Bayesian models

April 2025

Topics to be discussed

Part I. Introduction

1. Introduction to Artificial intelligence

Part II. Search and optimisation.

2. Search - basic approaches
3. Search - optimisation
4. Two-player deterministic games
5. Evolutionary and genetic algorithms

Part III. Machine learning and data analysis.

6. Regression, classification and clustering (Part I & II)
8. Artificial neural networks
9. Bayesian models
10. Reinforcement Learning

Part IV. Logic, Inference, Knowledge Representation

11. Propositional logic and predicate logic
12. Knowledge Representation

Part V. AI in Action: Language, Vision

13. AI in Natural language processing
14. AI in Vision and Graphics

Part VI. Summary

15. AI engineering, Explainable AI, Ethics,

Agenda

1. Introduction and reminders
2. Probability review - reminder
3. Bayesian Models
4. Bayesian Classifier
   - Introduction to Bayesian classifiers
   - Naive Bayes algorithm
   - Examples Naive Bayes
5. Bayesian networks
   - Inference in Bayesian Network
6. Summary

Estimation vs Prediction vs Optimization - reminder

- **Estimation**: focus on estimating the function f(x) that relates the input variable x to the output variable y.
  - Goal is to estimate the relationship between x and y based on available data.
  - Used in applications where understanding the underlying relationship between variables is important (e.g. determining the effect of a new drug on a disease).
  - Example question: What is the best estimate of the function f(x) that relates the input variable x to the output variable y?
- **Prediction**: focus on predicting the output variable y given the input variable x.
  - Goal is to minimize prediction error on new data.
  - Used in applications where accurate predictions are critical (e.g. predicting the stock market, medical diagnosis).
  - Example question: How well does the model predict the output variable y given the input variable x?
- **Optimization**: focus on finding the input variable x that maximizes or minimizes a certain objective function based on a fixed relationship between x and y.
  - Goal is to find the optimal input value x that achieves a desired outcome.
  - Used in applications where finding the best solution to a problem is important (e.g. optimizing the performance of a machine learning model).
  - Example question: What is the optimal input value x that maximizes the objective function f(x) given the fixed relationship between x and y?

In summary, in the formula y=f(x), prediction deals with estimating y given x, estimation deals with estimating f given data (x,y), and optimization deals with finding the value of x that optimizes f.

Interpretation vs Prediction - reminder

- **Prediction**: focus on making accurate predictions.
  - Goal is to minimize prediction error on new data.
  - Used in applications where accurate predictions are critical (e.g. predicting the stock market, medical diagnosis).
  - Sample questions:
    - How well does the model predict the outcome variable Y given the input variables X and Z in the model?
    - How does the model perform on new data compared to the training data?
    - What is the best metric to use for evaluating prediction accuracy, such as mean squared error (MSE) or accuracy score?
- **Interpretation**: focus on understanding how a model works.
  - Goal is to understand the underlying patterns and relationships in the data and potentially infer causality.
  - Used in applications where insights and explanations are important (e.g. social sciences, finance).
  - Sample questions:
    - What features (input variables) X and Z are most important for predicting the outcome variable Y in the model?
    - How do different features interact with each other in the model?
    - How can the model be simplified to improve interpretability without sacrificing too much prediction accuracy?
    - Can the model infer causality or just correlation between variables?
  - Causality is often more important in interpretation tasks, but can be difficult to infer from correlation alone.
  - Models that prioritize interpretability often sacrifice some prediction accuracy to achieve it.

Correlation vs Causation

- **Correlation**: a statistical relationship between two variables where changes in one variable are associated with changes in the other variable.
  - Example: Let X and Y be two random variables. Pearson's correlation coefficient ρ between X and Y measures the strength and direction of the linear relationship between them.
  - Correlation does not imply causation. Correlation can arise from confounding variables or spurious relationships. For example, there may be a positive correlation between ice cream sales and crime rates, but this does not mean that ice cream causes crime or vice versa.
  - Correlation can be measured using various correlation coefficients, such as Pearson's r, Spearman's rank correlation coefficient, and Kendall's tau.
- **Causation**: a relationship where changes in one variable directly cause changes in the other variable.
  - Example: Let X and Y be two random variables. X causes Y if and only if changing the value of X changes the value of Y.
  - Establishing causation requires experimental design and controlling for confounding variables. For example, to determine if smoking causes lung cancer, we would need to perform a randomized controlled trial where we randomly assign people to either smoke or not smoke, and then measure the incidence of lung cancer in both groups.
  - Causation can be inferred using causal models and techniques such as randomized controlled trials, instrumental variable analysis, and regression discontinuity design.

Probability Review

Random Variables

A random variable X is a variable whose possible values are outcomes of a random phenomenon. It can be discrete or continuous. The probability distribution of X is a function that assigns probabilities to each possible value of X.

X	$P(X)$
0	0.2
1	0.4
2	0.3
3	0.1

Joint Distribution

The joint distribution of two random variables X and Y is a function that assigns probabilities to each possible combination of values of X and Y. It is denoted as $P(X, Y)$.

X	Y	$P(X, Y)$
0	0	0.1
0	1	0.2
1	0	0.3
1	1	0.4

Probability Review (Cont'd) - reminder

Marginal Distribution

The marginal distribution of a random variable is the probability distribution of that variable, ignoring the values of other variables. For example, the marginal distribution of X is denoted as $P(X)$ and is computed by summing the joint probabilities over all possible values of Y:

$$P(X = x) = \sum_{y} P(X = x, Y = y)$$

X	$P(X)$
0	0.3
1	0.7

Conditional Distribution

The conditional distribution of a random variable given another variable is the probability distribution of the first variable, conditioned on specific values of the second variable. For example, the conditional distribution of X given $Y = y$ is denoted as $P(X|Y = y)$ and is computed as:

$$P(X = x|Y = y) = \frac{P(X = x, Y = y)}{P(Y = y)}$$

X	$Y = 0$	$Y = 1$
0	0.25	0.5
1	0.75	0.5

Conditioning vs Marginalization- reminder

- **Conditioning**: focus on updating our belief about a variable based on new information.
  - Used in frequentist statistics.
  - Example question: Given that we know X=x, what is the probability of Y=y?
  - Denoted as $P(Y = y|X = x)$.
  - Formula: $P(Y = y|X = x) = \frac{P(X=x, Y=y)}{P(X=x)}$
- **Marginalization**: focus on integrating over a variable to obtain the probability of another variable.
  - Used in Bayesian statistics.
  - Example question: What is the probability of Y=y, regardless of the value of X?
  - Denoted as $P(Y = y)$.
  - Formula: $P(Y = y) = \sum_x P(X = x, Y = y)$

In summary, conditioning involves updating our belief about one variable given another variable, while marginalization involves integrating over one variable to obtain the probability of another variable. Conditioning is used in frequentist statistics while marginalization is used in Bayesian statistics.

Probabilistic Inference - an example in Medical Diagnosis

Joint distribution (patient health database):

$$\mathbb{P}(C, S, F, T)$$

where: C represents whether the patient has a cold, S represents the symptom of a sore throat, F indicates whether it's the flu season, and T stands for the patient's temperature.

Probabilistic inference:

- **Condition** on evidence (sore throat, flu season): $S = 1$, $F = 1$
- Interested in **query** $C = 1$, where we are conditioning on the evidence that the patient has a sore throat $S = 1$ and it's flu season $F = 1$, and marginalizing out temperature T.

$$\mathbb{P}(C = 1 \mid S = 1, F = 1)$$

Bayesian Models

- Overview of Bayesian models
- Bayesian models vs. frequentist models
- Basic concepts of Bayesian models
  - Prior and posterior distributions
  - Maximum likelihood and maximum a posteriori estimation
  - Bayesian decision theory

Introduction to Bayesian Models and inference

- Bayesian models are statistical models that incorporate prior knowledge or beliefs into the modeling process.
- Bayesian modeling is based on the Bayes' theorem, which provides a framework for updating probabilities as new data becomes available.
- The main idea is to use prior probabilities and likelihoods to compute posterior probabilities, which represent updated beliefs after incorporating new data.
- Mathematically, Bayes' theorem can be represented as:

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}$$

where:
$P(A|B)$ is the posterior probability of event A given event B, $P(B|A)$ is the likelihood of event B given event A, $P(A)$ is the prior probability of event A, and $P(B)$ is the marginal probability of event B.

Prior and Posterior Distributions

Definition: In Bayesian inference, the prior distribution is a probability distribution that represents our knowledge or beliefs about the parameters of a model before observing any data. The posterior distribution, on the other hand, is the updated probability distribution after incorporating the observed data.

Mathematical Formula:

$$\begin{aligned} \text{Prior Distribution:} \quad & P(\theta) \\ \text{Posterior Distribution:} \quad & P(\theta|D) \propto P(D|\theta) \cdot P(\theta) \end{aligned}$$

where θ represents the parameters of the model and D represents the observed data.

Example: Suppose we have a coin and we want to estimate the probability of getting heads (θ) based on some observed data D (e.g., flipping the coin 100 times and getting 70 heads). We can use a prior distribution (e.g., a uniform distribution or a Beta distribution) to represent our initial belief about the possible values of θ. After observing the data, we can update our belief using Bayes' theorem to obtain the posterior distribution of θ given the data.

Maximum Likelihood and Maximum a Posteriori Estimation

Definition:

- **Maximum likelihood estimation (MLE)** is a method used to estimate the parameters of a statistical model by maximizing the likelihood function, which measures the likelihood of the observed data given the model parameters.
- **Maximum a posteriori estimation (MAP)** is a Bayesian approach that combines the likelihood function with a prior distribution to estimate the parameters of a model.

Mathematical Formula:

$$\text{MLE:} \quad \hat{\theta}_{\text{MLE}} = \arg\max_{\theta} P(D|\theta)$$

$$\text{MAP:} \quad \hat{\theta}_{\text{MAP}} = \arg\max_{\theta} P(\theta|D) \propto P(D|\theta) \cdot P(\theta)$$

where $\hat{\theta}_{\text{MLE}}$ and $\hat{\theta}_{\text{MAP}}$ represent the estimated parameters using MLE and MAP, respectively, $P(D|\theta)$ is the likelihood function, and $P(\theta)$ is the prior distribution.

Example:

- Consider a Gaussian distribution with unknown mean μ and known variance σ^2. We can estimate μ using MLE by maximizing the likelihood function $P(D|\mu)$, where D represents the observed data.
- Alternatively, we can use MAP to estimate μ by incorporating a prior distribution $P(\mu)$, such as a normal distribution with mean 0 and variance 1, and maximizing the posterior distribution $P(\mu|D)$.

Bayesian Models vs. Frequentist Models

Description: Bayesian models and frequentist models are two different approaches to statistical modeling and inference. They differ in their philosophical interpretations, assumptions, and methods of estimation.

Definitions:

- Bayesian Models: Bayesian models use Bayesian inference, which incorporates prior knowledge or beliefs about the parameters of a model into the analysis. The posterior distribution is updated based on the observed data using Bayes' theorem.
- Frequentist Models: Frequentist models use frequentist or classical inference, which does not incorporate prior knowledge or beliefs. It relies solely on the observed data to estimate model parameters and make inferences.

Mathematical Formula:

$$\text{Bayesian Inference:} \quad P(\theta|D) = \frac{P(D|\theta) \cdot P(\theta)}{P(D)}$$

$$\text{Frequentist Inference:} \quad \text{Estimate of } \theta \text{ based on maximum likelihood estimation (MLE)}$$

where θ represents the parameters of the model, D represents the observed data, $P(\theta|D)$ is the posterior distribution in Bayesian inference, $P(D|\theta)$ is the likelihood function, $P(\theta)$ is the prior distribution, and $P(D)$ is the marginal likelihood or evidence.

Example: Suppose we want to estimate the mean of a population based on a sample of data. In a Bayesian approach, we would incorporate prior information about the population mean into the analysis, while in a frequentist approach, we would rely solely on the observed sample data to estimate the population mean.

Comparison:

- Bayesian models allow for the incorporation of prior knowledge or beliefs, which can be useful in situations where there is limited data or when expert knowledge is available.
- Frequentist models rely solely on the observed data, which can be advantageous when prior information is not available or when a large amount of data is available.
- Bayesian models provide posterior distributions that represent the updated uncertainty about the parameters, while frequentist models provide point estimates without quantifying the uncertainty.
- Bayesian models tend to be more computationally intensive due to the need to estimate and update prior distributions, while frequentist models may be computationally simpler.

Bayesian Classifier, Naive Bayes

- Bayesian classifier, also known as naive Bayes classifier, is a popular supervised learning algorithm used for classification tasks.
- Given a set of training data with class labels, Naive Bayes estimates the prior probability $P(\text{class})$ and the conditional probability $P(\text{feature}|\text{class})$ for each class and feature, where:
  - A **class** is a category or label that a data point can be assigned to. For example, in a spam filtering problem, the classes might be "spam" and "not spam".
  - A **feature** is an attribute or characteristic of a data point that can be used to help classify it. For example, in a spam filtering problem, the features might be the presence or absence of certain words in an email.
- Bayesian classifier is based on the Bayesian inference principles:

$$P(class|feature) = \frac{P(feature|class) \times P(class)}{P(feature)}$$

- It assumes that the features are conditionally independent given the class label.
- To classify a new data point, Naive Bayes computes the posterior probability $P(\text{class}|\text{feature})$ for each class using Bayes' theorem and assigns the class with the highest probability as the predicted class.
- Bayesian classifier is simple, computationally efficient, and can handle categorical and continuous data.

Definition of Naive Bayes (cont.)

- The posterior probabilities $P(y|x)$ are calculated for each class y, based on the formula:

$$P(y|x) = P(y) \prod_{i=1}^{m} P(x_i|y)$$

  where m is the number of features
- The class with the highest posterior probability is selected as the predicted class. That is, $\hat{y}$ is given by:

$$\hat{y} = \arg\max_{y} P(y|x)$$

- Here, $\arg\max_y$ denotes the argument that maximizes the function $P(y|x)$ over all possible values of y.
- We select the class y that has the highest probability given the evidence provided by the features of the new data point x.
- This prediction rule is often referred to as the "maximum a posteriori" (MAP) decision rule.

Naive Bayes algorithm

Algorithm Naive Bayes Algorithm - Pseudocode

Input: Training set $\{(\mathbf{x}_1, y_1), ..., (\mathbf{x}_n, y_n)\}$ where $\mathbf{x}_i$ is a feature vector and y_i is the class label

Output: Prediction $\hat{y}$ for a new input vector $\mathbf{x}$

Estimate class prior probabilities: $P(y)$ for all classes y

Estimate conditional feature probabilities: $P(x_i|y)$ for all features x_i and classes y

Calculate the posterior probabilities $P(y|x)$ for all classes y and the new input vector $\mathbf{x}$ using Bayes' theorem:

for *each class* y **do**

$P(y|x) = P(y) \prod_{i=1}^{m} P(x_i|y)$ where m is the number of features

Predict the class with the highest posterior probability: $\hat{y} = \arg\max_y P(y|x)$

return $\hat{y}$

Examples of Naive Bayes

- **Spam filtering with Naive Bayes**: Given a set of emails labeled as spam or not spam, Naive Bayes estimates the probability of each word appearing in spam or non-spam emails.
  - **Step 1: Training**: For each class (spam and non-spam), Naive Bayes estimates the prior probability of the class $P(\text{class})$ and the conditional probability of each feature (word) given the class $P(\text{feature} \mid \text{class})$ using the training data.
  - **Step 2: Prediction**: To classify a new email, Naive Bayes computes the posterior probability of each class given the words in the email using Bayes' theorem:

$$P(\text{class} \mid \text{words}) = \frac{P(\text{words} \mid \text{class}) \times P(\text{class})}{P(\text{words})}$$

where $P(\text{words})$ is the normalizing constant that makes the probabilities sum to 1.

To avoid computing this constant, Naive Bayes uses proportional probabilities:

$$P(\text{class} \mid \text{words}) \propto P(\text{class}) \times \prod_{i=1}^{n} P(\text{word}_i \mid \text{class})$$

where n is the number of words in the email.

The predicted class is the one with the highest posterior probability.

Assumptions for Using Proportional Probabilities

When using proportional probabilities in Naive Bayes, it is important to make the following assumptions:

- The features are conditionally independent given the class label.
- The training set is representative of the true population of data.
- The prior probabilities of the classes are known or can be estimated accurately.
- The likelihoods of the features given the class are known or can be estimated accurately.
- The features are not redundant or highly correlated.

Advantages and disadvantages of Naive Bayes

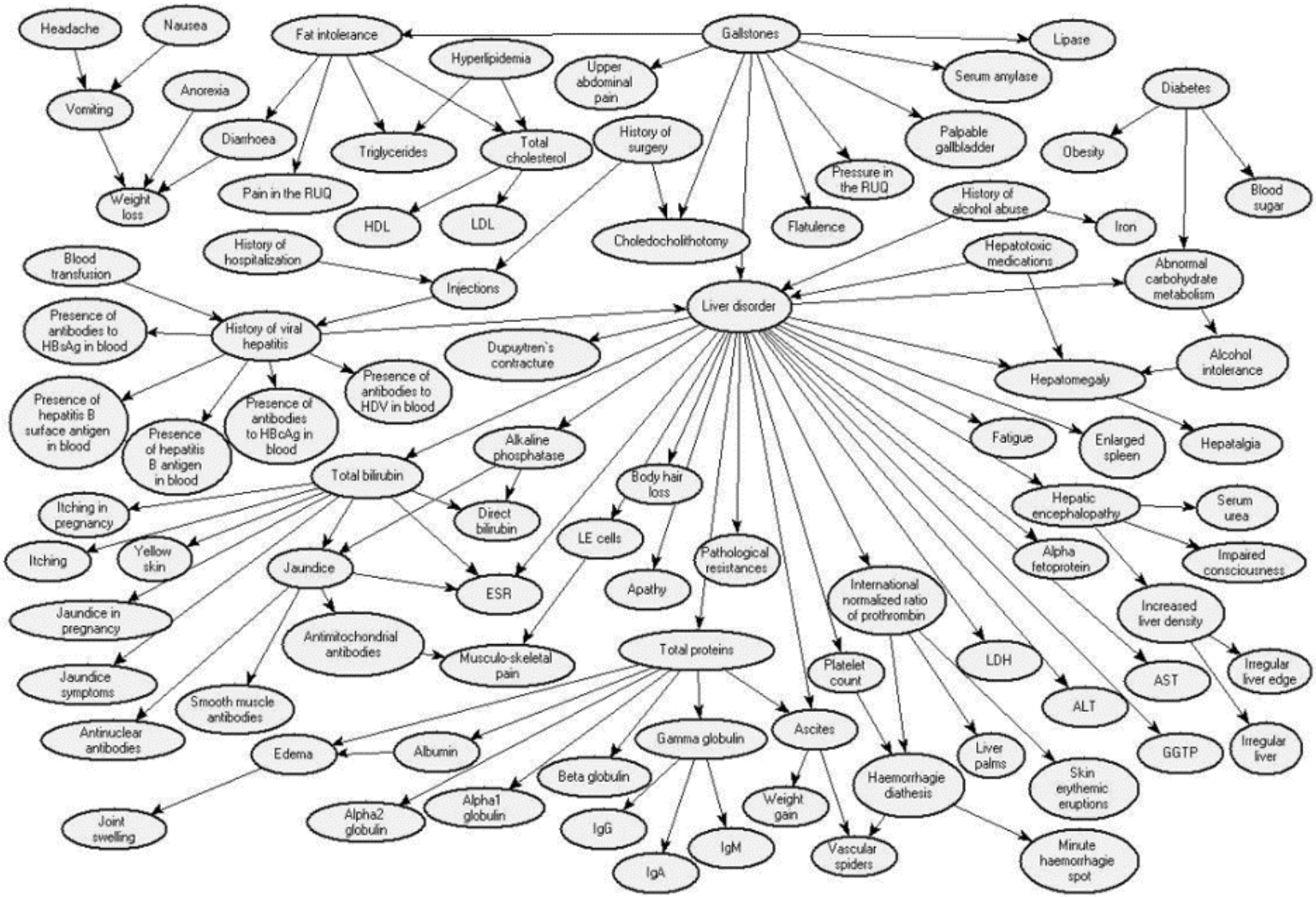
- Advantages:
  - Fast and simple to implement
  - Can handle large feature spaces and noisy data
  - Provides interpretable probability scores for each class
- Disadvantages:
  - Assumes conditional independence between features, which may not hold in reality
  - Limited expressiveness compared to more complex models such as decision trees or neural networks
  - Sensitivity to irrelevant features in the data
  - Requires a sufficient amount of training data to estimate probabilities accurately

Overall, Naive Bayes is a useful algorithm for classification and prediction tasks, especially in cases where the assumptions of conditional independence and large feature spaces hold. However, it is important to be aware of its limitations and potential weaknesses in certain scenarios.

Bayesian Networks

- Introduction to Bayesian networks
- Directed acyclic graphs (DAGs)
- Conditional probability tables (CPTs)
- Inference in Bayesian networks
  - Variable elimination
  - Belief propagation
  - Sampling-based methods

Bayesian Network - example from Health Sector



https://sites.pitt.edu/ druzdzel/psfiles/cbmi99a.pdf

Directed Acyclic Graphs (DAGs)

- Directed acyclic graphs (DAGs) are used to represent Bayesian networks.
- A DAG is a graph that consists of nodes (representing variables) and directed edges (arcs) that indicate the direction of probabilistic influence between nodes.
- DAGs are acyclic, meaning there are no cycles or loops in the graph.
- Mathematically, a DAG is denoted as $G = (V, E)$, where V is the set of nodes and E is the set of directed edges.

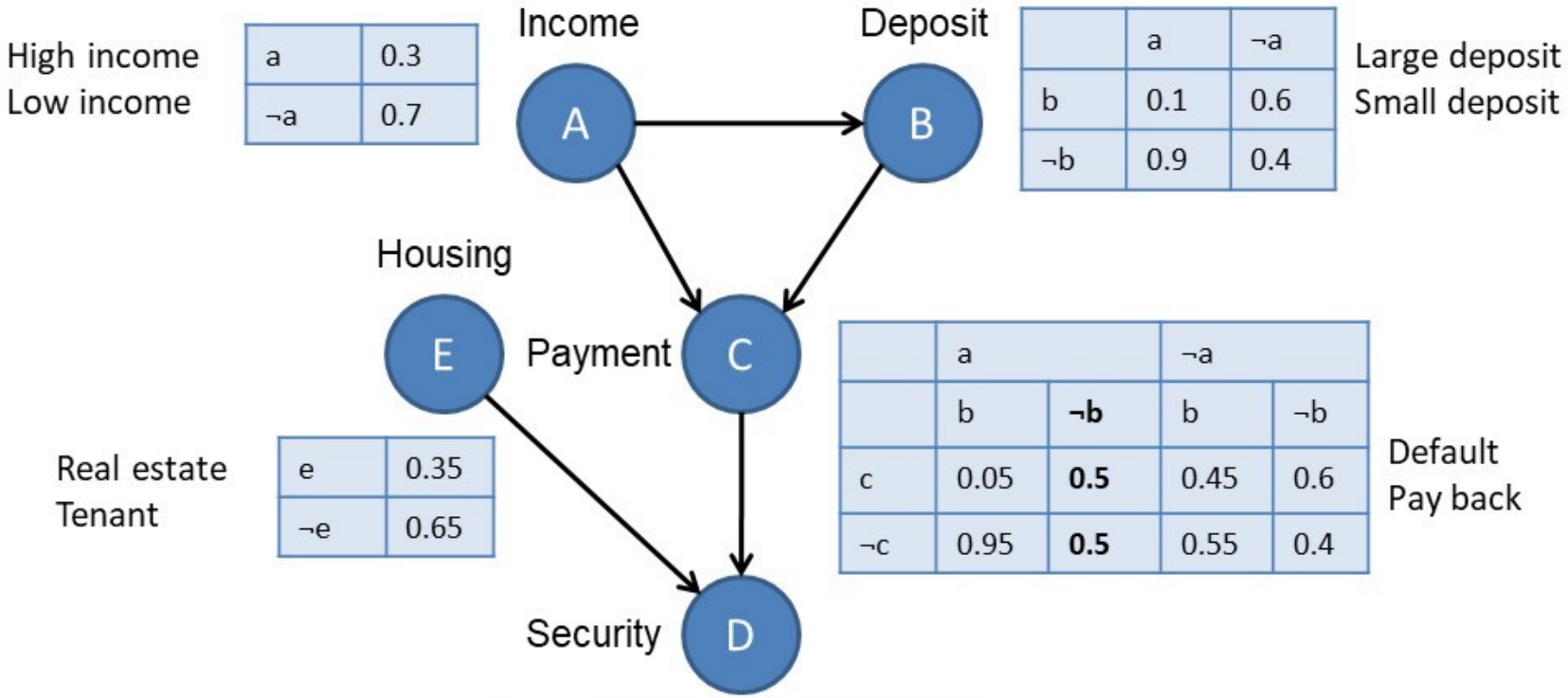
Conditional Probability Tables (CPTs)

- Conditional Probability Tables (CPTs) are used to represent the probabilistic relationships among variables in Bayesian networks.
- A CPT is a table that specifies the conditional probability distribution of a node given its parent nodes.
- Mathematically, a CPT for a node X_i with parents $X_{\text{pa}(i)}$ is denoted as $P(X_i \mid X_{\text{pa}(i)})$, where X_i represents the node and $X_{\text{pa}(i)}$ represents its parent nodes.
- CPTs capture the uncertainty or probabilistic dependencies between variables in the Bayesian network.

Bayesian Networks

- Bayesian networks, also known as belief networks or graphical models, are graphical representations of probabilistic relationships among a set of variables.
- Bayesian networks are represented as directed acyclic graphs (DAGs), where nodes represent variables and edges represent probabilistic dependencies between variables.
- Conditional probability tables (CPTs) are used to specify the conditional probabilities associated with each node given its parents in the graph.
- Bayesian networks can be used for probabilistic reasoning, inference, and prediction in various fields, such as healthcare, finance, and artificial intelligence.
- Example: A Bayesian network can be used to model the relationships between symptoms, diseases, and test results in a medical diagnosis system.

Bayesian Network - an example



a	0.3
¬a	0.7

	a	¬a
b	0.1	0.6
¬b	0.9	0.4

	a		¬a	
	b	¬b	b	¬b
c	0.05	**0.5**	0.45	0.6
¬c	0.95	**0.5**	0.55	0.4

	e	¬e
e	0.35	
¬e	0.65	

	c		¬c	
	e	¬e	e	¬e
d	0.01	0.5	0.75	0.31
¬d	**0.99**	0.5	**0.25**	0.69

https://www.dataminingapps.com/2019/09/bayesian-networks-lessons-learned-from-the-past/

Factorization in Bayesian Networks

Bayesian Network Structure:

- Directed acyclic graph (DAG) representation.
- Nodes represent variables, edges signify dependencies.

Local Probabilities:

- Each variable X_i has a conditional probability $p(x_i \mid x_{\text{pa}(i)})$.
- For root nodes, this is simply $p(x_i)$.

Joint Probability Factorization:

$$\mathbb{P}(X_1 = x_1, X_2 = x_2, \ldots, X_n = x_n) = \prod_{i=1}^{n} p(x_i \mid x_{\text{pa}(i)})$$

Utility in Applications:

- Simplifies computation of complex probabilities.
- Essential in fields like medical diagnosis and risk assessment.

Special Properties of Bayesian Networks

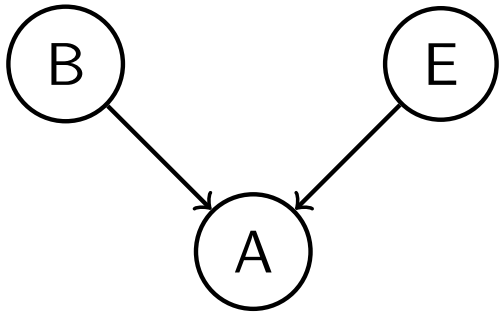
Key idea: locally normalized

All factors (local conditional distributions) in a Bayesian network are locally normalized. This means that for each variable X_i, the sum over all possible values of X_i, given its parents, is equal to 1:

$$\forall x_{\mathrm{Pa}(i)} \sum_{x_i} p(x_i \mid x_{\mathrm{Pa}(i)}) = 1$$

This property ensures that the local conditional distributions of each variable, given its parents, behave as proper probability distributions.

Consistency of Sub-Bayesian Networks - an example



A sample calculation:

$$\mathbb{P}(B = b, E = e) \stackrel{\text{def}}{=} \sum_a \mathbb{P}(B = b, E = e, A = a)$$

$$\stackrel{\text{def}}{=} \sum_a p(b)p(e)p(a|b, e)$$

$$\stackrel{\text{def}}{=} p(b)p(e) \sum_a p(a|b, e)$$

$$\stackrel{\text{def}}{=} p(b)p(e)$$

Bayesian Network and Probabilistic Inference

Bayesian network:

$$\mathbb{P}(X = x) = \prod_{i=1}^{n} p(x_i | x_{\mathrm{Pa}(i)})$$

Where X is a set of random variables, and $x_{\mathrm{Pa}(i)}$ are the values of the parent nodes in the network.

Probabilistic inference:

$$\mathbb{P}(Q|E = e)$$

Here, Q represents the set of query variables for which we want to compute the probability given evidence E, which is the set of observed variables denoted by e.

Inference in Bayesian Networks

- **Purpose of Inference:** To compute the posterior distribution of a subset of nodes given observed values for other nodes (evidence).
- **Types of Inference:**
  - Exact Inference: Utilizes algorithms like Variable Elimination and Junction Tree.
  - Approximate Inference: Includes methods such as Monte Carlo simulations and Belief Propagation.
- **Variable Elimination:**
  - Simplifies the computation by systematically reducing the Bayesian network into a simpler form.
  - Combines and marginalizes over variables not of interest.
- **Belief Propagation (Message Passing):**
  - Useful in networks that are tree-like.
  - Computes local messages that are propagated through the network to perform inference.
- **Challenges in Inference:**
  - Scalability with network size.
  - Computational complexity, especially in networks with many interdependencies.

Variable Elimination

- Variable elimination is a common exact inference algorithm used in Bayesian networks.
- It computes the marginal probability of a target variable by summing out the other variables in the network.
- Variable elimination exploits the structure of the Bayesian network to reduce the computational complexity of inference.
- Mathematically, variable elimination can be represented as:

$$P(X_i \mid e) = \sum_{X_{\text{non-elim}}} P(X_i, X_{\text{non-elim}} \mid e)$$

where X_i is the target variable, e is the evidence or observed variables, and $X_{\text{non-elim}}$ represents the non-eliminated variables in the network.

Variable Elimination in a Bayesian Network

Context: Epidemiological Model of Virus Transmission

To illustrate variable elimination with a simple chain X→Y→Z in a real-world context, let's use a simple example from epidemiology, specifically the transmission of a virus. Here, X represents whether an individual is infected, Y represents whether they show symptoms, and Z represents the spread to a close contact.

- **Variables:**
  - X Infected status (1 for infected, 0 for not infected)
  - Y Symptom presentation (1 for symptomatic, 0 for asymptomatic)
  - Z Transmission to close contact (1 for transmission, 0 for no transmission)

- **Probabilities:**
  - $P(X = 1) = 0.1$, $P(X = 0) = 0.9$
  - $P(Y = 1|X = 1) = 0.8$, $P(Y = 0|X = 1) = 0.2$
  - $P(Y = 0|X = 0) = 1$ (No symptoms if not infected)
  - $P(Z = 1|Y = 1) = 0.3$, $P(Z = 0|Y = 1) = 0.7$
  - $P(Z = 1|Y = 0) = 0.05$, $P(Z = 0|Y = 0) = 0.95$

Calculation of $P(Z = 1)$ Using Variable Elimination

Step-by-Step Elimination Process

- **Step 1: Joint Probability**

$$P(X, Y, Z) = P(Z \mid Y)P(Y \mid X)P(X)$$

- **Step 2: Eliminate Y**

$$P(X, Z) = \sum_{Y} P(Z \mid Y)P(Y \mid X)P(X)$$

Specifically for $Z = 1$:

$$P(X = 1, Z = 1) = (0.3 \times 0.8 \times 0.1) + (0.05 \times 0.2 \times 0.1) = 0.0245$$

$$P(X = 0, Z = 1) = 0.05 \times 1 \times 0.9 = 0.045$$

- **Step 3: Eliminate X**

$$P(Z = 1) = \sum_{X} P(X, Z = 1) = P(X = 1, Z = 1) + P(X = 0, Z = 1)$$

$$P(Z = 1) = \sum_{X} P(X, Z = 1) = 0.0245 + 0.045 = 0.0695$$

- **Conclusion** The final probability of transmission to a close contact, $P(Z = 1)$, is 6.95%.

Belief Propagation

- Belief propagation, also known as sum-product algorithm or message passing, is an efficient algorithm for exact or approximate inference in Bayesian networks.
- It uses message passing between nodes in the network to compute marginal probabilities or beliefs.
- Belief propagation is based on the idea of passing messages along the edges of the graph, updating the beliefs at each node based on the messages received from its neighbors.
- Mathematically, belief propagation can be represented as:

$$\text{bel}(X_i) \propto \prod_{j \in \text{ne}(X_i)} \text{msg}_{j \to i}(X_i)$$

where $\text{bel}(X_i)$ is the belief or marginal probability of node X_i, $\text{ne}(X_i)$ represents the neighbors of node X_i, and $\text{msg}_{j \to i}(X_i)$ is the message sent from node X_j to node X_i.

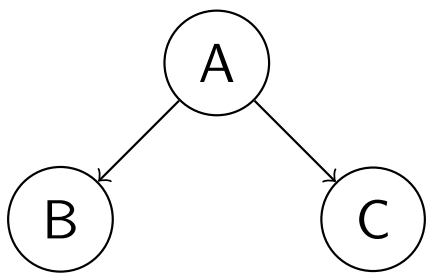
Belief Propagation with Bayesian Networks - Example

Bayesian Network Structure:

- Consider a Bayesian network with nodes A, B, and C, where A influences both B and C.
- These relationships are denoted by directed edges from A to B and from A to C.

Belief Propagation:

- An efficient algorithm for exact or approximate inference in Bayesian networks.
- Utilizes message passing between nodes to compute marginal probabilities.
- Each node updates its belief based on messages from its neighbors.



Belief Propagation Calculation Example

Belief Propagation Mechanics:

- Messages are sent along the edges of the graph.
- The belief or marginal probability of node X_i, denoted as $\text{bel}(X_i)$, is updated as:

$$\text{bel}(X_i) \propto \prod_{j \in \text{ne}(X_i)} \text{msg}_{j \to i}(X_i)$$

- $\text{ne}(X_i)$ represents the neighbors of node X_i.
- $\text{msg}_{j \to i}(X_i)$ is the message sent from node X_j to node X_i.

Example Calculation:

- For node B, the marginal probability $\text{bel}(B)$ incorporates messages from node A.
- The message from A to B, given no evidence, is the prior of A times the conditional probability of B given A.

$$\text{bel}(B) = \sum_A \text{msg}_{A \to B}(B)$$

$$\text{msg}_{A \to B}(B) = P(A)P(B|A)$$

Numerical Calculations in Belief Propagation

Example Numerical Calculation for Node B:

- Given $P(A = \mathit{true}) = 0.6$ and $P(A = \mathit{false}) = 0.4$,
- $P(B = \mathit{true}|A = \mathit{true}) = 0.7$, and $P(B = \mathit{true}|A = \mathit{false}) = 0.2$,

Message from A to B (no evidence):

$$\text{msg}_{A\to B}(B = \mathit{true}) = P(A = \mathit{true})P(B = \mathit{true}|A = \mathit{true}) + P(A = \mathit{false})P(B = \mathit{true}|A = \mathit{false})$$

$$= 0.6 \times 0.7 + 0.4 \times 0.2 = 0.42 + 0.08 = 0.50$$

Belief at Node B:

$$\text{bel}(B = \mathit{true}) \propto \text{msg}_{A\to B}(B = \mathit{true})$$

Since no other nodes influence B, the belief is proportional to the message:

$$\text{bel}(B = \mathit{true}) = \text{msg}_{A\to B}(B = \mathit{true}) = 0.50$$

Normalization of Belief:

$$\text{bel}(B = \mathit{false}) = 1 - \text{bel}(B = \mathit{true}) = 1 - 0.50 = 0.50$$

Sampling-based Methods

- Sampling-based methods are approximate inference algorithms in Bayesian networks.
- They generate samples from the joint distribution of the variables in the network to estimate marginal probabilities or beliefs.
- Sampling-based methods include Markov Chain Monte Carlo (MCMC) methods, such as Gibbs sampling and Metropolis-Hastings, and likelihood weighting.
- Sampling-based methods are useful when exact inference is computationally infeasible or when approximate solutions are sufficient.

Learning Parameters in Bayesian Networks

Learning Objective: To determine the parameters θ which maximize the likelihood of the observed data given a Bayesian network structure.
Parameter Set: The set $\theta = \{\theta_1, \theta_2, \ldots, \theta_m\}$ represents the parameters of the conditional probability distributions associated with the network.
Learning Process:

- Typically involves finding the set of parameters θ that maximizes the likelihood function $L(\theta) = P(Data|Model, \theta)$.
- This can be done using optimization techniques like gradient descent or expectation-maximization (EM) when the data is incomplete or the model includes latent variables.

Parameter Estimation:

- For fully observed data, parameter estimation can often be reduced to counting and normalization:

$$\theta_{x_i|x_{\mathrm{Pa}(i)}} = \frac{N(x_i, x_{\mathrm{Pa}(i)})}{N(x_{\mathrm{Pa}(i)})}$$

  where $N(x_i, x_{\mathrm{Pa}(i)})$ is the count of data instances where $X_i = x_i$ and $X_{\mathrm{Pa}(i)} = x_{\mathrm{Pa}(i)}$, and $N(x_{\mathrm{Pa}(i)})$ is the count of data instances where $X_{\mathrm{Pa}(i)} = x_{\mathrm{Pa}(i)}$.
- For incomplete data or with hidden variables, iterative algorithms like EM are used to estimate the parameters.

Result: The resulting parameters θ can then be used for various inference tasks within the Bayesian network.

Maximum Likelihood Estimation for Bayesian Networks

Algorithm Parameters Estimation for Bayesian Networks

Input: training examples $\mathcal{D}_{\text{train}}$ of full assignments
Output: parameters $\Theta = \{p_d : d \in \mathcal{D}\}$

for each $x \in \mathcal{D}_{\text{train}}$ **do**
  for each variable x_i **do**
    Increment $\text{count}_{d_i}(x_{\text{Parents}(i)}, x_i)$

for each d_i and local assignment $x_{\text{Parents}(i)}$ **do**
  Set $p_{d_i}(x_i | x_{\text{Parents}(i)}) \propto \text{count}_{d_i}(x_{\text{Parents}(i)}, x_i)$

Conclusion

- Bayesian models, including Bayesian networks and Bayesian classifiers, provide a powerful framework for incorporating prior knowledge or beliefs into the modeling and inference process.
- Bayesian models allow for handling uncertainty, making more informed decisions, and providing probabilistic predictions.
- Bayesian networks are graphical representations of probabilistic relationships among variables, and can be used for probabilistic reasoning, inference, and prediction in various fields.
- Bayesian classifiers are popular supervised learning algorithms used for classification tasks, and they are simple, computationally efficient, and can handle categorical and continuous data.
- Bayesian modeling is widely used in diverse areas such as healthcare, finance, artificial intelligence, and many others.

Summary

1. Introduction and reminders
2. Probability review - reminder
3. Bayesian Models
4. Bayesian Classifier
   - Introduction to Bayesian classifiers
   - Naive Bayes algorithm
   - Examples Naive Bayes
5. Bayesian networks
   - Inference in Bayesian Network
6. Summary

References

1. S. J. Russell, P. Norvig, "Artificial Intelligence: A Modern Approach", Financial Times Prentice Hall, 3rd ed., 2019.
2. J. Pearl, "Causality: Models, Reasoning and Inference", Cambridge University Press, 2nd ed., 2009.
3. K.B. Korb, A.N. Nicholson, "Bayesian Artificial Intelligence", CRC Press, 2011.
4. M. Flasiński, "Introduction to Artificial Intelligence", Springer Verlang, 2016
5. M. Muraszkiewicz, R. Nowak (ed.), "Sztuczna Inteligencja dla inżynierów", Oficyna Wydawnicza PW, 2022