CS 229 Autumn 2016
Problem Set#3:Theory & Unsupervised learning

Due Wednesday, November 16 at 11:00 am on Gradescope.

Notes: (1) These questions require thought, but do not require long answers. Please be as concise as possible. (2) If you have a question about this homework, we encourage you to post your question on our Piazza forum, at `https://piazza.com/stanford/autumn2016/cs229`. (3) If you missed the first lecture or are unfamiliar with the collaboration or honor code policy, please read the policy on Handout #1 (available from the course website) before starting work. (4) For problems that require programming, please include in your submission a printout/scan of your code (with comments) and any figures that you are asked to plot.

If you are skipping a question, please include it on your PDF/photo, but leave the question blank and tag it appropriately on Gradescope. This includes extra credit problems. If you are scanning your document by cellphone, please check the Piazza forum for recommended cellphone scanning apps and best practices.

1. **[23 points] Uniform convergence**

   You are hired by CNN to help design the sampling procedure for making their electoral predictions for the next presidential election in the (fictitious) country of Elbania.

   The country of Elbania is organized into states, and there are only two candidates running in this election: One from the Elbanian Democratic party, and another from the Labor Party of Elbania. The plan for making our electorial predictions is as follows: We'll sample m voters from each state, and ask whether they're voting democrat. We'll then publish, for each state, the estimated fraction of democrat voters. In this problem, we'll work out how many voters we need to sample in order to ensure that we get good predictions with high probability.

   One reasonable goal might be to set m large enough that, with high probability, we obtain uniformly accurate estimates of the fraction of democrat voters in every state. But this might require surveying very many people, which would be prohibitively expensive. So, we're instead going to demand only a slightly lower degree of accuracy.

   Specifically, we'll say that our prediction for a state is "highly inaccurate" if the estimated fraction of democrat voters differs from the actual fraction of democrat voters within that state by more than a tolerance factor γ. CNN knows that their viewers will tolerate some small number of states' estimates being highly inaccurate; however, their credibility would be damaged if they reported highly inaccurate estimates for too many states. So, rather than trying to ensure that all states' estimates are within γ of the true values (which would correspond to no state's estimate being highly inaccurate), we will instead try only to ensure that the number of states with highly inaccurate estimates is small.

   To formalize the problem, let there be n states, and let m voters be drawn IID from each state. Let the actual fraction of voters in state i that voted democrat be ϕ_i. Also let X_{ij} $(1 \leq i \leq n, 1 \leq j \leq m)$ be a binary random variable indicating whether the j-th randomly chosen voter from state i voted democrat:

$$X_{ij} = \begin{cases} 1 & \text{if the } j^{th} \text{ example from the } i^{th} \text{ state voted democrat} \\ 0 & \text{otherwise} \end{cases}$$

We assume that the voters correctly disclose their vote during the survey. Thus, for each value of i, we have that X_{ij} are drawn IID from a Bernoulli(ϕ_i) distribution. Moreover, the X_{ij}'s (for all i, j) are all mutually independent.

After the survey, the fraction of democrat votes in state i is estimated as:

$$\hat{\phi}_i = \frac{1}{m}\sum_{j=1}^{m} X_{ij}$$

Also, let $Z_i = 1\{|\hat{\phi}_i - \phi_i| > \gamma\}$ be a binary random variable that indicates whether the prediction in state i was highly inaccurate.

(a) Let ψ_i be the probability that $Z_i = 1$. Using the Hoeffding inequality, find an upper bound on ψ_i.

(b) In this part, we prove a general result which will be useful for this problem. Let V_i and W_i $(1 \le i \le k)$ be Bernoulli random variables, and suppose

$$\mathrm{E}[V_i] = P(V_i = 1) \le P(W_i = 1) = \mathrm{E}[W_i] \qquad \forall i \in \{1, 2, \ldots k\}$$

Let the V_i's be mutually independent, and similarly let the W_i's also be mutually independent. Prove that, for any value of t, the following holds:

$$P\left(\sum_{i=1}^{k} V_i > t\right) \le P\left(\sum_{i=1}^{k} W_i > t\right)$$

[Hint: One way to do this is via induction on k. If you use a proof by induction, for the base case $(k = 1)$, you must show that the inequality holds for $t < 0$, $0 \le t < 1$, and $t \ge 1$.]

(c) The fraction of states on which our predictions are highly inaccurate is given by $\overline{Z} = \frac{1}{n}\sum_{i=1}^{n} Z_i$. Prove a reasonable closed form upper bound on the probability $P(\overline{Z} > \tau)$ of being highly inaccurate on more than a fraction τ of the states.

[Note: There are many possible answers, but to be considered reasonable, your bound must decrease to zero as $m \to \infty$ (for fixed n and $\tau > 0$). Also, your bound should either remain constant or decrease as $n \to \infty$ (for fixed m and $\tau > 0$). It is also fine if, for some values of τ, m and n, your bound just tells us that $P(\overline{Z} > \tau) \le 1$ (the trivial bound).]

2. **[15 points] More VC dimension**

Let the domain of the inputs for a learning problem be $\mathcal{X} = \mathbb{R}$. Consider using hypotheses of the following form:

$$h_\theta(x) = 1\{\theta_0 + \theta_1 x + \theta_2 x^2 + \cdots + \theta_d x^d \ge 0\},$$

and let $\mathcal{H} = \{h_\theta : \theta \in \mathbb{R}^{d+1}\}$ be the corresponding hypothesis class. What is the VC dimension of $\mathcal{H}$? Justify your answer.

[Hint: You may use the fact that a polynomial of degree d has at most d real roots. When doing this problem, you should not assume any other non-trivial result (such as that the VC dimension of linear classifiers in d-dimensions is $d + 1$) that was not formally proved in class.]

3. **[12 points] MAP estimates and weight decay**

   Consider using a logistic regression model $h_\theta(x) = g(\theta^T x)$ where g is the sigmoid function, and let a training set $\{(x^{(i)}, y^{(i)}); i = 1, \ldots, m\}$ be given as usual. The maximum likelihood estimate of the parameters θ is given by

$$\theta_{\mathrm{ML}} = \arg\max_\theta \prod_{i=1}^{m} p(y^{(i)}|x^{(i)}; \theta).$$

   If we wanted to regularize logistic regression, then we might put a Bayesian prior on the parameters. Suppose we chose the prior $\theta \sim \mathcal{N}(0, \tau^2 I)$ (here, $\tau > 0$, and I is the $n+1$-by-$n+1$ identity matrix), and then found the MAP estimate of θ as:

$$\theta_{\mathrm{MAP}} = \arg\max_\theta p(\theta) \prod_{i=1}^{m} p(y^{(i)}|x^{(i)}, \theta)$$

   Prove that

$$||\theta_{\mathrm{MAP}}||_2 \leq ||\theta_{\mathrm{ML}}||_2$$

   [Hint: Consider using a proof by contradiction.]
   Remark. For this reason, this form of regularization is sometimes also called **weight decay**, since it encourages the weights (meaning parameters) to take on generally smaller values.

4. **[15 points] KL divergence and Maximum Likelihood**

   The Kullback-Leibler (KL) divergence between two discrete-valued distributions $P(X), Q(X)$ is defined as follows:[1]

$$KL(P\|Q) = \sum_x P(x) \log \frac{P(x)}{Q(x)}$$

   For notational convenience, we assume $P(x) > 0, \forall x$. (Otherwise, one standard thing to do is to adopt the convention that "$0 \log 0 = 0$.") Sometimes, we also write the KL divergence as $KL(P||Q) = KL(P(X)||Q(X))$.

   The KL divergence is an assymmetric measure of the distance between 2 probability distributions. In this problem we will prove some basic properties of KL divergence, and work out a relationship between minimizing KL divergence and the maximum likelihood estimation that we're familiar with.

   (a) Nonnegativity. Prove the following:

$$\forall P, Q \quad KL(P\|Q) \geq 0$$

   and

$$KL(P\|Q) = 0 \quad \text{if and only if } P = Q.$$

[1] If P and Q are densities for continuous-valued random variables, then the sum is replaced by an integral, and everything stated in this problem works fine as well. But for the sake of simplicity, in this problem we'll just work with this form of KL divergence for probability mass functions/discrete-valued distributions.

[Hint: You may use the following result, called **Jensen's inequality**. If f is a convex function, and X is a random variable, then $E[f(X)] \geq f(E[X])$. Moreover, if f is strictly convex (f is convex if its Hessian satisfies $H \geq 0$; it is *strictly* convex if $H > 0$; for instance $f(x) = -\log x$ is strictly convex), then $E[f(X)] = f(E[X])$ implies that $X = E[X]$ with probability 1; i.e., X is actually a constant.]

(b) **Chain rule for KL divergence.** The KL divergence between 2 conditional distributions $P(X|Y), Q(X|Y)$ is defined as follows:

$$KL(P(X|Y)\|Q(X|Y)) = \sum_y P(y)\left(\sum_x P(x|y)\log\frac{P(x|y)}{Q(x|y)}\right)$$

This can be thought of as the expected KL divergence between the corresponding conditional distributions on x (that is, between $P(X|Y=y)$ and $Q(X|Y=y)$), where the expectation is taken over the random y.

Prove the following chain rule for KL divergence:

$$KL(P(X,Y)\|Q(X,Y)) = KL(P(X)\|Q(X)) + KL(P(Y|X)\|Q(Y|X)).$$

(c) **KL and maximum likelihood.**

Consider a density estimation problem, and suppose we are given a training set $\{x^{(i)}; i = 1, \ldots, m\}$. Let the empirical distribution be $\hat{P}(x) = \frac{1}{m}\sum_{i=1}^{m} 1\{x^{(i)} = x\}$. ($\hat{P}$ is just the uniform distribution over the training set; i.e., sampling from the empirical distribution is the same as picking a random example from the training set.)

Suppose we have some family of distributions P_θ parameterized by θ. (If you like, think of $P_\theta(x)$ as an alternative notation for $P(x;\theta)$.) Prove that finding the maximum likelihood estimate for the parameter θ is equivalent to finding P_θ with minimal KL divergence from $\hat{P}$. I.e. prove:

$$\arg\min_\theta KL(\hat{P}\|P_\theta) = \arg\max_\theta \sum_{i=1}^{m} \log P_\theta(x^{(i)})$$

Remark. Consider the relationship between parts (b-c) and multi-variate Bernoulli Naive Bayes parameter estimation. In the Naive Bayes model we assumed P_θ is of the following form: $P_\theta(x,y) = p(y)\prod_{i=1}^{n} p(x_i|y)$. By the chain rule for KL divergence, we therefore have:

$$KL(\hat{P}\|P_\theta) = KL(\hat{P}(y)\|p(y)) + \sum_{i=1}^{n} KL(\hat{P}(x_i|y)\|p(x_i|y)).$$

This shows that finding the maximum likelihood/minimum KL-divergence estimate of the parameters decomposes into $2n+1$ independent optimization problems: One for the class priors $p(y)$, and one for each of the conditional distributions $p(x_i|y)$ for each feature x_i given each of the two possible labels for y. Specifically, finding the maximum likelihood estimates for each of these problems individually results in also maximizing the likelihood of the joint distribution. (If you know what Bayesian networks are, a similar remark applies to parameter estimation for them.)

5. **[20 points] K-means for compression**

In this problem, we will apply the K-means algorithm to lossy image compression, by reducing the number of colors used in an image.

We will be using the following files:

- `http://cs229.stanford.edu/ps/ps3/mandrill-small.tiff`
- `http://cs229.stanford.edu/ps/ps3/mandrill-large.tiff`

The `mandrill-large.tiff` file contains a 512x512 image of a mandrill represented in 24-bit color. This means that, for each of the 262144 pixels in the image, there are three 8-bit numbers (each ranging from 0 to 255) that represent the red, green, and blue intensity values for that pixel. The straightforward representation of this image therefore takes about $262144 \times 3 = 786432$ bytes (a byte being 8 bits). To compress the image, we will use K-means to reduce the image to $k = 16$ colors. More specifically, each pixel in the image is considered a point in the three-dimensional (r, g, b)-space. To compress the image, we will cluster these points in color-space into 16 clusters, and replace each pixel with the closest cluster centroid.

Follow the instructions below. Be warned that some of these operations can take a while (several minutes even on a fast computer)![2]

(a) Start up MATLAB, and type `A = double(imread('mandrill-large.tiff'));` to read in the image. Now, `A` is a "three dimensional matrix," and `A(:,:,1)`, `A(:,:,2)` and `A(:,:,3)` are 512x512 arrays that respectively contain the red, green, and blue values for each pixel. Enter `imshow(uint8(round(A)));` to display the image.

(b) Since the large image has 262144 pixels and would take a while to cluster, we will instead run vector quantization on a smaller image. Repeat (a) with `mandrill-small.tiff`. Treating each pixel's (r, g, b) values as an element of $\mathbb{R}^3$, run K-means[3] with 16 clusters on the pixel data from this smaller image, iterating (preferably) to convergence, but in no case for less than 30 iterations. For initialization, set each cluster centroid to the (r, g, b)-values of a randomly chosen pixel in the image.

(c) Take the matrix `A` from `mandrill-large.tiff`, and replace each pixel's (r, g, b) values with the value of the closest cluster centroid. Display the new image, and compare it visually to the original image. Hand in all your code and a printout of your compressed image (printing on a black-and-white printer is fine).

(d) If we represent the image with these reduced (16) colors, by (approximately) what factor have we compressed the image?

[2] In order to use the `imread` and `imshow` commands in octave, you have to install the Image package from octave-forge. This package and installation instructions are available at: `http://octave.sourceforge.net`

[3] Please implement K-means yourself, rather than using built-in functions from, e.g., MATLAB or octave.