. PERCEPTRON)



OVA: LINEAR CLASSIFIERS (E.G. PERCEPTRON)



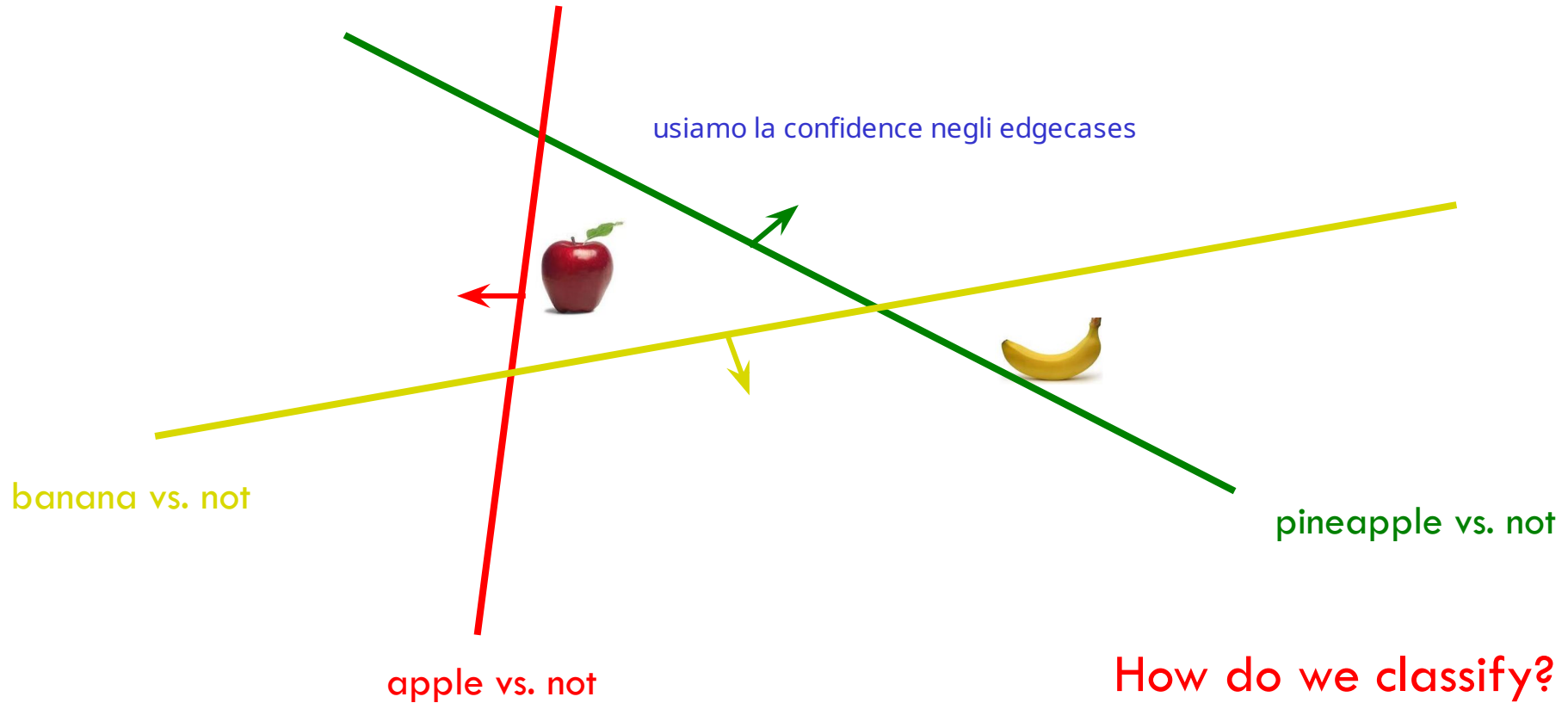
banana vs. not

apple vs. not

pineapple vs. not

How do we classify?

OVA: LINEAR CLASSIFIERS (E.G. PERCEPTRON)



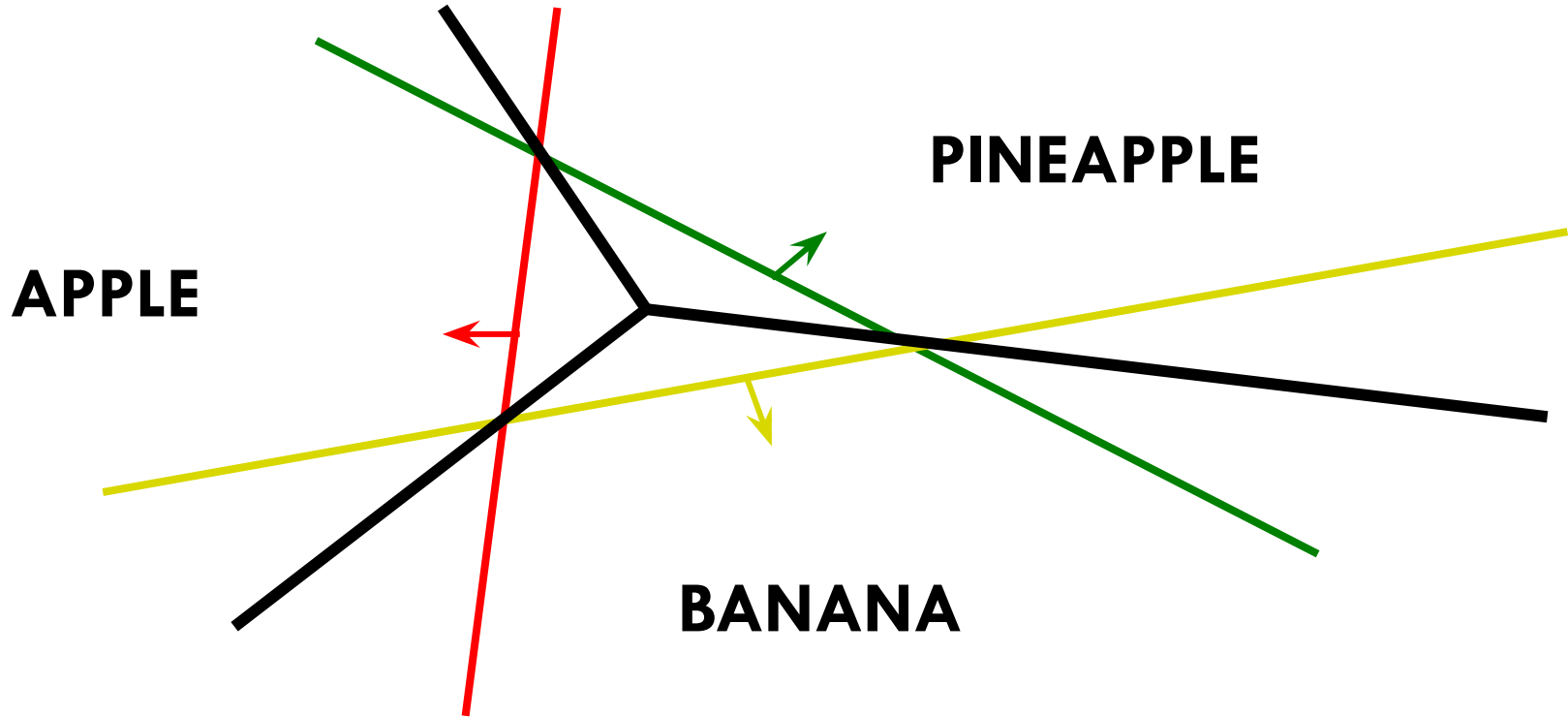
OVA: CLASSIFY

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

OVA: LINEAR CLASSIFIERS (E.G. PERCEPTRON)

decision boundary representation



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:    $\mathbf{D}^{bin} \leftarrow$  relabel  $\mathbf{D}^{multiclass}$  so class  $i$  is positive and  $\neg i$  is negative
3:    $f_i \leftarrow$  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:    $y \leftarrow f_i(\hat{x})$ 
4:    $score_i \leftarrow score_i + y$ 
5: end for
6: return  $\operatorname{argmax}_k score_k$        $\operatorname{argmax}$  = index corresponding to the max class ???
```




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)

	apple
	orange
	apple
	banana
	banana




apple vs orange

	+1
	+1
	-1

orange vs banana

	+1
	-1
	-1

apple vs banana

	+1
	+1
	-1
	-1

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: k 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
 - $y = F_{ij}(x)$
 - $\text{score}_j += y$
 - $\text{score}_k -= y$

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

- $y = F_{ij}(x)$
- $score_j += y$
- $score_i -= y$

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($D^{\text{multiclass}}$, BINARYTRAIN)

```
1:  $f_{ij} \leftarrow \emptyset, \forall 1 \leq i < j \leq K$ 
2: for  $i = 1$  to  $K-1$  do
3:    $D^{\text{pos}} \leftarrow$  all  $x \in D^{\text{multiclass}}$  labeled  $i$    extract all samples of label  $i$ 
4:   for  $j = i+1$  to  $K$  do
5:      $D^{\text{neg}} \leftarrow$  all  $x \in D^{\text{multiclass}}$  labeled  $j$    extract all samples of label  $j$ 
6:      $D^{\text{bin}} \leftarrow \{(x, +1) : x \in D^{\text{pos}}\} \cup \{(x, -1) : x \in D^{\text{neg}}\}$    relabel the samples
7:      $f_{ij} \leftarrow \text{BINARYTRAIN}(D^{\text{bin}})$ 
8:   end for
9: end for
10: return all  $f_{ij}$ s   tutti i classifier
```

Algorithm 16 ALLVERSUSALLTEST(all f_{ij} , \hat{x})

```
1:  $\text{score} \leftarrow \langle 0, 0, \dots, 0 \rangle$    // initialize  $K$ -many scores to zero
2: for  $i = 1$  to  $K-1$  do
3:   for  $j = i+1$  to  $K$  do
4:      $y \leftarrow f_{ij}(\hat{x})$ 
5:      $\text{score}_i \leftarrow \text{score}_i + y$ 
6:      $\text{score}_j \leftarrow \text{score}_j - y$ 
7:   end for
8: end for
9: return  $\text{argmax}_k \text{score}_k$ 
```

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)

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

EVALUATION

MULTICLASS EVALUATION



label

prediction

apple

orange



orange

orange



apple

apple



banana

pineapple



banana

banana



pineapple

pineapple

How should we evaluate?

Accuracy: 4/6

non funziona bene un accuracy globale
perchè non mi da informazioni sulla performance
per le singole classi

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: $4/6$

macroaveraging:

apple = $1/2$

orange = $1/1$

banana = $1/2$

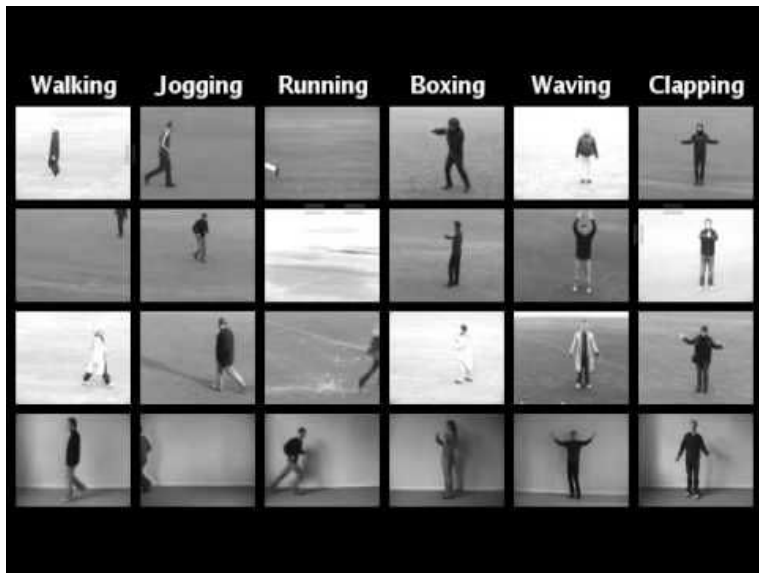
pineapple = $1/1$

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	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
	box	clap	wave	jog	Run	Walk

QUESTIONS?



Some slides are taken from David Kauchak