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

<table>
<thead>
<tr>
<th>Terrain</th>
<th>Unicycle-type</th>
<th>Weather</th>
<th>Go-For-Ride?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trail</td>
<td>Normal</td>
<td>Rainy</td>
<td>NO</td>
</tr>
<tr>
<td>Road</td>
<td>Normal</td>
<td>Sunny</td>
<td>YES</td>
</tr>
<tr>
<td>Trail</td>
<td>Mountain</td>
<td>Sunny</td>
<td>YES</td>
</tr>
<tr>
<td>Road</td>
<td>Mountain</td>
<td>Rainy</td>
<td>YES</td>
</tr>
<tr>
<td>Trail</td>
<td>Normal</td>
<td>Snowy</td>
<td>NO</td>
</tr>
<tr>
<td>Road</td>
<td>Normal</td>
<td>Rainy</td>
<td>YES</td>
</tr>
<tr>
<td>Road</td>
<td>Mountain</td>
<td>Snowy</td>
<td>YES</td>
</tr>
<tr>
<td>Trail</td>
<td>Normal</td>
<td>Sunny</td>
<td>NO</td>
</tr>
<tr>
<td>Road</td>
<td>Normal</td>
<td>Snowy</td>
<td>NO</td>
</tr>
<tr>
<td>Trail</td>
<td>Mountain</td>
<td>Snowy</td>
<td>YES</td>
</tr>
</tbody>
</table>

Where do they come from?

UCI Machine Learning Repository

Provided features

Predicting the age of abalone from physical measurements

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Measurement Unit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sex</td>
<td>nominal</td>
<td>--</td>
<td>M, F, and I (Infant)</td>
</tr>
<tr>
<td>Length</td>
<td>continuous</td>
<td>mm</td>
<td>Longest shell measurement</td>
</tr>
<tr>
<td>Diameter</td>
<td>continuous</td>
<td>mm</td>
<td>perpendicular to length</td>
</tr>
<tr>
<td>Height</td>
<td>continuous</td>
<td>mm</td>
<td>with meat in shell</td>
</tr>
<tr>
<td>Whole weight</td>
<td>continuous</td>
<td>grams</td>
<td>whole abalone</td>
</tr>
<tr>
<td>Shucked weight</td>
<td>continuous</td>
<td>grams</td>
<td>weight of meat</td>
</tr>
<tr>
<td>Viscera weight</td>
<td>continuous</td>
<td>grams</td>
<td>gut weight (after bleeding)</td>
</tr>
<tr>
<td>Shell weight</td>
<td>continuous</td>
<td>grams</td>
<td>after being dried</td>
</tr>
<tr>
<td>Rings</td>
<td>integer</td>
<td>--</td>
<td>+1.5 gives the age in years</td>
</tr>
</tbody>
</table>

Provided features

Predicting breast cancer recurrence

1. Class: no-recurrence-events, recurrence-events
3. menopause: premeno, postmeno.
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

<table>
<thead>
<tr>
<th>Raw data</th>
<th>Features?</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Feature examples

<table>
<thead>
<tr>
<th>Raw data</th>
<th>Features</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Clinton said banana repeatedly last week on tv, &quot;banana, banana, banana&quot;</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>(4, 1, 1, 0, 1, 0, 0, ...)</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Frequency of word occurrence</th>
</tr>
</thead>
</table>

Do we retain all the information in the original document?
Feature examples

<table>
<thead>
<tr>
<th>Raw data</th>
<th>Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clinton said banana repeatedly last week on tv, &quot;banana, banana, banana&quot;</td>
<td>(1, 1, 0, 0, 1, 0, 0, ...)</td>
</tr>
</tbody>
</table>

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?

- Images are made up of pixels
- For a color image, each pixel corresponds to an RGB value (i.e., three numbers)
Do we retain all the information in the original document?

Other features for images?

- 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

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

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

training data (labeled examples) → learn → model/classifier
Pre-process training data

What types of preprocessing might we want to do?

Outlier detection

What is an outlier?

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

How do we identify these?

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

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?

- **Mean**: average value, often written as $\mu$

- **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 $\sigma$)

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 $k \gg 3$)

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

\[ f_1, f_2, \ldots, \text{label} \]

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

<table>
<thead>
<tr>
<th>$f_1$</th>
<th>$f_2$</th>
<th>...</th>
<th>label</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td></td>
<td></td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td></td>
<td></td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td></td>
<td></td>
<td>1</td>
</tr>
</tbody>
</table>