FEATURES

David Kauchak
CS 451 – Fall 2013

Admin

Assignment 2

- This class will make you a better programmer!
- How did it go?
- How much time did you spend?

Assignment 3 out

- Implement perceptron variants
- See how they differ in performance
- Take a break from implementing algorithms after this (for 1-2 weeks)

Features

Terrain	Unicycle-type	Weather	Go-For-Ride?
Trail	Normal	Rainy	NO
Road	Normal	Sunny	YES
Trail	Mountain	Sunny	YES
Road	Mountain	Rainy	YES
Trail	Normal	Snowy	NO
Road	Normal	Rainy	YES
Road	Mountain	Snowy	YES
Trail	Normal	Sunny	NO
Road	Normal	Snowy	NO
Trail	Mountain	Snowy	YES

Where do they come from?

UCI Machine Learning Repository

UCI Machine Learning Repository
Center for Machine Learning and Intelligent Systems

http://archive.ics.uci.edu/ml/datasets.html

Provided features

Predicting the age of abalone from physical measurements

Name / Data Type / Measurement Unit / Description

Sex / nominal / -- / M, F, and I (infant)
Length / continuous / mm / Longest shell measurement
Diameter / continuous / mm / perpendicular to length
Height / continuous / mm / with meat in shell
Whole weight / continuous / grams / whole abalone
Shucked weight / continuous / grams / weight of meat
Viscera weight / continuous / grams / gut weight (after bleeding)
Shell weight / continuous / grams / after being dried
Rings / integer / -- / +1.5 gives the age in years

Provided features

Predicting breast cancer recurrence

1. Class: no-recurrence-events, recurrence-events
2. age: 10-19, 20-29, 30-39, 40-49, 50-59, 60-69, 70-79, 80-89, 90-99.
3. menopause: lt40, ge40, premeno.
4. tumor-size: 0-4, 5-9, 10-14, 15-19, 20-24, 25-29, 30-34, 35-39, 40-44, 45-49, 50-54, 55-59.
5. inv-nodes: 0-2, 3-5, 6-8, 9-11, 12-14, 15-17, 18-20, 21-23, 24-26, 27-29, 30-32, 33-35, 36-39.
6. node-caps: yes, no.
7. deg-malig: 1, 2, 3.
8. breast: left, right.
9. breast-quad: left-up, left-low, right-up, right-low, central.
10. irradiated: yes, no.

Provided features

In many physical domains (e.g. biology, medicine, chemistry, engineering, etc.)

- the data has been collected and the *relevant* features identified
- we cannot collect more features from the examples (at least “core” features)

In these domains, we can often just use the provided features

Raw data vs. features

In many other domains, we are provided with the raw data, but must extract/identify features

For example

- image data
- text data
- audio data
- log data
- …

Text: raw data

Raw data

Features?

Feature examples

Raw data

Features

Clinton said banana repeatedly last week on tv, “banana, banana, banana”

(1, 1, 1, 0, 0, 1, 0, 0, …)

banana clinton said california across tv wrong capital

Occurrence of words

Feature examples

Raw data

Features

Clinton said banana repeatedly last week on tv, “banana, banana, banana”

(4, 1, 1, 0, 0, 1, 0, 0, …)

banana clinton said california across tv wrong capital

Frequency of word occurrence

Do we retain all the information in the original document?

Feature examples

Raw data

Features

Clinton said banana repeatedly last week on tv, “banana, banana, banana”

(1, 1, 1, 0, 0, 1, 0, 0, …)

banana repeatedly, clinton said, said banana, california schools, across the, tv banana, wrong way, capital city

Occurrence of bigrams

Feature examples

Raw data

Features

Clinton said banana repeatedly last week on tv, "banana, banana, banana"

(1, 1, 1, 0, 0, 1, 0, 0, …)

banana repeatedly
clinton said
said banana
california schools
across the
tv banana
wrong way
capital city

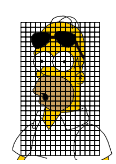
Other features?

Lots of other features

- POS: occurrence, counts, sequence
- Constituents
- Whether 'V1agra' occurred 15 times
- Whether 'banana' occurred more times than 'apple'
- If the document has a number in it
- …
- Features are very important, but we're going to focus on the models today

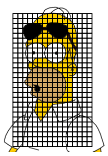
How is an image represented?

How is an image represented?



- images are made up of pixels
- for a color image, each pixel corresponds to an RGB value (i.e. three numbers)

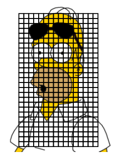
Image features



for each pixel: R[0-255]
G[0-255]
B[0-255]

Do we retain all the information in the original document?

Image features

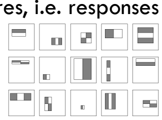


for each pixel: R[0-255]
G[0-255]
B[0-255]

Other features for images?

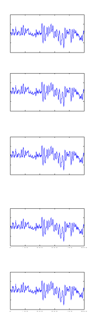
Lots of image features

- Use “patches” rather than pixels (sort of like “bigrams” for text)
- Different color representations (i.e. L*A*B*)
- Texture features, i.e. responses to filters



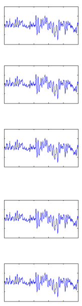
- Shape features
- …

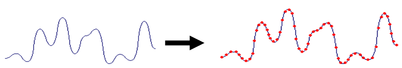
Audio: raw data



How is audio data stored?

Audio: raw data



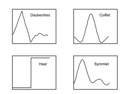


Many different file formats, but some notion of the frequency over time

Audio features?

Audio features

- frequencies represented in the data (FFT)
- frequencies over time (STFT)/responses to wave patterns (wavelets)



- beat
- timber
- energy
- zero crossings
- …

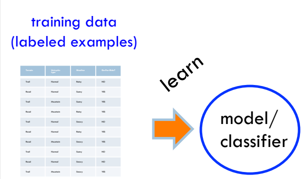
Obtaining features

Very often requires some domain knowledge

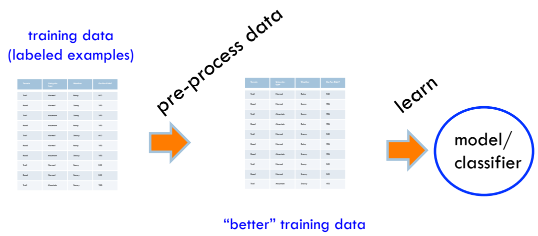
As ML algorithm developers, we often have to trust the "experts" to identify and extract reasonable features

That said, it can be helpful to understand where the features are coming from

Current learning model

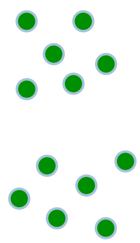


Pre-process training data



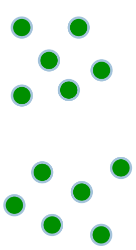
What types of preprocessing might we want to do?

Outlier detection



What is an outlier?

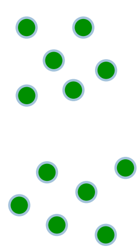
Outlier detection



An example that is inconsistent with the other examples

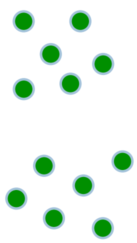
What types of inconsistencies?

Outlier detection



An example that is inconsistent with the other examples
- extreme feature values in one or more dimensions
- examples with the same feature values but different labels

Outlier detection



An example that is inconsistent with the other examples
- extreme feature values in one or more dimensions
- examples with the same feature values but different labels

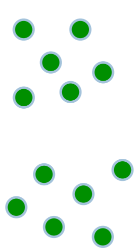
Fix?

Removing conflicting examples

Identify examples that have the same features, but differing values
- For some learning algorithms, this can cause issues (for example, not converging)
- In general, unsatisfying from a learning perspective

Can be a bit expensive computationally (examining all pairs), though faster approaches are available

Outlier detection



An example that is inconsistent with the other examples
- extreme feature values in one or more dimensions
- examples with the same feature values but different labels

How do we identify these?

Removing extreme outliers

Throw out examples that have extreme values in one dimension

Throw out examples that are very far away from any other example

Train a probabilistic model on the data and throw out "very unlikely" examples

This is an entire field of study by itself! Often called outlier or anomaly detection.

Quick statistics recap

What are the mean, standard deviation, and variance of data?

Quick statistics recap

mean: average value, often written as μ

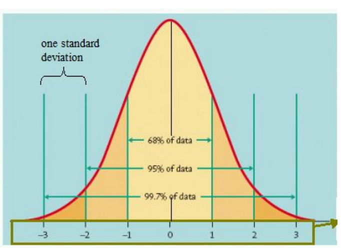
variance: a measure of how much variation there is in the data. Calculated as:

$$\sigma^2 = \frac{\sum_{i=1}^{n}(x_i - \mu)^2}{n}$$

standard deviation: square root of the variance (written as σ)

How can these help us with outliers?

Outlier detection

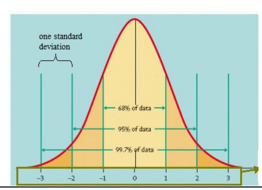


If we know the data is distributed normally (i.e. via a normal/gaussian distribution)

Outliers in a single dimension

Examples in a single dimension that have values greater than $|k\sigma|$ can be discarded (for k >>3)

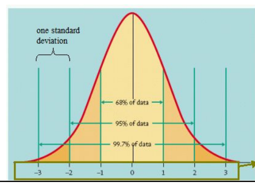
Even if the data isn't actually distributed normally, this is still often reasonable



Outliers in general

- Calculate the centroid/center of the data
- Calculate the average distance from center for all data
- Calculate standard deviation and discard points too far away

Again, many, many other techniques for doing this



Outliers for machine learning

Some good practices:

- Throw out conflicting examples
- Throw out any examples with obviously extreme feature values (i.e. many, many standard deviations away)
- Check for erroneous feature values (e.g. negative values for a feature that can only be positive)
- Let the learning algorithm/other pre-processing handle the rest

Feature pruning

Good features provide us information that helps us distinguish between labels

However, not all features are good

What makes a bad feature and why would we have them in our data?

Bad features

Each of you are going to generate a feature for our data set: pick 5 random binary numbers

f_1 f_2 … label

I've already labeled these examples and I have two features

Bad features

Each of you are going to generate some a feature for our data set: pick 5 random binary numbers

f_1 f_2 …	label
	1
	0
	1
	1
	0

Is there any problem with using your feature in addition to my two real features?