The University of Texas at Austin
Department of Electrical and Computer Engineering

460J: Data Science Lab — Fall 2019

Lab Five

Caramanis/Dimakis Due: Tuesday October 8th, 3:00pm 2019.

Problem 1
Read Shannon's 1948 paper 'A Mathematical Theory of Communication'. Focus on pages 1-19 (up to Part II), the remaining part is more relevant for communication.
`http://math.harvard.edu/~ctm/home/text/others/shannon/entropy/entropy.pdf`
Summarize what you learned briefly (e.g. half a page).

Problem 2: Scraping, Entropy and ICML papers.

ICML – the International Conference on Machine Learning – is a top research conference in Machine learning. Scrape all the pdfs of all ICML 2019 papers from `http://proceedings.mlr.press/v97/`.

1. What are the top 10 common words in the ICML papers?
2. Let Z be a randomly selected word in a randomly selected ICML paper. Estimate the entropy of Z.
3. Synthesize a random paragraph using the marginal distribution over words.
4. (Optional) Synthesize a random paragraph using an n-gram model on words. Synthesize a random paragraph using any model you want. Top five synthesized text paragraphs win bonus!

Problem 3: Logistic Regression.

The following is a logistic regression problem using a real data set, made available by the authors of the book "Applied Regression and Muiltilevel Modeling" by Gelman and Hill.

Download the data from the book, which you can find here `http://www.stat.columbia.edu/~gelman/arm/software/`. In particular, we are interested in the `arsenic` data set. The file `wells.dat` contains data on 3,020 households in Bangladesh. For each family, the natural arsenic level of each well was measured. In addition, the distance to the nearest safest well was measured. Each family is also described by a feature that relates to their community involvement, and a feature that gives the education level of the head of household. We are interested in building a model that predicts whether the family decided to switch wells or not, based on being informed of the level of arsenic in the well. Thus the "label" for this problem is the binary vector that is the first column of the dataset, labeled "switch."

- Fit a logistic regression model using only an offset term and the distance to the nearest safe well.
- Plot your answer: that is, plot the probability of switching wells as a function of the distance to the nearest safe well.

- Interpreting logistic regression coefficients: Use the "rule-of-4" discussed in class on Thursday, to interpret the solution: what can you say about the change in the probability of switching wells, for every additional 100 meters of distance?
- Now solve a logistic regression incorporating the constant term, the distance and also arsenic levels. Report the coefficients
- Next we want to answer the question of which factor is more significant, distance, or arsenic levels? This is not a well specified question, since these two features have different units. One natural choice is to ask if after normalizing by the respective standard deviations of each feature, if moving one unit in one (normalized) feature predicts a larger change in probability of switching wells, than moving one unit in the other (also normalized) feature. Use this reasoning to answer the question.
- Now consider all the features in the data set. Also consider adding interaction terms among all features that have a large main effect. Use cross validation to build the best model you can (using your training set only), and then report the test error of your best model.[1]
- (Optional) Now also play around with ℓ_1 and ℓ_2 regularization, and try to build the most accurate model you can (accuracy computed on the test data).

Problem 4: Logistic Regression and CIFAR-10. In this problem you will explore the data set CIFAR-10, and you will use multinomial (multi-label) Logistic Regression to try to classify it. You will also explore visualizing the solution.

- (Optional) You can read about the CIFAR-10 and CIFAR-100 data sets here: `https://www.cs.toronto.edu/~kriz/cifar.html`.
- (Optional) OpenML curates a number of data sets. You will use a subset of CIFAR-10 provided by them. Read here for a description: `https://www.openml.org/d/40926`.
- Use the `fetch_openml` command from `sklearn.datasets` to import the CIFAR-10-Small data set.
- Figure out how to display some of the images in this data set, and display a couple. While not high resolution, these should be recognizable if you are doing it correctly.
- There are 20,000 data points. Do a train-test split on 3/4 - 1/4.
- You will run multi-class logistic regression on these using the cross entropy loss. You have to specify this specifically (`multi_class='multinomial'`). Use cross validation to see how good your accuracy can be. In this case, cross validate to find as good regularization coefficients as you can, for ℓ_1 and ℓ_2 regularization (called penalties), which are naturally supported in `sklearn.linear_model.LogisticRegression`. I recommend you use the solver `saga`.
- Report your training and test loss from above,
- How sparse can you make your solutions without deteriorating your testing error too much? Here, I am asking you to try to obtain a sparse solution that has test accuracy that is close to the best solution you found.

[1]Note that since you have essentially unlimited access to your test set, this opens the door for massive overfitting. In contrast, Kaggle competitions try to mollify this by giving you only limited access to the test set.

Problem 5: Multi-class Logistic Regression – Visualizing the Solution.

You will repeat the previous problem but for the MNIST data set which you will find here: `https://www.openml.org/d/554`. MNIST is a data set of handwritten digits, and is considered one of the "easiest" image recognition problems in computer vision. We will see here how well logistic regression does, as you did above on the CIFAR-10 subset. In addition, we will see that we can visualize the solution, and that in connection to this, sparsity can be useful.

- Use the `fetch_openml` command from `sklearn.datasets` to import the MNIST data set,
- Choose a reasonable train-test split, and again run multi-class logistic regression on these using the cross entropy loss, as you did above. Try to optimize the hyperparameters.
- Report your training and test loss from above,
- Choose an ℓ_1 regularizer (penalty), and see if you can get a sparse solution with almost as good accuracy.
- Note that in Logistic Regression, the coefficients returned (i.e., the $\boldsymbol{\beta}$'s) are the same dimension as the data. Therefore we can pretend that the coefficients of the solution are an image of the same dimension, and plot it. Do this for the 10 sets of coefficients that correspond to the 10 classes. You should observe that, at least for the sparse solutions, these "kind of" look like the digits they are classifying.