CMSC 25025 / STAT 37601

Machine Learning and Large Scale Data Analysis

HW 1

Due: Tuesday, March 28, 2023 (at 11:00 a.m.)

Homework submission.
Upload to Gradescope `HW1`: A pdf of the theoretical homework.
Upload the ipynb file for problems 4,5 to Gradescope `HW1 programming`.

Jupyter notebook guidelines.

1. Make sure most of your code is arranged in **functions.** One function per cell.
2. You then can use one or several cells for running the code on data with different parameter settings etc.
3. Make sure to have a markdown cell above each cell explaining what that cell is doing. Sometimes you will need some mathematical formulas and you can use latex in the markdown.
4. Remember that the variables you set in a non-function cell become global variables in the notebook.

Problems.

1. *Maximum likelihood* (15 points)

   Maximum likelihood is a method to estimate parameters of a distribution from observations. Let $f(x, \theta), \theta \in \Theta$ be a family of distributions. Assume $X_1, \ldots, X_n$ are i.i.d samples from $f(x, \theta^*)$. For any value of $\theta \in \Theta$ the log-likelihood is

   $$\ell(X_1, \ldots, X_n; \theta) = \ell(\mathbf{X}, \theta) = \sum_{i=1}^{n} \log f(X_i; \theta).$$

   This is minimized over Θ by solving the score equation

   $$\frac{\partial \ell(\theta)}{\partial \theta} = 0.$$

   The solution $\widehat{\theta}$ is call the *maximum likelihood* estimate of θ. In most situations we will deal with this has a unique solution that is indeed the maximum.

   (a) The Negative Binomial distribution with parameters r and p - $NB(r, p)$ - provides the number of independent Bernoulli(p) trials until the first r successes.

   $$P(X = k) = \binom{k + r - 1}{k} (1 - p)^k p^r.$$

(b) Let $X_1, \ldots, X_n$ be i.i.d $NB(r, p)$, show that $\sum_{i=1}^n X_i \sim NB(nr, p)$.

(c) Let $X_1, \ldots, X_n$ be independent draws from $NB(r, p)$ with r known. Write the log-likelihood $\ell(\mathbf{X}, p)$, derive the score equation and find $\widehat{p}$ that solves the score equation.

(d) Show that this estimate is biased i.e. $E(\widehat{p}) \neq p$.. Hint: try a slight modification of the ML estimate and show that it is unbiased.

(e) $X_1, \ldots, X_n \in R^d$ be i.i.d draws from the normal $N(\mu, \Sigma)$, with $\mu \in R^d$ and Σ a positive definite $d \times d$ matrix. Write the score equation for μ and solve it (note: the solution does not depend on Σ).

(f) Assume Σ is diagonal $\Sigma = diag(\sigma_1, \ldots, \sigma_d)$. Write the score equations for μ and $\sigma_1, \ldots, \sigma_d$. Denote by $\widehat{\mu}$ the maximum likelihood estimate that you obtained in part (b). Solve the score equations for $\widehat{\sigma}_i, i = 1, \ldots, d$.

(g) Assume $\Sigma_0 \in R^{d \times d}$ is positive definite and it is known and that $X_i \sim N(\mu, \alpha\Sigma_0)$, with $\theta = (\mu, \alpha)$ unknown. Write the score equation for α and solve for $\widehat{\alpha}$.

2. *Regression* (15 points)

Let X be an $n \times d$ design matrix in a linear regression setting with $X_{i,1} = 1, i = 1, \ldots, n$. Let $X_{(i)}$ be the i'th row of X. Assume $Y_i \sim N(X_{(i)} \cdot \beta, \sigma^2)$ are independent, with $\beta \in R^d$ and σ known.

(a) Show that the maximum likelihood estimate of β is given by, $\widehat{\beta} = (X^t X)^{-1} X^t Y$.

(b) The fitted values are defined to be $\widehat{\mathbf{y}} = X\widehat{\beta}$ where $\widehat{\mathbf{y}} = HY$ and

$$H = X(X^T X)^{-1} X^T.$$

assuming $n > d$ and $X^T X$ is nonsingular. The matrix H is called the "hat matrix." Define $\mathcal{L}$ to be the set of vectors that can be obtained as linear combinations of the columns of X. Show that the hat matrix satisfies the following properties:

i. $\widehat{y} = HY = X\widehat{\beta}$ are the least squares estimates.
ii. $HX = X$.
iii. H is symmetric: $H = H^T$.
iv. H is idempotent: $H^2 = H$.
v. $\widehat{\mathbf{y}} = H\mathbf{y}$ is the projection of $\mathbf{y}$ onto the column space $\mathcal{L}$.
vi. $\mathrm{rank}(X) = \mathrm{tr}(H) = d$.
vii. Let $e = Y - \widehat{y}$ be the vector of residuals. Show that $\sum_{i=1}^n e_i = 0$.

3. *Singular value decomposition* (15 points)

Let $X \in \mathbb{R}^{m \times n}$ have rank$(X) = r \leq \min(m, n)$ and let $X = U\Sigma V^\top$ be the SVD of A where $U \in \mathbb{R}^{m \times m}$, $V \in \mathbb{R}^{n \times n}$ are orthogonal matrices and $\Sigma \in \mathbb{R}^{m \times n}$ is the diagonal matrix of singular values, with $\Sigma_{ii} = \sigma_i,\ i = 1, \ldots, \min(m, n)$.

(a) Show that the columns of U are eigenvectors of $XX^\top$ and the columns of V are eigenvectors of $X^\top X$. Determine what are the corresponding eigenvalues in terms of $\sigma_1, ..., \sigma_r$.

(b) If $u_1, ..., u_m$ and $v_1, ..., v_n$ are the columns of U and V, show that $Xv_i = \sigma_i u_i$ and $X^\top u_i = \sigma v_i$.

(c) Express the Frobenius norm $\|X\|_F = \sqrt{\sum_{ij} X_{ij}^2}$ in terms of the singular values σ_i.

(d) The L2 operator norm of a matrix X is defined as: $\|X\| = \max_{v:|v|_2=1} |Xv|_2$. Show that the operatornorm $\|X\| = \sigma_1$

(e) Assuming X is square, express $|\det(X)|$ in terms of the singular values σ_i

(f) Assuming $X^\top X$ is invertible, express the hat matrix $H = X(X^\top X)^{-1}X^\top$ of linear regression in terms of the SVD.

4. *Self-fulfilling prophecies* (20 points)

We want to show what kind of issues can arise from clustering when it is applied to real people and affects their choices. In the sample code `diagonal_clusters.ipynb` you can see how to create plots.

(a) Simulate 1000 points from a bivariate normal distribution $N(\mu, \Sigma)$ with

$$\mu = 0, \Sigma = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}.$$

Plot the data. Pretend these two variables represent the features an app uses to characterizes people's preferences and subsequently cluster them into 3 clusters.

(b) Using the python package `sklearn.cluster.KMeans` fit a 3 cluster model to the data. Plot the three cluster centers and color the data points according to the clusters they are assigned. This is the assignment the app has chosen for this population.

(c) Now modify each point in the data to move 1% closer to its assigned cluster center. $x_i = .99 * x_i + .01 c_i$, where c_i is the cluster center assigned to the i'th data point. This corresponds to a tiny indirect effect of the choice of cluster on the features of the people in the sample. Now repeat the clustering on the modified data.

(d) Imagine the app repeats the clustering analysis every week based on the modified data. Repeat this process 50 times. Plot the original data cloud, and the final data cloud you obtained side by side. Describe what has happened to your original population of diverse individuals after a year (50 weeks).

5. *Presidential logorrhea* (35 points)

In this problem, you will analyze the lengths of the State of the Union addresses by all US presidents.

(a) The transcripts of all State of the Union addresses can be accessed in the `sou` subfolder of the `NLP` subfolder of the `LSDA_data` folder in the course `Modules` page. File name: `speeches.json`.

Once you download this to your machine you can load the speeches into python using the json package as follows:

```
import json
speeches=[]
with open('speeches.json') as f:
  for line in f:
    speeches.append(json.loads(line))
```

Each speech will now be a string in the list `speeches`.

Write Python code that parses each SOU address, finding end-of-sentence markers. Don't worry about being too precise about sentence boundaries—as a first approximation, you could find words ending in a period. (But what about "Mr."?) You can use the file `text_processing.ipynb` for some clues on how to process strings in python.

(b) Which President has the longest sentences on average? Which has the shortest sentences? Compute the median, 25% and 75% quantiles across all Presidents. What was the longest and shortest sentence ever spoken (or written) in a SOU?

(c) For each year, compute the number of sentences in the address, and the mean sentence length in words for that year. Plot these data and two linear regressions, one plot for the number of sentences by year, another for the average sentence length by year. Note that the definition of "word" and "sentence" is imprecise. You can experiment with different parsing rules, and see if the results change qualitatively. Describe the trends that you see, and give some explanation for them. You should compute the linear regressions directly—for example, you may use the linear algebra routine `numpy.linalg.solve` but do not use a package that computes the regression.

(d) Now, we want to fit two linear models for the number of words in a SOU versus year– one for the years 1790 to 1912, another for the years 1913 to the present **using one multiple linear regression model**.

i. How would you set up the multiple regression model?

ii. Explain the advantage of using multiple regression vs. estimating a separate linear regression for each group.

iii. Compute the estimated coefficients yourself (again, do not use a linear regression package) and plot the two lines you get. What trends do you see? Lookup the history of the State of the Union addresses (for example on Wikipedia) to explain the regressions.