Assignments 7

Next Monday: project proposal presentations
- informal
- 1 minute
- See the final project handout for details

Hack week QA session from OpenAI engineer
(Friday @ 12:30pm)
- https://5chack.com/#hack-week

Schedule for the rest of the semester

Machine Learning: A Geometric View
<table>
<thead>
<tr>
<th>Weight</th>
<th>Color</th>
<th>Label</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>Red</td>
<td>Apple</td>
</tr>
<tr>
<td>5</td>
<td>Yellow</td>
<td>Apple</td>
</tr>
<tr>
<td>6</td>
<td>Yellow</td>
<td>Banana</td>
</tr>
<tr>
<td>3</td>
<td>Red</td>
<td>Apple</td>
</tr>
<tr>
<td>7</td>
<td>Yellow</td>
<td>Banana</td>
</tr>
<tr>
<td>8</td>
<td>Yellow</td>
<td>Banana</td>
</tr>
<tr>
<td>6</td>
<td>Yellow</td>
<td>Apple</td>
</tr>
</tbody>
</table>

Can we visualize this data?

Turn features into numerical values

We can view examples as points in an $n$-dimensional space where $n$ is the number of features called the feature space.

Examples in a feature space

Test example: what class?
Test example: *what class?*

Another classification algorithm?

To classify an example $d$:
Label $d$ with the label of the closest example to $d$ in the training set

What about this example?

What about this example?
What about this example?

Most of the next closest are blue

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

How do we measure “nearest”?

Euclidean distance

$$D(a, b) = \sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2 + ... + (a_n - b_n)^2}$$
Decision boundaries

The decision boundaries are places in the features space where the classification of a point/example changes.

Where are the decision boundaries for k-NN?

K Nearest Neighbour (k-NN) Classifier

What is the decision boundary for k-NN for this one?
Machine learning models

Some machine learning approaches make strong assumptions about the data:
- If the assumptions are true, this can often lead to better performance.
- If the assumptions aren’t true, they can fail miserably.

Other approaches don’t make many assumptions about the data:
- This can allow us to learn from more varied data.
- But, they are more prone to overfitting (biasing too much to the training data).
- And generally require more training data.

What is the data generating distribution?
What is the data generating distribution?

Actual model
Model assumptions

If you don't have strong assumptions about the model, it can take you a longer to learn

Assume now that our model of the blue class is two circles
What is the data generating distribution?

Knowing the model beforehand can drastically improve the learning and the number of examples required.
What is the data generating distribution?

Make sure your assumption is correct, though!

Machine learning models

What were the model assumptions (if any) that k-NN and NB made about the data?

Are there training data sets that could never be learned correctly by these algorithms?

k-NN model
Linear models

A strong assumption is **linear separability**:
- In 2 dimensions, you can separate labels/classes by a line.
- In higher dimensions, need hyperplanes.

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

---

Hyperplanes

A hyperplane is line/plane in a high dimensional space.

What defines a line?
What defines a hyperplane?

---

Defining a line

Any pair of values \((w_1, w_2)\) defines a line through the origin:

\[ 0 = w_1 f_1 + w_2 f_2 \]

What does this line look like?

---

Defining a line

Any pair of values \((w_1, w_2)\) defines a line through the origin:

\[ 0 = w_1 f_1 + w_2 f_2 \]

\[ 0 = 1 f_1 + 2 f_2 \]

What does this line look like?
Defining a line

Any pair of values \((w_1, w_2)\) defines a line through the origin:

\[ 0 = w_1 f_1 + w_2 f_2 \]

\[ 0 = 1 f_1 + 2 f_2 \]

We can also view it as the line perpendicular to the weight vector

Classifying with a line

Mathematically, how can we classify points based on a line?

\[ 0 = 1 f_1 + 2 f_2 \]

We can classify points based on whether they are on one side of the line or the other.
Classifying with a line

Mathematically, how can we classify points based on a line?

$$0 = f_1^1 + 2f_2$$

(1,1): \[1 \times 1 + 2 \times 1 = 3\]

(1,-1): \[1 \times 1 + 2 \times (-1) = -1\]

The sign indicates which side of the line

Defining a line

Any pair of values \((w_1, w_2)\) defines a line through the origin:

$$a = w_1f_1 + w_2f_2$$

$$-1 = f_1 + 2f_2$$

Now intersects at -1
Linear models

A linear model in $n$-dimensional space (i.e. $n$ features) is defined by $n+1$ weights:

In two dimensions, a line:
\[ 0 = w_1 f_1 + w_2 f_2 + b \]  
(where $b \equiv -a$)

In three dimensions, a plane:
\[ 0 = w_1 f_1 + w_2 f_2 + w_3 f_3 + b \]

In $n$-dimensions, a hyperplane:
\[ 0 = b + \sum_{i=1}^{n} w_i f_i \]

Classifying with a linear model

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

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

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

Learning a linear model

Geometrically, we know what a linear model represents.

Given a linear model (i.e. a set of weights and $b$) we can classify examples.

How do we learn a linear model?

Which hyperplane would you choose?
Large margin classifiers

Choose the line where the distance to the nearest point(s) is as large as possible.

Large margin classifiers attempt to maximize this margin.

Large margin classifier setup

Select the hyperplane with the largest margin where the points are classified correctly!

Setup as a constrained optimization problem:

\[
\max_{w,b} \quad \text{margin}(w,b) \\
\text{subject to:} \\
y_i(w \cdot x_i + b) > 0 \quad \forall i
\]

what does this say?

- \(y_i\): label for example \(i\), either 1 (positive) or -1 (negative)
- \(x_i\): feature vector for example \(i\)

Measuring the margin

How do we calculate the margin?
Support vectors

For any separating hyperplane, there exist some set of “closest points”
These are called the support vectors

Measuring the margin

The margin is the distance to the support vectors, i.e. the “closest points”, on either side of the hyperplane

Support vector machine problem

Posed as a quadratic optimization problem
Maximize/minimize a quadratic function
Subject to a set of linear constraints
Many, many variants of solving this problem
One of the most successful classification approaches

Support vector machines

One of the most successful (if not the most successful) classification approach:

- decision tree
- Support vector machine
- k nearest neighbor
- Naive Bayes

Google Scholar
Other successful classifiers in NLP

<table>
<thead>
<tr>
<th>Perceptron algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Linear classifier</td>
</tr>
<tr>
<td>- Trains “online”</td>
</tr>
<tr>
<td>- Fast and easy to implement</td>
</tr>
<tr>
<td>- Often used for tuning parameters (not necessarily for classifying)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Logistic regression classifier (aka Maximum entropy classifier)</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Probabilistic classifier</td>
</tr>
<tr>
<td>- Doesn’t have the NB constraints</td>
</tr>
<tr>
<td>- Performs very well</td>
</tr>
<tr>
<td>- More computationally intensive to train than NB</td>
</tr>
</tbody>
</table>