**DSE 210: Probability and statistics** **Winter 2015**

# Worksheet 6 — Generative models 2

1. A man has two possible moods: `happy` and `sad`. The prior probabilities of these are:

$$\pi(\texttt{happy}) = \frac{3}{4}, \quad \pi(\texttt{sad}) = \frac{1}{4}.$$

   His wife can usually judge his mood by how talkative he is. After much observation, she has noticed that:

   - When he is happy,

$$\Pr(\text{talks a lot}) = \frac{2}{3}, \;\; \Pr(\text{talks a little}) = \frac{1}{6}, \;\; \Pr(\text{completely silent}) = \frac{1}{6}$$

   - When he is sad,

$$\Pr(\text{talks a lot}) = \frac{1}{6}, \;\; \Pr(\text{talks a little}) = \frac{1}{6}, \;\; \Pr(\text{completely silent}) = \frac{2}{3}$$

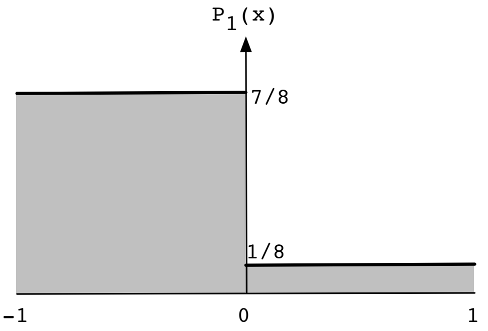
   (a) Tonight, the man is just talking a little. What is his most likely mood?

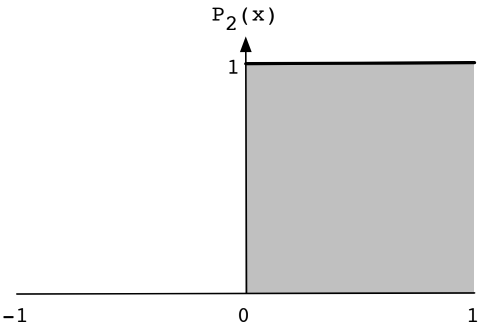
   (b) What is the probability of the prediction in part (a) being incorrect?

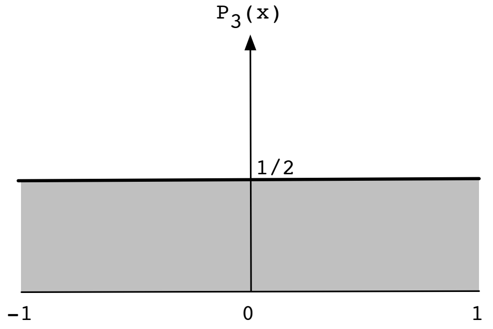
2. Suppose $\mathcal{X} = [-1, 1]$ and $\mathcal{Y} = \{1, 2, 3\}$, and that the individual classes have weights

$$\pi_1 = \frac{1}{3}, \;\; \pi_2 = \frac{1}{6}, \;\; \pi_3 = \frac{1}{2}$$

   and densities $P_1, P_2, P_3$ as shown below.







What is the optimal classifier $h^*$? Specify it exactly, as a function from $\mathcal{X}$ to $\mathcal{Y}$.

3. Would you expect the following pairs of random variables to be uncorrelated, positively correlated, or negatively correlated?

   (a) The weight of a new car and its price.

   (b) The weight of a car and the number of seats in it.

   (c) The age in years of a second-hand car and its current market value.

4. Consider a population of married couples in which every wife is exactly 0.9 of her husband's age. What is the correlation between husband's age and wife's age?

5. Each of the following scenarios describes a joint distribution $(x, y)$. In each case, give the parameters of the (unique) bivariate Gaussian that satisfies these properties.

   (a) $x$ has mean 2 and standard deviation 1, $y$ has mean 2 and standard deviation 0.5, and the orrelation between $x$ and $y$ is $-0.5$.

   (b) $x$ has mean 1 and standard deviation 1, and $y$ is equal to $x$.

6. Roughly sketch the shapes of the following Gaussians $N(\mu, \Sigma)$. For each, you only need to show a representative contour line which is qualitatively accurate (has approximately the right orientation, for instance).

   (a) $\mu = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$ and $\Sigma = \begin{pmatrix} 9 & 0 \\ 0 & 1 \end{pmatrix}$

   (b) $\mu = \begin{pmatrix} 0 \\ 0 \end{pmatrix}$ and $\Sigma = \begin{pmatrix} 1 & -0.75 \\ -0.75 & 1 \end{pmatrix}$

7. For each of the two Gaussians in the previous problem, check your answer using Python: draw 100 random samples from that Gaussian and plot it.

8. Consider the linear classifier $w \cdot x \geq \theta$, where

$$w = \begin{pmatrix} -3 \\ 4 \end{pmatrix} \quad \text{and} \quad \theta = 12.$$

Sketch the decision boundary in $\mathbb{R}^2$. Make sure to label precisely where the boundary intersects the coordinate axes, and also indicate which side of the boundary is the positive side.

9. *Handwritten digit recognition using a Gaussian generative model.* In class, we mentioned the MNIST data set of handwritten digits. You can obtain it from:

   `http://yann.lecun.com/exdb/mnist/index.html`

   In this problem, you will build a classifier for this data, by modeling each class as a multivariate (784-dimensional) Gaussian.

   (a) Upon downloading the data, you should have two training files (one with images, one with labels) and two test files. Unzip them.

   In order to load the data into Python you will find the following code helpful:

   `http://cseweb.ucsd.edu/~dasgupta/dse210/loader.py`

   For instance, to load in the training data, you can use:

   ```
   x,y = loadmnist('train-images-idx3-ubyte', 'train-labels-idx1-ubyte')
   ```

   This will set $x$ to a $60000 \times 784$ array where each row corresponds to an image, and $y$ to a length-60000 array where each entry is a label (0-9). There is also a routine to display images: use `displaychar(x[0])` to show the first data point, for instance.

   (b) Split the training set into two pieces – a training set of size 50000, and a separate *validation set* of size 10000. Also load in the test data.

   (c) Now fit a Gaussian generative model to the training data of 50000 points:

   - Determine the class probabilities: what fraction $\pi_0$ of the training points are digit 0, for instance? Call these values $\pi_0, \ldots, \pi_9$.
   - Fit a Gaussian to each digit, by finding the mean and the covariance of the corresponding data points. Let the Gaussian for the $j$th digit be $P_j = N(\mu_j, \Sigma_j)$.

   Using these two pieces of information, you can classify new images $x$ using Bayes' rule: simply pick the digit $j$ for which $\pi_j P_j(x)$ is largest.

   (d) One last step is needed: it is important to smooth the covariance matrices, and the usual way to do this is to add in $cI$, where $c$ is some constant and $I$ is the identity matrix. What value of $c$ is right? Use the validation set to help you choose. That is, choose the value of $c$ for which the resulting classifier makes the fewest mistakes on the validation set. What value of $c$ did you get?

   (e) Turn in an iPython notebook that includes:

   - All your code.
   - Error rate on the MNIST test set.
   - Out of the misclassified test digits, pick five at random and display them. For each instance, list the posterior probabilities $\Pr(y|x)$ of each of the ten classes.