

CS573: Homework 4 Solution

Part of the solution is based on Wahbeh Q.'s submission

1 Between cluster distances (4 pts)

Let cluster C_i contain n_i samples, and let d_{ij} be some measure of distance between two clusters C_i and C_j . In general, one might expect that if C_i and C_j are merged to form a new cluster C_k , then the distance from C_k to some other cluster C_h is not simply related to d_{hi} and d_{hj} . However, consider the equation:

$$d_{hk} = \alpha_i d_{hi} + \alpha_j d_{hj} + \beta d_{ij} + \gamma |d_{hi} - d_{hj}|$$

Show that the following choices for the coefficients $\alpha_i, \alpha_j, \beta, \gamma$ lead to the distance functions indicated.

1. Single-link: $\alpha_i = \alpha_j = 0.5, \beta = 0, \gamma = -0.5$

Sample solution:

$$\begin{aligned} d_{hk} &= \alpha_i d_{hi} + \alpha_j d_{hj} + \beta d_{ij} + \gamma |d_{hi} - d_{hj}| \\ &= 0.5d_{hi} + 0.5d_{hj} - 0.5|d_{hi} - d_{hj}| \\ &= \begin{cases} 0.5d_{hi} + 0.5d_{hj} - 0.5d_{hi} + 0.5d_{hj}, & \text{if } d_{hi} \geq d_{hj} \\ 0.5d_{hi} + 0.5d_{hj} - 0.5d_{hj} + 0.5d_{hi}, & \text{if } d_{hi} < d_{hj} \end{cases} \\ &= \begin{cases} d_{hj}, & \text{if } d_{hi} \geq d_{hj} \\ d_{hi}, & \text{if } d_{hi} < d_{hj} \end{cases} \\ &= \min\{d_{hj}, d_{hi}\} \\ &= \min\left\{\min_{u \in C_h, v \in C_j} d(u, v), \min_{u \in C_h, v \in C_i} d(u, v)\right\} \\ &= \min_{u \in C_h, v \in C_j \cup C_i} \{d(u, v)\} \\ &= \min_{u \in C_h, v \in C_k} \{d(u, v)\} = d_{hk} \end{aligned}$$

2. Complete-link: $\alpha_i = \alpha_j = 0.5, \beta = 0, \gamma = +0.5$

Sample solution:

$$\begin{aligned}d_{hk} &= \alpha_i d_{hi} + \alpha_j d_{hj} + \beta d_{ij} + \gamma |d_{hi} - d_{hj}| \\&= 0.5d_{hi} + 0.5d_{hj} + 0.5|d_{hi} - d_{hj}| \\&= \begin{cases} 0.5d_{hi} + 0.5d_{hj} + 0.5d_{hi} - 0.5d_{hj}, & \text{if } d_{hi} \geq d_{hj} \\ 0.5d_{hi} + 0.5d_{hj} + 0.5d_{hj} - 0.5d_{hi}, & \text{if } d_{hi} < d_{hj} \end{cases} \\&= \begin{cases} d_{hi}, & \text{if } d_{hi} \geq d_{hj} \\ d_{hj}, & \text{if } d_{hi} < d_{hj} \end{cases} \\&= \max\{d_{hj}, d_{hi}\} \\&= \max\left\{\max_{u \in C_h, v \in C_j} d(u, v), \max_{u \in C_h, v \in C_i} d(u, v)\right\} \\&= \max_{u \in C_h, v \in C_j \cup C_i} \{d(u, v)\} \\&= \max_{u \in C_h, v \in C_k} \{d(u, v)\} = d_{hk}\end{aligned}$$

3. Average-link: $\alpha_i = \frac{n_i}{n_i + n_j}, \alpha_j = \frac{n_j}{n_i + n_j}, \beta = \gamma = 0$

Sample solution:

$$\begin{aligned}d_{hk} &= \alpha_i d_{hi} + \alpha_j d_{hj} + \beta d_{ij} + \gamma |d_{hi} - d_{hj}| \\&= \frac{n_i}{n_i + n_j} d_{hi} + \frac{n_j}{n_i + n_j} d_{hj} \\&= \frac{n_i}{n_i + n_j} \left(\frac{\sum_{u \in C_h, v \in C_i} d(u, v)}{n_i} \right) + \frac{n_j}{n_i + n_j} \left(\frac{\sum_{u \in C_h, v \in C_j} d(u, v)}{n_j} \right) \\&= \frac{\sum_{u \in C_h, v \in C_i \cup C_j} d(u, v)}{n_i + n_j} \\&= \frac{\sum_{u \in C_h, v \in C_k} d(u, v)}{n_k} = d_{hk}\end{aligned}$$

4. Between-cluster distance (i.e., squared Euclidean distance between centroids):
 $\alpha_i = \frac{n_i}{n_i + n_j}, \alpha_j = \frac{n_j}{n_i + n_j}, \beta = -\alpha_i \alpha_j, \gamma = 0$

Sample solution:

Let m_i denote the centroid of C_i , which is $\frac{1}{n_i} \sum_{u \in C_i} u$. Furthermore, let d_{Eu}^2 denote the squared Euclidean distance

$$\begin{aligned}
d_{hk} &= \alpha_i d_{hi} + \alpha_j d_{hj} + \beta d_{ij} + \gamma |d_{hi} - d_{hj}| \\
&= \frac{n_i}{n_i + n_j} d_{hi} + \frac{n_j}{n_i + n_j} d_{hj} - \frac{n_i n_j}{(n_i + n_j)^2} d_{ij} \\
&= \frac{n_i d_{Eu}^2(m_h, m_i)}{n_i + n_j} + \frac{n_j d_{Eu}^2(m_h, m_j)}{n_i + n_j} - \frac{n_i n_j d_{Eu}^2(m_i, m_j)}{(n_i + n_j)^2} \\
&= \frac{n_i (m_h - m_i)^2}{n_i + n_j} + \frac{n_j (m_h - m_j)^2}{n_i + n_j} - \frac{n_i n_j (m_i - m_j)^2}{(n_i + n_j)^2} \\
&= \frac{n_i m_h^2 - 2n_i m_h m_i + n_i m_i^2 + n_j m_h^2 - 2n_j m_h m_j + n_j m_j^2}{n_i + n_j} - \frac{n_i n_j (m_i - m_j)^2}{(n_i + n_j)^2} \\
&= \frac{m_h^2 (n_i + n_j) + n_i m_i^2 + n_j m_j^2 - 2m_h (\sum_{u \in C_i} u + \sum_{u \in C_j} u)}{n_i + n_j} - \frac{n_i n_j (m_i - m_j)^2}{(n_i + n_j)^2} \\
&= \frac{m_h^2 (n_i + n_j) + n_i m_i^2 + n_j m_j^2 - 2m_h (\sum_{u \in C_i \cup C_j} u)}{n_i + n_j} - \frac{n_i n_j (m_i - m_j)^2}{(n_i + n_j)^2} \\
&= m_h^2 - 2m_h m_k + \frac{n_i m_i^2 + n_j m_j^2}{n_i + n_j} - \frac{n_i n_j (m_i - m_j)^2}{(n_i + n_j)^2} \\
&= m_h^2 - 2m_h m_k + \frac{(n_i + n_j) n_i m_i^2 + (n_i + n_j) n_j m_j^2 - n_i n_j (m_i - m_j)^2}{(n_i + n_j)^2} \\
&= m_h^2 - 2m_h m_k + \frac{n_i^2 m_i^2 + 2n_i n_j m_i m_j + n_j^2 m_j^2}{(n_i + n_j)^2} \\
&= m_h^2 - 2m_h m_k + \frac{(n_i m_i + n_j m_j)^2}{(n_i + n_j)^2} \\
&= m_h^2 - 2m_h m_k + \left(\frac{\sum_{u \in C_i \cup C_j} u}{n_i + n_j} \right)^2 \\
&= m_h^2 - 2m_h m_k + m_k^2 = (m_h - m_k)^2 \\
&= d_{Eu}(m_h, m_k) = d_{hk}
\end{aligned}$$

2 Clustering theorem (4 pts)

Read the paper: J. Kleinberg (2002). An Impossibility Theorem for Clustering. In Proceedings of the 16th conference on Neural Information Processing Systems. Explain its main result. Is there a hope for providing a good clustering framework/algorithm?

Sample solution:

The main result of this paper is an impossibility theorem stating that there is no clustering function satisfying a set of three desirable properties for clustering together. Namely, there exists no clustering function that satisfies all of the following properties:

- Scale Invariance: the clustering function should not be sensitive for changes in the units of distance measurement.
- Richness: every partition of the input set can be produced by the clustering function using some distance measure.
- Consistency: If the dataset was transformed in such a way that the distances between points inside a cluster shrink and distances between points in different clusters expand, then the clustering function should return the same result.

Nonetheless, the paper shows that there exists clustering functions that can satisfy any two of the three properties. An example is a single-linkage clustering procedure for which the paper shows that by choosing different stopping conditions, different subsets of the properties can be satisfied. The paper also shows that none of the functions in the class of centroid-based clustering satisfies the consistency property. Furthermore, the paper shows that some clustering functions do satisfy relaxations of the different properties.

This however, does not eliminate hope for a good clustering algorithm. It is all relative to the definition of ‘good’ which is often tied to the domain (or dataset) to which the clustering is applied. While the above properties are indeed desirable if one requires a general clustering framework, some properties might not be needed in some domains. For instance, if there exists a fixed distance function that needs to be used in a particular domain, then scale-invariance might not be a needed property. Similarly, if some partitions of the input dataset are not desirable or do not make sense, then requiring richness might not be needed. Ultimately, the efficacy of the clustering algorithm is closely tied to how well it performs on a particular dataset rather than whether it satisfies a particular property. Moreover, one can run different clustering algorithms on the same dataset, each algorithm satisfying a different set of properties, and use some combination of the results of each in the analysis depending on what properties are desirable. Finally, the paper shows that we can consider some relaxations of the properties; in some cases, these relaxations might be sufficient for the dataset under consideration.

3 Spectral Clustering (8 pts)

In this problem we will analyze the operation of a variant of spectral clustering methods on two datasets shown in Figure 1. For each of the datasets (unless directed otherwise) please answer the following questions.

1. The first step is to build an affinity matrix. The matrix defines the degree of similarity between points.
 - (a) Suppose we use the L2 norm to construct the following affinity matrix (let x_i denote the i th datapoint):

$$A(i, j) = A(j, i) = \begin{cases} 1 & \text{if } |x_i - x_j|_2 < \theta \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

What θ value would you choose and why?

Sample solution:

In general, you want to choose such a parameter that the similarity between points in different clusters is close to 0, while the similarity between neighboring points in the same cluster is close to 1. The answers to this question are based purely on eye-balling. So, for the dataset in Figure 1(a), θ between about 2.5 and 3 will result in an ideal case. For the dataset in Figure 1(b) it is less clear, but a value of $\theta = 0.5$, for example, will give us what we want.

- (b) Suppose instead we use Gaussian kernel for our affinity matrix:

$$A(i, j) = \exp\left(-\frac{|x_i - x_j|_2}{2\sigma^2}\right) \quad (2)$$

What σ value would you choose and why?

Sample solution:

For the Gaussian kernel, we want to set θ to obtain the same effect. So, for example, for the dataset in Figure 1(a) $\theta = 0.5$ and for the dataset in Figure 1(a) $\theta = 0.3$ should separate points reasonably.

2. The second step is to compute first k dominant eigenvectors of the affinity matrix, where k is the number of clusters we want to have. For the dataset in Figure 1(a) and the affinity matrix defined by equation 1 is there a value of θ for which you can compute analytically eigenvalues corresponding to the first two dominant eigenvectors? If not, explain why not. If yes, compute and write these eigenvalues down, and describe the corresponding eigenvectors.

Sample solution:

Consider $\theta = 2.6$. It will result in the affinity matrix that is a block matrix:

$$A = \begin{bmatrix} 1_{n \times n} & 0 \\ 0 & 1_{m \times m} \end{bmatrix}$$

where $1_{n \times n}$ is a block of ones of size n by n , and $1_{m \times m}$ a block of ones of size m by m (n is the number of points in left cluster, and m is the number of points in the right cluster). A has a first eigenvalue $\lambda_1 = n$ (with a corresponding eigenvector $v_1 = [1 \dots 1_n 0 \dots 0]^T$), and a second eigenvalue $\lambda_2 = m$ (with a corresponding eigenvector $v_2 = [0 \dots 0_n 1 \dots 1]^T$). These are the only eigenvalues of A , since $\text{rank}(A)$ is clearly 2.

3. The third step is to construct a matrix Y by placing k dominant eigenvectors (from above) into columns and re-normalizing the rows (to make each row a unit vector). Then we can cluster the rows of Y into k clusters using K-means (or a similar algorithm). For the dataset in Figure 1(a), the affinity matrix defined by equation 1, and the eigenvectors from above, write down your best guess for the coordinates of $k = 2$ cluster centers.

Sample solution:

Given the eigenvectors: $v_1 = [1 \dots 1_n 0 \dots 0]^T$ and $v_2 = [0 \dots 0_n 1 \dots 1]^T$, we have:

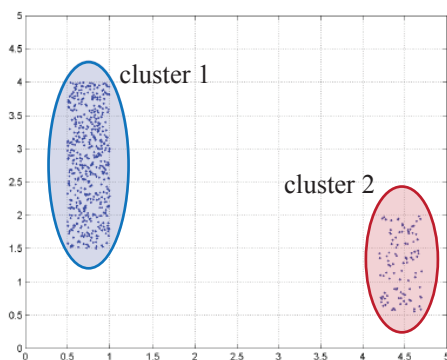
$$Y = \begin{pmatrix} 1 & 0 \\ \vdots & \vdots \\ 1_n & 0_n \\ 0 & 1 \\ \vdots & \vdots \\ 0_m & 1_m \end{pmatrix}$$

If we run K-means on the rows of Y , we get two clusters with centers at $\langle 1, 0 \rangle$ and $\langle 0, 1 \rangle$.

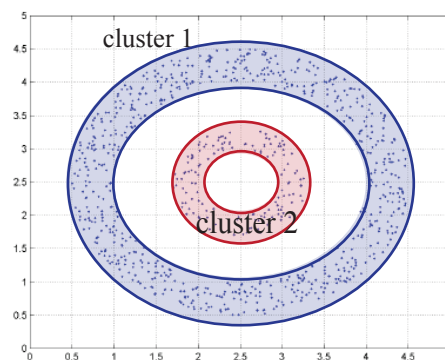
4. Finally, given the clusters on matrix Y , a point x_i is declared to be in cluster j iff the i th row of Y is in cluster j .
 - (a) What are the final clusters you would expect to obtain for each of the datasets? Provide a rough sketch of the clusters to give an idea.

Sample solution:

It will cluster connected components together resulting in perfect clustering.



(a)

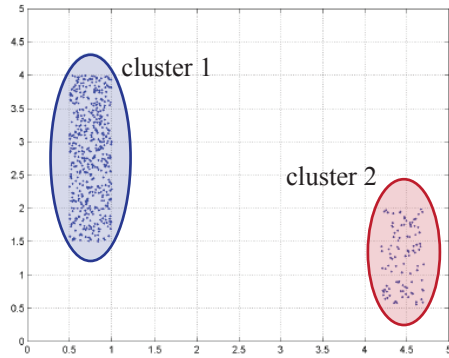


(b)

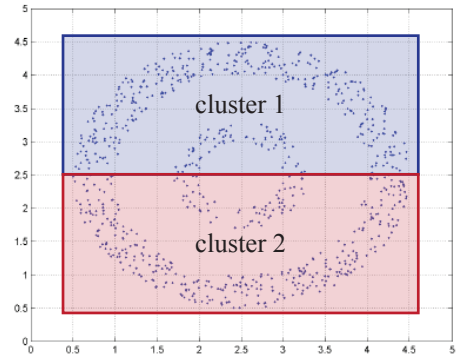
- (b) What are the clusters you would expect to obtain if using EM algorithm for Gaussian Mixture Models with 2 clusters? Also provide a rough sketch of the clusters.

Sample solution:

The dataset in Figure 1(a) should be clustered perfectly, while the dataset in Figure 1(b) will probably be split in two clusters - half of the inner and outer circles in one cluster, and the other halves of the circles in the second cluster. Thus, the dataset in Figure 1(b) is clustered incorrectly.



(a)



(b)

4 Gaussian Mixture Models and K-means (6 pts)

1. You are given a Gaussian mixture model, and all its class probabilities and Gaussian mean locations are learned using EM, but the covariance matrices are forced to be the identity matrices for each class. Rather than using a mixture of Gaussians, you evaluate the probability of each of the K classes given the datapoint and take the the cluster with the highest probability to be the cluster that produced the point. Is this equivalent to doing using a K-means model?

Sample solution:

No. GMM uses prior class probabilities $P(w_i)$ for evaluating the probability of each of the K classes given the datapoint, while K-means does not consider prior class probabilities (or it can be deemed as all the prior class probabilities are equal.)

2. Suppose you've done K-means and your K is equal to your number of data points with each cluster defined by a single datapoint. Say that you classify test data points as part of the cluster that they would belong to according to your distance metric. What model is this equivalent to?

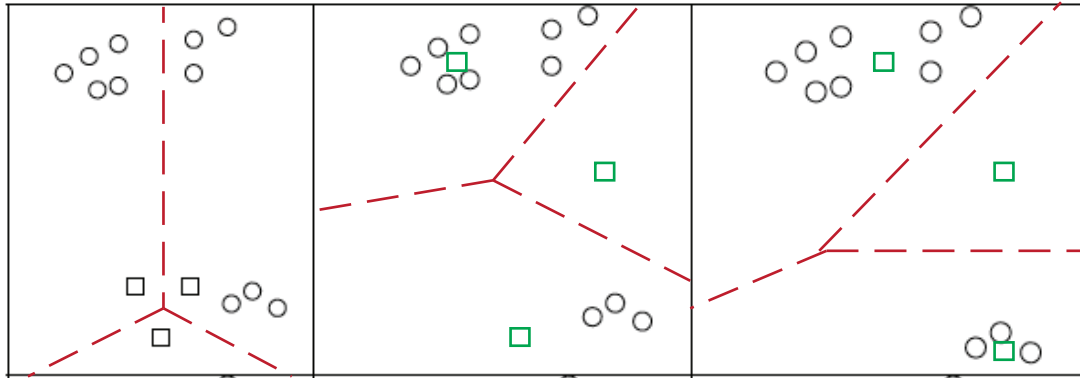
Sample solution:

This is equivalent to the k -nearest neighbor model where k is equal to 1. The 1-nearest neighbor would classify each point to the 1 point that is closest to it according to the distance metric. This would have the same effect in K -means where $K = n$, as the model will be testing closeness to each data point which itself represents the cluster.

3. Run K-means manually for the following dataset. Circles are data points and squares are the initial cluster centers. Draw the cluster centers and the decision boundaries that define each

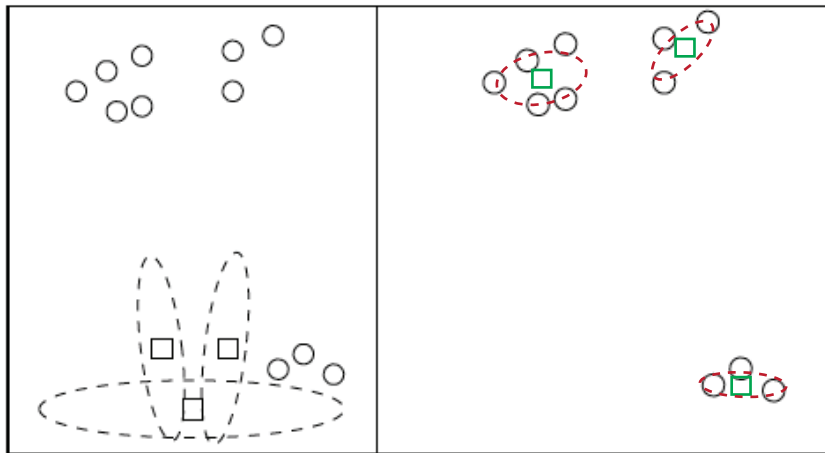
cluster. Use as many pictures as you need until convergence. *Note:* Execute the algorithm such that if a mean has no points assigned to it, it stays where it is for that iteration.

Sample solution:



- Now run a Gaussian mixture model of three Gaussians on the same dataset. The initial cluster centers are the same as for the k-means problem, and the dashed-line ellipses represent the size and shape of the initial covariance matrices. Assume that the model puts no restrictions on the form of the covariance matrices and that EM updates both the means and covariance matrices. Draw (approximately) the cluster centers and the size/shape of the covariance matrices of the final converged GMM.

Sample solution:



- Are the group memberships given by the mixture model the same as the memberships given by k-means? Why or why not?

Sample solution:

No. In the mixture model, soft associations (through the weights) are made with every data point by every gaussian, so it can't happen that the cluster center isn't associated with any data point. Also, the algorithms use different distance metrics, and that mixture models with full covariance matrices allow more flexibility in fitting a cluster.

5 Bayes nets (8 pts)

A analyst notices that people that drive SUVs (S) consume large amounts of gas (G) and are involved in more accidents (A) than the national average. They construct the Bayesian network in Figure 2.

1. Compute $P(A)$ in two ways:
 - (a) By generating the *entire joint distribution* over these variables and explicitly summing the appropriate entries.

Sample solution:

$$\begin{aligned}
 P(A) &= P(a|g, s)P(g, s) + P(a|g, \neg s)P(g, \neg s) + P(a|\neg g, s)P(\neg g, s) + P(a|\neg g, \neg s)P(\neg g, \neg s) \\
 &= P(a|g, s)P(g|s)P(s) + P(a|g, \neg s)P(g|\neg s)P(\neg s) + P(a|\neg g, s)P(\neg g|s)P(s) \\
 &\quad + P(a|\neg g, \neg s)P(\neg g|\neg s)P(\neg s) \\
 &= 0.75 \times 0.8 \times 0.4 + 0.20 \times 0.30 \times 0.60 + 0.75 \times 0.20 \times 0.40 + 0.20 \times 0.70 \times 0.60 \\
 &= 0.42
 \end{aligned}$$

- (b) Using the variable elimination algorithm.

Sample solution:

$$\begin{aligned}
 P(A) &= P(A|S)P(S) + P(A|\neg S)P(\neg S) \\
 &= 0.75 \times 0.40 + 0.20 \times 0.60 \\
 &= 0.42
 \end{aligned}$$

2. Using conditional independence, compute $P(\neg g, a|s)$ and $P(\neg g, a|\neg s)$. Then use Bayes rule to compute $P(s|\neg g, a)$.

Sample solution:

$$\begin{aligned}
 P(\neg g, a|s) &= P(\neg g|s)P(a|s) \\
 &= (1 - P(g|s))P(a|s) \\
 &= 0.20 \times 0.75 = 0.15
 \end{aligned}$$

$$\begin{aligned}
P(\neg g, a|\neg s) &= P(\neg g|\neg s)P(a|\neg s) \\
&= (1 - P(g|\neg s))P(a|\neg s) \\
&= 0.70 \times 0.20 = 0.14
\end{aligned}$$

$$\begin{aligned}
P(s|\neg g, a) &= \frac{P(\neg g, a|s)P(s)}{P(\neg g, a)} \\
&= \frac{P(\neg g, a|s)P(s)}{P(\neg g, a|s)P(s) + P(\neg g, a|\neg s)P(\neg s)} \\
&= \frac{0.15 \times 0.40}{0.15 \times 0.40 + 0.14 \times 0.60} = 0.417
\end{aligned}$$

3. The analyst then notices that there are two types of people that drive SUVs, people from California (C) and people with large families (F). After collecting some statistics, the analyst arrives at the Bayesian network in Figure 3. Using the chain rule from probability, compute the probability $P(\neg g, a, s, c, \neg f)$.

Sample solution:

$$\begin{aligned}
P(\neg g, a, s, c, \neg f) &= P(\neg g|a, s, c, \neg f)P(a|s, c, \neg f)P(s|c, \neg f)P(c|\neg f)P(\neg f) \\
&= P(\neg g|s)P(a|s)P(s|c, \neg f)P(c)P(\neg f) \\
&= 0.20 \times 0.75 \times 0.60 \times 0.3 \times 0.6 \\
&= 0.0162
\end{aligned}$$

4. Without explicitly generating the entire probability distribution compute $P(s)$ and $P(s|c)$ and $P(s|\neg c)$. (Hint: consider different values of F.)

Sample solution:

$$\begin{aligned}
P(s|c) &= P(s|c, f)P(f) + P(s|c, \neg f)P(\neg f) \\
&= 0.8 \times 0.4 + 0.60 \times 0.60 \\
&= 0.68
\end{aligned}$$

$$\begin{aligned}
P(s|\neg c) &= P(s|\neg c, f)P(f) + P(s|\neg c, \neg f)P(\neg f) \\
&= 0.50 \times 0.40 + 0.25 \times 0.60 \\
&= 0.35
\end{aligned}$$

$$\begin{aligned}P(s) &= P(s|c) * P(c) + P(s|\neg c)P(\neg c) \\ &= 0.68 \times 0.30 + 0.35 \times 0.70 \\ &= 0.449\end{aligned}$$

5. Show how you can compute the value of $P(G)$ in the complete distribution without doing any additional work (i.e., based solely on the work you have already done). Explain.

Sample solution:

We can use the probability $P(S)$ (from question 4) since S is the parent of G to compute the value of $P(G)$ as follows: Since S is the only parent of G , we can use the variable elimination algorithm given the joint distribution in order to get the following equation

$$\begin{aligned}P(G) &= P(g|s)P(s) + P(g|\neg s)P(\neg s) \\ &= 0.80 \times 0.449 + 0.30 \times (1 - 0.449) \\ &= 0.5245\end{aligned}$$