Homework 1 PY895 (Fall 2016)

Instructor: Pankaj Mehta (323 SCI, pankajm@bu.edu)
Problems: Due Thursday Oct 6th

1. We argued that as the number of data points N approaches infinity, the training and test errors must approach each other. In fact, we argued that the difference between these is controlled by a large-deviation function. We will calculate one such Large-Deviation function in this problem.

(a) Consider drawing numbers n random numbers, X_i $(i = 1 \dots n)$ from a Gaussian distribution

$$P(X_i = x_i) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x_i-\mu)^2}{2\sigma^2}}$$

We can form the sample mean

$$S_n = \frac{1}{n}\sum_{i=1}^{N} X_i$$

Show that for $n \gg 1$, that

$$P(S_n = s) \approx e^{-nI_n(s)}$$

with

$$I(s) = \frac{(s-\mu)^2}{2\sigma^2}$$

(b) Repeat the problem if the variables are distributed according to an exponential distribution

$$P(X_i = x_i) = \frac{1}{\mu} e^{\frac{-x_i}{u}}$$

(c) Now consider tossing a fair coin n times. If on the i-the toss, the coin lands heads, $X_i = 1$, if it lands tails $X_i = -1$. Calculate $I(s)$ now. The expression should be familiar to physicists. Explain your answer by thinking about the magnetization of system of non-interacting spins.

2. In this problem, we will use some statistical mechanics intuitions to think about the generalization errors of perceptrons. We will employ what is called the "annealed approximation" in this problem (calculate error after averaging over all possible fitted models). The real error should really be calculated using the "quenched disorder" (calculate error for different models and then average this error). However, the annealed approximation yields reasonable results for this problem. For those versed in the statistical physics of disordered systems, the training data play the role of disorder. The quenched average can be solved using Replica tricks or the Cavity Method.

Let us introduce some basic notation. Let us denote the by **S** a N-dimensional binary input vector where each $S_i = \pm 1$. Let us denote the real-valued weights of the perceptron by $\mathbf{J} = (J_1, J_2, \ldots, J_N)$. The binary classification of the perceptron (± 1) for a given input vector $\mathbf{S}_j$ is given by

$$y_j(\mathbf{J}) = \mathrm{sign}(\mathbf{J} \cdot \mathbf{x})$$

Since the norms of **J** and **S** do not effect the classification task, it is useful to normalize the coefficients and inputs so that

$$\begin{aligned} ||\mathbf{J}||_2^2 &= \sum_i^N J_i^2 = N \\ ||\mathbf{S}||_2^2 &= \sum_i^N S_i^2 = N \end{aligned}$$

With this normalization, both sets of vectors sit on a N-dimensional sphere (N-sphere) with radius $\sqrt{N}$. With this normalization, the space of possible inputs are vector on the N-dimensional sphere.

Imagine that the true data is generated using a "teacher" perceptron **T**. In other words, the data label for point j is given by $y_j = \mathrm{sign}(\mathbf{T} \cdot \mathbf{S}) = \mathrm{sign}(\mathbf{T} \cdot \mathbf{S})$. We will ask how well a "student perceptron" **J** that we are training can reproduce the results. Let us introduce the so-called teacher-student overlap

$$R = \frac{\mathbf{J} \cdot \mathbf{T}}{N} = \frac{1}{N} \sum_i T_i J_i \tag{1}$$

(a) Show that R is nothing more than the cosine of the angle between **T** and **J**. Draw a picture.

(b) Using the fact that the decision lines are orthogonal to the vector **T** and **J**, show that the generalization error $\epsilon = \theta/\pi$, where θ is the angle between **T** and **J**. In this way show that

$$\epsilon = \frac{1}{\pi} \arccos R \tag{2}$$

We will now train the student as follows. Consider all vectors **J** which score on the examples exactly like the teacher. The set of these *compatible* students is called the *version space.* We ask for the generalization error of a vector **J** drawn at random from this version space and characterize the *typical performance of* *a compatible student.*

(c)Show that all couplings $\mathbf{J}$ with overlap R, the chance of producing the same output on a randomly chosen input vector as the teacher is $1-\epsilon$. Let $\Omega_0(\epsilon)$ be volume of coupling vectors $\mathbf{J}$ with overlap R. Show that in the limit $N \to \infty$ that

$$\Omega_0(\epsilon) \sim \exp\left(\frac{N}{2}[1 + \ln 2\pi + \ln \sin^2(\pi\epsilon)]\right). \tag{3}$$

(Hint: Use saddle point approximation in the N-dimensional integral over all possible J. This calculation is very similar to derivation of Maxwell distribution).

(d)Argue that after we have trained on p examples, that the average volume, $\Omega_p(\epsilon)$, of compatible students with generalization error ϵ is given by

$$\Omega_p(\epsilon) = \Omega_0(\epsilon)(1-\epsilon)^p \tag{4}$$

(e) For the scaling $p = \alpha N$, show that in the limit $N \to \infty$ show that the generalization error is given by

$$\epsilon(\alpha) = \arg\max\left[\frac{1}{2}\ln\sin^2(\pi\epsilon) + \alpha\ln(1-\epsilon)\right]. \tag{5}$$

What is the error in $\alpha = 0$ and $\alpha \to \infty$? Make a graph of how the typical error ϵ as a function of α.

3. Derive the the generalization of the Bias-Variance tradeoff when data is noisy: $y_i = f(x_i) + \eta_i$ where η_i are white noise (independent variables drawn from Gaussian with mean zero and variance one).

4. Computational Exercise. Analyze the following datasets with Ridge Regression, LASSO, and Elastic Nets. We start by constructing the dataset with $n = 200$ examples each with dimension $d = 90$.

1. Construct a random design matrix X with dimensions $n \times d$.
2. Let us construct the true weighs $\mathbf{w}_{true}$ so that the first 20 components are $+20$ or -20 arbitrarily, and all other components are zero.
3. Let $\mathbf{y}$ be the response vector of length n with $\mathbf{y} = X\mathbf{w} + \eta$, where η is a $n \times 1$ random noise vector with mean zero and standard deviation 0.1.

Run ridge regression, LASSO, and Elastic Nets on this dataset. Choose the regularizer α that minimizes the square loss on the validation set. For the chosen α, examine the model coefficients. Report on how many components with true value 0 have been estimated to be non-zero, and vice-versa (dont worry if non-zero values are accurate). Now choose a small threshold (say 10^{-3} or smaller), count anything with magnitude smaller than the threshold as zero, and repeat the report. Explain your results. Turn this problem in as a Python notebook.

5. (Stolen from David Rosenberg's Machine Learning Class at NYU) In this problem, we will examine and compare the behavior of the Lasso and ridge regression in the case of an exactly repeated feature. That is, consider the design matrix $X \in R^{n\times d}$, where $X_{\cdot;i} = X_{\cdot;j}$ for some i and j, where $X_{\cdot;i}$ is the i-th th column of X. We will see that ridge regression divides the weight equally among identical features, while Lasso divides the weight arbitrarily. In the final part to this problem, we will consider what changes when $X_{\cdot;i}$ and $X_{\cdot;j}$ are highly correlated (e.g. exactly the same except for some small random noise) rather than exactly the same.

1. Derive the relation between w_i and w_j, the i-th and the j-th components of the optimal weight vector obtained by solving the Lasso optimization problem. [Hint: Assume that in the optimal solution, $w_i = a$ and $w_j = b$. First show that a and b must have the same sign. Then, using this result, rewrite the optimization problem to derive a relation between a and b.]

2. Derive the relation between w_i and w_j , the i-th and the j-th components of the optimal weight vector obtained by solving the ridge regression optimization problem.

3. What do you think would happen with Lasso and ridge when $X_{\cdot;i}$ and $X_{\cdot;j}$ are highly correlated, but not exactly the same. You may investigate this experimentally by adding small Gaussian noise to each component in the column.