PROBLEM SET 1: APPLIED MATHEMATICS 216

Due: Monday January 29 in class.

Goals for the week.

(1) Get some experience with coding in python. For those of you who do not know python, this is likely to be the main challenge in carrying out the homework problems below. Start by looking for commands (plotting, etc.) on stackoverflow. Then ask your classmates and us for help.
(2) Understand qualitatively how different types of norms can influence the results of regression problems, especially the robustness to outliers
(3) Experiment with different ways of clustering high dimensional datasets.

Before you start, make sure you have a python environment set up, that the packages `sklearn`, `keras`, `numpy` and `scipy` are installed, as well as `jupyter` notebooks. For your reference, the jupyter notebook that Yohai used in class is available online on the course web site.

Problems.

(1) *L1 vs L2 (vs L5!)*
   (a) Generate data according to $y = 4 + 15x + 17x^2$, where you draw 30 values of x uniformly distributed in the interval $[-20, 10]$, and add Gaussian random noise to the data (with mean zero and variance 0.5). Does the magnitude of the noise strength affect any conclusions that you might draw?
   (b) Fit the noisy data to a quadratic polynomial, and compare the fits for L_1, L_2 and L_5 (for varying magnotudes of noise?)
   (c) Add an outlier to the curve by perturbing one of the points significantly. Redo the optimization and observe whether or not the perturbation affects the fit. How large does the perturbation have to be for the fit to be significantly disrupted?

(2) *Compressed Sensing*
   (a) Generate an $n \times m$ matrix A full of random numbers. In class we chose the random numbers to be normally distributed. Investigate which random number generators that `numpy.random` has available and do this problem using a matrix made out of several of them. (This will both introduce you to the random number generators and make you contemplate whether the choice of random number generator matters for these algorithms to

work. Try your hardest to find a random number generator that causes the algorithm to fail.)

(b) In class we compared L_1 and L_2. Also consider the norm that is a linear combination of L_1 and L_2, so that $L_{new} = L_2 + \beta L_1$, where β is a free parameter.

(c) Carry out the exercise leading to signal reconstruction in L_1, L_2 and L_{new}. Comment on whether there is any change of behavior in L_{new} as a function of the parameter β. Is there a value of β where the character of the solution changes from L_1 like to L_2 like? Why?

(3) *Images!* In class we considered various ways of looking at the MNIST digits: random projection; PCA and Kmeans clustering. Here we will repeat a variant of this exercise on a classical image dataset. For your information, there are many repositories of datasets that you can play with – the two that we have used so far are (a) sklearn's `fetch_mldata`, which extracts data from `mrldata.org` and has a plethora of different datasets of different types, and (b) Keras's `datasets`. Below we recommend you download `CIFAR10` from Keras (`https://keras.io/datasets/#cifar10-small-image-classification`).

(a) The training set for `CIFAR10` consists of 50,000 images over 10 categories. Note that the images are 32 x 32 with 3 colors. Carry out the random projections we did in class for MNIST on this dataset.

(b) Carry out PCA

(c) Carry out Kmeans clustering.

(d) Train the neural network to classify this data. (use the same architecture for the one we described in class–with two stacked dense layers.)

(e) Compare the performance of Kmeans clustering to the neural network, as we did in class.

(4) *K-means Clustering*: Looking at the documentation or in some other source, outline the algorithm for that is used in scikit's K-means clustering algorithm. Briefly discuss the limitations of the algorithm: We saw that it worked only modestly well for the MNIST dataset of handwritten digits. Give an example of a dataset where the algorithm will work well.

(5) *Random Matrix theory*: Consider a data matrix X of size $p \times n$, where p is the number of variables and n is the number of experiments. We might choose $p = 300$ and $n = 50$ so there are 50 observations of 300 variables.

(a) Construct such a matrix of random numbers chosen from a gaussian distribution with zero mean and unit variance.

(b) Now compute the $p \times p$ *covariance matrix*, defined as

$$C = \frac{1}{n} X X^T.$$

(c) Using numpy, compute the eigenvalues of this matrix. Plot a histogram of the eigenvalue distribution.

(d) Compare the histogram with the so-called Marcenko-Pastur law

$$\rho_{\mathrm{MP}}(\lambda) = \frac{\sqrt{(\lambda_+ - \lambda)(\lambda - \lambda_-)}}{\lambda 2\pi p/n}, \quad \text{where} \quad \lambda_\mp = (1 \mp \sqrt{p/n})^2$$