CSE 512: Homework 3 Due Monday, Oct. 7

1. **Conditional independence vs independence.**

   Tom is a blue-gray cat with a bushy tail, and Jerry is a brown mouse with a rope-like tail. After many years of fighting, they both decided to settle down, and now have thriving families. Tom has 10 kids and Jerry has 40 kids. Tom's kids are all cats like him, with bushy tails. Half of Tom's kids are blue, while the other half is gray. Jerry's kids are all brown mice, with rope-like tails.

   (a) I pick up a baby animal at random. What is the probability that ... (fill in the table)

fur \ tail	furry	rope-like
blue		
gray		
brown		

   Ans. (0.5 pts)

fur \ tail	furry	rope-like
blue	10%	0
gray	10%	0
brown	0	80%

   (b) Are the features "fur color" and "tail texture" correlated, without knowing the type of animal? (Show mathematically.)

   Ans. (0.25 pts) Yes, they are correlated. To take as an example,

$$\begin{aligned} \mathbf{Pr}(\text{blue}, \text{fuzzy}) &= 5/50 = 1/10 \\ \mathbf{Pr}(\text{blue}) &= 5/50 = 1/10 \\ \mathbf{Pr}(\text{fuzzy}) &= 10/50 = 1/5 \\ \mathbf{Pr}(\text{blue}) \cdot \mathbf{Pr}(\text{fuzzy}) &= 1/50 \neq 1/10 \end{aligned}$$

   (c) Now Tom comes over and says, "I'm very proud of my baby girl, of whom you are holding." What is the probability that (fill in the table)

fur \ tail	furry	rope-like
blue		
gray		
brown		

   Ans. (0.5 pts)

fur \ tail	furry	rope-like
blue	50%	0
gray	50%	0
brown	0	0

(d) Are the features "fur color" and "tail texture" correlated, now that I know the animal is Tom's cherished baby daughter? (Show mathematically.)

Ans. (0.25 pts) No, now the features are uncorrelated. Specifically,

$$\begin{aligned}
\underbrace{\mathbf{Pr}(\text{blue})}_{1/2} \cdot \underbrace{\mathbf{Pr}(\text{fuzzy})}_{1} &= \underbrace{\mathbf{Pr}(\text{blue}, \text{fuzzy})}_{1/2} \\
\underbrace{\mathbf{Pr}(\text{gray})}_{1/2} \cdot \underbrace{\mathbf{Pr}(\text{fuzzy})}_{1} &= \underbrace{\mathbf{Pr}(\text{gray}, \text{fuzzy})}_{1/2} \\
\underbrace{\mathbf{Pr}(\text{blue})}_{1/2} \cdot \underbrace{\mathbf{Pr}(\text{rope-like})}_{0} &= \underbrace{\mathbf{Pr}(\text{blue}, \text{fuzzy})}_{0} \\
\underbrace{\mathbf{Pr}(\text{gray})}_{1/2} \cdot \underbrace{\mathbf{Pr}(\text{rope-like})}_{0} &= \underbrace{\mathbf{Pr}(\text{gray}, \text{fuzzy})}_{0}
\end{aligned}$$

2. **Exponential distribution.** Wait time is often modeled as an exponential distribution, e.g.

$$\mathbf{Pr}(\text{I wait} < x \text{ hours at the DMV}) = \begin{cases} 1 - e^{-\lambda x}, & x > 0 \\ 0, & x \leq 0, \end{cases}$$

and this cumulative density function is parametrized by some constant $\lambda > 0$. A random variable X distributed according to this CDF is denoted as $X \sim \exp[\lambda]$.

(a) In terms of λ, give the probability distribution function for the exponential distribution.

Ans. (0.25 pts) The PDF can be computed as just the derivative of the CDF, which comes to

$$p_\lambda(x) = \begin{cases} \lambda e^{-\lambda x}, & x > 0 \\ 0, & x \leq 0. \end{cases}$$

(b) Show that if $X \sim \exp(\lambda)$, then the mean of X is $1/\lambda$ and the variance is $1/\lambda^2$.
(You may use a symbolic integration tool such as Wolfram Alpha. If you do wish to do the integral by hand, my hint is to review integration by parts.)

Ans. (0.25 pts) To compute the mean,

$$\mathbb{E}[X] = \int_0^\infty x p_\lambda(x) dx = \int_0^\infty \lambda x e^{-\lambda x} dx$$

Now the rest is an exercise in integration. Using Wolfram Alpha,

$$\int_0^\infty \lambda x e^{-\lambda x} dx = \frac{e^{-\lambda x}}{\lambda}(\lambda x - 1)|_0^\infty = 0 - \left(-\frac{1}{\lambda}\right) = 1/\lambda$$

The same result can be arrived at by using integration by parts.
To compute the variance, recall that $\mathbf{var}(X) = \mathbb{E}[(X - \mathbb{E}[X])^2] = \mathbb{E}[X^2] - (\mathbb{E}[X])^2$. Then

$$\mathbb{E}[X^2] = \int_0^\infty x^2 p_\lambda(x) dx = \int_0^\infty \lambda x^2 e^{-\lambda x} dx = \exp(-\lambda x)\left(-\frac{2}{\lambda^2} - \frac{2x}{\lambda} - x^2\right)|_0^\infty = \frac{2}{\lambda^2}$$

$$\mathbb{E}[X^2] - (\mathbb{E}[X])^2 = \frac{2}{\lambda^2} - \frac{1}{\lambda^2} = \frac{1}{\lambda^2}.$$

(c) Now suppose I run a huge server farm, and I am monitoring the server's ability to respond to web requests. I have m observations of delay times, $x_1, ..., x_m$, which I assume are i.i.d., distributed according to exp[λ] for some λ. Given these m observations, what is the maximum likelihood estimate $\hat{\lambda}$ of λ?

Ans. (0.5 pts) First, we compute the likelihood of observations $x_1, ..., x_m$ given that they are i.i.d., distributed as exp[λ]:

$$\mathbf{Pr}(x_1, ..., x_m) = \prod_{i=1}^{m} \mathbf{Pr}(x_i) = \prod_{i=1}^{m} (\lambda e^{-\lambda x_i}) = \lambda^m \exp\left(-\lambda \sum_{i=1}^{m}\right).$$

I would like to find λ which maximizes this quantity. However, this expression looks pretty complicated–not convex or concave.

Let's use a trick that we are now pretty familiar with: take the log.

$$\log\left(\mathbf{Pr}(x_1, ..., x_m)\right) = m\log(\lambda) - \lambda \sum_{i=1}^{m} x_i.$$

This is a concave function of λ, so now we can find the maximum of the log probability by taking the derivative and setting it to 0:

$$\frac{\partial}{\partial \lambda} \log\left(\mathbf{Pr}(x_1, ..., x_m)\right) = \frac{m}{\lambda} - \sum_{i=1}^{m} x_i = 0 \Rightarrow \frac{1}{\hat{\lambda}} = \frac{1}{m} \sum_{i=1}^{m} x_i$$

(d) Given the estimate of $\hat{\lambda}$ in your previous question, is $1/\hat{\lambda}$ an unbiased estimate of the mean wait time? Is $1/\hat{\lambda}^2$ an unbiased estimate of the variance in wait time?

Ans. (0.5 pts) The term $1/\hat{\lambda}$ is indeed an unbiased estimator of the mean:

$$\mathbb{E}\left[\frac{1}{\hat{\lambda}}\right] = \mathbb{E}\left[\frac{1}{m}\sum_{i=1}^{m} x_i\right] = \frac{1}{m}\sum_{i=1}^{m} \mathbb{E}[x_i] = \lambda$$

The term $1/\hat{\lambda}^2$ is a biased estimator of the variance:

$$\begin{aligned}
\mathbb{E}\left[\frac{1}{\hat{\lambda}^2}\right] &= \mathbb{E}\left[\left(\frac{1}{m}\sum_{i=1}^{m} x_i\right)^2\right] \\
&= \frac{1}{m^2}\sum_{i=1}^{m}\sum_{j=1}^{m} \underbrace{\mathbb{E}[x_i x_j]}_{\text{i.i.d.}} \\
&= \frac{1}{m^2}\left(\sum_{i\neq j} \underbrace{\mathbb{E}[x_i]}_{1/\lambda}\underbrace{\mathbb{E}[x_j]}_{1/\lambda} + \sum_{i=j} \underbrace{\mathbb{E}[x_i^2]}_{\mathbf{var}(X)+(\mathbb{E}[X])^2}\right) \\
&= \frac{1}{m^2}\left(m(m-1)\frac{1}{\lambda^2} + m\cdot(\frac{1}{\lambda^2}+\frac{1}{\lambda^2})\right) \\
&= \frac{1}{m^2}\left(m(m+1)\cdot\frac{1}{\lambda^2}\right) \\
&= \left(1+\frac{1}{m^2}\right)\cdot\frac{1}{\lambda^2}
\end{aligned}$$

(e) Now let's consider $x_1, ..., x_m$ drawn i.i.d. from a *truncated* exponential distribution, e.g.

$$p_{\lambda,c}(x) = \begin{cases} 0 & \text{if } x > c \text{ or } x < 0 \\ \dfrac{\lambda\exp(-\lambda x)}{1-\exp(-\lambda c)} & \text{else.} \end{cases}$$

Using Hoefdding's inequality, give a range of values that account for the uncertainty in your guess. That is, as a function of x_i, m and δ, give a range of values $[\hat{\lambda}_{\min}, \hat{\lambda}_{\max}]$ such that

$$\mathbf{Pr}(\hat{\lambda}_{\min} \leq \mathbb{E}[X] \leq \hat{\lambda}_{\max}) \geq 1 - \delta.$$

Ans. (0.5 pts) Actually, we can see from Hoeffding's inequality that the only quality we need to extract from the distribution is the range of values that x_i can take with nonzero probability. Via this truncation, we see that each x_i must be between 0 and c; otherwise, it occurs with 0 probability.

Therefore, we can think of a new variable $z_i := x_i/c$, and in fact for the random variable $Z = X/c$, $\mathbb{E}[Z] = \mathbb{E}[X/c] = \frac{1}{c}\mathbb{E}[X]$.

Then, the random variable Z satisfies all the conditions needed for Hoeffding's inequality; e.g.,

$$\mathbf{Pr}\left(\frac{1}{m}\sum_{i=1}^{m} z_i - \mathbb{E}[Z] \geq \epsilon\right) \leq e^{-2m\epsilon^2}.$$

Plugging in $x_i = cz_i$ and $X = cZ$, then

$$\mathbf{Pr}\left(\frac{1}{m}\sum_{i=1}^{m} \frac{x_i}{c} - \frac{\mathbb{E}[X]}{c} \geq \epsilon\right) \leq e^{-2m\epsilon^2}.$$

Since I want an estimator for $\mathbb{E}[X]$ and not $\mathbb{E}[X]/c$, we need to rescale things:

$$\mathbf{Pr}\left(\frac{1}{m}\sum_{i=1}^{m} x_i - \mathbb{E}[X] \geq c\epsilon\right) \leq e^{-2m\epsilon^2}.$$

Ok, now we're almost there! What we basically can see now is that

$$\frac{1}{m}\sum_{i=1}^{m} x_i - c\epsilon \leq \mathbb{E}[X] \leq \frac{1}{m}\sum_{i=1}^{m} x_i + c\epsilon$$

with probability $1 - e^{-2m\epsilon^2}$. Taking $\delta = e^{-2m\epsilon^2}$, we see that

$$\epsilon = \sqrt{\frac{\log(2/\delta)}{2m}}.$$

The final answer should look like

$$\hat{\lambda}_{\min} = \frac{1}{m}\sum_{i=1}^{m} x_i - c\sqrt{\frac{\log(2/\delta)}{2m}}, \qquad \hat{\lambda}_{\max} = \frac{1}{m}\sum_{i=1}^{m} x_i + c\sqrt{\frac{\log(2/\delta)}{2m}}$$

3. **Decision theory.** I run a factory that makes widgets and gadgets. Despite best efforts, manufacturing defects can always occur. I would like to inspect each of these items individually, but the cost of inspection is pretty high, so I cannot inspect each individual widget and gadget.

   The widgets and gadgets are printed on disks. A disk has a 10% chance of being warped. There are two printing presses, a blue one and a red one. The table below gives the possibility that, given a disk of a particular state printed by a particular press, a widget or gadget printed on that disk is defective.

disk \ press	red	blue
warped	30%	85%
normal	5%	0 %

(To interpret the table, the probability that a gadget is defective if it were on a disk that is not warped, and printed by a red press, is 10%.)

(a) First, we consider only the loss of quality in a product. That is, if we ship a widget or gadget is defective, we incur a loss of +1. Otherwise, we incur no losses. **Ans.** It took me a few tries to arrive at an answer here that I'm convinced is correct. This problem is definitely a good example of why it's super important to write out the probabilities that you are interested in, as explicitly as possible, and to avoid taking shortcuts everywhere.

It seems the best way to do this is to break out the problem in terms of all possible events. We denote D, W, and I as the random variables for whether an item is defective, the disk the item is on is warped, or the disk the item is on is inspected. We write $D = 1$ if it is defective and $D = 0$ otherwise. (Similarly, with W and I.) Then

$$\text{risk}_{\text{Bayes}}(item) = \sum_{D,W,I \in \{0,1\}} \mathbf{Pr}(D, W, I)\text{cost}(D, W, I).$$

There are 8 terms in this sum. Some of them we can reduce a bit; namely, whether an item is inspected or not doesn't depend on the disk being warped, or the item is defective. Then

$$\mathbf{Pr}(D, W, I) = \mathbf{Pr}(D|W)\mathbf{Pr}(W)\mathbf{Pr}(I).$$

i. Without inspecting anything (that is, we ship out everything we make), what is the Bayes risk of using a red press? a blue press? Which machine would I use to minimize the Bayes' risk?
Ans. (0.5 pts) At this point, the loss is simply

$$\mathbf{loss}(D) = \begin{cases} +1 & \text{if } D = 1 \\ 0 & \text{if } D = 0. \end{cases}$$

Therefore,

$$\text{Bayes' risk} = \mathbf{loss}(D = 1) \cdot \mathbf{Pr}(D = 1) + \mathbf{loss}(D = 0) \cdot \mathbf{Pr}(D = 0) = \mathbf{Pr}(D = 1).$$

For the red press, the probability of an item being defective is

$$\begin{aligned} \mathbf{Pr}(D = 1|\text{red}) &= \mathbf{Pr}(D = 1|W = 1, \text{red})\mathbf{Pr}(W = 1|\text{red}) + \mathbf{Pr}(D = 1|W = 0, \text{red})\mathbf{Pr}(W = 0|\text{red}) \\ &= 0.3 \cdot 0.1 + 0.05 \cdot 0.9 = 0.075 \end{aligned}$$

For the blue press, the probability of an item being defective is

$$\begin{aligned} \mathbf{Pr}(D = 1|\text{blue}) &= \mathbf{Pr}(D = 1|W = 1, \text{blue})\mathbf{Pr}(W = 1|\text{blue}) + \mathbf{Pr}(D = 1|W = 0, \text{blue})\mathbf{Pr}(W = 0|\text{blue}) \\ &= 0.85 \cdot 0.1 + 0.0 \cdot 0.9 = 0.085. \end{aligned}$$

Therefore, the Bayes risk is

$$\text{Bayes' risk, red machine} = 0.075, \qquad \text{Bayes' risk, blue machine} = 0.085.$$

For the lowest Bayes' risk, I would use the red machine.

ii. Without inspecting anything, what is the Minimax risk of using a red press? a blue press?
Ans. (0.5 pts) The question here is based more on logic. Recall that since $X = D$ and $X = \bar{D}$ are nonzero probability events, then

$$\text{minimax risk} = \max_{D \in \{0,1\}} [\mathbf{loss}(D))].$$

Since, as calculated above, the probability that a widget or gadget is defective given red or blue machine is greater than 0, then our minimax risk is 1 in both cases. So, using either machine gives same loss.

iii. Suppose I invest the effort into inspecting disks, and remove all warped disks. What is the Bayes risk of using a red press? a blue press?
Ans. (.5 pts) This in fact changes the loss function a bit. Specifically, the loss per item is now a function of defection and warped.

$$\mathbf{loss}(D, W) = \begin{cases} +1 & \text{if } D = 1 \text{ and } W = 0 \\ 0 & \text{if } D = 0 \text{ or } W = 1 \end{cases}$$

Now,the Bayes risk is

$$\text{risk}_{\text{Bayes}}(\text{item}) = 1 \cdot \mathbf{Pr}(D = 1, W = 0) = \mathbf{Pr}(D = 1|W = 0)\mathbf{Pr}(W = 0).$$

For the red press, the numbers are

$$\text{risk}_{\text{Bayes}}(\text{item}) = 5\% \cdot 90\% = 4.5\%$$

and for the blue press, 0%. Then the Bayes risk is simply this quantity:

$$\text{Bayes' risk, red machine} = 0.045, \qquad \text{Bayes' risk, blue machine} = 0.0.$$

iv. After removing all warped disks, what is the Minimax risk of using a red press? a blue press?
Ans. (0.5 pts) Since the chance of a defection with the red press is still nonzero, then the minimax loss is 1. For the blue press, since the chance of defection is 0 once all warped disks are removed, the minimax loss is 0.

(b) Widgets are primarily used in online advertising. If they are defective, they will end up sending an ad that is undesirable. However, if they are removed, then no ad is sent out. Therefore, the revenue gained from a widget is estimated at

$$\text{revenue per widget} = \begin{cases} \$1 & \text{if widget is sold and is not defective} \\ -\$1 & \text{if widget is sold and is defective} \\ 0 & \text{if the widget is not sold.} \end{cases}$$

There is no cost to rejecting a disk, but the cost of inspection is \$1 per percent of disks inspected. (In other words, if I inspect all the disks, that cost me \$100 per widget/gadget.) Every widget that is not on a disk that was found to be warped is sold.

i. Compute the Bayes reward (e.g. the expected profit per widget) as a function of $x = \mathbf{Pr}(\text{inspection})$ for widgets, when using the blue press. Compute the same for the red press.
Ans. (0.5 pts) At this point, we should be able to do everything by just adjusting our reward function. The new reward goes like this:

$$\mathbf{reward}(D, W, I) = \begin{cases} +1 & \text{if } D = 0 \text{ and } (W = 0 \text{ or } I = 0) \\ -1 & \text{if } D = 1 \text{ and } (W = 0 \text{ or } I = 0) \\ 0 & \text{if } W = 1 \text{ and } I = 1. \end{cases}$$

Additionally, there is the added overhead cost of inspection.
Then we can tally up our numbers as

$$\begin{aligned} \text{risk}_{\text{Bayes}}(\text{item}) &= 1 \cdot \mathbf{Pr}(D = 0, W = 0 \text{ or } I = 0) - 1 \cdot \mathbf{Pr}(D = 1, W = 0 \text{ or } I = 0) - 100x \\ &= \mathbf{Pr}(D = 0, W = 0, I = 0) + \mathbf{Pr}(D = 0, W = 1, I = 0) + \mathbf{Pr}(D = 0, W = 0, I = 1) \\ &\quad -(\mathbf{Pr}(D = 1, W = 0, I = 0) + \mathbf{Pr}(D = 1, W = 1, I = 0) + \mathbf{Pr}(D = 1, W = 0, I = 1)) - 100x \\ &= \mathbf{Pr}(D = 0|W = 0)\mathbf{Pr}(W = 0)\mathbf{Pr}(I = 0) \\ &\quad +\mathbf{Pr}(D = 0|W = 1)\mathbf{Pr}(W = 1)\mathbf{Pr}(I = 0) \\ &\quad +\mathbf{Pr}(D = 0|W = 0)\mathbf{Pr}(W = 0)\mathbf{Pr}(I = 1) \\ &\quad -\mathbf{Pr}(D = 1|W = 0)\mathbf{Pr}(W = 0)\mathbf{Pr}(I = 0) \\ &\quad -\mathbf{Pr}(D = 1|W = 1)\mathbf{Pr}(W = 1)\mathbf{Pr}(I = 0) \\ &\quad -\mathbf{Pr}(D = 1|W = 0)\mathbf{Pr}(W = 0)\mathbf{Pr}(I = 1) - 100x \end{aligned}$$

For the red press, this results in

$$\begin{aligned} \text{risk}_{\text{Bayes}}(\text{item}) &= 95\% \cdot 0.9(1 - x) + 70\% \cdot 0.1(1 - x) + 95\% \cdot 0.9x \\ &\quad -5\% \cdot 0.9(1 - x) - 30\% \cdot 0.1(1 - x) - 5\% \cdot 0.9x \\ &= 0.85 - 100.80x. \end{aligned}$$

For the blue press, this results in

$$\begin{aligned}\text{risk}_{\text{Bayes}}(\text{item}) &= 100\% \cdot 0.9(1-x) + 15\% \cdot 0.1(1-x) + 100\% \cdot 0.9x \\ &\quad -0\% \cdot 0.9(1-x) - 85\% \cdot 0.1(1-x) - 0\% \cdot 0.9x \\ &= 0.83 - 98.27x.\end{aligned}$$

ii. If you were a consultant for my factory, how much inspection would you recommend? Would you recommend using one press over the other for widgets?
Ans. (0.25 pts) From a purely profit point of view, I make the most money if I don't inspect anything at all. The same logic applies to both machines.

(c) Gadgets are primarily used in medical care. If they are defective, someone will die. However, if they are removed, then someone waits a day longer to get a much-needed test. While we can never assign monetary value to a human life, in terms of insurance costs experts have estimated the following value:

$$\text{revenue per gadget} = \begin{cases} \$500 & \text{if gadget is sold and is not defective} \\ -\$10,000 & \text{if gadget is sold and is defective} \\ \$0 & \text{if the gadget is not sold.} \end{cases}$$

Again, there is no cost to rejecting a disk, and again, the cost of inspection is \$1 per percent of disks inspected. Every gadget that is not on a disk that was found to be warped is sold.

i. Compute the Bayes reward (e.g. the expected profit per day) as a function of $x = \mathbf{Pr}(\text{inspection})$ for gadgets, when using the blue press. Compute the same for the red press.
Ans. (0.5 pts)
With the new reward construct, I have

$$\mathbf{reward}(D, W, I) = \begin{cases} +500 & \text{if } D = 0 \text{ and } (W = 0 \text{ or } I = 0) \\ -10000 & \text{if } D = 1 \text{ and } (W = 0 \text{ or } I = 0) \\ 0 & \text{if } W = 1 \text{ and } I = 1. \end{cases}$$

Following very similar work as to the previous problem, we get
For the red press, this results in

$$\begin{aligned}\text{risk}_{\text{Bayes}}(\text{item}) &= 500 \cdot (95\% \cdot 0.9(1-x) + 70\% \cdot 0.1(1-x) + 95\% \cdot 0.9x) \\ &\quad -10,000 \cdot (5\% \cdot 0.9(1-x) + 30\% \cdot 0.1(1-x) + 5\% \cdot 0.9x) - 100x \\ &= -287.5 + 215x\end{aligned}$$

For the blue press, this results in

$$\begin{aligned}\text{risk}_{\text{Bayes}}(\text{item}) &= 100\% \cdot 0.9(1-x) + 15\% \cdot 0.1(1-x) + 100\% \cdot 0.9x \\ &\quad -10,000 \cdot (0\% \cdot 0.9(1-x) + 85\% \cdot 0.1(1-x) + 0\% \cdot 0.9x) - 100x \\ &= -392.5 + 742.5x\end{aligned}$$

ii. If you were a consultant for my factory, how much inspection would you recommend? Would you recommend using one press over the other for gadgets?
Ans. (0.25 pts) In fact, I stand to get the most profit if I inspect all of the disks, with a clear preference toward using exclusively the blue machine.

4. ***K*-nearest neighbors** We will now try to use the KNN classifier to classify MNIST digits. Bear in mind that the full MNIST datasets has 60000 training symbols, so we will need to use every trick in the book to avoid memory issues.

- Load your data. It should have 4 parts: `X_train`, `X_test`, `y_train`, `y_test`. The labels `y_train` and `y_test` should have labels 1,2,...,10. *Do not prune away* data of different labels; in this exercise, we will use *all* the labels. We will also *not need to normalize any data.*

- However, to make life easier, you will need to typecast your data. The data loaded are all 8-bit integers, and can only take values 0,...,255. (Try adding 255 + 255; you won't get 510.) You need to convert them from type uint8 to type float. In MATLAB, you can typecast by just typing

  ```
  X_train = float(X_train);
  X_test = float(X_test);
  ```

  In Python, you can use

  ```
  X_train = X_train.astype(float)
  X_test = X_test.astype(float)
  ```

- First, to understand the memory issues involved, write a function that takes in `X_train` and `y_test`, and a number $m < 60000$, and returns a train data matrix and label vector that contains only m data samples, sampled uniformly. Implement a 1-nearest-neighbor classifier, which takes a test data point, finds the closest point (in terms of Euclidean distance) in the train data set, and returns the label of that closest point.

  There are two ways you can implement the 1-NN classifier:

  - **Sacrifice memory, save computation.** Create a *distance matrix* D where D_{ij} stores the Euclidean distance between the ith training sample and jth test sample. Note that this requires all the test data samples to be known ahead of time.
  - **Sacrifice computational time, save memory.** The other approach is to compute everything "on the fly". For each incoming new data sample, compute its distance with every training sample, sort the distances, and return the label of the closest point. This is the more "realistic" approach, but for our purposes will require too much runtime, so we will forgo it.

  Using the first approach, and for $m = 10, 100, 1000, 10000$, return the error rate over the test data set.

- $K > 1$**.** Now increase K slowly from 1 to 10, and pick the label based on a majority voting system amongst the K closest neighbors. Do you see the computational time or memory becoming more burdensome with greater K? Again, for $m = 10, 100, 1000, 10000$, return the error rate over the test data set. Was the extra pain worth it?

- **Analyzing results.** For 10 test samples (one per digit) use `imshow` to plot, side by side, the digit, the training sample furthest from that digit but with the same label, and the training sample closest to that digit but with a different label. In the title, print the Euclidean distance between the test sample and the selected train sample. Interpret what you see.

- In your opinion, is KNN a reasonable way of performing handwriting digit classification? How does it compare against logistic regression or SVM?

Ans. (3 pts) So here are some observations based on my own implementation. For 1-NN, this is what happened on my tiny laptop. I definitely prefer the batch method, but of course in the real world, it may not be super realistic, and on-the-fly may be more feasible.

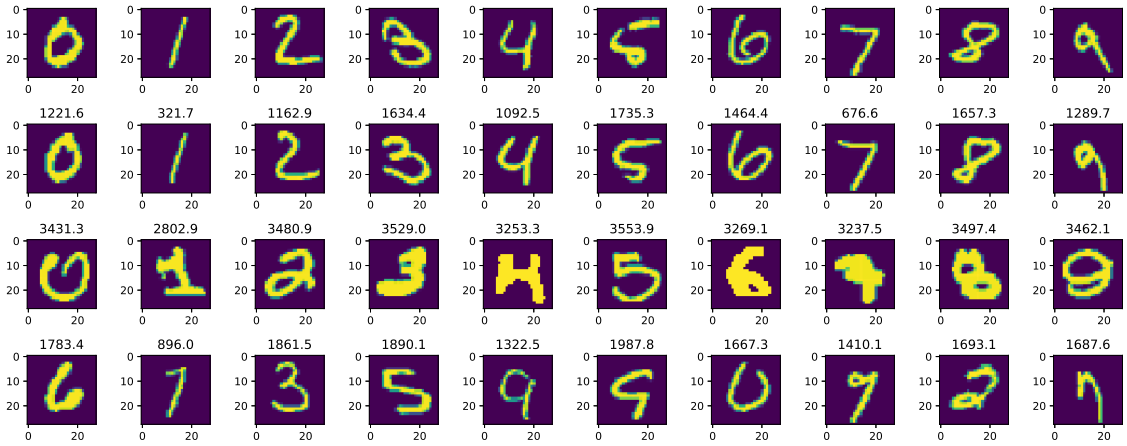
m	misclass rate	runtime using batch method	runtime using on-the-fly method
10	0.66	0.219	2.55
100	0.29	0.298	22.9
1000	0.10	1.93	too long!
10000	0.052	12.9	too long!

For KNN with $K = 10$,

m	misclass rate	runtime using batch method	runtime using on-the-fly method
10	0.90	0.441	3.73
100	0.43	0.495	22.7
1000	0.13	1.51	too long!
10000	0.055	16.3	too long!

Perhaps because using numpy's argmin and argsort are approximately the same complexity, I did not observe a huge difference in runtime between $K = 1$ and $K = 10$. But, one thing to note is that if we wanted to employ tricks, like data hashing, which can approximate and shorten the "nearest neighbor search" step, those approaches tend to be more efficient when K is small vs when K is large.

To give an example of what some of these data samples look like, below I have plotted (from top to bottom) a test candidate, the closest same label training sample, the furthest same label training sample, and the closest wrong label training sample. It certainly seems that if the train set is large enough, the test set can find a "doppleganger", e.g. somehow KNN has some "memorization capability." However, it would not be fair to say that each cluster is well-separated, as the furthest correct label has a distance much further than the closest wrong label, showing that the clusters are quite interleaved.



Challenge!

In lecture we saw that if R_{NN} is the Bayes risk of a 1-NN classifier and R^* the Bayes risk of a Bayes classifier, then the Bayes risk can be used to bound the 1-NN classifier in that $R^* \leq R_{NN} \leq 2R^*(1 - R^*)$. Here we will investigate this bound more carefully, to make sure we really understand all the components of the proof.

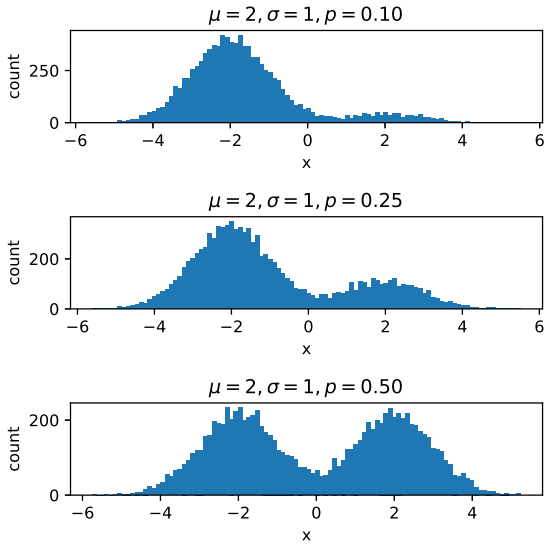
We will analyze this bound in terms of a 2-cluster model, defined as follows:

- Y is a random variable taking values in $\{+1, -1\}$, and $\mathbf{Pr}(Y = 1) = p$.
- $X \in \mathbb{R}$ is a random variable taking any value in $\mathbb{R}$, defined by a scalar Gaussian distribution $X \sim \mathcal{N}(Y \cdot \mu, \sigma)$. That is, if $Y = 1$ then X has mean μ and variance σ^2; if $Y = -1$, then X has mean $-\mu$ and variance σ^2.

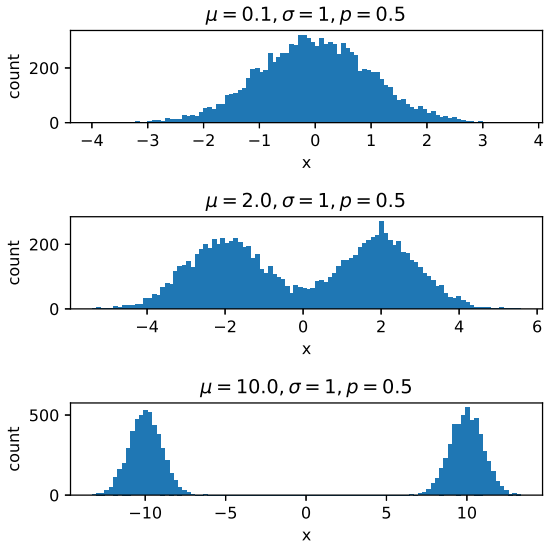
The goal will be to perform binary classification on X, and analyze how the performance of 1-NN and Bayes classifier works as we increase / decrease μ and σ.

1. **Exploring the model.** Plot some histograms of X, by drawing m datapoints and labels $x_1, ..., x_m$ and $y_1, ..., y_m$ according to our model. Use a "large enough" value of m so that the histogram well represents the model at the limit $m \to +\infty$.

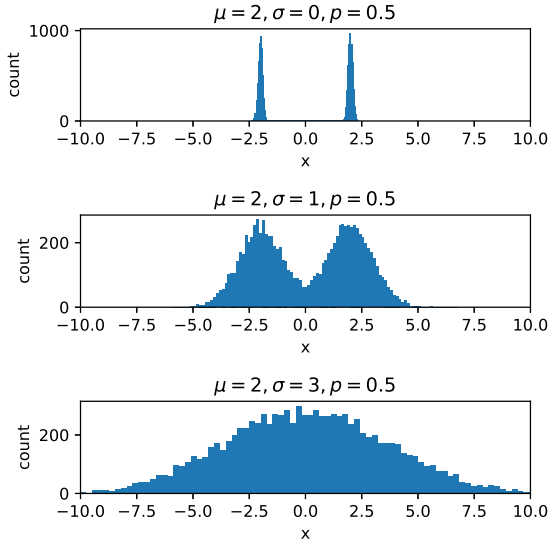
   (a) **Sweep label balance.** Do this for $\mu = 2$, $\sigma = 1$ and $p = 0.1, 0.25, 0.5$
   Ans.



   (b) **Sweep separation width.** Repeat for $\sigma = 1$, $p = 0.5$ and $\mu = 0.1, 2.0, 10.0$.
   Ans.



(c) **Sweep cluster variance.** Repeat for $\mu = 2$, $p = 0.5$ and $\sigma = 0.1, 1.0, 3.0$.
Ans.



2. **Bayes classifier.** Recall that we denote $\eta(x) = \mathbf{Pr}(Y = 1|x)$.

   (a) Write out this probability (that is, find an expression for η) in terms of μ, σ, p, and x.
   Hint: Use Bayes' rule. Additionally, you can use the property that, for two different PDFs $p_{g(X)}$ and $p_{h(X)}$, that

$$\frac{\mathbf{Pr}(g(X) = g(x))}{\mathbf{Pr}(h(X) = h(x))} = \frac{p_{g(x)}}{p_{h(x)}}.$$

   Note that in general, $\mathbf{Pr}(g(X) = g(x)) \neq p_{g(x)}$ when X is a continuous random variable! [1] **Ans.** Ok, let's use the hint!

$$\mathbf{Pr}(Y = 1|X = x) = \frac{\mathbf{Pr}(X = x|Y = 1)}{\mathbf{Pr}(X = x|Y = -1) + \mathbf{Pr}(X = x|Y = 1)}$$

[1] While in general the PDF does not tell us about the probability of a continuous variable taking a specific value, we can arrive at this equivalence of their ratios through the use of Radon-Nikodym derivatives. Think of it as something similar to chain rule: for two functions F and G applied over a measurable set A, assuming F and G are absolutely continuous, then $F(A) = G(A) \cdot \frac{\partial F(A)}{\partial G(A)}$. Anyway, we do not need to go into measure theory in this class, just rest assured that this operation in the hint is allowed!

And, using the hint further,

$$\begin{aligned}\mathbf{Pr}(Y=1|X=x) &= \frac{p_{X|y}(x,y=1)\cdot\mathbf{Pr}(y=1)}{p_{X|y}(x,y=-1)\cdot\mathbf{Pr}(y=-1)+p_{X|y}(x,y=1)\cdot\mathbf{Pr}(y=1)} \\ &= \frac{\exp(\frac{(x-\mu)^2}{2\sigma^2})}{\exp(\frac{(x-\mu)^2}{2\sigma^2})+\exp(\frac{(x+\mu)^2}{2})}\end{aligned}$$

(b) For $\mu=1, \sigma=1, p=0.25$, fill out to 2 significant digits the first three columns in this table

x	$\mathbf{Pr}(y=1\|x)$ $(\eta(x))$	$\mathbf{Pr}(y=-1\|x)$ $(1-\eta(x))$	Bayes risk $(\beta(x))$	1-NN risk
-2				
-1				
-0.5				
0				
0.5				
1				
2				

Ans.

x	$\mathbf{Pr}(y=1\|x)$ $(\eta(x))$	$\mathbf{Pr}(y=-1\|x)$ $(1-\eta(x))$	Bayes risk $(\beta(x))$	1-NN risk
-2.000	0.00607	0.994	0.00607	0.00603
-1.000	0.0432	0.957	0.0432	0.0413
-0.500	0.109	0.891	0.109	0.0973
0.000	0.250	0.750	0.250	0.188
0.500	0.475	0.525	0.475	0.249
1.000	0.711	0.289	0.289	0.205
2.000	0.948	0.0521	0.0521	0.0494

(c) Given a new vector x drawn from this distribution, describe the action of the Bayes classifier. What is the rule in choosing a label $\hat{y}=1$ or $\hat{y}=-1$?

Ans. The Bayes classifier picks the label for which the probability $\Pr(X=x|Y=y)$ is larger. The rule is therefore

$$\hat{y}(x)=\begin{cases}1 & \text{if } \eta(x)>1-\eta(x)\\ -1 & \text{if } \eta(x)\le 1-\eta(x).\end{cases}$$

(d) Write out an expression for the Bayes risk (R^*) , in terms of μ, σ, and p. Your answer may involve an integral that you do not need to evaluate.

Ans. Using this Bayes classifier, we therefore know that our Bayes risk is $\min(\eta(x), 1-\eta(x))$. Taken as an expectation over all x, this can be computed as

$$R^* = \mathbb{E}_x[\min(\eta(x), 1-\eta(x))] = \int_{-\infty}^{\infty} \frac{\min(\eta(x), 1-\eta(x))}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right) dx$$

(e) Numerically estimate the Bayes risk (R^*) for $p = 0.25$, $\sigma = 1$, $\mu = 1$. You can do this by generating 1000 points according to this distribution, calculating the Bayes risk for each point, and reporting the average.
Ans. $R^* \approx 0.126$

3. **1-NN classifier.** Now we consider the limiting case of a 1-NN classifier; that is, we consider a scenario where, for any test data point x, there exists a labeled training point z_x that is arbitrarily close to x. We assume that the training and test data are drawn i.i.d., so, conditioned on their respective labels, x and z_x are not correlated.

(a) In this regime, the probability of error should somehow be high in regions where the label could be 1 or -1 with equal probability, but pretty low when the label is more likely 1 or -1. Write an expression, in terms of p, μ, σ, and x, of the "limiting error", e.g. the error of 1-NN if there always exists a labeled point arbitrarily close to x. **Ans.** Hopefully by this point you are convinced that the 1-NN classifier has Bayes risk

$$\mathbf{Pr}(Y(Z_X) = 1|Z_X)\mathbf{Pr}(Y(X) = -1|X) = \mathbf{Pr}(Y(X) = 1|X)\mathbf{Pr}(Y(X) = -1|X) = \eta(x)(1-\eta(x)),$$

as it characterizes the "ambiguity" of a label for a particular point x. Since we already have the expression for $\eta(x)$, we're just piecing the two parts together:

$$\text{Risk}_{NN}(x) \quad = \quad \frac{\exp(\frac{(x-\mu)^2}{2\sigma^2})}{\exp(\frac{(x-\mu)^2}{2\sigma^2}) + \exp(\frac{(x+\mu)^2}{2})} \cdot \left(1 - \frac{\exp(\frac{(x-\mu)^2}{2\sigma^2})}{\exp(\frac{(x-\mu)^2}{2\sigma^2}) + \exp(\frac{(x+\mu)^2}{2})}\right).$$

The purpose of this question is to make it obvious that these quantities depend on the distributional parameters on x and y, rather than on $\eta(x)$.

(b) Fill out the last column in the table above, again for $\mu = 1, \sigma = 1, p = 0.25$.
Ans. See table above.

(c) Write out an expression for the 1-NN risk (R_{NN}) , in terms of μ, σ, and p. Your answer may involve an integral that you do not need to evaluate.
Ans. The average Bayes risk of the nearest neighbor classifier is just the expected risk over the distribution of X. This can simply be written as

$$R_{NN} = \mathbb{E}_x[\eta(x)(1-\eta(x))] = \int_{-\infty}^{\infty} \text{Risk}_{NN}(x)dx$$

(d) Numerically estimate the 1-NN risk (R_{NN}) for $p = 0.25$, $\sigma = 1$, $\mu = 1$. You can do this by generating 1000 points according to this distribution, calculating the Bayes risk for each point, and reporting the average.
Ans. $R_{NN} \approx 0.181$

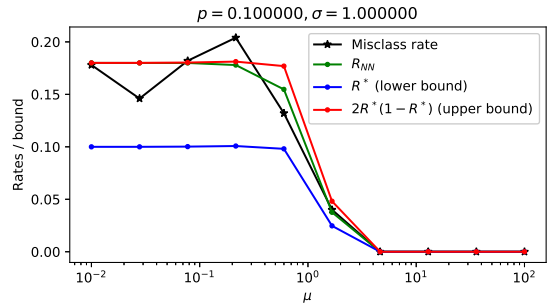
4. Ok, now we have all the pieces of the puzzle, and all the code snippets needed to do a more involved analysis! In particular, we want to see under what regimes we would expect the 1-NN risk to approach its lower bound (Bayes risk) or upper bound ($2R^*(1-R^*)$).

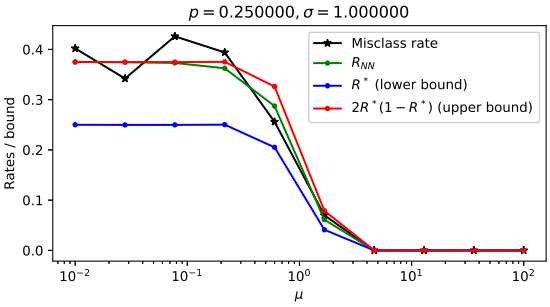
- Write a function that takes a value σ, p, and μ, returns a numerical estimate of the Bayes risk R^* and 1-NN risk R_{NN}. Pick m "large enough" (I find $m = 1000$ works fine, but for certain regimes even fewer points is sufficient.)
- Write a function that takes a value σ, p, and μ, generates $x_1, ..., x_m$ and $y_1, ..., y_m$ according to this 2-cluster model, and using the first half of the data as a train set and the second half as a test set, returns the test misclassification error for a 1-NN classifier.

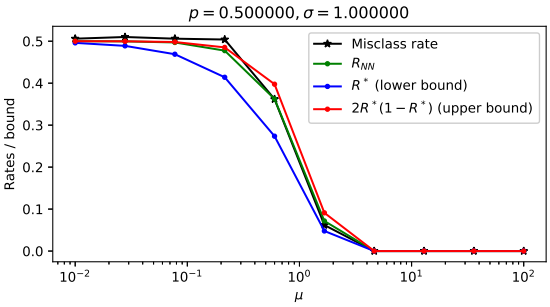
- For $\sigma = 1.$, $p = 0.25$, sweep μ as `logspace(-2,2,10) * sigma`. Plot as a function of μ the quantities R_{NN}, R^*, the upper bound $2R^*(1-R^*)$, and the misclassification rate of the implemented 1-NN. Comment on what you see.
- Pick either 2 other values of σ or 2 other values of p and repeat this experiment. How do these other parameters affect the bound and tightness? Can you venture a guess as to what kind of scenarios would hit the upper bound, and what would hit the lower bound?

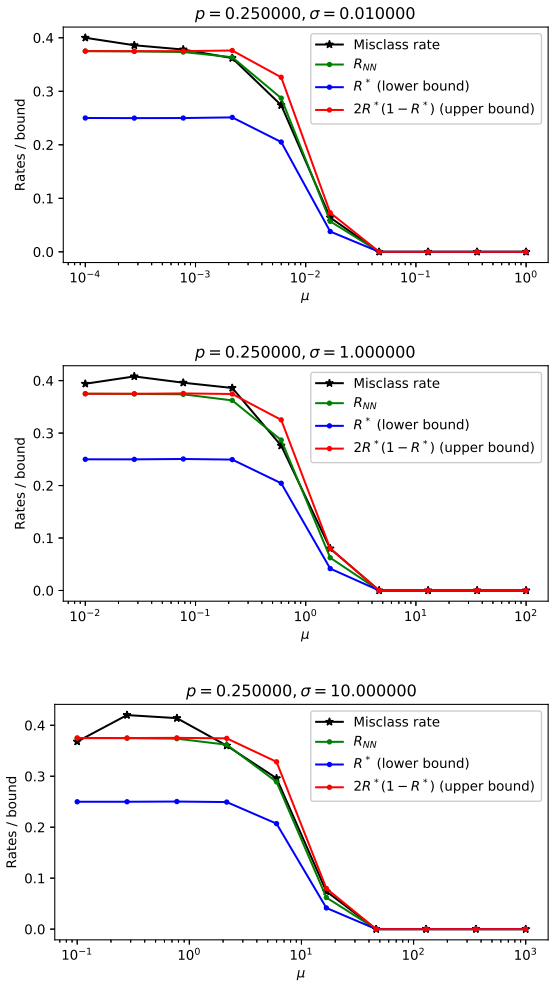
Ans.

This part is more exploratory, just to give you guys a chance to play around with parameters and test bounds. I have my plots here, where the misclassification rate is given for an experiment with $m = 1000$. Since this isn't very close to the asymptotic limit, we don't expect the misclassification rate to necessarily fall between the upper and lower bounds. However, were I to keep increasing m, this rate would converge to R_{NN}, which is squarely characterized in the bound.









In fact, it seems like all the quantities are hugging the upper bound. I was a bit surprised by this, but recall that the gap between the upper bound and the risk is exactly the variance of $\min\{\eta(x), 1 - \eta(x)\}$. For Gaussian distributions, this quantity is often very low. Perhaps for a distribution where this variance is higher, we would expect the lower bound to be more "active."