Week 3 Lab

Bookmark this page

Week 3 Lab

(1.0 points possible)

A data folder for doing this lab can be found **here**. Download this to your computer.

In this lab we will look at feature engineering for car data, written food reviews, images, and sound. Please make sure you read the lecture notes covering feature representations for this lab.

1) Feature engineering for car data

Last week, given a dataset of feature vectors and their corresponding labels, we saw how to implement two algorithms for building linear classifiers. This week we are looking at how these features are defined from raw collected data and some implications of those feature choices on the learning algorithm.

In this part of the lab, we are going to explore different ways of defining features for data.

Open `auto-mpg.tsv` in a text editor (or view on the web here). This file is in a common format, called "tab separated values". In the first line you will find the names of the columns. Each row is a data point; **the first column is the label for that point (1 or -1).**

"*The data concerns city-cycle fuel consumption in miles per gallon, to be predicted in terms of 3 multivalued discrete and 5 continuous attributes.*" (Quinlan, 1993)

The original data came from the StatLib library from CMU. It was modified by Ross Quinlan to remove entries with unknown mpg (miles per gallon). We have modified it further by removing six entries with unknown horsepower. We have also binarized the mpg column to turn this into a classification problem (later in the term, we will look at predicting continuous values, i.e. regression problems). Here are the nine columns in the dataset:

```
1. mpg:          continuous (modified by us to be -1 for mpg < 23, 1 for others)
2. cylinders:    multi-valued discrete
3. displacement: continuous
4. horsepower:   continuous
5. weight:       continuous
6. acceleration: continuous
7. model year:   multi-valued discrete
8. origin:       multi-valued discrete
9. car name:     string (many values)
```

There are 392 entries in the dataset, evenly split between positive and negative labels. The field names should be self-explanatory except possibly `displacement` and `origin`. You can read about `displacement` here; in this data set `displacement` is in cubic inches. `origin` is 1=USA, 2=Europe, 3=Asia. We'll ignore `car name` in this assignment.

A new student, Hiper Playne, suggests that we should just use all the numeric features in their raw form to predict whether the car will get good or bad gas mileage. He will use the Perceptron algorithm to learn the classifier. Once he trains the model on this dataset, he wants to predict whether cars in 2019 will get good gas mileage.

1A) What problems do you think Hiper might have with this method?

- [x] Because `weight` values and cylinders values are on different scales, perceptron might take many iterations
- [ ] `cylinders` is a discrete valued feature
- [x] `origin` is a discrete valued feature
- [ ] There are too many features and the classifier will overfit
- [x] `model year` is in the 70s, so a classifier based on this data might not perform well in 2019

Ask for Help 100.00%

You have infinitely many submissions remaining.

Solution:

✔ Because `weight` values and cylinders values are on different scales, perceptron might take many iterations

✘ `cylinders` is a discrete valued feature

✔ `origin` is a discrete valued feature

✘ There are too many features and the classifier will overfit

✔ `model year` is in the 70s, so a classifier based on this data might not perform well in 2019

Explanation:

1. We do not want to values on different scales (here we have weights of the order of 3000 or 4000, but cylinders are of the order of 5 or 10), as the learning algorithm will find it harder to get a good classifier. It is important to put all values on an equal basis.

2,3. Number of cylinders can stay as a discrete-valued feature because the magnitude conveys useful comparative information. However, the origin should not be a discrete-valued feature as its magnitude conveys no information. Discrete-valued numbers were used in the data set simply as indicators of different origins, and they actually could have been in any random order.

4. Here, the fact that the data has many features helps in classification. (The number of features becomes a problem if there are so many of them that some data points could have been identified by just looking at very few of their coordinates. Here it is not the case.)

5. Your model fits the training data you provide. So a model trained on 70's data would be less performant on 2019 data, compared to a model trained on data from the late 00's.

1B) For each feature from the following:
`[cylinders, displacement, horsepower, weight, acceleration, model_year, origin]`
indicate how you can represent it so as to make classification easier and get good generalization on unseen data, by choosing one of:

- `'drop'` - leave the feature out,
- `'raw'` - use values as they are,
- `'standard'` - standardize values by subtracting out average value and dividing by standard deviation,
- `'one-hot'` - use a one-hot encoding.

There could be multiple answers that make sense for each feature; please mention the tradeoffs between each answer. Write down your choices.

1C) How can `car name`, a textual feature, be transformed into a feature which can be used by the perceptron algorithm?

1D) Given a learning algorithm (example: perceptron), say you found 3 ideas for how to represent the textual feature from 1C. Talk with your partner to determine which of the following options would be appropriate, for determining the best feature representation out of the 3 ideas:

- For each feature representation, compute a classifier by running the algorithm on all the data and compare the number of errors of that classifier on the data.
- Split the data into two sets: training is 90%, test is 10%. For each feature representation, compute a classifier by running the algorithm on the training data, compare the sum of the number of errors of that classifier on the training and test data.
- Split the data into two sets: training is 90%, test is 10%. For each feature representation, compute a classifier by running the algorithm on the training data, compare the number of errors of that classifier on the test data.
- For each feature representation, perform 10-fold cross-validation and compare the resulting average error rate.

2) Feature engineering for food reviews

In this part of the assignment, we are going to explore ways of defining features for textual data.

Open `reviews.tsv` in a text editor (or view on the web here). This file is in "tab separated values" (tsv) format, and it consists of 10,000 fine food reviews from Amazon. You can find additional information here. We will be focusing on the `text` field and use it to predict the `sentiment` field (-1 or 1).

We will convert review texts into fixed-length feature vectors using a bag-of-words (BOW) approach. We start by compiling a list of all the words that appear in a **training** set of reviews into a *dictionary*, thereby producing a list of d *unique* words.

2A) For instance, consider two simple documents "Mary is selling apples." and "Tom is buying apples to eat". If we had only these two sentences in our training set of reviews, which option corresponds to the dictionary that we build when using the bag-of-words approach discussed above?

- ○ $(Mary, is, selling, apples, Tom, is, buying, apples, to, eat)$
- ◉ $(Mary, is, selling, apples, Tom, buying, to, eat)$
- ○ $(Mary, is, selling, red, apples, Tom, buying, blue, to, eat)$

Submit View Answer Ask for Help 100.00%

You have infinitely many submissions remaining.

We can then transform each of the reviews into a feature vector of length d by setting the i^{th} coordinate of the feature vector to 1 if the i^{th} word in the dictionary appears in the review or 0 if it does not.

2B) Hence, how would we represent the sentence "Tom is buying apples to eat" as a feature vector using the bag-of-words approach and the dictionary we found above?

- ○ $(1, 1, 0, 0, 1, 1, 1, 1)$
- ○ $(0, 1, 0, 1, 1, 0, 1, 1)$
- ◉ $(0, 1, 0, 1, 1, 1, 1, 1)$

Submit View Answer Ask for Help 100.00%

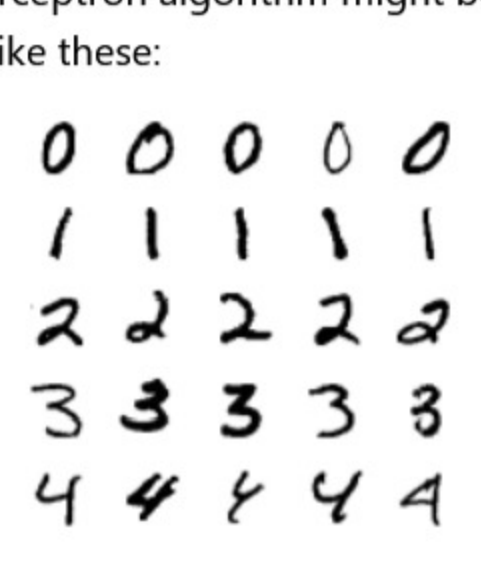
You have infinitely many submissions remaining.

2C) Talk with your partner about the weaknesses of the bag-of-words approach seen above. For instance, how would you interpret the feature vector $(1, 1, 1, 1, 1, 0, 1, 0)$? (Who is selling what to whom?)

2D) Words like "the", "to", "a", "and" and so on occur with very high frequency in English sentences. Do they help in detecting the sentiment in a review? Talk to your partner about how to deal with these words, when building your dictionary, using the bag-of-words approach.

3) Image features

We will be exploring in the homework how the perceptron algorithm might be applied to classify images of handwritten digits, from a well-known ("MNIST") dataset, with items like these:

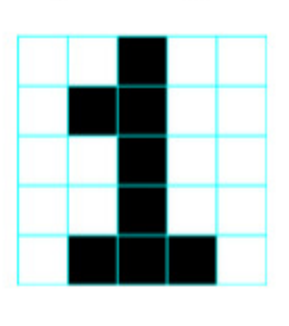


For today, assume we only have images of digits '1' and '3' and we would like to find a linear classifier which can classify these images correctly, giving label $+1$ for the digit '1' and label -1 for the digit '3'.

For simplicity, assume each image provided is a 5-pixel wide and 5-pixel tall (5x5) black and white image.

3A) Assume images from the MNIST dataset are provided to you as 5x5 matrices. An image is represented as a matrix, say A, with the entry of the i^{th} row and j^{th} column ($A_{i,j}$) representing the image pixel found at the i^{th} row and j^{th} column of pixels. Black pixels have value 0 while while pixels have value 1 in the matrix.

With the help of your partner, write down the matrix corresponding to the image shown here:



3B) Imagine we want to use the perceptron algorithm to perform the task of distinguishing between images of digits '1' and '3'. Each image is a data point, which we need in **vector** form, to feed to the perceptron algorithm. One approach is to take the two-dimensional matrix representation of an image and concatenate the rows one after the other, into a one-dimensional vector of the form $[A_{1,1}, A_{1,2}, A_{1,3}, A_{1,4}, A_{1,5}, A_{2,1}, A_{2,2}, ..., A_{5,5}]$

For the image shown above, what will be the last 3 entries of the one-dimensional vector representing that image? Enter a python list of length 3 for your answer. [0,0,1]

Submit View Answer Ask for Help 100.00%

You have infinitely many submissions remaining.

3C) How well would you expect the perceptron algorithm to work on classifying images if you simply represented them as vectors? Is there any information lost when representing images this way?

3D) What approaches might you suggest to extract more meaningful features from the images, such that the perceptron algorithm might do a good job of classification? (Hint: What makes the image of the digit '1' different compared to the image of the digit '3'?)

In the homework for this week we will investigate these datasets in more detail. And later in the semester, we'll explore other algorithms, including neural networks and convolutional neural networks, which can essentially do feature extraction on their own.

4) Acoustic features

Hiper Playne now has access to a large number of one-second sound clips, each one sounding like a dog bark or a cat meow. Each sound clip is a vector of length 100, with an amplitude value in the range [0,1] assigned for discrete time steps $0.00, 0.01, 0.02, ..., 0.99$. He wants to use the perceptron algorithm to learn a classifier which can separate dog bark from cat meow sound clips.

4A) How well would you expect the perceptron algorithm to work, if it was given these raw sound clips as input?

4B) What makes a cat's meow different from a dog's bark? If you had a black box which can take in a sound clip and return the 5 most dominant frequencies heard in the sound clip, how would you use the black box and extract features from the sound clips? How would you represent those features as a vector which will then be fed to the perceptron algorithm?