Assignment 4: Probabilistic ML

Marcus Hansen

December 2024

Exercise 1

Consider the factor graph where a and b can be either true or false. The definition of the factor f is provided in the table, and the prior distribution for a follows the Bernoulli distribution Bern(0.3).

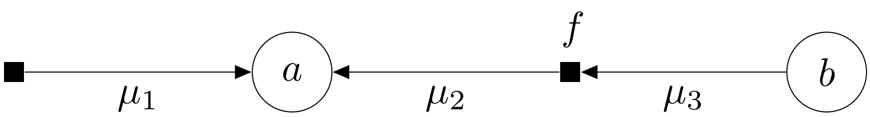
a	b	$f(a,b)$
true	true	0.9
true	false	0.1
false	true	0.2
false	false	0.8

The factor f can be formulated as

$$f(a,b) = 0.9\delta(a)\delta(b) + 0.2\delta(\bar{a})\delta(b) + 0.1\delta(a)\delta(\bar{b}) + 0.8\delta(\bar{a})\delta(\bar{b}),$$

where $\delta(a)$ is the Kronecker delta function, and $\bar{a}$ represents the negation of a.

Assume that we have observed that b is true. Compute the distribution of a using message passing.



For clarification, the arrows indicate the direction of the message passing in the graph. So the graph is an undirected graph from the beginning.

Solution:

Given:

$$p(a) = \text{Bernoulli}(a; 0.3)$$

The message μ_1 is:

$$\mu_1 = p(a) = \text{Bernoulli}(a; 0.3) = 0.3\delta(a - a_{\text{true}}) + 0.7\delta(a - a_{\text{false}})$$

Since b is observed as true, the message μ_3 is:

$$\mu_3 = \delta(b - b_{\text{true}})$$

To calculate μ_2:

$$\mu_2 = \int f(a,b)\delta(b - b_{\text{true}})\, db$$

Substituting $b = \text{true}$ into $f(a,b)$:

$$\mu_2 = f(a, b = \text{true}) = 0.9\delta(a - a_{\text{true}}) + 0.2\delta(a - a_{\text{false}})$$

The posterior distribution $p(a \mid b = \text{true})$ is given by:

$$p(a \mid b = \text{true}) = \frac{1}{Z}\mu_1 \cdot \mu_2$$

Substitute the expressions for μ_1 and μ_2:

$$p(a \mid b = \text{true}) = \frac{1}{Z}\left(0.9\delta(a - a_{\text{true}}) + 0.2\delta(a - a_{\text{false}})\right) \cdot \left(0.3\delta(a - a_{\text{true}}) + 0.7\delta(a - a_{\text{false}})\right)$$

The normalization constant Z is:

$$Z = \sum_a \left(0.9\delta(a - a_{\text{true}}) + 0.2\delta(a - a_{\text{false}})\right) \cdot \left(0.3\delta(a - a_{\text{true}}) + 0.7\delta(a - a_{\text{false}})\right)$$

Calculating Z:

$$Z = 0.9 \cdot 0.3 + 0.2 \cdot 0.7 = 0.27 + 0.14 = 0.41$$

The posterior probabilities are:

$$p(a = \text{true} \mid b = \text{true}) = \frac{0.9 \cdot 0.3}{Z} = \frac{0.27}{0.41} \approx 0.3415$$

$$p(a = \text{false} \mid b = \text{true}) = \frac{0.2 \cdot 0.7}{Z} = \frac{0.14}{0.41} \approx 0.6585$$

Thus, the posterior distribution is:

$$p(a \mid b = \text{true}) = \text{Bernoulli}(a; 0.3415)$$

Exercise 2

Q1: Compute the marginal distribution of a

The marginal distribution of a is defined as $g_a(a)$:

Where:

$$g_a(a) = \mu_{f_1 \to a}(a) \cdot \mu_{f_2 \to a}(a)$$

To find the marginal distribution of a, we need to compute the messages $\mu_{f_1 \to a}(a)$ and $\mu_{f_2 \to a}(a)$.

$$\mu_{f_1 \to a}(a) = \int f_1(a)\, db = f_1(a)$$

Since $f_1(a) = \mathcal{N}(a; \mu_1, \sigma_1^2)$:

$$\mu_{f_1 \to a}(a) = \mathcal{N}(a; \mu_1, \sigma_1^2)$$

$$\mu_{b \to f_2}(b) = 1 \quad \text{(leaf node)}$$

$$\mu_{f_2 \to a}(a) = \int \mu_{b \to f_2}(b) \cdot f_2(a, b)\, db$$

Substitute $f_2(a, b) = \mathcal{N}(b; \alpha a, \sigma_2^2)$ which is a PDF and therfore integrates to 1:

$$\mu_{f_2 \to a}(a) = \int \mathcal{N}(b; \alpha a, \sigma_2^2)\, db = 1$$

Compute $g_a(a)$

$$g_a(a) = \mu_{f_1 \to a}(a) \cdot \mu_{f_2 \to a}(a)$$

Substitute the results:

$$g_a(a) = \mathcal{N}(a; \mu_1, \sigma_1^2) \cdot 1 = \mathcal{N}(a; \mu_1, \sigma_1^2)$$

Final Answer:

$$g_a(a) = \mathcal{N}(a; \mu_1, \sigma_1^2)$$

Q2: Compute the marginal distribution of b

The marginal distribution of b is defined as $g_b(b)$:

$$g_b(b) = \mu_{f_2 \to b}(b)$$

Compute $\mu_{f_2 \to b}(b)$

$$\mu_{f_1 \to a}(a) = f_1(a)$$

$$\mu_{a \to f_2}(a) = \mu_{f_1 \to a}(a)$$

$$\mu_{f_2 \to b}(b) = \int \mu_{a \to f_2}(a) \cdot f_2(a, b)\, da$$

Substitute $\mu_{a \to f_2}(a) = \mathcal{N}(a; \mu_1, \sigma_1^2)$ and $f_2(a, b) = \mathcal{N}(b; \alpha a, \sigma_2^2)$:

$$g_b(b) = \int \mathcal{N}(a; \mu_1, \sigma_1^2) \cdot \mathcal{N}(b; \alpha a, \sigma_2^2)\, da$$

The above integral is a product of two Gaussian distributions, which results in a Gaussian distribution. The mean and variance of the resulting Gaussian distribution. Therfore we can write the marginal distribution of b as:

$$g_b(b) = \mathcal{N}(b; \mu, \sigma^2)$$

We can find the mean of the Gaussian distribution by calculating the expected value of $\mathcal{N}(b; \alpha a, \sigma_2^2)$ where a comes from the $p(a) = \mathcal{N}(a; \mu_1, \sigma_1^2)$:

$$\begin{aligned} \mathbb{E}\left[p(a)\right] =& \mu_1 \\ \mu =& \mathbb{E}\left[\mathcal{N}(b; \alpha a, \sigma_2^2)\right] = \mathbb{E}\left[\mathcal{N}(b; \alpha \mathbb{E}\left[p(a)\right], \sigma_2^2)\right] = \alpha \mu_1 \end{aligned}$$

And finding the variance of the Gaussian distribution

$$\sigma^2 = \mathbb{E}\left[(b - \alpha \mu_1)^2\right],$$

where b depends on a and follows the conditional distribution:

$$b \sim \mathcal{N}(\alpha a, \sigma_2^2).$$

Using the law of total variance

$$\mathrm{Var}(b) = \mathbb{E}[\mathrm{Var}(b \mid a)] + \mathrm{Var}(\mathbb{E}[b \mid a]).$$

Calculate each term separately

- $\mathbb{E}[b \mid a] = \alpha a$,
- $\mathrm{Var}(b \mid a) = \sigma_2^2$,
- $\mathbb{E}[b] = \alpha \mu_1$.

Resulting in

$$\mathrm{Var}(\mathbb{E}[b \mid a]) = \mathrm{Var}(\alpha a) = \alpha^2 \mathrm{Var}(a) = \alpha^2 \sigma_1^2.$$
$$\mathbb{E}[\mathrm{Var}(b \mid a)] = \mathbb{E}[\sigma_2^2] = \sigma_2^2.$$

$$\mathrm{Var}(b) = \mathrm{Var}(\mathbb{E}[b \mid a]) + \mathbb{E}[\mathrm{Var}(b \mid a)] = \sigma^2 = \alpha^2 \sigma_1^2 + \sigma_2^2.$$

Resulting marginal distribution of b as:

$$g_b(b) = \mathcal{N}(b; \alpha\mu_1, \alpha^2\sigma_1^2 + \sigma_2^2)$$

Final Answer:

$$g_b(b) = \mathcal{N}(b; \alpha\mu_1, \alpha^2\sigma_1^2 + \sigma_2^2)$$

Exercise 3

A company using two different machines produces random amounts of clips and pins each day. Suppose that the production follows a Poisson distribution $P(\lambda)$, where the rate λ depends on the quality of the steel the company is using on a specific day. The Poisson distribution is given by:

$$x \sim P(\lambda) \quad \Leftrightarrow \quad p(x) = P(x; \lambda) = \frac{\lambda^x \exp(-\lambda)}{x!}$$

For high-quality steels, $\lambda = 10$, and if the steel is of low quality, $\lambda = 7$. The probability that the company will receive high-quality steel on a specific day is 0.25.
Suppose that at the end of the day the company has produced 10 clips and 8 pins.

Q1 Identify the conditional distributions in the model and draw the Bayesian Network.

We are given two different machines that each produce a random number of clips and pins every day. The production of both clips and pins follows a Poisson distribution with parameter λ, which represents the quality of the steel used (high quality, $\lambda = 10$, with probability 0.25 and low quality, $\lambda = 7$, with probability 0.75). Therefore, we can model the outputs of both machines and λ as random variables:

- X_c represents the number of clips produced by the first machine, which follows a Poisson distribution with parameter λ.
- X_p represents the number of pins produced by the second machine, also following a Poisson distribution with the same parameter λ.
- λ following a Bernoulli distribution.

We therefore have the following:

$$X_c \sim \text{Poisson}(\lambda)$$

$$X_p \sim \text{Poisson}(\lambda)$$

$$\lambda \sim \text{Bernoulli}(\lambda, 0.25)$$

From this, we can construct the Bayesian Network as shown in Figure 1. In this network, the filled circles represent observed random variables. In our case, these observed values are the number of clips and pins produced at the end of the day. Specifically, we observed 10 clips and 8 pins.

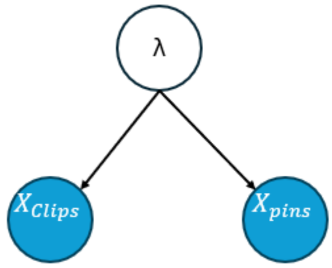


Figure 1: Bayesian Network

The Bayesian Netowrk gives us the joint probability, where we now have identified the condinitonal probabilities.

$$p(\lambda, X_{clips}, X_{pins}) = p(\lambda)p(X_{clips}|\lambda)p(X_{pins}|\lambda)$$

Q2 Transform the Bayesian network into a factor graph.

To transform our Bayesian Network into a factor graph, we need to define the factors corresponding to each node in the Bayesian Network and then connect the factor nodes to the variable nodes according to the edges. Additionally, we will introduce factor nodes to represent the observed values of clips and pins. The factors for each node are defined as follows:

- $f_A = p(\lambda)$: Prior distribution of the parameter λ.
- $f_B = p(X_c \mid \lambda)$: Conditional distribution of the number of clips given λ.
- $f_C = p(X_p \mid \lambda)$: Conditional distribution of the number of pins given λ.
- $f_D = \delta(X_c - X_c^{\text{observed}})$: Function for the observed number of clips.
- $f_E = \delta(X_p - X_p^{\text{observed}})$: Function for the observed number of pins.

In figure 2 you can see the corresponding factor graph.

Q3 Using message passing, compute the probability that the company was using high-quality steel.

We want to infer $p(\lambda = 10|X_c = X_c^{\text{observed}}), X_p = X_p^{\text{observed}}))$ using message passing.

- **Step 1:** Initialize the messages from the factor nodes to the variable nodes:

$$\mu_{f_A \to \lambda} = f_A(\lambda) = p(\lambda)$$

$$\mu_{f_D \to X_c} = f_D(X_c) = \delta(X_c - X_c^{\text{observed}})$$
$$\mu_{f_E \to X_p} = f_E(X_p) = \delta(X_p - X_p^{\text{observed}})$$

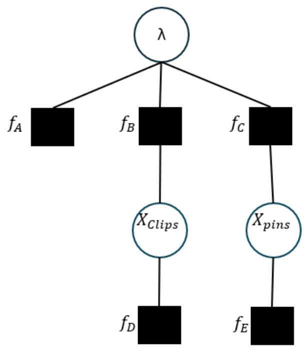


Figure 2: Factor graph

- **Step 2:** Pass messages from the variable nodes to their respective connected factor nodes:

$$\mu_{X_c \to f_B} = \mu_{f_D \to X_c}$$

$$\mu_{X_p \to f_C} = \mu_{f_E \to X_p}$$

- **Step 3:** Compute the messages from the factor nodes f_B and f_C to the node λ:

$$\mu_{f_B \to \lambda} = \sum_{X_c} f_B(X_c, \lambda)\, \mu_{X_c \to f_B} = p(X_c^{\text{observed}} \mid \lambda)$$

$$\mu_{f_C \to \lambda} = \sum_{X_p} f_C(X_p, \lambda)\, \mu_{X_p \to f_C} = p(X_p^{\text{observed}} \mid \lambda)$$

- **Step 4:** Calculate the posterior distribution of λ given the observed values of clips and pins:

$$p(\lambda \mid X_c^{\text{observed}}, X_p^{\text{observed}}) \propto p(\lambda)\, p(X_c^{\text{observed}} \mid \lambda)\, p(X_p^{\text{observed}} \mid \lambda)$$

To obtain the full posterior distribution, we need to normalize our result using the joint probability $p(X_c^{\text{observed}}, X_p^{\text{observed}})$. This gives:

$$p(\lambda \mid X_c^{\text{observed}}, X_p^{\text{observed}}) = \frac{p(\lambda)\, p(X_c^{\text{observed}} \mid \lambda)\, p(X_p^{\text{observed}} \mid \lambda)}{p(X_c^{\text{observed}}, X_p^{\text{observed}})}$$

We are asked to compute the probability that the steel has high quality given the observed number of clips and pins. For this, we substitute $\lambda = 10$, $X_c^{\text{observed}} = 10$, and $X_p^{\text{observed}} = 8$:

$$\begin{aligned}
p(\lambda = 10 \mid X_c^{\text{observed}} = 10, X_p^{\text{observed}} = 8) &= \frac{p(\lambda = 10)\, p(X_c^{\text{observed}} = 10 \mid \lambda = 10)\, p(X_p^{\text{observed}} = 8 \mid \lambda = 10)}{p(X_c^{\text{observed}} = 10, X_p^{\text{observed}} = 8)} \\
&= \frac{0.25 \cdot \text{Pois}(10; 10) \cdot \text{Pois}(8; 10)}{0.25 \cdot \text{Pois}(10; 10) \cdot \text{Pois}(8; 10) + 0.75 \cdot \text{Pois}(10; 7) \cdot \text{Pois}(8; 7)} \\
&= \frac{0.25 \cdot 0.13 \cdot 0.11}{0.25 \cdot 0.13 \cdot 0.11 + 0.75 \cdot 0.13 \cdot 0.07} \\
&= \frac{0.0051}{0.01} \\
&\approx 0.33
\end{aligned}$$

Answer: The probability that the company was using high-quality steel is approximately 33%.

Q4: Write a program to verify the calculated probability using Monte Carlo simulation. The Monte Carlo simulation is implemented as follows:

1. Generate λ by sampling from a Bernoulli distribution: $\lambda = 10$ with probability 0.25 and $\lambda = 7$ with probability 0.75.
2. Draw one sample from the Poisson distribution to get the number of clips produced and one for pins.
3. Count all matches where 10 clips and 8 pins are produced, and track how many of these matches come from $\lambda = 10$.
4. Run the simulation for the desired number of iterations.
5. Estimate the probability by dividing the number of matches from $\lambda = 10$ by the total number of matches.

We found that the probability that the company was using high-quality steel is approximately 33.7% after running 10^6 simulations and 32.4% after running 10^5 simulations.

See the Jupyter notebook for the program!

Exercise 4

Figure 3 shows a model for capturing the inter dependencies between 5 discrete random variables related to a hypothetical student taking a class. The "student" Bayesian network models the dependencies between the following variables:

• D: Difficulty of the class (Easy, Hard),

• I: Intelligence of the student (Low, High),

• G: Grade of the student in the class (A, B, C),

• S: SAT score of the student (Bad, Good),

• L: Letter of recommendation (Bad, Good).

The conditional probability tables (CPTs) and graph structure for this network are shown in Figure 4.2. The joint probability distribution can be expressed as:

$$p(L, S, G, D, I) = p(L|G)p(S|I)p(G|D, I)p(D)p(I).$$

Q1 Write the full joint probability p(L, S, G,D, I) using the given graph and CPTs.

The full joint probability p(L, S, G,D, I) is by using the Bayesian net factorization rules given according to:

$$p(L, S, G, D, I) = p(L|G)p(S|I)p(G|D, I)p(D)p(I).$$

and this is also given in the assignment description.

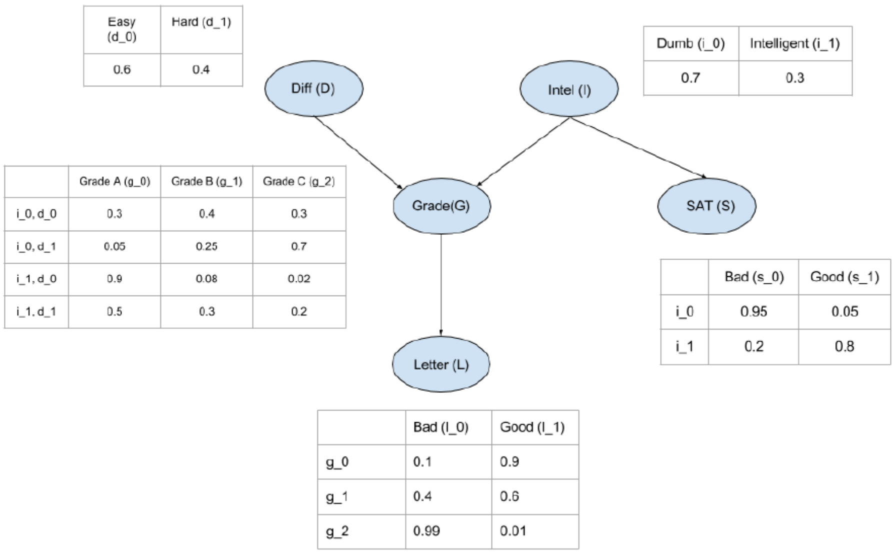


Figure 3: (From assignment description) The (simplified) student network. “Diff” is the difficulty of the class. “Intel” is the intelligence of the student. “Grade” is the grade of the student in this class. “SAT” is the score of the student on the SAT exam. “Letter” is whether the teacher writes a good or bad letter of recommendation. The circles (nodes) represent random variables, the edges represent direct probabilistic dependencies.

Q2 Identify two sets of variables that are conditionally independent given another variable in the network. We have that if all paths between two random variables are blocked, then these random variables are independent given that the path is blocked. A path is blocked if we have a chain or a fork where a middle RV is observed. If we have a inverted fork a path is blocked if that node or any of its descendants is NOT observed.

In figure 3 we have a chain: $D, I- > G- > L$, a fork: $G < -D- > S$ and an inverted fork $D- > G < -I, S$.

Hence once example is that D, I, S are all three conditionally independent of L given G since G is in a chain and blocks all the paths (only one for each RV in this Bayes net) by being observed.

Another is that D, G is conditionally independent of S given I since I blocks the inverted fork path between G and S, D would however be independent of S even though we did not observe I, as the path is blocked by the inverted fork G (or descendant L) as long as we do NOT observe them.

Q3 Calculate the probability that a student is intelligent (I = High) given that the

student received a grade of C (G = C) using the CPTs provided. That is, compute: $p(I = High|G = C)$.

From the graph and CPTs We have the following pmf's:

$$\begin{aligned}
&p(L \mid G)\\
&p(S \mid I)\\
&p(G \mid D, I)\\
&p(D)\\
&P(I)
\end{aligned} \tag{1}$$

Using Bayes theorem we have that:

$$P(I = High|G = C) = \frac{P(G = C \mid I = High)P(I = High)}{P(G = C)} \tag{2}$$

Where we - using the law of total probability and chain rule can rewrite and evaluate the unknown $P(G = C \mid I = High)$ as:

$$\begin{aligned}
P(G = C \mid I = High) &= \sum_D P(G, D \mid I) = \\
&= \sum_D P(G = C \mid D, I = High)P(D) \\
&\approx 0.092
\end{aligned} \tag{3}$$

Similarly for the unknown $P(G = C)$:

$$\begin{aligned}
P(G = C) &= \sum_{D,I} P(G, D, I) \\
&= \sum_{D,I} P(G, D \mid I)P(I) \\
&= \sum_{D,I} P(G \mid D, I)P(I)P(D) = \quad \{G = C\} \\
&\approx 0.3496
\end{aligned}$$

Inserting these values together with the knowns from the CPT we from 2 get:

$$P(I = High|G = C) \approx \frac{0.092 \cdot 0.3}{0.3496} \approx 0.079$$

Q4 Suppose we observe that the student has a good SAT score (S = Good) in addition to a grade G = C. Calculate the probability that a student is intelligent (I = High).

Again, using Bayes theorem we have that:

$$P(I = High|G = C, S = Good) = \frac{P(G = C, S = Good|I = High)P(I = High)}{P(G = C, S = Good)} \tag{4}$$

Where due to the conditional independence that arises can rewrite the unknown $P(G = C, S = Good|I = High)$ as:

$$P(S, G \mid I) = P(S \mid I)P(G \mid I)$$

where $P(S \mid I)$ is known from 1 and $P(G = C \mid I = High)$ from equation 3. We rewrite in terms of knowns and evaluate $P(S = Good, G = C)$:

$$\begin{aligned}
P(S = Good, G = C) &= \sum_I P(S, G, I) = \\
&= \sum_I P(S, G \mid I)P(I) = \quad \{S \perp G \mid I\} = \\
&= \sum_I P(S \mid I)P(G \mid I)P(I) = \quad \{\text{equation 3}\} \\
&= \sum_I P(S \mid I)P(I) \sum_D P(G \mid D, I)P(D) = \{S = Good, G = C\} \\
&\approx 0.03818
\end{aligned} \tag{5}$$

Inserting these values together with the knowns from the CPT by evaluating 4 get:

$$P(I = High|G = C, S = Good) \approx \frac{0.0736 \cdot 0.3}{0.03818} \approx 0.578$$

Q5 by calculating $p(D = Hard|G = C, S = Good)$, explain what cause this observation influences our belief.

Similarly to before, using Bayes theorem we have that:

$$P(D = Hard|G = C, S = Good) = \frac{P(G = C, S = Good \mid D = Hard)P(D = Hard)}{P(G = C, S = Good)} \tag{6}$$

We can again using the law of total probability and chain rule to rewrite $p(G, S \mid D)$ in terms of knowns and evaluate using the CPT's:

$$\begin{aligned}
p(G, S \mid D) =& \frac{p(G, S, D)}{p(D)} = \\
=& \ \frac{1}{p(D)} \sum_I p(G, S, D, I) = \\
=& \ \frac{1}{p(D)} \sum_I p(G, S, D \mid I)p(I) = \quad \{G, D \perp\mid I\} \\
=& \ \frac{1}{p(D)} \sum_I p(G, D \mid I)p(S \mid I)p(I) = \\
=& \ \frac{1}{\cancel{p(D)}} \sum_I p(G \mid D, I)\cancel{p(D)}p(S \mid I)p(I) = \\
=& \ \sum_I p(G \mid D, I)p(S \mid I)p(I) = \quad \{G = C, D = Hard, S = Good\} \\
\approx& \ 0.0725
\end{aligned}$$

We have $P(G = C, S = Good)$ from equation 5 and hence we can evaluate equation 6 as:

$$P(D = Hard|G = C, S = Good) \approx \frac{0.0725 \cdot 0.4}{0.03818} \approx 0.756$$

CLARIFICATION FROM CANVAS: The question asks why we get different results in Q3 and Q4 for the probability of a student being intelligent. You need to answer this by considering the probability P(D = Hard — G = C, S = Good). From Q3:

$$P(I = High|G = C) \approx \frac{0.092 \cdot 0.3}{0.3496} \approx 0.079$$

and Q4 we have:

$$P(I = High|G = C, S = Good) \approx \frac{0.0736 \cdot 0.3}{0.03818} \approx 0.578$$

Hence we have a low probability that a person with grade C is intelligent, but given that the person also has a good SAT increases the probability quite a lot. Meaning that good SAT provides a lot of information about a persons intelligence. From Q5 we have that it is also a quite high probability that given a grade C and good SAT score the class is hard. So it is likely that the student is intelligent but attends a hard class and therefore has a low grade. So the probability of the student being intelligent increases from Q3 to Q4 as in Q3 we only know the grade, which is low, which makes it unlikely that the student is intelligent. But in Q4 we get the additional information about the SAT, and since there is a causal relationship between intelligence and SAT this gives us additional information about the intelligence, moving the belief towards the grade being low not because the student is dumb but because the class is hard.

Q6 explain the difference between causality and correlation using an example from the figure 3.

Correlation is defined $\approx$ as a linear measure of the relationship between random variables. Causality is $\approx$ that one event has a effect on another event so that changing the first event, there will also be a change in the other. In the graph above, we can easily identify causal (direct and indirect) relationships by going from random variables to others on paths constrained that we have to follow the directions of the arrows.

For example according to this model/graph, starting at intelligence, there is a causal (direct because its not through any other random variables) path from intelligence to both Grade and SAT and therefore also a causal relationship relationship between both intelligence and grade and intelligence and SA. If we change the observed intelligence, it will directly impact the probabilities of the outcomes of Grade and SAT. There is also a causal path between Intelligence and Letter even though this path goes through Grade (indirect path) where Grade is also influenced by Difficulty. Changing the observed Difficulty or Intelligence will hence effect grade which in turn will effect the probabilities of the outcomes of Letter.

There is however no causal path between SAT and grade, so there is no causal relationship between these two. Meaning that if we change the observed SAT it will not have an effect on the probabilities of the outcomes of Grade. But there is a correlation, meaning there is a linear relationship between them. In this case that linear relationship is due to intelligence having a causal relationship to both of them. Greater intelligence will cause a higher probability of both good Grades and a good SAT, so both will increase if we change intelligence from low to high, which due to them being correlated. But since there is no causal relationship between them, changing the outcome of SAT or Grade wont effect the other probabilities of the other.

Exercise 5

One application of Markov Random Fields (MRFs) is to denoise noisy images. In this task, we minimize the energy function using the **Iterated Conditional Mode (ICM)**.

Q1: Simulate noisy images with varying noise levels (e.g., Gaussian noise with different standard deviations or salt-and-pepper noise with varying probabilities).

We applied salt-and-pepper noise to the ground truth image. In Figure 4, the ground truth image is shown with varying noise levels. As the noise level increases, the image becomes increasingly distorted, with higher noise levels causing nearly every pixel to be changed.

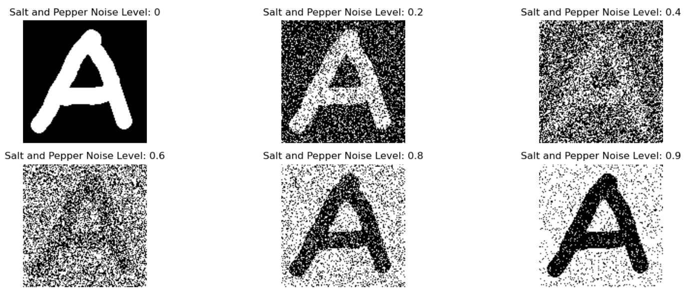


Figure 4: Salt and Pepper noise with different noise levels.

Q2: Implement the ICM algorithm to minimize the energy function.

See the jupyter notebook for the implementation of the ICM algorithm!

Q3: Plot the denoised image after iterations from the following list: [0, 1, 5, 15, 50].

We used the ICM algorithm to denoise the salt-and-pepper image, applying a noise level of 0.2, meaning approximately 20% of the pixels were changed. In Figure 5, you can observe that after one iteration, minimal changes occurred. However, after five iterations, the result appears to have converged, as further iterations show little to no additional changes.

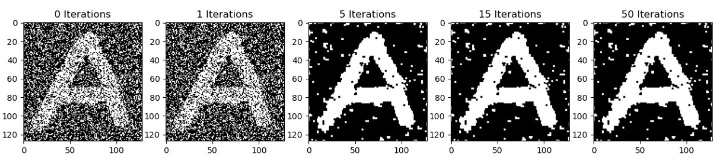


Figure 5: Denoised image using the ICM algorithm.

Q4: Compute and plot the **Normalized Mean Squared Error (NMSE) between the denoised image (D) and the original ground truth image (G) for each noise level:**

We calculated the NMSE between the ground truth image and the denoised image for varying noise levels. As shown in Figure 6, the error increases as the noise level rises. We used 10 iterations for each noise level.

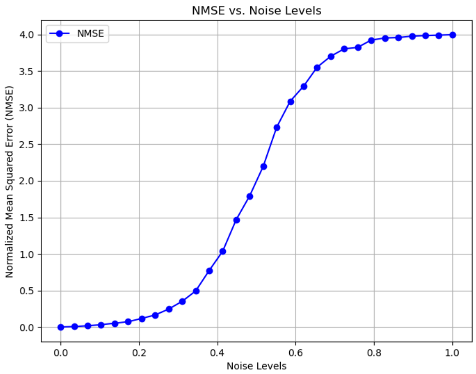


Figure 6: NMSE error between ground truth image and denoised image with varying noise level.

Q5: Analyze the performance of the denoising algorithm as the noise level increases.

As shown in Figure 6, the algorithm performs well when the noise level is between 0 and 0.3. However, as the noise level increases, the algorithm struggles more to reconstruct the image. Between 0.4 and 0.6, its performance declines rapidly, and around 0.8, it stabilizes with an NMSE of around 4, likely because the algorithm converges to the same denoised version regardless of further increases in noise. When the noise level is high and salt-and-pepper noise is applied, the image almost becomes reversed, with black pixels turning white and vice versa. The algorithm then denoises the image to the reversed version, where the letter 'A' is now black, and the background is white.

Q6: Vary the parameters h, β, and η in the energy function and evaluate their impact on denoising performance.

For this question, we will vary one parameter and keep the rest fixed. The initial values are: $h = 0$, $\beta = 1$, and $\eta = 1$.

In Figure 7, you can see the performance of the algorithm when varying h. We can clearly see that when h is negative, we get a white image because we then favor white pixels. When h is greater than 0, we get a black image, as it favors black pixels in the minimization.

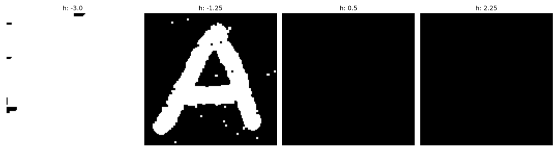


Figure 7: Denoised image with different h.

In Figure 8, you can see the performance of the algorithm when varying β. We can see that increasing β does not affect the performance of the algorithm in our case. This is probably because it is beneficial to favor nearby pixels, as our image has many similar pixels.

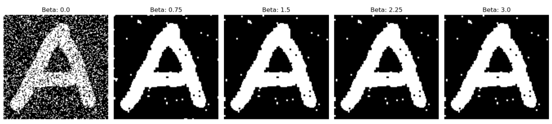


Figure 8: Denoised image with different β.

In Figure 9, you can see the performance of the algorithm when varying η. We can clearly see that η affects the performance. When η is zero, the algorithm performs poorly, but when η is between 0.625 and 1.875, it does not significantly affect the performance. A greater η, like 2.5, affects the performance and outputs a quite noisy image. This is because a larger η takes more information from the observed image, which, in our case, is noisy.

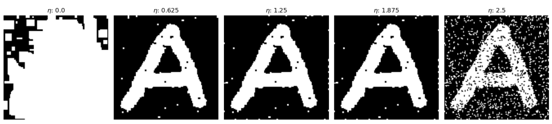


Figure 9: Denoised image with different η.

Q7 For each parameter, plot:

For this question, we will vary one parameter and keep the rest fixed. The initial values are: $h = 0$, $\beta = 1$, and $\eta = 1$.

NMSE vs h

For this, we varied h between -3 and 4 and plotted it against the NMSE, as shown in Figure **??**. Here, we can see that large negative values of h increase the NMSE, and the closer we get to 0, the lower the NMSE becomes. For h-values greater than 1, we get a larger NMSE, but not as large as when h is negative. This is likely because when h is greater than 0, it favors black pixels, and the original image has more black pixels as the background is black, with only the letter "A" in white. Therefore, there is a smaller error, but the image becomes completely black.

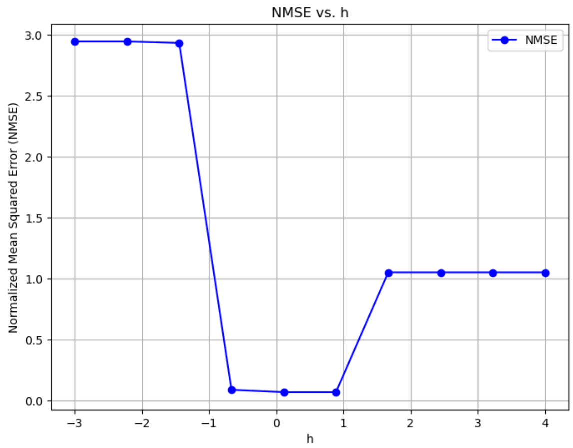


Figure 10: NMSE vs h

NMSE vs β

For this, we varied β between 0 and 2.5 and plotted it against the NMSE, as shown in Figure 11. We can clearly see that a higher β yields better results, but after $\beta = 1$, the performance remains the same. It is reasonable that a small β results in worse performance because less information is used from neighboring pixels. This image is very similar in many areas, so the pixels should take more information from their neighbors.

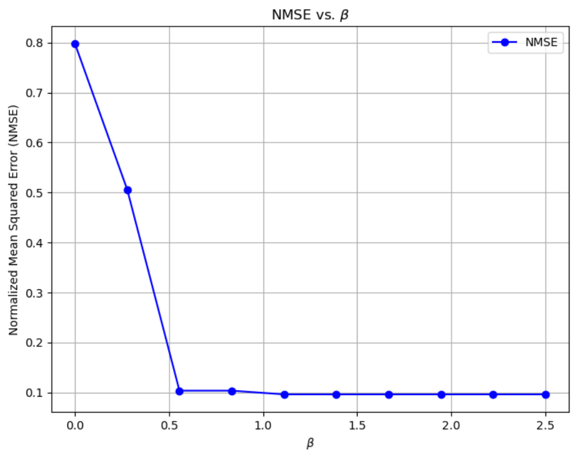


Figure 11: NMSE vs β

NMSE vs η

For this, we varied η between 0 and 2.5 and plotted it against the NMSE, as shown in Figure 12. Here, we can see that when $\eta = 0$, we get a high error because we don't consider the observed image at all. When η is around 1, the error is small, and when η exceeds 2, the error increases. This could be because, at higher values of η, we focus more on the observed image, and if the observed image has a lot of noise, it can lead to poor denoising.

Q8:Parameter influences

$$E(x, y) = h \sum_i x_i - \beta \sum_{i,j} x_i x_j - \eta \sum_i x_i y_i,$$

When choosing the parameters h, β, and η, there is always a trade-off, as their significance is only relevant in comparison to one another, given that all contribute to the energy function.

Thus, when selecting these parameters, one needs to consider what is important in the task: the degree of denoising, how much of the observed image should be preserved, and the desired relationship between neighboring pixels in the final image.

For example, the relationship between η and β is crucial. A high η in relation to β ensures that the observed image is strongly preserved, while a high β in relation to η smooths the image by prioritizing uniformity among neighboring pixels/smoothness.

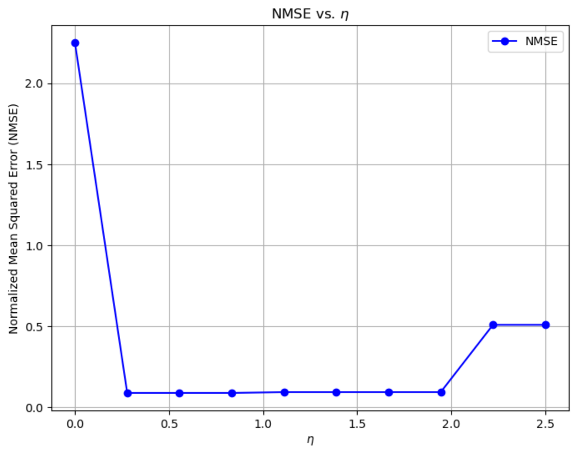


Figure 12: NMSE vs η

Additionally, the h parameter, often referred to as the bias term, incorporates prior knowledge about the expected distribution of labels. For instance, a larger positive h biases the result towards more black pixels, while a negative h biases it towards more white pixels.

Choosing appropriate values for these parameters involves making a trade-off between these priorities to achieve the desired outcome for the specific task.