

1. [20 pts] Bayesian Networks: Do the following statements hold in each of the above networks? Please explain your reasoning
   1. A⊥C|B, D
   2. B⊥D|A, C
2. [80 pts] Expectation Maximization (EM): In this question you will implement the EM algorithm for Gaussian Mixture Models. A good read on gaussian mixture EM can be found [here](https://www.ics.uci.edu/~smyth/courses/cs274/notes/Notes7_Mixtures_and_EM.pdf). For this problem:
   1. n is the number of training points
   2. f is the number of features
   3. k is the number of gaussians
   4. X is an n × f matrix of training data
   5. w is an n × k matrix of membership weights. w(i, j) is the probability that xi was generated by gaussian j
   6. π is a k × 1 vector of mixture weights (gaussian prior probabilities). πi is the prior probability that any point belongs to cluster i
   7. μ is a k × f matrix containing the means of each gaussian
   8. Σ is an f × f × k tensor of covariance matrices. Σ(:, :, i) is the covariance of gaussian i

2.1) [15 pts] **Expectation**: Complete the function [w] = Expectation(X, k, π, μ, Σ). This function takes in a set of parameters of a gaussian mixture model, and outputs the membership weights of each data point

2.2) [15 pts] **Maximization of Means**: Complete the function [μ] = MaximizeMean(X, k, w). This function takes in the training data along with the membership weights, and calculates the new maximum likelihood mean for each gaussian.

2.3) [15 pts] **Maximization of Covariances**: Complete the function [Σ] = MaximizeCovariance(X, k, w, μ). This function takes in the training data along with membership weights and means for each gaussian, and calculates the new maximum likelihood covariance for each gaussian.

2.4) [15 pts] **Maximization of Mixture Weights:** Complete the function [π] = MaximizeMixtures(k, w). This function takes in the membership weights and calculates the new maximum likelihood mixture weight for each gaussian.

2.5) [10 pts] **EM:** Put everything together and implement the function [π, μ, Σ] = EM(X, k, π0, μ0, Σ0, nIter). This function runs the EM algorithm for nIter steps and returns the parameters of the underlying GMM. Note: Since this code will call your other functions, make sure that they are correct first. A good way to test your EM function offline is to check that the log likelihood, log P (X|π, μ, Σ) is increasing for each iteration of EM.

2.6) [10 pts] Train the EM model using the text data file provided. Then train using the Scikit-learn GMM. Compare the results.