Homework 2: Least Squares and Feature Engineering

Due: 9/30/2021

1. *Matrix calculus.* Recall that the gradient of a scalar function $f : \mathbf{R}^n \to \mathbf{R}$ is the vector whose components are the partial derivatives of f.

$$\nabla f(x) = \left(\frac{\partial f}{\partial x_1}, \frac{\partial f}{\partial x_2}, ..., \frac{\partial f}{\partial x_n} \right) \in \mathbf{R}^n$$

Let $x \in \mathbf{R}^n$. Consider the function

$$f(x) = \log \left(\sum_{i=1}^{n} e^{x_i} \right).$$

(a) Calculate the partial derivatives $\dfrac{\partial f}{\partial x_j}$ for $j = 1, \ldots, n$.

(b) What is the gradient of f?

(c) Based on the solution to the above problem, do you think there is a matrix calculus equivalent of the "chain rule" from calculus? Formulate this rule. That is, give a formula for $\nabla g(h(x))$ where $g : \mathbf{R} \to \mathbf{R}$ and $h : \mathbf{R}^n \to \mathbf{R}$ when the functions are smooth so all derivatives are well-defined.

Verify this equation for the results you obtained in parts (a) + (b).

(d) Now consider $f(x) = x^T Ax$. Calculate the partial derivatives $\dfrac{\partial f}{\partial x_i}$ for $i = 1, \ldots, n$.

(e) What is the gradient of f?

(f) Based on the solution to the above problem, do you think there is a matrix calculus equivalent of the "product rule" from calculus? Formulate this rule. That is, give a formula for $\nabla(g(x)^T h(x))$ where $g, h : \mathbf{R}^n \to \mathbf{R}^n$ when the functions are smooth so all derivatives are well-defined.

Verify this equation for the results you obtained in parts (d) + (e)

Hint: What are g and h if $g(x)^T h(x) = x^T Ax$?

2. Now we will verify your calculation by checking that the gradient $\nabla f(x)$ is tangent to $f(x)$ at the point x. Since we can only plot in two dimensions, we'll check that this is true by looking at how the function varies in a random direction v.
   Let $x \in \mathbf{R}^5$ and
   $$f(x) = \log\left(\sum_{i=1}^{5} e^{x_i}\right).$$
   (a) Pick a random point x and a random direction v. Plot $f(x+\alpha v)$ and $f(x) + \alpha(\nabla f(x))^T v$ for $\alpha \in [-1, 1]$. Repeat this for a few different v and a few different x. What do you observe? Submit a plot for at least one point x and two random directions v.

   (b) Now plot the same thing using $v = \nabla f(x)$. What do you observe? Submit at least one plot with $v = \nabla f(x)$.

   (c) In which direction v does the function f decrease the fastest? More formally, consider $\{v \mid \|v\| = 1\}$. For which v is
   $$\frac{d}{d\alpha} f(x+\alpha v)|_{\alpha=0}$$
   most negative?

3. *Average error of least squares.* In this problem, we consider whether a linear model is accurate, on average, for certain groups of examples. Given a data set $\mathcal{D} = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ with $x_i \in \mathbf{R}^d$, $y_i \in \mathbf{R}$ for $i = 1, \ldots, n$, define the ith prediction $\hat{y}_i(w)$ made by a linear model with parameters w to be
   $$\hat{y}_i(w) = x_i^T w.$$
   We say that a model with parameters w is accurate, on average, for the set of examples $S \subseteq \{1, \ldots, n\}$ if
   $$\sum_{i \in S} (y_i - \hat{y}_i(w)) = 0.$$
   (a) Let's suppose that we are fitting a model with an offset: $(x_i)_d$, the last entry of x_i, is equal to 1, for each $i = 1, \ldots, n$. We will compute w using least squares, by solving
   $$\text{minimize} \quad \sum_{i=1}^{n} (y_i - x_i^T w)^2$$
   with variable $w \in \mathbf{R}^d$. Show that the resulting linear model is accurate, on average, for the full set of examples $S = \{1, \ldots, n\}$.

(b) Let's suppose that we are again fitting a model with an offset, as above, so that $(x_i)_d = 1$ for each $i = 1, \ldots, n$. Let's also suppose that the first entry of each feature vector is Boolean: $(x_i)_1 \in \{0, 1\}$ for each $i = 1, \ldots, n$. We will again compute w using least squares, by solving

$$\text{minimize} \quad \sum_{i=1}^{n} (y_i - x_i^T w)^2$$

with variable $w \in \mathbf{R}^d$. Show that the resulting linear model is accurate, on average, for the set of examples $S_1 = \{i : (x_i)_1 = 1\}$ for which the Boolean attribute has value 1.

(c) Consider again the setting in the previous part. Show that the resulting linear model is accurate, on average, for the set of examples $S_0 = \{i : (x_i)_1 = 0\}$ for which the Boolean attribute has value 0.

(d) Suppose we want to make a model to predict the scores of students on the final exam. We have n students. The input x_i for each student is a vector of covariates representing things we know about the student; for example, one of the entries is a Boolean which takes value 1 if the student enjoyed homework 2, and 0 if the student hated homework 2; other covariates might include the student's grades on each previous homework assignment, the number of lectures the student attended, the number of times the student went to office hours and to section, the list of related classes the student had taken previously, *etc.*, all encoded using a variety of feature transformations into the numerical vector x_i. The output y is the student's score on the final exam.

We build a model (with an offset) using least squares to predict the student's score from these covariates. Is the model accurate, on average? Is it accurate on average for students who enjoyed homework 2? Is it accurate on average for students who hated homework 2?

(e) Suppose that students who enjoyed homework 2 earned higher exam scores, on average, than students who hated homework 2. What can you say, if anything, about the difference in the average predicted scores for these two groups (using the same model as above)?

4. *Data exploration.* See the Jupyter notebook `incomeTax.ipynb`.
In this problem, we will be exploring income tax data for the state of NY from 1999-2013. Income is declared to the internal revenue service (IRS) by each person or family, and taxes are calculated, in a document called an *income tax return.* Our goal will be to create a model to predict the average income tax paid per return.

(a) First, let's take a look at the data and generate some plots. Plot the number of returns in Tompkins County from each income class bracket over time on a single graph.

Plot the average income tax per return in Tompkins County (disregarding income class). What kind of plot (bar, scatter, histogram, . . .) did you choose to make? Why? *Hint: For the first plot, draw the plot for each income class, ignoring the rows with class 'Total'. For the second plot, add up the all income classes (except 'Total') to calculate the weighted average tax OR just plot the rows of 'Total'. The sum might differ from the number in 'total'; either answer is ok. Messy data!)*

(b) Continuing to look only at Tompkins County, fit a linear model, minimizing the square error, to predict `avg_tax` using the year. That is, your output space $\mathcal{Y} = \mathbf{R}$ is the average tax `avg_tax`, and the feature space $\mathcal{X} = \mathbf{R}$ is the year. Transform the input $x \in \mathcal{X}$ to include an offset term in the model: $\phi(x) = [x, 1]$. Call the coefficients from this model w^b.

(c) Now we'll change the feature space $\mathcal{X}$: the features will be the year and the `avg_tax` from the previous year (and offset).
Fit another model using these features. Interpret the coefficients w^{c} of this model. What do they mean?

Plot your model's prediction for each year against the real average tax return that year. Discuss how your model fits.

(d) Add two new features to your model. Each new feature could be a column, a transformation of a column, or a new column formed from another. The one requirement is that the prediction in year t depends only on data available by the end of year $t - 1$, for every year t.

State the feature space $\mathcal{X}$ for your new model, and why you think those features predict the data well. Fit your model and interpret the coefficients w^{d}.

(e) Compare the coefficients w^{c} and w^{d} in your models from part c and part d. Does the coefficient of `avg_tax` from the previous year differ in the two models? If so, how do you interpret this difference?

(f) Now we want to see how your model performs in predicting income tax in other counties. Define the mean error of a least squares model $h(x) = w^T x$ on a data set $\mathcal{D} = \{(x_1, y_1), \ldots, (x_n, y_n)\}$ to be the mean value of the least squares objective

$$\frac{1}{n}\sum_{i=1}^{n}(y_i - h(x_i))^2 = \frac{1}{n}\sum_{i=1}^{n}(y_i - w^T x_i)^2.$$

Consider the data from each county as a different data set: for example, $\mathcal{D}^{\text{Tompkins}}$, $\mathcal{D}^{\text{Brooklyn}}$, $\mathcal{D}^{\text{Queens}}$, etc. Only consider counties that have data for all the years from 1999-2013. Make sure to remove "fake" counties like `NYS Unclassified +`, `Residence Unknown ++`, and `Grand Total, Full-Year Resident`.

Apply the model you fit in part c, with coefficients w^{c}, to data from the other counties, and compute the error of this model on each data set. Plot a histogram of the mean errors the model makes.

Compare to the error of the model on the data from Tompkins County. Are they higher, or lower? Are there major outliers?

(g) Using the same features you chose in part d, fit a model to the data for each of the other counties. We'll call these models the county-specific models. In general, they will have coefficient vectors that differ from w^{d}. Plot a histogram of the mean errors the county-specific models make for their respective counties.

How does this error distribution compare to that of the model you fit on Tompkins County? Are the coefficients of the model about the same for each county, or do they differ significantly?

(h) If you wanted to predict the income tax in each county in future years, do you think the county-specific models or the Tompkins model would perform better? Why? What concerns might you have about each model?

(i) What other information would you want to use to make your model even better?

5. *Calibration.* How long did you spend on each problem in this homework assignment, and on the homework assignment, in total?