Homework #3

CSE 446: Machine Learning
Prof. Sewoong Oh
Due: **Thursday** 11/21/2019 11:59 PM
100 points

Please review all homework guidance posted on the website before submitting to Gradescope. Please provide succinct answers along with succinct reasoning for all your answers. Points may be deducted if long answers demonstrate a lack of clarity. Similarly, when discussing the experimental results, concisely create tables and/or figures when appropriate to organize the experimental results. In other words, all your explanations, tables, and figures for any particular part of a question must be grouped together.

Classification

1. *[5 points]* Suppose we run the Perceptron algorithm with $\mathbf{w}$ initialized to be an arbitrary unit vector. Suppose that the algorithm is then given the same vector $\mathbf{x}$ (whose label is 1) over and over again. How many mistakes can it make in the worst case? Express your answer in terms of $||\mathbf{x}||_2$. (Hint: derive the result from first principles, do not attempt to use the general result we proved in class).

2. Consider using a linear decision boundary for classification (labels in $\{-1, 1\}$) of the form $\mathbf{w} \cdot \mathbf{x} = 0$ (i.e., no offset). Now consider the following loss function evaluated at a data point $(\mathbf{x}, y)$ which is a variant on the hinge loss.

$$\ell((\mathbf{x}, y), \mathbf{w}) = \max(0, -y(\mathbf{w} \cdot \mathbf{x})).$$

a. *[2 points]* Given a dataset of $(\mathbf{x}_i, y_i)$ pairs, write down a single step of subgradient descent with a step size of η if we are trying to minimize

$$\frac{1}{n} \sum_{i=1}^{n} \ell((\mathbf{x}_i, y_i), \mathbf{w})$$

for $\ell(\cdot)$ defined as above. That is, given a current iterate $\widetilde{\mathbf{w}}$ what is an expression for the next iterate?

b. *[2 points]* Use what you derived to argue that the Perceptron (as formulated in class) can be viewed as implementing SGD applied to the loss function just described (for what value of η)?

c. *[1 points]* Suppose your data was drawn iid and that there exists a $\mathbf{w}^*$ that separates the two classes perfectly. Provide an explanation for why hinge loss is generally preferred over the loss given above.

3. We've talked a lot about binary classification, but what if we have $k > 2$ classes, like the 10 digits of MNIST? Concretely, suppose you have a dataset $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$ where $\mathbf{x}_i \in \mathbb{R}^d$ and $y_i \in \{1, \ldots, k\}$. Like in our least squares classifier of homework 1 for MNIST, we will assign a separate weight vector $\mathbf{w}^{(\ell)}$ for each class $\ell = 1, \ldots, k$; let $W = [\mathbf{w}^{(1)}, \ldots, \mathbf{w}^{(k)}] \in \mathbb{R}^{d \times k}$. We can generalize the binary classification probabilistic model to multiple classes as follows: let

$$\mathbb{P}_W(y_i = \ell | W, \mathbf{x}_i) = \frac{\exp(\mathbf{w}^{(\ell)} \cdot \mathbf{x}_i)}{\sum_{j=1}^{k} \exp(\mathbf{w}^{(j)} \cdot \mathbf{x}_i)}$$

The negative log-likelihood function is equal to

$$\mathcal{L}(W) = -\sum_{i=1}^{n} \sum_{\ell=1}^{k} \mathbf{1}\{y_i = \ell\} \log \left(\frac{\exp(\mathbf{w}^{(\ell)} \cdot \mathbf{x}_i)}{\sum_{j=1}^{k} \exp(\mathbf{w}^{(j)} \cdot \mathbf{x}_i)} \right)$$

Define the $\texttt{softmax}(\cdot)$ operator to be the function that takes in a vector $\theta \in \mathbb{R}^d$ and outputs a vector in $\mathbb{R}^d$ whose ith component is equal to $\frac{\exp(\theta_i)}{\sum_{j=1}^d \exp(\theta_j)}$. Clearly, this vector is nonnegative and sums to one. If for any i we have $\theta_i \gg \max_{j \neq i} \theta_j$ then $\texttt{softmax}(\theta)$ approximates $\mathbf{e}_i$, a vector of all zeros with a one in the ith component. For each y_i let $\mathbf{y}_i$ be the one-hot encoding of y_i (i.e., $\mathbf{y}_i \in \{0,1\}^k$ is a vector of all zeros aside from a 1 in the y_ith index).

a. *[5 points]* If $\widehat{\mathbf{y}}_i^{(W)} = \texttt{softmax}(W^\top \mathbf{x}_i)$, show that $\nabla_W \mathcal{L}(W) = -\sum_{i=1}^n \mathbf{x}_i(\mathbf{y}_i - \widehat{\mathbf{y}}_i^{(W)})^\top$.

b. *[5 points]* Recall problem 6 of Homework 1 (*Ridge Regression on MNIST*) and define $J(W) = \frac{1}{2}\sum_{i=1}^n \|\mathbf{y}_i - W^\top \mathbf{x}_i\|_2^2$. If $\widetilde{\mathbf{y}}_i^{(W)} = W^\top \mathbf{x}_i$ show that $\nabla_W J(W) = -\sum_{i=1}^n \mathbf{x}_i(\mathbf{y}_i - \widetilde{\mathbf{y}}_i^{(W)})^\top$. Comparing the least squares linear regression gradient step of this part to the gradient step of minimizing the negative log likelihood of the logistic model of part a may shed light on why we call this classification problem *logistic regression*.

c. *[15 points]* Using the original representations of the MNIST flattened images $\mathbf{x}_i \in \mathbb{R}^d$ ($d = 28 \times 28 = 764$) and all $k = 10$ classes, implement gradient descent (or stochastic gradient descent) for both $J(W)$ and $\mathcal{L}(W)$ and run until convergence on the training set of MNIST. For each of the two solutions, report the classification accuracy of each on the training and test sets using the most natural $\arg\max_j \mathbf{e}_j W^\top \mathbf{x}_i$ classification rule.

Kernels

4. *[5 points]* Suppose that our inputs are one dimensional and that our feature map is infinite dimensional: $\phi(x)$ is a vector whose ith component is

$$\frac{1}{\sqrt{i!}} e^{-\frac{x^2}{2}} x^i.$$

for all nonnegative integers i. (Thus, ϕ is an infinite dimensional vector.) Show that $K(x, x') = e^{-\frac{(x-x')^2}{2}}$ is a kernel function for this feature map, i.e.,

$$\phi(x) \cdot \phi(x') = e^{-\frac{(x-x')^2}{2}}.$$

Hint: Use the Taylor expansion of e^z. (This is the one dimensional version of the Gaussian (RBF) kernel).

5. This problem will get you familiar with kernel ridge regression using the polynomial and RBF kernel. First let's generate some data. Let $n = 30$ and $f_*(x) = 4\sin(\pi x)\cos(6\pi x^2)$. For $i = 1, \ldots, n$ let each x_i be drawn uniformly at random on $[0, 1]$ and $y_i = f_*(x_i) + \epsilon_i$ where $\epsilon_i \sim \mathcal{N}(0, 1)$. For any function f, the true error is defined as

$$\mathcal{E}_{true}(f) = E_{XY}[(f(X) - Y)^2]$$

whereas the training error is defined as

$$\widehat{\mathcal{E}}_{train}(f) = \frac{1}{n}\sum_{i=1}^n (f(x_i) - y_i)^2.$$

Using kernel ridge regression, construct a predictor

$$\widehat{\alpha} = \arg\min_\alpha ||K\alpha - y||^2 + \lambda \alpha^T K \alpha \ , \qquad \widehat{f}(x) = \sum_{i=1}^n \widehat{\alpha}_i k(x_i, x)$$

where $K_{i,j} = k(x_i, x_j)$ is a kernel evaluation and λ is the regularization constant.

a. *[10 points]* Using leave-one-out cross validation, find a good λ and hyperparameter settings for the following kernels:

- $k_{poly}(x, z) = (1 + x^T z)^d$ where $d \in \mathbb{N}$ is a hyperparameter,

- $k_{rbf}(x, z) = \exp(-\gamma\|x - z\|^2)$ where $\gamma > 0$ is a hyperparameter[1].

Report the values of d, γ, and the λ values for both kernels.

b. *[10 points]* Let $\widehat{f}_{poly}(x)$ and $\widehat{f}_{rbf}(x)$ be the functions learned using the hyperparameters you found in part a. For a single plot per function $\widehat{f} \in \{\widehat{f}_{poly}(x), \widehat{f}_{rbf}(x)\}$, plot the original data $\{(x_i, y_i)\}_{i=1}^n$, the true $f(x)$, and $\widehat{f}(x)$ (i.e., define a fine grid on $[0, 1]$ to plot the functions).

c. *[5 points]* We wish to build Bootstrap percentile confidence intervals for $\widehat{f}_{poly}(x)$ and $\widehat{f}_{rbf}(x)$ for all $x \in [0, 1]$ from part b. Use the non-parametric bootstrap with $B = 300$ datasets to find 5% and 95% percentiles at each point x on a fine grid over $[0, 1]$ (see Hastie, Tibshirani, Friedman Ch. 8.2 for a review). Specifically, for each dataset $b \in \{1, \ldots, B\}$, draw uniformly at randomly with replacement n samples from $\{(x_i, y_i)\}_{i=1}^n$, train an $\widehat{f}_b$ using the bth resampled dataset, compute $\widehat{f}_b(x)$ for each x in your fine grid; let the 5th percentile at point x be the largest value ν such that $\frac{1}{B}\sum_{b=1}^B \mathbf{1}\{\widehat{f}_b(x) \leq \nu\} \leq .05$, define the 95% analogously. Plot the 5 and 95 percentile curves on the plots from part b.

d. *[5 points]* Repeat all parts of this problem with $n = 300$ (you may just use 10-fold CV instead of leave-one-out)

e. *[5 points]* For this problem, use the $\widehat{f}_{poly}(x)$ and $\widehat{f}_{rbf}(x)$ learned in part d. Suppose $m = 1000$ additional samples $(x'_1, y'_1), \ldots, (x'_m, y'_m)$ are drawn i.i.d. the same way the first n samples were drawn. Use the non-parametric bootstrap with $B = 300$ to construct a confidence interval on $\mathbb{E}[(Y - \widehat{f}_{poly}(X))^2 - (Y - \widehat{f}_{rbf}(X))^2]$ (i.e. randomly draw with replacement m samples denoted as $\{(\widetilde{x}'_i, \widetilde{y}'_i)\}_{i=1}^m$ from $\{(x'_i, y'_i)\}_{i=1}^m$ and compute $\frac{1}{m}\sum_{i=1}^m \left((\widetilde{y}'_i - \widehat{f}_{poly}(\widetilde{x}'_i))^2 - (\widetilde{y}'_i - \widehat{f}_{rbf}(\widetilde{x}'_i))^2\right)$, repeat this B times) and find 5% and 95% percentiles. Using this confidence interval, is there statistically signicant evidence to suggest that one of $\widehat{f}_{rbf}$ and $\widehat{f}_{poly}$ is better than the other at predicting Y from X? (Hint: does the confidence interval contain 0?)

Context: With binary labels like in the extra credit below, we can easily compute means and variances and appeal to the CLT to compute confidence intervals. However, in almost all other settings computing nearly-exact confidence intervals is impossible because there is information we simply do not have. The Bootstrap gives us an ability to construct confidence intervals on nearly any quantity of interest with minimal side-knowledge.

k-means clustering

6. Given a dataset $\mathbf{x}_1, ..., \mathbf{x}_n \in \mathbb{R}^d$ and an integer $1 \leq k \leq n$, recall the following k-means objective function

$$\min_{\pi_1,\ldots,\pi_k} \sum_{i=1}^k \sum_{j\in\pi_i} \|\mathbf{x}_j - \mu_i\|_2^2 \ , \quad \mu_i = \frac{1}{|\pi_i|}\sum_{j\in\pi_i} \mathbf{x}_j \ . \tag{1}$$

Above, $\{\pi_i\}_{i=1}^k$ is a partition of $\{1, 2, ..., n\}$. The objective (1) is NP-hard[2] to find a global minimizer of. Nevertheless the commonly used heuristic which we discussed in lecture, known as Lloyd's algorithm, typically works well in practice. Implement Lloyd's algorithm for solving the k-means objective (1). Do not use any off the shelf implementations, such as those found in `scikit-learn`.

a. *[5 points]* Run the algorithm on the *training* dataset of MNIST with $k = 10$, plotting the objective function (1) as a function of iteration. Visualize (and include in your report) the cluster centers as a 28×28 image.

b. *[5 points]* For $k = \{2, 5, 10, 20, 40, 80, 160, 320, 640, 1280\}$ run the algorithm on the *training* dataset to obtain centers $\{\mu_i\}_{i=1}^k$. If $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$ and $\{(\mathbf{x}'_i, y'_i)\}_{i=1}^m$ denote the training and test sets, respectively, plot the training error $\frac{1}{n}\sum_{i=1}^n \min_{j=1,\ldots,k} \|\mu_j - \mathbf{x}_i\|_2^2$ and test error $\frac{1}{m}\sum_{i=1}^m \min_{j=1,\ldots,k} \|\mu_j - \mathbf{x}'_i\|_2^2$ as a function of k on the same plot. (If the large values of k are taking unresonably long to compute, just go up to the largest value of k you can.)

[1] Given a dataset $x_1, \ldots, x_n \in \mathbb{R}^d$, a heuristic for choosing a range of γ in the right ballpark is the inverse of the median of all $\binom{n}{2}$ squared distances $||x_i - x_j||_2^2$.

[2] To be more precise, it is both NP-hard in d when $k = 2$ and k when $d = 2$. See the references on the wikipedia page for k-means for more details.

Multivariate Gaussians

7. For a matrix $A \in \mathbb{R}^{n \times n}$ we denote $|A|$ as the determinant of A. A multivariate Gaussian with mean $\mu \in \mathbb{R}^n$ and covariance $\Sigma \in \mathbb{R}^{n \times n}$ has a probability density function $p(x|\mu, \Sigma) = \frac{1}{\sqrt{(2\pi)^n |\Sigma|}} \exp(-\frac{1}{2}(x-\mu)^T \Sigma^{-1}(x-\mu))$ which we denote as $\mathcal{N}(\mu, \Sigma)$. For background on multivariate Gaussians, see Murphy 2.5.2, 2.6.1, and 4.1. Let

- $\mu_1 = \begin{bmatrix} 1 \\ 2 \end{bmatrix}$ and $\Sigma_1 = \begin{bmatrix} 1 & 0 \\ 0 & 2 \end{bmatrix}$
- $\mu_2 = \begin{bmatrix} -1 \\ 1 \end{bmatrix}$ and $\Sigma_2 = \begin{bmatrix} 2 & -1.8 \\ -1.8 & 2 \end{bmatrix}$
- $\mu_3 = \begin{bmatrix} 2 \\ -2 \end{bmatrix}$ and $\Sigma_3 = \begin{bmatrix} 3 & 1 \\ 1 & 2 \end{bmatrix}$

For each $i = 1, 2, 3$ on a separate plot:

a. *[5 points]* Draw $n = 100$ points $X_{i,1}, \ldots, X_{i,n} \sim \mathcal{N}(\mu_i, \Sigma_i)$ and plot the points as a scatter plot with each point as a triangle marker (Hint: use `numpy.random.randn(d)` to generate a d-dimensional vector $Z \sim \mathcal{N}(0, I)$, then use the fact that $AZ + b \sim \mathcal{N}(b, AA^\top)$. Be careful, if symmetric A satisfies $A^2 = \Sigma$ then A is the matrix square root of Σ which in general is *not* the square root of the entries of Σ. See Murphy references above)

b. *[5 points]* Compute the sample mean and covariance matrices $\widehat{\mu}_i = \frac{1}{n}\sum_{j=1}^n X_{i,j}$ and $\widehat{\Sigma}_i = \frac{1}{n-1}\sum_{j=1}^n (X_{i,j} - \widehat{\mu}_i)(X_{i,j} - \widehat{\mu}_i)^\top$. Compute the eigenvectors of $\widehat{\Sigma}_i$. Plot the unit-norm eigenvectors as line segments originating from $\widehat{\mu}_i$ and have magnitude equal to the square root of their corresponding eigenvalues. Make sure your axes are square (e.g., the x and y limits are both $[-r, r]$ for some appropriately large $r > 0$ to see the data.) (Hint: see `numpy.linalg.eig`.)

c. *[5 points]* If $(u_{i,1}, \lambda_{i,1})$ and $(u_{i,2}, \lambda_{i,2})$ are the eigenvector-eigenvalue pairs of the sample covariance matrix with $\lambda_{i,1} \geq \lambda_{i,2}$ and $||u_{i,1}||_2 = ||u_{i,2}||_2 = 1$, for $j = 1, \ldots, n$ let $\widetilde{X}_{i,j} = \begin{bmatrix} \frac{1}{\sqrt{\lambda_{i,1}}} u_{i,1}^T (X_{i,j} - \widehat{\mu}_i) \\ \frac{1}{\sqrt{\lambda_{i,2}}} u_{i,2}^T (X_{i,j} - \widehat{\mu}_i) \end{bmatrix}$. On a new figure, plot these new points as a scatter plot with each point as a circle marker. Make sure your axes are square.

Extra credit: Statistically Significant Improvement

8. *[2 points]* (Extra Credit) This problem explores the connection between quantiles (or confidence intervals) and hypothesis testing. See Murphy 2.2.6 and any introductory statistics text for background on hypothesis testing[3]. Let $X \in \mathcal{N}(\mu, 1)$ for some $\mu \in \mathbb{R}$. Consider the hypothesis test

$$H_0 : \mu = 0$$
$$H_1 : \mu \neq 0$$

so that under H_0 we have $\mathbb{E}[X] = \mu = 0$, and under H_1 we have $\mathbb{E}[X] = \mu \neq 0$. Define $z_{\alpha/2}$ to be the unique number such that $\int_{z_{\alpha/2}}^{\infty} \frac{1}{\sqrt{2\pi}} e^{-x^2/2} dx = \alpha/2$. We say we *reject the null hypothesis* H_0 in favor of H_1 if $|X| \geq z_{\alpha/2}$.

a. Under H_0, show that the probability of rejecting the null hypothesis is less than or equal to α.

b. If we do *not* reject the null hypothesis (i.e., $|X| < z_{\alpha/2}$) is this evidence to support that H_0 is true? Justify your answer.

[3] There are many resources online as well, e.g., https://machinelearningmastery.com/statistical-hypothesis-tests/

9. *[8 points]* (Extra Credit) Suppose you have two models f_1 and f_2 that predict Y from X where (X, Y) follow a joint distribution $\mathcal{P}$ (e.g., $X \in \mathbb{R}^d$, $Y \in \{-1, 1\}$). For $j = 1, 2$ define

$$p_j = \mathbb{P}_{XY}(f_j(X) \neq Y) \qquad \rho = \frac{\mathbb{E}_{XY}[(\mathbf{1}\{f_1(X) \neq Y\} - p_1)(\mathbf{1}\{f_2(X) \neq Y\} - p_2)]}{\sqrt{p_1(1-p_1)p_2(1-p_2)}}$$

noting that each variance $\mathbb{E}_{XY}[(\mathbf{1}\{f_j(X) \neq Y\} - p_j)^2] = p_j(1-p_j)$. The quantity $\rho \in [-1, 1]$ determines the degree to which the predictions of f_1 and f_2 are correlated. You wish to determine if one model versus the other is better at predicting Y from X in a statistically significant way. That is, you are faced with the hypothesis test

$$H_0 : p_1 = p_2$$
$$H_1 : p_1 \neq p_2.$$

We will leverage our result from problem 1 to construct a hypothesis test such that under H_0, the null hypothesis H_0 is rejected in favor of H_1 with probability less than α.
Let $(x_1, y_1), \ldots, (x_n, y_n) \overset{iid}{\sim} \mathcal{P}$ and $(x'_1, y'_1), \ldots, (x'_m, y'_m) \overset{iid}{\sim} \mathcal{P}$ be two independent fresh samples from $\mathcal{P}$ of size n and m, respectively. For $j = 1, 2$ define

$$\widehat{p}_j = \frac{1}{n}\sum_{i=1}^{n} \mathbf{1}\{f_j(x_i) \neq y_i\} \qquad \widehat{p}'_j = \frac{1}{m}\sum_{i=1}^{m} \mathbf{1}\{f_j(x'_i) \neq y'_i\}$$

a. (Means) Show that $\mathbb{E}[\widehat{p}_j] = \mathbb{E}[\widehat{p}'_j] = p_j$ for $j = 1, 2$.

b. (Variances) Show that $\mathbb{E}[(\widehat{p}_j - p_j)^2] = \frac{p_j(1-p_j)}{n}$ and $\mathbb{E}[(\widehat{p}'_j - p_j)^2] = \frac{p_j(1-p_j)}{m}$ for $j = 1, 2$.

c. (Empirical Variance) Argue *briefly* that $\widehat{p}_j(1-\widehat{p}_j) \to p_j(1-p_j)$ as $n \to \infty$ by appealing to the Law of Large Numbers (LLN).

d. (Independent Test statistic) Define $\theta := \frac{\widehat{p}_1 - \widehat{p}'_2}{\sqrt{\widehat{p}_1(1-\widehat{p}_1)/n + \widehat{p}'_2(1-\widehat{p}'_2)/m}}$. Using the approximations of part c (for example, replace $\widehat{p}_1(1-\widehat{p}_1)$ with $p_1(1-p_1)$), argue *briefly* that $\theta \sim \mathcal{N}(0, 1)$ under the null hypothesis H_0 (i.e., $p_1 = p_2$) as $n, m \to \infty$ by appealing to the central limit theorem (CLT). Note by above that this implies the probability of $|\theta| \geq z_{\alpha/2}$ is no greater than α. (Hint: what are $\mathbb{E}[\theta]$ and $\mathbb{E}[(\theta - \mathbb{E}[\theta])^2]$ as $n, m \to \infty$?)

e. (Dependent Test statistic) Define $\phi := \frac{\widehat{p}_1 - \widehat{p}_2}{\sqrt{\widehat{p}_1(1-\widehat{p}_1)/n + \widehat{p}_2(1-\widehat{p}_2)/n}}$. Using the approximations of part c, argue that under the null hypothesis H_0 (i.e., $p_1 = p_2$) as $n \to \infty$ we have that $\phi \sim \mathcal{N}(0, 1-\rho)$. Explain why the probability that $|\phi| \geq z_{\alpha/2}$ is no longer necessarily bounded by α

f. (Correcting the test statistic) Using the approximations of part c, argue that the probability that $|\phi|/\sqrt{2} \geq z_{\alpha/2}$ is no greater than α under the null hypothesis H_0 (i.e., $p_1 = p_2$) as $n \to \infty$. (Hint: the sum/difference of even dependent Gaussian random variables is still Gaussian distributed.)

g. (Dependent Test statistic revisited) Define

$$\begin{aligned}
\widehat{\Delta} &= \frac{1}{n}\sum_{i=1}^{n}(\mathbf{1}\{f_1(x_i) \neq y_i\} - \mathbf{1}\{f_2(x_i) \neq y_i\}) \\
&= \widehat{p}_1 - \widehat{p}_2 \\
\widehat{\sigma}^2 &= \frac{1}{n}\sum_{i=1}^{n}(\mathbf{1}\{f_1(x_i) \neq y_i\} - \mathbf{1}\{f_2(x_i) \neq y_i\} - \widehat{\Delta})^2 \\
&= \frac{1}{n}\sum_{i=1}^{n}(\mathbf{1}\{f_1(x_i) \neq y_i\} - \widehat{p}_1 + \widehat{p}_2 - \mathbf{1}\{f_2(x_i) \neq y_i\})^2 \\
&= \frac{1}{n}\sum_{i=1}^{n}(\mathbf{1}\{f_1(x_i) \neq y_i\} - \widehat{p}_1)^2 + (\mathbf{1}\{f_2(x_i) \neq y_i\} - \widehat{p}_2)^2 + (\mathbf{1}\{f_1(x_i) \neq y_i\} - \widehat{p}_1)(\mathbf{1}\{f_2(x_i) \neq y_i\} - \widehat{p}_2) \\
&= \frac{\widehat{p}_1(1-\widehat{p}_1)}{n} + \frac{\widehat{p}_2(1-\widehat{p}_2)}{n} + \frac{1}{n}\sum_{i=1}^{n}(\mathbf{1}\{f_1(x_i) \neq y_i\} - \widehat{p}_1)(\mathbf{1}\{f_2(x_i) \neq y_i\} - \widehat{p}_2)
\end{aligned}$$

so that $\mathbb{E}[\widehat{\Delta}] = p_1 - p_2$ and $\mathbb{E}[\widehat{\sigma}^2] = \mathbb{E}[(\widehat{\Delta} - \mathbb{E}[\widehat{\Delta}])^2]$. If $\psi := \frac{\widehat{\Delta}}{\sqrt{\widehat{\sigma}^2}}$ then as $n \to \infty$ we have that $\psi \sim \mathcal{N}(0,1)$ using the approximations of part *c*. Under the null hypothesis H_0 (i.e., $p_1 = p_2$), we have shown that both tests: $\{|\phi|/\sqrt{2} \geq z_{\alpha/2}\}$ and $\{|\psi| \geq z_{\alpha/2}\}$ reject the null hypothesis with probability at most α as $n \to \infty$. Define the *power* of a test to be the probability that the null hypothesis H_0 ($p_1 = p_2$) is rejected when the alternative hypothesis H_1 ($p_1 \neq p_2$) is true. Which of these two tests, $\{|\phi|/\sqrt{2} \geq z_{\alpha/2}\}$ or $\{|\psi| \geq z_{\alpha/2}\}$, do you think is more powerful? Justify your answer.

Context: It is common in machine learning to have standard benchmark datasets in order to evaluate whether your new algorithm performs better than past algorithms. If we have access to how both algorithms classified each individual test point, we could use the estimator of f. or g. (both are correct), but note that if we only have access to the empirical means reported in the paper, we can only (safely) use f.