1. (20 points)

Let $X$ be the time to failure of a widget and $F$ be the index of the factory that made the widget. The probability that a widget comes from factory $f \in \{1, \ldots, K\}$ is known to be $q_k = P(F = k)$. If $F = k$ is given, then $p(x|F = k) = p_k(x)$ is exponential with parameter $\alpha_k$. For each of the conditions below, describe how you would estimate $\alpha_k$, including equations.

(a) You are given i.i.d. training samples with factory labels $D_1 = \{(x_t, f_t); t = 1, \ldots, N_1\}$.

**Solution:** With labeled training data, you can use standard maximum likelihood estimation. You could look up the answer for the exponential distribution in your notes, or derive it. (The solution below doesn’t use sufficient statistics, but it is fine to use them.) Letting $D_{1,k}$ be the subset of data corresponding to label $k$, and $N_k = |D_{1,k}|$.

$$
\hat{\alpha}_k = \operatorname*{argmax}_\alpha L(D_{1,k}, \alpha) = \operatorname*{argmax}_\alpha \log p(D_{1,k}|\alpha) = \operatorname*{argmax}_\alpha \sum_{t: f_t=k} \log p(x_t|\alpha) = \operatorname*{argmax}_\alpha \sum_{t: f_t=k} \log(\alpha e^{-\alpha x_t}) = \operatorname*{argmax}_\alpha \left[ N_k \log \alpha - \sum_{t: f_t=k} \alpha x_t \right] = \frac{N_k}{\sum_{t: f_t=k} x_t} \tag{1}
$$

(b) You are given i.i.d. samples $D_2 = \{x_t; t = 1, \ldots, N_2\}$ with no knowledge of the factory that they came from.

**Solution:** In this case, you need to use the EM algorithm. The factory that they came from is the hidden variable, and the distribution of the observed data $p(x)$ is a mixture of exponentials: $p(x) = \sum_k q_k p_k(x)$. Although $q_k$ is known, you still need EM to maximize $p(x)$, but you can use the standard mixture update equations without re-estimating $q_k$.

**E step:** Estimate the factory probability for each $x_t$:

$$
\gamma_t^{(l)}(k) = p(F_t = k|x_t) = \frac{\alpha_k^{(l)} \exp(-\alpha_k^{(l)} x_t) q_k}{\sum_j \alpha_j^{(l)} \exp(-\alpha_j^{(l)} x_t) q_j}
$$

**M step:** Use weighted sum observations in the update of $\alpha_k$ for all $k$:

$$
\alpha_k^{(l)} = \frac{\sum_t \gamma_t^{(l)}(k)}{\sum_t \gamma_t^{(l)}(k) x_t}
$$

(c) You are given both $D_1$ and $D_2$, and they contain independent samples.
Solution: Because of $D_2$, you again need the EM algorithm. The idea of using $D_1$ for the initial estimate and $D_2$ for subsequent EM passes is not great, because you are not fully leveraging your labeled data. Averaging the two estimates from parts (a) and (b) is not great either, because (b) could be higher quality if it leverages labeled data. The best solution is to use EM with both data sets in each iteration, with probability 1 for the known factory cases.

E step: Estimate the factory probability for each $x_t \in D_2$ as in part (b). Use $\gamma_l^{(l)}(k) = 1$ for $x_t \in D_{1,k}$ and zero for all other $x_t$ in $D_1$.

M step: Use both data sets in the update of $\alpha_k$ for all $k$:

$$
\alpha_k^{(l)} = \frac{N_k + \sum_{x_t \in D_2} \gamma_l^{(l)}(k)}{\sum_{x_t \in D_{1,k}} x_t + \sum_{x_t \in D_2} \gamma_l^{(l)}(k)x_t}
$$

(d) In which of the above cases, can you use a class-dependent sample average of $\{x_t\}$ as a sufficient statistic?

Solution: You can use the sample average as a sufficient statistic for $D_1$ in both parts (a) and (c), though in part (c) it is only part of the solution.

2. (20 points)
Consider two families of models for modeling one-dimensional data:

- $H_1: N(\mu_1, 1), -5 \leq \mu_1 \leq 5$
- $H_2: N(\mu_2, 2), -1 \leq \mu_2 \leq 1$

In other words, one ($H_1$) is a family of “narrow” Gaussians with a larger range for possible mean values, whereas the other ($H_2$) is a family of “broad” Gaussians with a smaller range of possible mean values. Assume the priors $p(\mu_i|H_i)$ are uniform.

For training, we are given two observations $D = \{-2, 2\}$ and wish to compute the evidence for each model in order to do Bayesian model selection:

(a) What are the maximum likelihood estimates $\hat{\mu}_1$ and $\hat{\mu}_2$ for the means $\mu_1$ and $\mu_2$, respectively?

The maximum likelihood estimate of the mean of a Gaussian is simply the sample mean. So $\hat{\mu}_1 = \hat{\mu}_2 = \frac{1}{2}(2 - 2) = 0$.

(b) Show that the variance of the sample mean is $\frac{\sigma_x^2}{n}$ when the data samples are i.i.d. with variance $\sigma_x^2$. (Recall that $Var(X_1 + X_2) = Var(X_1) + Var(X_2) + Cov(X_1, X_2)$.)

\[
Var[\hat{\mu}] = Var\left[\frac{1}{n}\sum_{i=1}^{n} x_i\right]
\]

\[
= \frac{1}{n^2} Var\left[\sum_{i=1}^{n} x_i\right] = \frac{1}{n^2} \sum_{i=1}^{n} Var[x_i] \quad \text{(the cov terms are zero due to i.i.d. samples)}
\]

\[
= \frac{1}{n^2} \sum_{i=1}^{n} \sigma_x^2 = \frac{1}{n^2} n \sigma_x^2 = \frac{\sigma_x^2}{n}
\]
(c) Let $\sigma^2_{\mu_1|D}$ be the variance of the sample mean when the underlying distribution is $N(\mu, \sigma)$. Find $\sigma^2_{\mu_1|D}$ and $\sigma^2_{\mu_2|D}$, the variance of $\hat{\mu}_1$ and $\hat{\mu}_1$ given data $D$, respectively.

Use the above formula $\frac{\sigma^2}{n}$, where $n = 2$ for both cases and the model variances are plugged in for $\sigma^2$. Thus, $\sigma^2_{\mu_1|D} = 1/2$, and $\sigma^2_{\mu_2|D} = 2/2 = 1$.

(d) Using the following evidence equation: $P(D|H_i) = P(D|\hat{\mu}, H_i)P(\hat{\mu}|H_i)\sigma_{\mu|D}$, roughly compute the evidence for $H_1$ and $H_2$. Which model would you choose? (Note: $N(0,2)$ evaluated at $x = 2$ is 0.12; $N(0,1)$ evaluated at $x = 2$ is 0.05.)

For Model 1, $P(D|\hat{\mu}, H_1) = (0.05)^2$, $P(\hat{\mu}|H_1) = \frac{1}{10}$ (due to uniform over -5,5 range), $\sigma^2_{\mu_1|D} = 1/2$. For Model 2, $P(D|\hat{\mu}, H_2) = (0.12)^2$, $P(\hat{\mu}|H_1) = \frac{1}{2}$ (due to uniform over -1,1 range), $\sigma^2_{\mu_2|D} = 1$. $P(D|H_2)$ is clearly larger, so we select model 2.

3. (15 points)
Ridge regression is an example of regularization, where you include a penalty term in the objective to improve performance of a complex model with sparse training data. Another example for a Gaussian classifier would be to combine an estimated class-dependent covariance with the shared class-independent estimate, as in:

$$\hat{\Sigma}_k(\alpha) = \alpha \hat{\Sigma}_k + (1-\alpha)\hat{\Sigma}$$

(This is sometimes also called “smoothing”.)

(a) What is the effect on this strategy in terms of bias and variance of the resulting models?

The bias would possibly increase but the variance would decrease, because the shared class-independent covariance would have less variance due to using more data but would shift the estimate (increase bias) if the class-dependent models truly are different.

(b) For a 2-class scalar case with equal priors and Gaussian distributions illustrated below, indicate: the Bayes-optimal decision boundaries for the MAP decision rule, hypothetical decision boundaries for the class-dependent estimated distributions, and corresponding decision boundaries for the smoothed covariance. For the hypothetical estimates, imagine a case where the random sample is such that the smoothing helps.
For all three cases, you need to have two boundaries (one decision region is broken up), and the boundaries are at the point where the curves cross. For the hypothetical estimates, the left boundary of the smoothed case should be closer to the optimal boundary. In addition the smoothed case should have a bigger decision region for the second (narrow) Gaussian, and the two hypothetical cases should have boundaries that are symmetric about a common mean.

4. (10 points)
Sketch an example of three 2-dimensional Gaussian class distributions (with ovals to illustrate the Gaussians), where using principal components analysis (PCA) to transform to 1-dimension is better than using the original feature set and where the optimal classifier is quadratic in both the 1-D and 2-D spaces. Assume that you know the true form of the distribution, but need to estimate the parameters from training data. Explain why 1 dimension is better and give an example scenario where this situation might occur.

An example that requires a quadratic classifier is one where the covariances (or variances in 1 dimension) are not all the same, as illustrated in the figure below. In this example, the second dimension is a zero-mean Gaussian that is independent of all the classes and of the first dimension, so it has no information about the class. If you had the true distributions, then you would know it was uncorrelated and it wouldn't hurt to keep it, but if you didn't have the true distributions then the random statistical variation of the estimate might give you different covariances for the different classes that introduce errors in classification. This is related to the concept of having lower variance for a lower complexity model given a fixed amount of data. Another reason that 1-D is better is less computation in the classification process. (Note that if you have covariances with tilts (i.e. not a diagonal covariance), then the correlation between the two variables helps with classification, and then it would hurt performance to go to 1-D.) Two examples where a transformation would actually help performance include: i) the second variable has no information about the class (useless feature), or ii) the second variable is simply a noisy version of the other (i.e. \( y = ax + n \) where \( n \) is noise).
5. (15 points)
Given the same test set and two binary classifiers, is the following condition possible?

- Classifier 1 has higher accuracy than Classifier 2, BUT
- Classifier 2 has higher precision and higher recall than Classifier 1

If this is not possible, please clearly explain why. If this is possible, please provide an example of a 2 × 2 table for each classifier, each giving counts of true positives, false positives, true negatives, and false negatives.

**Solution:** It is possible for classifier 1 to have higher accuracy but lower precision and recall than classifier 2. In general, whether a classifier is better than the other depends on the measure used to judge “better”, so it is important to make sure that you choose the measure that matches your problem. Define the following quantities for a binary classification problem:

<table>
<thead>
<tr>
<th>Truth:</th>
<th></th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prediction: 0</td>
<td>$n_{00}$</td>
<td>$n_{01}$</td>
</tr>
<tr>
<td>Prediction: 1</td>
<td>$n_{10}$</td>
<td>$n_{11}$</td>
</tr>
</tbody>
</table>

- Accuracy = $\frac{n_{00} + n_{11}}{n_0 + n_1}$ (i.e. the fraction of correct classifications, regardless of class).
- Precision = $\frac{n_{11}}{N_1}$ (i.e. the fraction of correct (true) positives out of total predicted positives)
- Recall = $\frac{n_{11}}{n_1}$ (i.e. the fraction of true positives out of total positives in the dataset)

Note that the denominator for accuracy remains constant for classifier 1 and 2; this is the total number of samples. Similarly, the denominator for recall remains constant for both classifiers, since it is the total number of positive samples. Can we adjust the terms $n_{00}, n_{01}, n_{10}, n_{11}$ for both classifiers (while keeping $n_0, n_1$ constant) such that the above criteria holds? Observe that neither precision nor recall uses $n_{00}$: this is the key quantity to help us satisfy the criteria.

To be more methodological, let’s write out the constraints:

- Accuracy of classifier 1 = $\frac{n_{00} + n_{11}}{n_0 + n_1}$. Accuracy of classifier 2 = $\frac{n_{00} + n_{11}}{n_0 + n_1}$. Accuracy 1 is better than Accuracy 2 implies that $n_{11} > n_{11}$ since the denominators are equal.
- Recall of classifier 1 = $\frac{n_{11}}{n_1}$. Recall of classifier 2 = $\frac{n_{11}}{n_1}$. Recall 1 worse than Recall 2 implies that $n_{11} < n_{11}$ since the denominators are equal.
- Combine the above two constraints. If $n_{11} > n_{11}$ but $n_{11} < n_{11}$, then $n_{00} > n_{00}$.

More precisely, if $n_{11} = n_{11} + x$ for some integer $x > 0$, then $n_{00} = n_{00} + x + y$ for some integer $y > 0$. Put simply, classifier 2 has more true positives (lower right corner of the table) than classifier 1; classifier 1 has more true negatives (upper left corner of the table) than classifier 2.

Using the above intuition, we can start filling in the tables for $n_{00}$, etc. (We can also specify the remaining constraint involving Precision 2 > Precision 1, but it is simpler to just avoid violating it while filling in the tables.) To start, let’s set $n_0 = 5, n_1 = 5$ since these are easy numbers to work with. We want the true negatives to be high for classifier 1, and the true
positives to be high for classifier 2. One possibility is illustrated below, where the accuracy of
classifier 1 is \((4 + 0)/10 = 0.4\), and the accuracy of classifier 2 is \((0 + 3)/10 = 0.3\). We’ve
satisfied the accuracy criteria, so now we just fill in the blanks to ensure that \(n_0, n_1\) sum up
correctly:

<table>
<thead>
<tr>
<th>Truth</th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Classifier 1</td>
<td>Classifier 2</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Prediction: 0</td>
<td>4</td>
<td>(N_0)</td>
<td>0</td>
<td>(N_0)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Prediction: 1</td>
<td>0</td>
<td>(N_1)</td>
<td>3</td>
<td>(N_1)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note that precision and recall of classifier 1 is 0, since \(n_{00} = 0\). Precision and recall of
classifier 2 is \(\frac{3}{5}\) and \(\frac{3}{8}\), respectively. We have satisfied all criteria by having classifier 2 be
more biased to predict positives, while classifier 1 is more biased to predict negatives.

As a final remark, the situation demonstrated in this problem can occur often in practice,
especially for datasets with high skew (i.e. \(n_0 >> n_1\)). Since there are many negative samples,
a classifier that predicts negative 100 percent of the time can still achieve high accuracy.
However, it would lose in precision and recall. What is the right measure depends on the cost
of misclassification and the problem.

6. (10 points)
For which of the following problems would you want to use a single held out set of data vs.
cross-validation? Assume that your total amount of data is fixed. Justify your answers.

(a) Feature selection with a wrapper approach where you have a large amount of labeled
training data and an expensive learning algorithm.

**Solution:** Wrapper approach can be expensive, so to reduce amount of computation,
single held-out is preferred.

(b) Performance estimation when you have a small amount of labeled training data.

**Solution:** When there is little data, we cannot afford to carve out a held-out
set. Thus cross-validation is the preferred method for performance estimation.

(c) Performance estimation when you have a large amount of data and want an unbiased
estimate.

**Solution:** When there is a lot of data, a single held-out can give unbiased estimate in performance estimation. If one can afford the computational overhead,
then cross-validation is equally good in terms of unbiased performance estimate—just make sure the training segment is sufficiently large.

(d) Choosing a shrinkage parameter for smoothing covariance matrices as in problem 3 when you have a medium amount of data.

Solution: Either method would work.

(e) Choosing model order (e.g. number of mixtures, number of hidden nodes in a neural net) when the deadline for your design is in a few hours.

Solution: K-fold cross-validation is $K$ times the work of single held-out, so under stringent time constraints, single held-out is preferred.

7. (10 points)
Two classifiers are tested on the same test set, which has 500 points. They make the same decision for all of the test samples except 20. Classifier A is wrong on $K < 10$ of these samples, and classifier B is wrong on the rest. What is the maximum $K$ (if any) for which you could be 95% confident that classifier A is better than classifier B? You can use the figure below from DHS.

![Figure 9.10](image)

**FIGURE 9.10.** The 95% confidence intervals for a given estimated error probability $\hat{p}$ can be derived from a binomial distribution of Eq. 38. For each value of $\hat{p}$, the true probability has a 95% chance of lying between the curves marked by the number of test samples $n$. The larger the number of test samples, the more precise the estimate of the true probability and hence the smaller the 95% confidence interval. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

Solution: Since the two classifiers are tested on the same data, you need to look only at the number of cases that are different between the two. In other words, you want to test the significance of $\hat{p} = K/20 < 0.5$. The point on the $\hat{p}$ axis where the
upper $N = 20$ curve is below the 0.5 horizontal line is roughly around $\hat{p} = .27$, so $\hat{p} < 5/20 = .25$ or $K < 5$ will make you at least 95% confident that $A$ is better than $B$. 