Homework 2: Linear Classification

Writeup and code due 23:59 on Monday October 17, 2022

You will do this assignment individually and submit your answers as a PDF via the Blackboard course website. There is a mathematical component and a programming component to this homework.

This homework is about multi-class logistic regression. Whereas in typical logistic regression, we usually build classifiers that discriminate between two classes, in multi-class logistic regression we discriminate between three or more classes. As usual, we imagine that we have some inputs $\mathbf{x}_n \in \mathbb{R}^D$ (or perhaps some basis, without loss of generality) but that our outputs are now "one-hot coded". What that means is that, if there are K output classes, rather than representing the output labels as integers $1, 2, \ldots, K$, we represent them as a binary vectors of length K. These vectors are zero in each component except for the one corresponding to the correct label, and that entry has a one. So, if there are seven classes and a particular datum has label 3, then the target vector would be $[0, 0, 1, 0, 0, 0, 0]$.

There are various loss functions that can be used for classification, but logistic regression is a discriminative probabilistic model. Therefore, a prediction consists of a distribution over the different classes. Here that means a vector of nonnegative numbers that sum to one. To turn inputs into such a vector, it is most common to use a set of weights for each class, $\mathbf{w}_k$ and a *softmax* function:

$$\Pr(t_k = 1 \mid \mathbf{x}, \{\mathbf{w}_{k'}\}_{k'=1}^{K}) = \frac{\exp\{\mathbf{w}_k^\mathsf{T}\mathbf{x}\}}{\sum_{k'=1}^{K} \exp\{\mathbf{w}_{k'}^\mathsf{T}\mathbf{x}\}}\,.$$

Here we're using $t_k = 1$ to indicate the probability that the kth entry is one, i.e., that the label is class k. Across the different entries of $\mathbf{t}$, this produces a non-negative vector that sums to one based on the inner products of the weights with the input features.

See the readings: 3.6.1 from *Undergraduate Fundamentals of Machine Learning* and 4.3.2, 4.3.4 from Bishop.

1. Alternative Softmax Parameterization [8pts]

The softmax looks a bit different than the binary case, in which the logistic function $1/(1 + \exp\{-\mathbf{w}^\mathsf{T}\mathbf{x}\})$ is used and it appears that the quantity 1 is being used for one of the classes. This can be done with the multi-class case also:

$$\Pr(t_k = 1 \mid \mathbf{x}, \{\mathbf{w}_{k'}\}_{k'=1}^{K-1}) = \begin{cases} \dfrac{\exp\{\mathbf{w}_k^\mathsf{T}\mathbf{x}\}}{1 + \sum_{k'=1}^{K-1} \exp\{\mathbf{w}_{k'}^\mathsf{T}\mathbf{x}\}} & k < K \\ \dfrac{1}{1 + \sum_{k'=1}^{K-1} \exp\{\mathbf{w}_{k'}^\mathsf{T}\mathbf{x}\}} & k = K \end{cases}.$$

Show why this reparameterization does not change the classifiers that can be represented.

2. Maximum Likelihood Training Objective [10pts]

Having seen data $\{\mathbf{x}_n, \mathbf{t}_n\}_{n=1}^N$, with $\mathbf{t}_n$ being the one-hot label vectors for K classes, what is the likelihood function associated with this alternative softmax parameterization?

3. Gradient of Alternative Softmax Parameterization [10pts]

What is the gradient of the log likelihood in terms of one of the weight vectors $\mathbf{w}_k$?

4. Classifying Fruit [20pts]

You're tasked with classifying three different kinds of fruit, based on their heights and widths. Figure 1 is a plot of the data. Iain Murray collected these data and you can read more about this on his website at `http://homepages.inf.ed.ac.uk/imurray2/teaching/oranges_and_lemons/`. We have made a slightly simplified (collapsing the subcategories together) version of this available as `fruit.csv`, which you can download from the Blackboard website. The file has three columns: type (1=apple, 2=orange, 3=lemon), width, and height. The first few lines look like this:

```
fruit,width,height
1,8.4,7.3
1,8,6.8
1,7.4,7.2
1,7.1,7.8
...
```

Implement the three-class generalization of logistic regression for these data using the alternative parameterization given here. Use gradient ascent on the log likelihood (or a fancier optimizer like L-BFGS, if you like). Make a plot that shows the decision boundaries. Numpy and Matplotlib will be useful Python libraries. See HW0 Problem 15 for a Numpy review.

5. Optional: Generative Classifiers [50 bonus points]

Implement a simple generative classifier with Gaussian class-conditional densities. Fit each of the class-conditional distributions using maximum likelihood and plot the resulting decision boundaries. See the readings: 3.6.2 from *Undergraduate Fundamentals of Machine Learning* and 4.2.2 from Bishop. The Scipy library has a multivariate normal pdf function you can use to simplify coding.

Describe in words the differences that you see in the boundaries for the two models. Which one looks like it can fit these simple data better? Why do you think this is true?

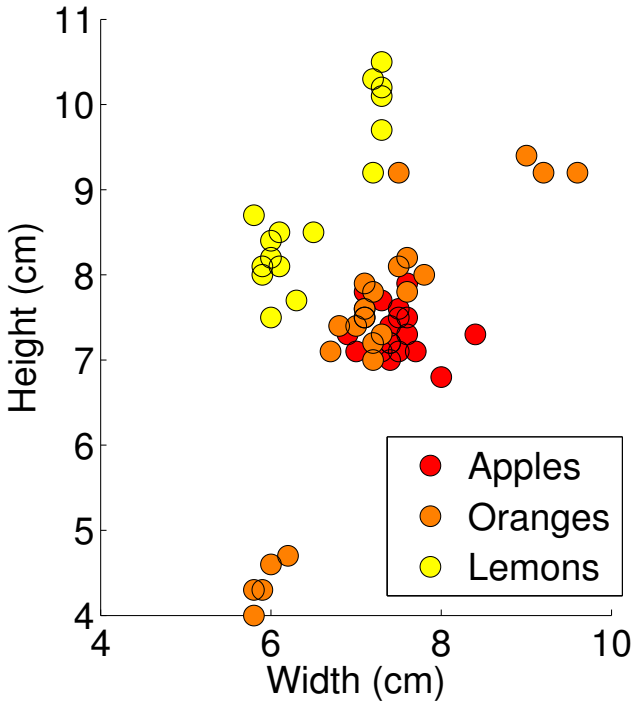

Figure 1: Heights and widths of apples, oranges, and lemons. These fruit were purchased and measured by Iain Murray: http://homepages.inf.ed.ac.uk/imurray2/teaching/oranges_and_lemons/.

Calibration [2pt]

Approximately how long did this homework take you to complete?