MMSE, Conditional Independence and MNIST

Submit a PDF of your answers to Canvas.

1. *Bayes for continuous feature vectors.* Let $\boldsymbol{x} \in \mathbb{R}^n \sim f(\boldsymbol{x})$ be vector of continuous random variables, and let y be a discrete random variable. Show from first principles that

$$p(y|\boldsymbol{x}) = \frac{f(\boldsymbol{x}|y)p(y)}{f(\boldsymbol{x})}.$$

Hint: Start with definition of conditional probability

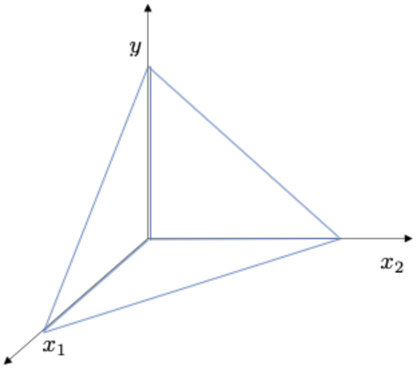
$$\mathbb{P}(A|B) = \frac{\mathbb{P}(A, B)}{\mathbb{P}(B)}$$

and use the definition of a multivariate pdf

$$f(\boldsymbol{x}) = \lim_{\Delta \to 0} \frac{\mathbb{P}(x_1 < X_1 \leq x_1 + \Delta, x_2 < X_2 \leq x_2 + \Delta, \ldots)}{\Delta^n}.$$

2. *Regression with MMSE.* Consider a feature vector $\boldsymbol{x} \in \mathbb{R}^2$ and $y \in \mathbb{R}$ with joint pdf

$$p(\boldsymbol{x}, y) = \begin{cases} 6 \text{ for } x_1, x_2, y \geq 0 \text{ and } x_1 + x_2 + y \leq 1 \\ 0 \text{ otherwise.} \end{cases}$$



You aim to find a function $f(\boldsymbol{x})$ to estimate y.

a) Find the function $f(\boldsymbol{x})$ that minimizes $E[\ell(f(\boldsymbol{x}), y)]$ under the squared error loss function.

b) What is the minimum true risk $E[\ell(f(\boldsymbol{x}), y)]$, under the squared error loss function? You can leave your answer as as integral.

3. *Bayes Nets on MNIST.* Note that Activity 12 will be helpful for completing this problem. Recall the MAP classification rule:

$$\widehat{y} = \arg\min_{y} p(\boldsymbol{x}|y)p(y)$$

By repeated application of the product rule of probability, a distribution $p(\boldsymbol{x}|y)$ can be written as:

$$p(\boldsymbol{x}|y) = p(x_1|y)p(x_2|x_1, y)p(x_3|x_1, x_2, y)p(x_4|x_1, x_2, x_3, y) \dots p(x_n|x_1, \dots, x_{n-1}, y).$$

Last time we used *Naïve Bayes* to estimate $p(\boldsymbol{x}|y)$. Recall that the class $y \in \{0, 1, \dots, 9\}$ represents the true digit, while $\boldsymbol{x} \in \{0, 1\}^{784}$ is the 28×28 black and white image that we'd like to classify. Naïve Bayes make the often poor assumption that $p(\boldsymbol{x}|y) \approx \prod_{i=1}^{n} p(x_i|y)$.

In this problem we will make a compromise to model relationships *between* pixels. More specifically, we will estimate $p(\boldsymbol{x}|y)$ by making a *conditional independence*: we will assume that the probability of a pixel, conditioned on the values of the four pixels that surround, it is independent of all the other pixels in the image.

a) Imagine the pixels are enumerated so that the x_1 is the pixel in the upper left corner of the image, x_2 is the pixel below x_1 (in the first column and second row) and so on. Simplify the expression

$$p(x_{34}|x_1, x_2, \dots, x_{33})$$

if the pixels are conditionally independent given their 4 neighbors (the pixel immediately to the right and left, and above and below).

b) Find an expression for $p(\boldsymbol{x}|y)$ given the conditional independence assumptions. You can ignore any discrepancy for edges cases (when x_i corresponds to a pixel in the first column or bottom row).

c) How many parameters do we need to estimate? How does this compare to the Naive Bayes case or the general case?

d) We can estimate $p(x_i = 0|x_j = 0, x_k = 0, y = 9)$ (for example) empirically by counting the number of times pixel i is equal to zero when $x_j = 0, x_k = 0$, vs. the number of times it is equal to 1 when $x_j = 0, x_k = 0$ among the images that are labeled $y = 9$:

$p(x_i = 0|x_j = 0, x_k = 0, y = 9) =$

$$\frac{1 + \text{count of examples with pixel } i \text{ equal to 0 where } x_j = 0, x_k = 0 \text{ from class 9}}{2 + \text{count of examples where } x_j = 0, x_k = 0 \text{ from class 9}}.$$

The 1 in the numerator and 2 in the denominator is called 'Laplace Smoothing', and deals with cases where there are no corresponding examples by assigning them an uncommitted value of 1/2.

Write code that estimates $p(x_i = 0|x_j, x_k, y)$ and $p(x_i = 0|x_j, x_k, y)$ by counting the occurrences. Note that for each pixel, you will have to consider 80 cases: 2 values for the pixel itself $x_i \in \{0, 1\}$, 4 cases for the conditionals, $x_j, x_j = \{(0, 0), (0, 1), (1, 0), (1, 1)\}$, times 10 cases for $y = 0, 1, \ldots, 9$. Store your estimates of $p(x_i|y)$ in a $2 \times 10 \times 4 \times 28 \times 28$ array.

e) Write code that computes log likelihood of a new image for each class $y = 1, \ldots, 9$. Recall that the log likelihood is given by $\log(p(\boldsymbol{x}|y))$ when provided with a test image.

f) What is the classification error of your classifier on the $10k$ test images?