MULTICLASS

David Kauchak
CS 451 – Fall 2013

Admin

Assignment 4

Assignment 2 back soon

If you need assignment feedback…

CS Lunch tomorrow (Thursday): 12:20pm in Ross

Student CS talks

Eclipse/IDEs

generate class files with package structure, etc.

auto-generate method stubs (auto generate getters/setters)

check syntax on the fly

auto-complete as you type

automatically add imports

automatically add try/catch blocks

debugging

variable, method, parameter, renaming

Javadoc

Javadoc

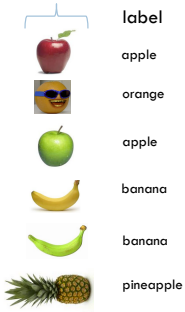
```
/**
 * Creates a new data set from a CSV file.  The file can start with any number
 * of "comment" lines which must start with a # sound.  Then the next line must
 * be a header (i.e. the features) then all following lines are treated as examples.
 *
 * @param csvFile comma separated file containing the examples WITH a header
 * @param labelIndex the index (0-based) where the label is at
 */
public DataSet(String csvFile, int labelIndex){
    try {
        BufferedReader in = new BufferedReader(new FileReader(csvFile));
```

- human readable
- easily generated in most IDEs
- can use tools to automatically generate documentation

http://www.cs.middlebury.edu/~dkauchak/classes/cs451/assignments/assign4/doc/

Multiclass classification

examples



label

apple

orange

apple

banana

banana

pineapple

Same setup where we have a set of features for each example

Rather than just two labels, now have 3 or more

real-world examples?

Real world multiclass classification

document classification

protein classification

handwriting recognition

face recognition

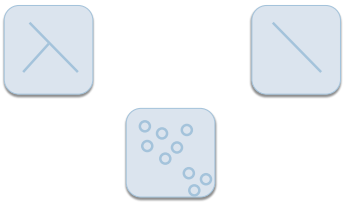
most real-world applications tend to be multiclass

sentiment analysis

autonomous vehicles

emotion recognition

Multiclass: current classifiers



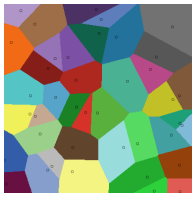
Any of these work out of the box?
With small modifications?

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!



Decision Tree learning

Base cases:

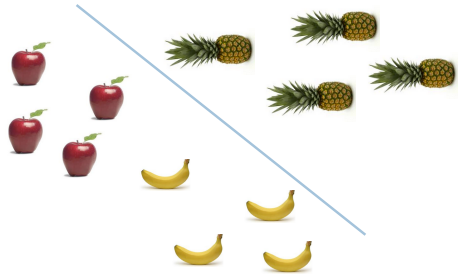
1. If all data belong to the same class, pick that label
2. If all the data have the same feature values, pick majority label
3. If we're out of features to examine, pick majority label
4. If the we don't have any data left, pick majority label of *parent*
5. *If some other stopping criteria* exists to avoid overfitting, pick majority label

Otherwise:

- calculate the "score" for each feature if we used it to split the data
- pick the feature with the highest score, partition the data based on that data value and call recursively

No algorithmic changes!

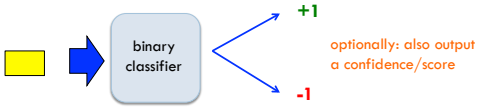
Perceptron learning



Hard to separate three classes with just one line ☹

Black box approach to multiclass

Abstraction: we have a generic binary classifier, how can we use it to solve our new problem



Can we solve our multiclass problem with this?