

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ НАЦІОНАЛЬНИЙ
УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»

Факультет інформатики

Кафедра інформатики

АНАЛІЗ ТА ДОСЛІДЖЕННЯ МЕТОДІВ ТА ЗАСОБІВ НАВЧАННЯ
БАЙЄСІВСЬКИХ МЕРЕЖ

КУРСОВА РОБОТА

студентки III курсу

Ніколаюк Дарії Андріївни

Науковий керівник:

к.ф.-м.н, ст. викладач Ющенко Ю.О.

КИЇВ 2020

Міністерство освіти і науки України
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «КИЄВО-МОГИЛЯНСЬКА АКАДЕМІЯ»
Кафедра інформатики факультету інформатики

ЗАТВЕРДЖУЮ
Зав.кафедри інформатики,
доцент, к.ф.-м.н.
_____ С. С. Гороховський
(підпис)
“ ____ ” _____ 202_ р.

ІНДИВІДУАЛЬНЕ ЗАВДАННЯ
на курсову роботу

студенту _____ Ніколаюк Дарії

_____ 3 _____ курсу факультету інформатики

ТЕМА: аналіз та дослідження методів та засобів навчання байєсівських мереж

Індивідуальне завдання

Вступ

1. Аналіз предметної області. Постановка завдання курсової роботи
2. Теоретичні відомості
3. Байєсівські мережі в мовах програмування
4. Демонстрування байєсівських мережі

Висновки

Список використаних джерел

Дата видачі „ ____ ” _____ 2020 р. Керівник: к.ф.-м.н, ст. викладач Ющенко Ю.О.

Завдання отримала _____

Тема: Аналіз та дослідження методів та засобів навчання байєсівських мереж

Календарний план виконання роботи:

№ п/п	Назва етапу дипломного проекту (роботи)	Термін виконання етапу	Примітка
1.	Отримання завдання на курсову роботу.	06.11.2019	
2.	Огляд технічної літератури за темою роботи.	20.01.2020	
3.	Виконання аналізу актуальності теми	03.02.2020	
4.	Написання пояснювальної роботи.	20.03.2020	
5.	Попередня демонстрація проекту науковому керівнику	20.04.2020	
7.	Створення презентації	09.05.2020	
8.	Захист курсової роботи		

Студент _____ Ніколаюк Д. А. _____

Керівник _____ Ющенко Ю. О. _____

Зміст

Перелік умовних позначень	4
Вступ.....	6
РОЗДІЛ 1: Аналіз предметної області. Постановка завдання курсової роботи ...	8
1.1 Аналіз сучасного стану питання та обґрунтування теми	8
1.2 Мережі Байєса у житті.....	8
1.3 Огляд існуючих методів навчання байєсівських мереж.....	9
РОЗДІЛ 2: Теоретичні відомості	10
2.1 Основні терміни теорії ймовірності	10
2.2 Байєсівський та частотний підхід до статистики.....	15
2.3 Поняття байєсівські мережі	19
2.4 Типи БМ	21
2.5 Навчання байєсівської мережі	23
2.6 Байєсове виведення.....	24
2.7 Наївний байєсівський класифікатор мережі	26
РОЗДІЛ 3: Байєсівські мережі в мовах програмування	27
3.1 Байєсівські мережі в мові програмування R	27
3.2 Байєсівські мережі в мові програмування Python	28
РОЗДІЛ 4: Демонстрування байєсівських мережі використовуючи пакет romegranate	33
4.1 Аналіз задачі	33
4.2 Мережа, що описує задачу	33
4.3 Опис коду програми.....	34
4.4 Результат виконання програми.....	36
Висновки	37
Список використаної літератури	38

Перелік умовних позначень

БМ – байєсівські мережі.

КГ – метод Купера та Гершковича.

ОМД - оцінка мінімальної довжини.

Апріорна ймовірність — апріор

Prior - апріор

Posterior – апостеріор

НБ - наївна байєсівська мережа

ПК – персональний комп'ютер

Вступ

Актуальність, наукове та практичне значення обраної теми

Працювати з ймовірностями в галузі інформатики може бути трохи незвично, оскільки більшість галузей мають справу з детермінованими та точними структурами. Однак, коли ми говоримо про штучний інтелект або науку даних взагалі, невизначеність та стохастичність можуть проявлятися у багатьох формах. Дані, безумовно, є основним джерелом невизначеності, але джерелом також може бути модель. Теорія ймовірностей надає інструменти для моделювання та подолання невизначеності, а байєсівські мережі це одні із таких інструментів. Програми в реальному світі мають ймовірнісний характер, а взаємозв'язок між кількома подіями описується простою та природною формулою Бйаєса. Сукупності пов'язаних подій реального світу можна можна модулювати за допомогою ланцюжків, які можуть перетинатися, та утворювати довільні, будь якої складності структури без циклів. Такий підхід дозволяє модулювати залежності в довільних предметних областях та використовуватись для розв'язку широкого кола задач. Залежності між подіями можна представляти дугами зображеними на відповідних графів. Подання пов'язаних подій за допомогою графів називають мережею Байєса. Байєсівські мережі (БС) тісно пов'язані з діаграмами впливу, які успішно використовуються для прийняття рішень. Байєсівські мережі допомагають вирішити складні проблеми з неточною та неповною інформацією. Основною ідеєю побудови графічної моделі є поняття модульності, тобто розкладання складної системи на прості елементи. Вони впроваджуються в найбільш прогресивних технологіях епохи, таких як штучний інтелект та машинне навчання. Байєсівські мережі успішно застосовуються у багатьох сферах та не мають обмежень сфер застосування. Широкого використання БМ набули в сферах діагностики захворювань, в пошукових системах при оптимізації пошуку, застосовуються для фільтрації спаму,

моделюють регуляторні мережі генів тощо. І цей список можна розширити задачами прогнозування, класифікації, прийняття рішень тощо. Основна мета цих мереж це модулювання причинно-наслідкових відносин, а головною задачею навчання БМ є її вдосконалення задля покращення результатів розв'язку задач. Таким чином, БМ являють собою вельми простий спосіб надання ознак розумності програмним системам штучного інтелекту. Однією з переваг БМ є швидкість та простота процесу навчання при порівнянні з іншими засобами розробки систем штучного інтелекту.

Існує низка продуктивних Байєсівських систем, які при відсутності датасетів успішно навчалися розв'язувати складні задачі навіть в тих областях, в яких відсутні спеціалісти.

Мета та завдання курсової роботи

Мета: Проаналізувати застосування байєсівських мереж та порівняння різних методів навчання у різних мовах програмування.

Завдання: Дослідження способів навчання байєсівських мереж у різних мовах програмування та демонстрування роботи байєсівської мережі на задачі.

Об'єкт дослідження

Бібліотеки різних мов програмування з байєсівськими мережами.

Методи дослідження

Дослідження методів та засобів навчання БМ буде здійснюватися за допомогою вивчення математичної складової проблеми та аналізу документацій різноманітних мов програмування.

Джерела дослідження

Робота побудована на ретельному аналізі та дослідженні багатьох сторін байєсовських мереж. Тож, було використано у навчальних цілях й опрацьовано багато наукових інтернет ресурсів, пов'язаних з темою роботи щодо машинного навчання, та для аналізу і порівняння наявних засобів у веб-просторі.

РОЗДІЛ 1: Аналіз предметної області. Постановка завдання курсової роботи

1.1 Аналіз сучасного стану питання та обґрунтування теми

На сьогоднішній день, засоби машинного навчання є майже у кожній мові програмування. Існують вже готові бібліотеки за допомогою яких існує можливість швидко та зручно розв'язувати поставлені задачі. Способів використання БМ у житті дуже багато, тому важливо розуміти якими методами та засобами можна побудувати правильну модель.

1.2 Мережі Байєса у житті

Мережі Байєса мають безліч застосувань у різноманітних сферах, включаючи біоінформатику, медицину, , пошук інформації тощо. ^[5]

1. Діагностика хвороб: Байєсові мережі зазвичай використовуються в галузі медицини для виявлення та профілактики захворювань. Вони можуть бути використані для моделювання можливих симптомів та прогнозування того, чи є людина хворою чи ні.
2. Фільтрація спаму: Байєсові моделі використовуються в алгоритмі фільтрації спаму Gmail вже багато років. Вони можуть ефективно класифікувати документи, розуміючи контекстуальне значення листів. Вони також використовуються в інших програмах класифікації документів.
3. Біомоніторинг: Байєсові мережі відіграють важливу роль у моніторингу кількості хімічних доз, що застосовуються у фармацевтичних препаратах.
4. Оптимізований пошук в Інтернеті: для покращення точності пошуку використовуються байєсівські мережі, розуміючи наміри пошуку та надаючи найрелевантніші результати пошуку. Вони можуть ефективно відображати наміри користувачів на відповідний вміст та доставляти результати пошуку.

5. Регуляторні мережі генів: GRN - це мережа генів, яка складається з багатьох сегментів ДНК. Вони ефективно використовуються для прямого або опосередкованого спілкування з іншими сегментами клітини. Для моделювання такої поведінки комірок та формування прогнозів використовуються математичні моделі, такі як байєсівські.

1.3 Огляд існуючих методів навчання байєсівських мереж

Завдання побудови байєсівської мережі по заданих навчальних даних є NP-важкою (тобто - це завдання нелінійної поліноміальної складності). Тому розробка методів, які будуть дозволяти зменшити обчислювальну складність, є актуальною і затребуваною при моделюванні процесів різної природи мережами Байєса. Основними методами навчання є метод опису мінімальної довжини, метод Купера та Гершковича, а також деякі модернізації цих методів, такі як K2 та TУО. Для зменшення обчислювальної складності пропонується використовувати попередню впорядкованість вершин та ставиться завдання розробки евристичного методу побудови байєсівської мереж, що складається з двох етапів. На першому етапі виконується обчислення значення обопільної інформації між усіма вершинами. На другому виконується цілеспрямований пошук, який використовує в якості оціночної функції оцінку мінімальної довжини (ОМД), засновану на принципі опису, який застосовується на кожній ітерації алгоритму навчання. Використання евристичного методу навчання істотно розширює можливості використання байєсівських мереж при проведенні аналізу в різних областях людської діяльності, особливо там, де доводиться працювати з великими обсягами інформації. Алгоритм навчання БМ з використанням ОМД буде виглядати наступним чином, відбувається перебір у циклі усіх можливих не циклічних мережевих структур, далі найоптимальніша зберігається. Такою структурою буде та, в якій значення функції буде найменше.

Інший метод навчання байєсівських мереж називається методом Купера та Гершковича(КГ). В методі ОМД шукається структура з мінімальною довжиною опису, тоді як в методі КГ потрібно знайти структуру з максимальним значенням функції. На початку по циклу відбувається перебір усіх можливих не циклічних мережевих структур та зберігається найоптимальніша. Оптимальною структурою буде та, у якій значення функції буде найбільше.

РОЗДІЛ 2: Теоретичні відомості

2.1 Основні терміни теорії ймовірності

Для того, щоб почати роботу з байєсівськими методами потрібно розуміти основні терміни теорії ймовірності: випадкова подія, незалежна подія, умовно залежні події, повна ймовірність та інші. [2]

Стохастичний експеримент - експеримент, наслідки якого неможливо передбачити до проведення експерименту, але його можна повторити будь-яке число разів, він буде мати стійку поведінку частот

Простір елементарних подій – це сукупність елементарних подій, тобто множина всіх можливих наслідків стохастичного експерименту
Випадкова подія – це подія, яка може відбутися, а може не відбутися за певних заданих умовах.

Події називаються несумісними, якщо вони не можуть одночасно відбутися в одному експерименті.

Події називаються незалежними, якщо поява однієї з подій не впливає на ймовірність появи іншої.

Випадкова величина – величина, можливими значеннями якої є результати випробувань чи спостережень явищ або процесів, що носять випадковий характер.

Умовна ймовірність - ймовірність події А, обчислена при умові, що подія В відбулася ($P(A|B)$) Умовна ймовірність – це концепція, яка відповідає питання: як нам оновити наше вірування у світлі доказів, які ми спостерігаємо?

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

Подія А -подія, невизначеність якої ми хочемо оновити

В – спостереження, яке нам дано

$P(A)$ – апіорна ймовірність події А(апіор)

$P(A|B)$ - апостеріорна ймовірність події А
Тобто, апіор (prior) – наша модель до оновлення, а апостеріор (posterior) - наше переконання щодо гіпотези після оновлення на основі наших спостережень.

$$P(A \cap B) = P(A|B) \cdot P(B)$$

Спільна ймовірність (joint probability) - це міра двох подій, які відбуваються одночасно, і застосовується, коли одночасно може відбуватися більше одного спостереження. Спільна ймовірність лише визначає ймовірність настання обох подій. Умовна ймовірність використовується для обчислення ймовірності спільного, як видно з цієї формули:

$$P(A \cap B) = P(A|B) \cdot P(B)$$

Наприклад, з колоди 52 карт ймовірність витягнути червону шістку дорівнює $\frac{1}{26}$, оскільки колода карт містить чотири карти шісток та тільки дві з них червоні, розв'язок можна за показати за формулою спільної ймовірності:

$$P(6 \cap \text{red}) = P(6) \cdot P(\text{red}) = \frac{4}{52} \cdot \frac{26}{52} = \frac{1}{26}$$

Символ « $\cap$ » у спільній ймовірності називається перетином. Ймовірність того, що подія А та подія В відбудуться - це те саме, що і точка, де А та В перетинаються.

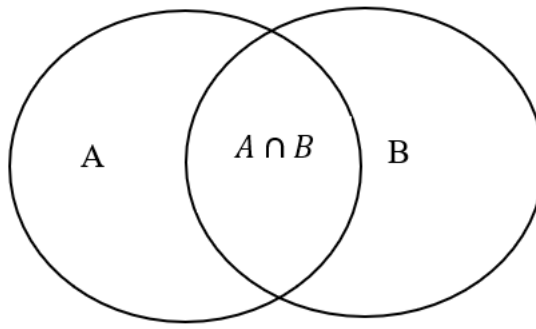


Рисунок 2.1 – перетин двох подій

Тому спільною ймовірністю також називають перетин двох і більше подій. Аналітики використовують спільну ймовірність як інструмент, коли дві або більше спостережуваних подій можуть статися одночасно. [3]

Повна ймовірність - пов'язує умовну ймовірність з незалежними подіями . Вона показує, що умовна ймовірність може бути використана для розкладання складних задач на простіші частини, і вона часто використовується в тандемі з правилом Байєса.

Нехай існує повна група несумісних подій(гіпотез): $B_1, B_2, \dots, B_n$

Ймовірність настання події А за умовою настання гіпотези B_i – відомі. Щоб порахувати безумовну вірогідність події А без знання яка саме гіпотеза трапилася, потрібно використати Формулу повної ймовірності.

$$P(A) = \sum_{j=1}^n P_{B_j}(A) \cdot P(B_j)$$

$P(A)$ – вірогідність настання події А

$P_{B_j}(A)$ ймовірність настання події А за умовою настання гіпотези B_j

$P(B_j)$ – вірогідність настання гіпотези B_j

Теорема Байєса:

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

Важливою властивістю правила Байєса є те, що воно є узгодженим, тобто, якщо ми отримуємо декілька частин інформації і хочемо оновити наші ймовірності, щоб включити всю інформацію, не має значення, чи ми оновлюємо послідовно, беручи до уваги кожну частину інформації один за одним, або одночасно, використовуючи всю інформацію одночасно. Головна ідея, що лежить в теоремі Байєса, це те, що нова інформація не повинна повністю визначати наші переконання, а лише оновлювати наші апріорні. Розглянемо приклад використання теореми Байєса та формули повної ймовірності на прикладі діагностування рідкої хвороби. Нехай в нас є пацієнт, який пройшов тест на хворобу, яка можлива лише у 1% населення. Результат тесту виявився позитивним. Нехай D – подія, що в пацієнта є хвороба, а T – подія, що тест позитивний. Припустимо, що тест правильно надає результат у 95% , тобто $P(T|D)=0,95$. Щоб знайти ймовірність того, що у пацієнта дійсно є ця хвороба, якщо тест на неї позитивний, застосуємо теорему Байєса та формулу повної ймовірності:

$$P(D|T) = \frac{P(T|D) \cdot P(D)}{P(T)} = \frac{P(T|D) \cdot P(D)}{P(T|D)P(D) + P(T|D^c)P(D^c)} = \frac{0,95 \cdot 0,01}{0,95 \cdot 0,01 + 0,05 \cdot 0,99} \approx 0,16$$

Тобто, ймовірність того, що пацієнт хворий, якщо тест на хворобу позитивний лише 16%. Такий результат може здатися неочікуваним для багатьох людей, адже надійність тесту складала 95%. Треба розуміти 2 фактори: надійність тесту та попередня(апріорна) інформація щодо поширеності захворювання. Хоча надійність тесту і була дуже висока, вірогідність захворіти дуже маленька, саме тому умовна ймовірність $P(D|T)$ відображає баланс між цими двома факторами: зважуючи рідкість хвороби проти рідкості помилкового результату тесту маємо відповідний результат. Для наглядного зображення використаємо такий малюнок

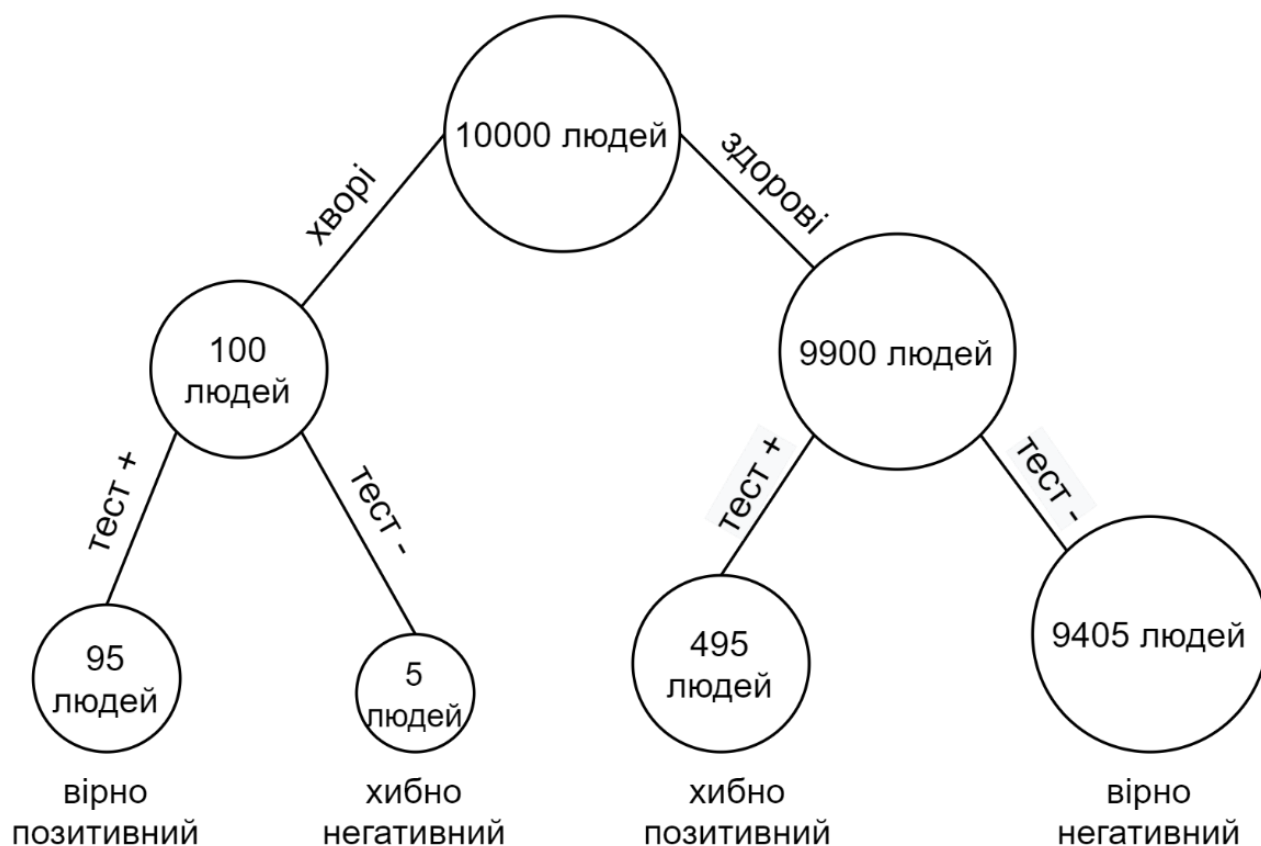


Рисунок 2.2 - відношення людей у задачі про хворобу

Даний малюнок наглядно ілюструє відношення людей у яких виявили позитивно негативний результат(95 людей) та тих, у кого виявили хибно негативний результат(495 людей), це означає, що більшість людей, які отримали позитивний тест на хворобу насправді не хворі. відношення людей ^[4] Де: статистичні показники діагностичного діагнозу щодо виявлення хворих та здорових, які виходять із помилок першого та другого роду у бінарній класифікації. Їх значення лежать у межах від 0 до 1.

Таблиця 2.1 – статистичні показники

	Захворювання є	Захворювання нема
Тест позитивний	Вірно позитивні	Хибно позитивні
Тест негативний	Хибно негативні	Вірно негативні

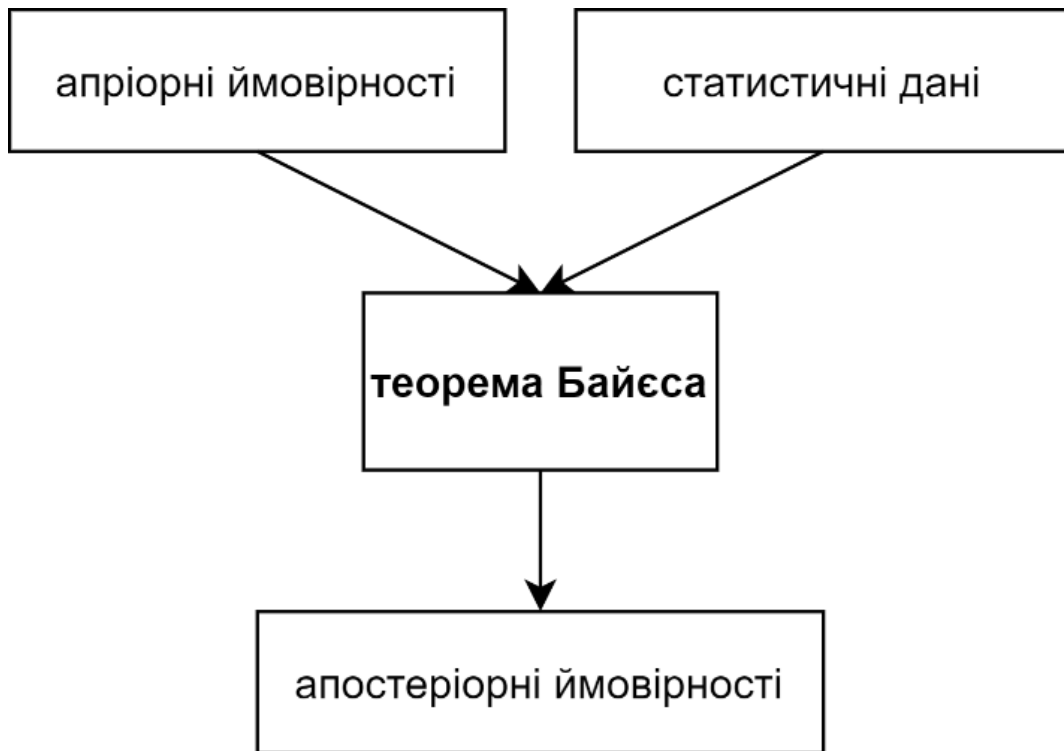


Рисунок 2.3 - основна байєсівська парадигма

2.2 Байєсівський та частотний підхід до статистики

Розглянемо два підходи до статистики: частотний (frequentist) та байєсівський. Тож вони різняться у багатьох аспектах. А найважливішим є те, що частотний це об'єктивний, а байєсівський - суб'єктивний. Уявимо, що у вас є монета, і ви кидаєте її. Частотний підхід каже, що ймовірність, того, що випаде орел буде одна друга і ймовірність, що випаде герб буде одна друга. І ви нічого не можете з цим зробити. Однак байєсівський каже, що якби хтось знав початкову швидкість і всі початкові параметри кидання монети, то результат експерименту можна було передбачити. І для цієї людини експеримент не був би випадковим. Ще одна відмінність між частотними та байєсівськими підходами це те, як вони обробляють параметри даних. Частковий говорить, що параметри фіксовані, а дані - випадкові. І вони хотіли б знайти цю оптимальну точку. Однак байєсівськи сказали би навпаки, що параметри є випадковими і дані фіксовані. І це насправді

має сенс. Коли модель тренується, дані вже відомо, і вони вже не випадкові. Однак може бути декілька хороших точок, і хочеться визначити розподіл ймовірностей над ними. Також байєсівські методи працюють для довільної кількості літерних точок. Частковий підхід працює лише тоді, коли кількість точок даних набагато більша, ніж кількість параметрів. Може здатися, що це не проблема, коли ми маємо великі дані, однак треба пам'ятати, що для нейронної мережі у нас є мільйони параметрів, а точок даних лише тисячі. Ще одна відмінність полягає в тому, як частковий та байєсівські підходи навчають свої моделі. Частковий підхід тренує свою модель за методом максимальної правдоподібності, намагаючись знайти параметри тети, що максимально збільшують ймовірність даних (likelihood), заданих параметрів. Однак, байєсівські намагаються обчислити апостеріорну ймовірність (posterior), вірогідність параметрів за даними. Це вони роблять за допомогою формули Байєса. ^[6]

Байєсівські підходи чудово підходять для машинного навчання. До байєсівських відносять групу методів ймовірнісно-статистичного оцінювання даних, які розроблено в результаті формулювання та пошуку оптимальних розв'язків задач аналізу поведінки процесів і систем різної природи. Більшість статистичних задач, незалежно від методів їх розв'язування, мають деякі загальні властивості, а саме: до отримання конкретної вибірки даних стосовно поведінки досліджуваного процесу потенційно можливими вважаються кілька ймовірнісно-статистичних моделей. Після отримання даних виникають виражені у числовому вигляді знання щодо відносної прийнятності цих моделей для розв'язання конкретної задачі. Одним зі способів аналізу відносної прийнятності ймовірнісних моделей є байєсівський підхід. ^[7]

Є дві школи статистичного виводу: байєсівська і частотна. Обидва підходи дозволяють оцінити докази щодо конкуруючих гіпотез. Байєсівська школа моделює невизначеність шляхом розподілу ймовірностей за гіпотезами. Можливість робити висновки залежить від ступеня впевненості у вибраному

апріорі, і надійність вказівки щодо чергування попередніх апріорів може бути доречною та важливою. Школа частотного виводу(frequentist school) використовує лише умовні розподіли даних, задані конкретними гіпотезами. Припущення полягає в тому, що деяка гіпотеза (параметр, що визначає умовний розподіл даних) є істинною, і що спостережувані дані відбираються з цього розподілу. Зокрема, частотний підхід не залежить від суб'єктивного апріора, який може відрізнятися від одного дослідника до іншого. Ці дві школи можуть бути додатково протиставлені наступним чином: [8]

Байєсівський вивід (Bayesian inference):

- використовує ймовірності як для гіпотез, так і для даних
- залежить від апріора і вірогідності спостережуваних даних
- вимагає того, хто може побудувати "суб'єктивний апріор"
- може бути складно обчислювальний через інтегрування за багатьма параметрами

Частотний вивід (Frequentist inference):

- ніколи не використовують або не дають імовірність гіпотези (апріорної чи апостеріорної)
- залежить від ймовірності $P(A|B)$ як для спостережуваних, так і для незабезпечених даних
- не вимагає апріора
- має тенденцію бути менш складно обчислювальним.

Наприклад, маємо певну інформацію(датасет), яка надходить невеликими частинами. Для цього можна використати цю нову інформацію для оновлення наших параметрів, вона і буде нашим новим апостеріором, який буде використовуватися як апріор для наступного нового експерименту. Для наглядності подивимося на графік, де θ це параметр, який оцінюється. Апріорний графік:

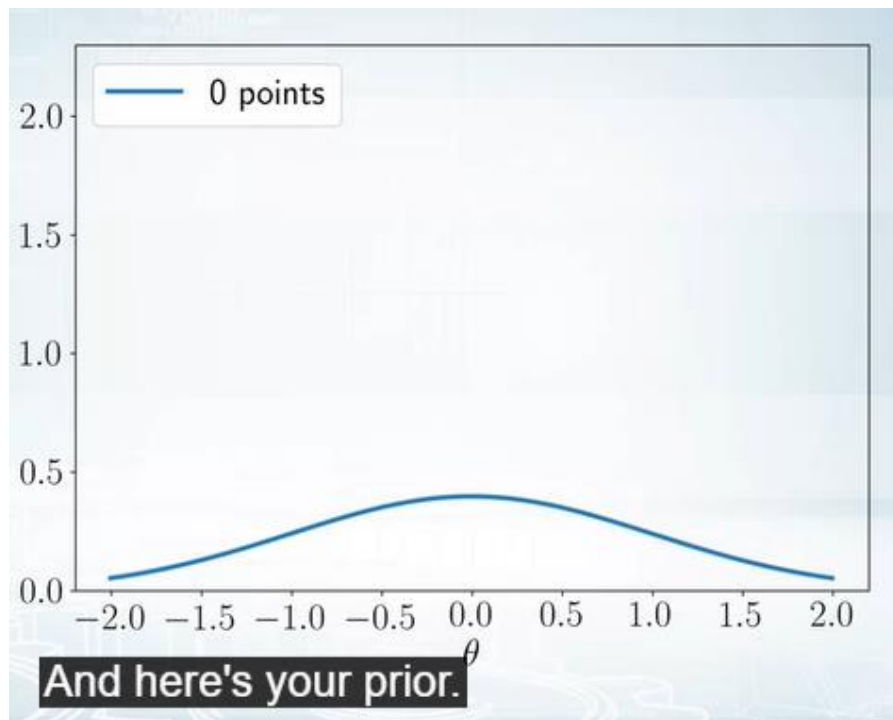


Рисунок 2.4 – апріорні дані

Оновимо наш графік, візьмемо 10 точок:

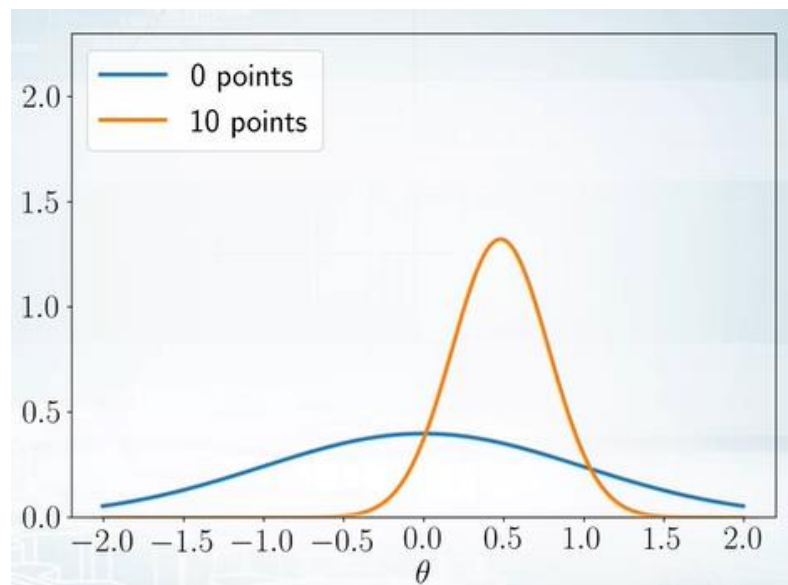


Рисунок 2.5 – оновлений апріор

Можна побачити, що з оновленням середня значення функції змінилося та дисперсія розподілу зменшилася. Таким чином, чим більше точок ми маємо, тим точніше становиться наша функція і тим краще ми можемо передбачити значення параметрів.

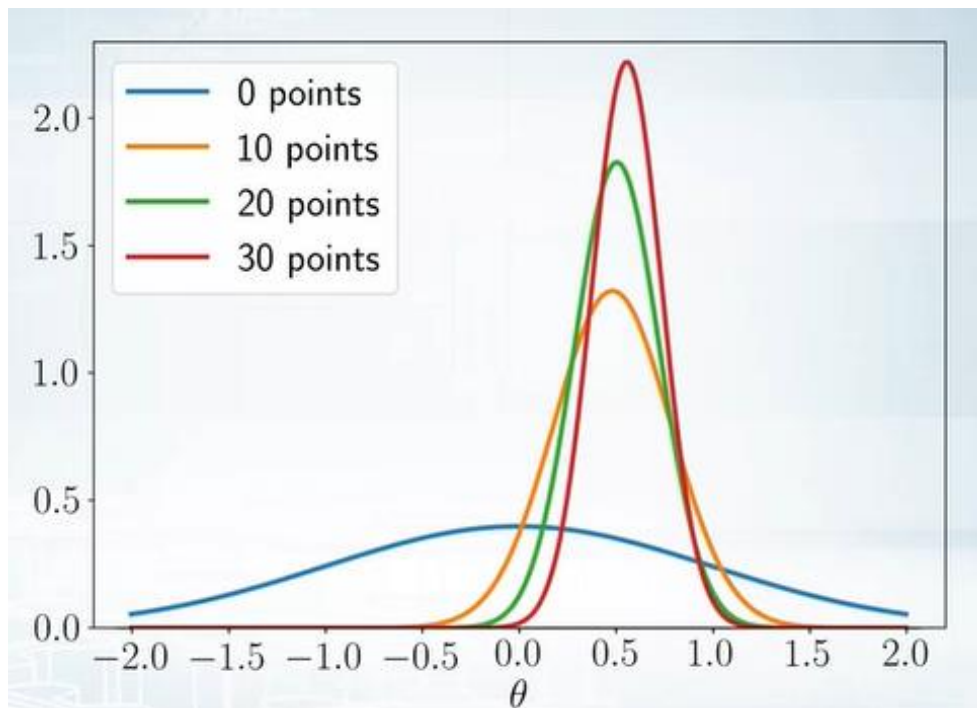


Рисунок 2.6 – оновлений апріор (30 точок)

2.3 Поняття байєсівські мережі

Більшість інструментів обробки даних базується на двох технологіях: машинне навчання та візуалізація (наочне представлення інформації). Це дві технології просто поєднуються у байєсівських мережах (БМ). Це порівняно молода тенденція в науці, що виникла на перетині теорії ймовірностей та теорії графів

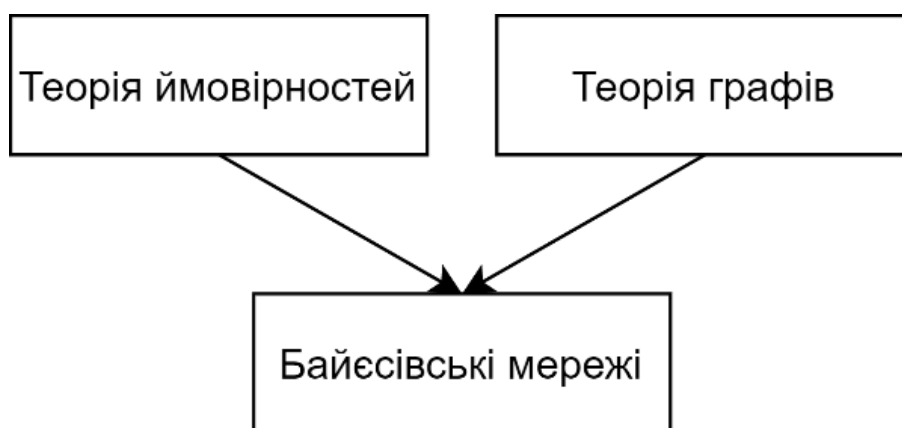


Рисунок 2.7 - БМ на стику двох наук

Байєсові мережі (БМ) - зручний імовірнісний інструмент для опису динаміки та статистики процесів різного характеру з метою подальшого аналізу особливостей їх функціонування, виявлення причинно-наслідкових зв'язків між змінними та

передбачення поведінки їх подальшого розвитку, розпізнавання ситуацій та закономірностей тощо. Графічні моделі представляють собою графи, вузли яких відповідають випадковим змінним. Якщо ці вузли (змінні) не з'єднані дугами, то їх вважають умовно незалежними. Ненаправлені графічні моделі називають також марковськими випадковими полями (МВП). Для МВП незалежність можна сформулювати наступним чином: дві множини (вузлів) A і B є умовно незалежними при наявності в моделі третьої множини C , якщо всі шляхи між вузлами множини A і B розділені вузлами множини C . Байєсівська мережа (БМ) довіри - це направлений ациклічний граф. БМ - це пара, в якій перший компонент G , є направленим нециклічним графом, який відповідає випадковим змінним. З БМ пов'язано більш складне поняття незалежності, яке враховує направлення дуг. Граф записаний, як набір умов незалежності: кожна змінна незалежна від її батьків G . Друга компонента пари, являє собою набір параметрів, які визначають мережу. З математичної точки зору БМ - це модель для представлення імовірнісних залежностей, а також відсутність цих залежностей. При цьому зв'язок $A \rightarrow B$ є причиною, а подія A є причиною виникнення B , впливає на значення, прийняте B . БМ називають причиною (каузальною), коли всі її зв'язки є причинними. Є й ряд наступних причин використання причинних моделей в штучному інтелекті: [9]

- Як правило, людина інтерпретує події з точки зору причина-результат, це спрощує для користувача розуміння причинних моделей
- ідентифікація інваріантних причинних зв'язків в конкретній задачі, дозволяє прогнозувати ефекти, що виникають в наслідок випадкових подій (випадкових змінних), і ефекти, зумовлені вже визначеними діями (тобто, різними маніпуляціями або інтервенціями)
- причинність і ймовірність тісно пов'язані між собою, тому що причинність зазвичай передбачає існування імовірнісних взаємозалежностей, що забезпечують розуміння причинності; фактично, необхідною умовою існування причинності є кореляція;

Деякі методи пояснення, що використовуються в БМ, розроблені спеціально для причинних мереж або навіть для конкретних канонічних моделей, а інші методи є загальними в тому сенсі, що вони не припускають причинної інтерпретації мережі. Найпоширенішим завданням, яке вирішується за допомогою байєсівської мережі, є ймовірнісний вивід.

2.4 Типи БМ

Дискретні БМ - мережі, в яких змінні використовуються дискретними величинами. Дискретні БМ мають наступні властивості:

- кожна вершина представляє собою подію, описане випадковою величиною, яка може мати кілька станів;
- всі вершини, пов'язані з «батьківськими», визначаються таблицею умовних ймовірностей або функцією умовних ймовірностей;
- для вершин без «батьків» ймовірності її станів є безумовними (маргінальними).

Іншими словами, в байєсівських мережах довіри вершини представляють собою випадкові змінні, а дуги - ймовірнісні залежності, які визначаються за допомогою таблиць умовних ймовірностей.

Динамічні БМ — мережі, в яких значення вузлів змінюється з часом. Динамічні БМ ідеально підходять для моделювання тимчасових процесів. Їх плюс в тому, що вони використовують табличне представлення умовних ймовірностей, що полегшує представлення різних нелінійних явищ. Термін «тимчасова байєсівська мережа» (temporal Bayesian network) краще підходить ніж «динамічна байєсівська мережа» (dynamic Bayesian network), так як мається на увазі, що структура моделі не змінюється. Зазвичай параметри моделі не змінюється з часом, але завжди можна додавати додаткові скриті вузли, щоб описати поточний стан. Найпростіший тип динамічної БМ - це прихована модель Маркова (Hidden Markov Model), у якій в кожному шарі є один дискретний прихований вузол і один дискретний або безперервний спостережуваний вузол.

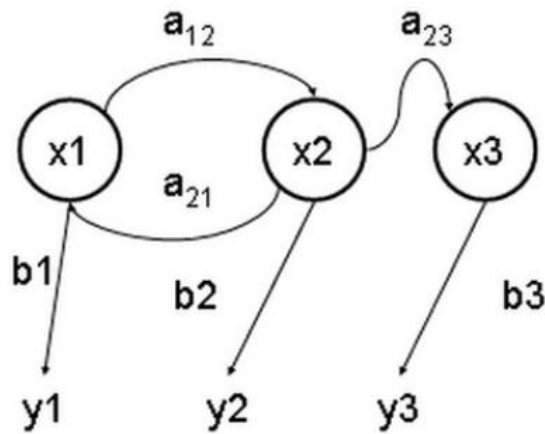


Рисунок 2.8 - діаграма переходів в прихованій Марківського моделі

Де:

x - приховані стани

y - спостережувані результати

a - ймовірності переходів

b - ймовірність результату

У звичайній марковській моделі стан видимий спостерігачеві, тому ймовірності переходів - єдиний параметр. У прихованій марковській моделі ми можемо спостерігати лише за змінними, на які впливає даний стан. Кожен стан має імовірнісний розподіл серед всіх можливих вихідних значень. Тому послідовність символів, згенерована приховано. Марківською моделлю, дає інформацію про послідовності станів. Мережі такого виду використовують у розпізнаванні мовлення, криптоаналізу, машинного перекладу тощо.

Неперервні БМ – змінні вузлів мережі є неперервними величинами. У багатьох випадках події можуть приймати будь-які стани із деякого діапазону. Тому розподіл ймовірності для неперервної випадкової величини визначаються інакше, ніж у дискретному випадку і для їх опису використовуються функції розподілу ймовірностей і густини розподілу ймовірностей. Безперервні БМ використовуються для моделювання стохастичних процесів в просторі станів з безперервним часом.

Гібридні БМ – мережі, які мають вузли як з дискретними змінними так і з неперервними.

2.5 Навчання байєсівської мережі

Для опису байєсової мережі необхідно визначити топологію графа і параметра кожного вузла. Цю інформацію ми можемо отримати з нашого датасету. Але виявлення правильної топології не легка задача, особливо, коли деякі вузли приховані або дані некоректні чи недостатні. Можна окреслити 4 випадки:

Таблиця 2.2 – методи навчання

Структура	Спостереження	Метод
Відома	Повне	Метод максимальної правдоподібності
Відома	Часткове	Жадібний метод пошуку екстремуму
Невідома	Повне	Пошук в просторі моделей
Невідома	Часткове	Структурний алгоритм максимізації математичного сподівання або стиснення кордонів

Відома структура, повне спостереження. В цьому випадку обчислюються значення параметрів кожного умовного імовірнісного розподілу, які максимізують правдоподібність навчальних даних. Нормалізоване логарифмічне рівняння правдоподібності має вигляд:

$$L = \frac{1}{N} \sum_{i=1}^m \sum_{l=1}^s \log (P(X_i | Pa(X_i), D_l))$$

$Pa(X_i)$ – набір батьків

D – навчальна множина

D_l – l - й елемент навчальної множини

N – кількість подій в навчальній множині

Відома структура, часткове спостереження. Коли деякі з вузлів приховані, то можна застосувати алгоритм максимізації математичного очікування (ММО), для знаходження локальної оптимальної оцінки максимальної правдоподібності (ОМП) параметрів. Основна ідея алгоритму ММО полягає в тому, що якби ми знали значення всіх вузлів, навчання було б простим, оскільки знайомі з попередніми. Так на наступному етапі, ми обчислюємо очікувані значення вузлів, що використовують алгоритм виведення, і потім використовуємо ці значення так, як якби вони були отримані зі спостережень.

Невідома структура, повне спостереження. Найбільш імовірною моделлю в даному випадку є повний граф, тому що в цьому випадку буде задіяно найбільшу кількість параметрів, отже така модель буде найбільше відповідати даним.

Невідома структура, часткове спостереження. Це найскладніший випадок, коли структура невідома і є некоректні дані і приховані змінні. У цьому випадку використовують структурний алгоритм максимізації математичного очікування (СММО) або алгоритм стиснення кордонів (Bound and Collapse) Алгоритм СММО поєднує в собі стандартний алгоритм ММО, який оптимізує параметри, з структурним пошуком моделі відбору. Цей алгоритм навчає мережі, ґрунтуючись на штрафних імовірнісних значеннях, які включають значення, отримані за допомогою байєсівського інформаційного критерію, принципу мінімальної довжини опису, а також значення інших критеріїв. Метод стиснення кордонів (СК) моделює відсутність даних, припускаючи, що ймовірність відсутності даних знаходиться в інтервалі від 0 до 1. ^[10]

2.6 Байєсове виведення

Після того як ми побудували байєсівську мережу (за попередніми знаннями, даними або комбінацією), нам зазвичай потрібно визначити різні ймовірності, що цікавлять модель. Ймовірності не зберігаються безпосередньо в моделі, і

тому їх потрібно обчислювати. Взагалі, обчислення ймовірності інтересу за даною моделлю називається

Розглянемо імовірнісний вивід у байєсівських мережах. Кілька дослідників розробили ймовірнісні алгоритми виводу для байєсівських мереж з дискретними змінними. Наприклад, Говард і Матесон у 1981 році, Олмстед у 1983 році та Шахтер у 1988 році розробили алгоритм, який реверсує дуги в структурі мережі, поки відповідь на даний імовірнісний запит не можна буде прочитати безпосередньо з графа. У цьому алгоритмі кожне реверсування дуги відповідає застосуванню теореми Байєса. Схема передачі повідомлень, яка оновлює розподіли ймовірностей для кожного вузла в байєсівській мережі у відповідь на спостереження однієї або декількох змінних була розроблена Перлом у 1986 році. Дженсен(1990) та Лаурітцен та Шпігельхальтер(1988) створили алгоритм, який спочатку перетворює байєсівську мережу в дерево, де кожному вузлу в дереві відповідає підмножина змінних у X . Потім алгоритм використовує декілька математичних властивостей цього дерева для здійснення імовірнісного виводу. Найпопулярніший алгоритм дискретних змінних є алгоритмом Лаурітцена та Шпігельхальтера (1988), Дженсена та ін. Методи точного виводу в мережах Байєса, що кодують багатовимірні гауссівські розподіли були розроблені Шатчером та Кенлі у 1989 році та Лаурітzenом у 1992. Ці методи також використовують твердження про умовну незалежність для спрощення виводу.

Хоча ми використовуємо умовну незалежність для спрощення імовірнісного виводу, точний висновок у довільній байєсівській мережі для дискретних змінних є NP важкою задачею. Навіть приблизний вивід такий як методи Монте-Карло є NP важкою задачею. Так виходить через ненаправлені цикли в структурі мережі Байєса - цикли в структурі, де ми ігноруємо спрямованість дуг. Коли структура байєсівської мережі містить безліч ненаправлених циклів, вивід є непереборним. Однак для багатьох застосувань

структури є досить простими або можуть бути спрощені, не жертвуючи великої точності, для ефективного виводу.^[11]

2.7 Наївний байєсівський класифікатор мережі

Наївна байєсівська мережа (НБ) - це проста, але потужна байєсівська мережа. Навіть із сильним припущенням незалежності серед даних, вона демонструє конкурентоспроможність у порівнянні з іншими найсучаснішими класифікаторами, такими як векторні машини підтримки (support vector machines). Він широко використовується в таких областях, як класифікація тексту і фільтрації спаму. НБ передбачає умовну незалежність серед змінних чи атрибутів. При використанні у класифікації НБ прогнозує нову точку даних як клас з найбільшою апіорною ймовірністю. Цей підхід моделює спільну ймовірність у кожній підмножині окремо, а потім застосовує правило Байєса для побудови правила апіорної класифікації. Цей фреймворк здається неповним, оскільки ця процедура побудови фактично відкидає важливу дискримінаційну інформацію для класифікації.

Не враховуючи інших класів даних, цей метод намагається лише приблизити інформацію у кожній підмножині. З іншого боку, дискримінаційні класифікатори дуже добре зберігають цю інформацію, будуючи правила прийняття рішень серед усіх даних. Для наївного байєсівського класифікатора недостатньо просто наблизити дані у кожній підмножині окремо. Він повинен забезпечити глобальну схему збереження дискримінаційної інформації серед усіх даних. Одне з вирішень цієї проблеми – безпосередньо навчити модель апіорної ймовірності, а не спільно ймовірнісну модель.^[12]

Наївного класифікатора дуже легкий класифікатор у реалізації, не потребує великої вибірки для навчання, дає високі результати класифікації, але через припущення взаємозалежності можлива втрата точності.

РОЗДІЛ 3: Байєсівські мережі в мовах програмування

Як можна було зрозуміти з того, що писалося вище, байєсівські мережі невід’ємна частина машинного навчання, а тому дуже часто використовуються у мовах програмування.

3.1 Байєсівські мережі в мові програмування R

R - мова та середовище для статистичних обчислень та графіків. Це проект GNU, подібний до мови та середовища S, який був розроблений в лабораторіях Bell Laboratories. R може розглядатися як інша імплементація мови S. R забезпечує широкий спектр статистичних та графічних утилітів (лінійне та нелінійне моделювання, класичні статистичні тести, аналіз часових рядів, класифікація, кластеризація ...).

Bnlearn являє собою пакет мови програмування R для вивчення графічної структури байєсівських мереж, оцінки їх параметрів та виконання корисних виводів. Вперше пакет був випущений в 2007 році, він постійно розвивався протягом більше 10 років і досі постійно оновлюється. ^[13]

bnlearn реалізує такі алгоритми навчання структури на основі обмежень:

- ПК
- Зростати-усаджувати (Grow-Shrink)
- Напівпроменені Hiton-PC (SI-HITON-PC)
- Поступова асоціація Марківська ковдра (Incremental Association Markov Blanket)
- Інтерментована інкрементальна асоціація (Interleaved Incremental Association, Inter-IAMB)
- Гібридні батьки та діти (Hybrid Parents & Children)
- Швидка інкрементальна асоціація (Fast-IAMB)
- Поступова асоціація з корекцією FDR (IAMB-FDR)
- Max-Min батьки та діти (MMPC)

такі алгоритми навчання структури на основі балів:

- Hill Climbing (HC)
- Пошук Tabu (Tabu)

наступні алгоритми навчання гібридної структури:

- Max-Min Hill Climbing (MMHC)
- Гібридна HPC (H2PC)
- Загальна двофазна обмежена максимізація (RSMAX2)

наступні локальні алгоритми виявлення:

- Чоу-Лю;
- ARACNE;

та наступні байєсівські класифікатори мережі:

- Наївний Байєс;
- Деревопоширений наївний Байєс (Tree-Augmented naïve Bayes).

3.2 Байєсівські мережі в мові програмування Python

Python- не складна мова програмування, яка дуже подобається багатьом початковим програмістам і це не дивно. Серед роботодавців Python користується попитом та документація цієї мови написана детально та зрозуміло. Стандартні бібліотеки та модули, які контрибує спільнота відкривають безмежні можливості у застосуванні.

Scikit-learn - найпоширеніший вибір для вирішення завдань класичного машинного навчання. Вона надає широкий вибір алгоритмів навчання з учителем і без вчителя. Навчання з учителем передбачає наявність розміченого датасета, в якому відомо значення цільового показника. У той час як навчання без вчителів не передбачає наявності розмітки в датасеті - потрібно навчитися отримувати корисну інформацію з довільних даних. Одна з основних переваг бібліотеки полягає в тому, що вона працює на основі декількох поширених математичних

бібліотек, і легко інтегрує їх один з одним. Ще однією перевагою є широка спільнота і докладна документація. Scikit-learn широко використовується для промислових систем, в яких застосовуються алгоритми класичного машинного навчання, для досліджень, а так само для новачків, які тільки робить перші кроки в області машинного навчання.

Бібліотека реалізує наступні основні методи:

Метричні: моделі, які обчислюють відстань по одній з метрик між об'єктами вибірки, і приймають рішення в залежності від цього відстані (K найближчих сусідів).

Ансамблеві методи: методи, засновані на деревах рішень, які комбінують міць безлічі дерев, і таким чином підвищують їх якість роботи, а також дозволяють проводити відбір ознак (бустінг, беггінг, випадковий ліс, мажоритарну голосування).

Лінійні: моделі, завдання яких побудувати розділяє (для класифікації) або апроксимує (для регресії) .

Дерева рішень: навчання моделей, що базуються на безлічі умов, оптимально обраних для вирішення завдання.

Нейронні мережі: комплексний нелінійний метод для задач регресії і класифікації.

SVM: нелінійний метод, який навчається визначати межі прийняття рішень.

Наївний Байєс: пряме ймовірносне моделювання для задач класифікації.

K-середніх: найпоширеніший метод для кластеризації, требують на вхід число кластерів, за якими повинні бути розподілені дані.

Крос-валідація: метод, при якому для навчання використовується весь датасет (на відміну від розбиття на вибірки train / test), проте навчання відбувається багаторазово, і в якості валідаційної вибірки на кожному кроці виступають різні

частини датасета. Підсумковий результат уявляють собою усереднення отриманих результатів.

Це - лише базовий список. Крім цього, Scikit-learn містить функції для розрахунку значень метрик, вибору моделей, препроцесінга даних та інші. ^[14]

Pomegranate - це пакет Python, який реалізує швидкі та гнучкі ймовірнісні моделі, починаючи від індивідуального розподілу ймовірностей і закінчуючи композиційними моделями, такими як байєсівські мережі та приховані моделі Маркова. Основна філософія пакету pomegranate полягає в тому, що всі ймовірнісні моделі можна розглядати як розподіл ймовірностей, оскільки всі вони оцінюють ймовірність урожайності для зразків і можуть бути оновлені за даними зразками та відповідними вагами. Основним наслідком цього погляду є те, що компоненти, які реалізовані в pomegranate, можна складати більш гнучко, ніж в інших пакетах. Наприклад, можна побудувати Гауссєву сумішеву модель так само просто, як побудувати експоненціальну модель або звичайну модель суміші. Можна створити класифікатор Байєса, який використовує різні типи розподілів для кожної функції, моделюючи пов'язані з часом функції, використовуючи експоненціальний розподіл, і рахує за допомогою розподілу Пуассона. Композиційні моделі можуть розглядатися як розподіли ймовірностей, можна створити суміш байєсівських мереж або прихований класифікатор Маркова моделі Байєса, який робить передбачення щодо послідовностей.

Окрім різноманітності розподілу ймовірностей та моделей, pomegranate має різноманітні вбудовані функції, які реалізовані для всіх моделей. До них відносяться різні стратегії навчання, такі як напіваавтоматичне навчання, навчання з відсутніми значеннями та mini-batch навчання. ^[15]

Інший пакет, який допомагає застосовувати байєсівські мережі у Python є BayesPy. Перша версія цього пакету була створена у 2013 році. BayesPy надає

інструменти для байєсівського висновку. Користувач може побудувати байєсівську мережу, спостерігати за даними та виконувати апріорні виводи. Метою створення ефективного, гнучкого та зручного інструменту є використання його як для експертів так і для менш продвинутих користувачів.

На даний момент реалізовано лише варіаційний байєсівський вивід для спільно-експоненціальної родини (conjugate-exponential family, передача варіаційного повідомлення). Майбутня робота включає варіативні наближення для інших типів розподілів та, можливо, інших приблизних методів виведення, таких як поширення очікування, наближення Лапласа, ланцюг Маркова Монте-Карло (MCMC) та інші методи.

Тобто даний пакет не такий розвинений як Romenate, а більш звужено направлений. Але він лише у процесі розробки, остання версія була оновлена у грудні 2019 року (puri package 0.5.19).^[16]

PyMC3 - пакет Python для байєсівського статистичного моделювання та ймовірнісного машинного навчання, який фокусується на просунутому ланцюзі Маркова Монте-Карло та варіативних алгоритмах підгонки, побудований на Theano. Theano - це пакет, який дозволяє нам визначати функції, що включають операції масиву та лінійну алгебру. На відміну від PyMC2, який використовував розширення Fortran для виконання обчислень, PyMC3 покладається на Theano для автоматичної диференціації, а також для оптимізації обчислень та динамічної компіляції C. PyMC3 використовувався для вирішення задач вивду в кількох наукових сферах, включаючи астрономію молекулярну біологію, кристалографію, хімію, екологію та психологію. Попередні версії PyMC також широко використовувалися, наприклад, в галузі кліматичних наук, охорони здоров'я, нейронауки та паразитології^[17]

BayesiaLab

Побудований на фундаменті байєсівського мережевого формалізму, BayesiaLab - це потужний десктопний додаток (Windows, macOS, Linux / Unix) із дуже наповненим графічним інтерфейсом для потреб користувача. Він надає вченим

комплексне "лабораторне" середовище для машинного навчання, моделювання знань, діагностики, аналізу, моделювання та оптимізації. Завдяки BayesiaLab стало можливим, щоб прикладні дослідники в багатьох галузях, а не просто комп'ютерні фахівці, скористалися перевагами формалізму байесівської мережі.

Bayesware Discoverer

Bayesware Discoverer - програмне забезпечення розроблене англійською компанією Bayesware. Серед його особливостей слід відзначити можливість автоматичної побудови мереж Байєса за навчальними даними, порівняння декількох моделей, побудованих на основі мереж Байєса, шляхом моделювання, використання алгоритму Bound and Collapse у випадку неповних даних, заповнення таблиці умовних ймовірностей та побудови ймовірнісного висновку, роботи з категоріальними даними та неперервними змінними. ^[18]

WinBUGS

WinBUGS - це програмне забезпечення для байесівських аналізів, яке використовує Марківський ланцюг Монте-Карло (MCMC), щоб відповідати статистичним моделям. WinBUGS підходить для моделей з фіксованим ефектом та багаторівневим використанням байєсового підходу. WinBUGS був випущений під Майкрософт у далекому 1989 році і остання його версія була випущена у 2007 році. На разі це одне з найстаріших програмних забезпечень для байєсівської статистики і має обмежені можливості.

РОЗДІЛ 4: Демонстрування байєсівських мережі використовуючи пакет romegranate

4.1 Аналіз задачі

Уявимо ситуацію на ігровому шоу: є троє дверей, за однією з них знаходиться приз(припустимо автомобіль), а за іншими козли. За якою із дверей знаходиться приз знає тільки ведучий(Монті Холл). Ви обираєте будь-яку із дверей і тоді ведучий завжди відкриває іншу, де є козел. Тоді ведучий вам пропонує змінити свій вибір. Що ви будете робити? Виберете інші двері, чи залишите ваш перший варіант?

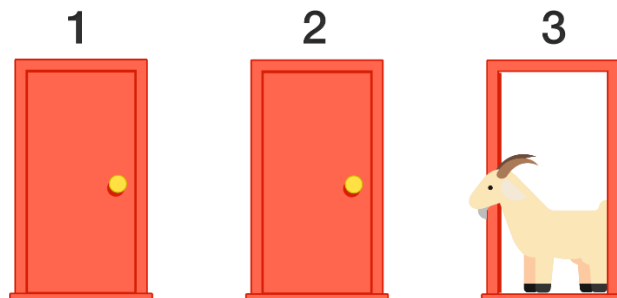


Рисунок 4.1 – представлення задачі

4.2 Мережа, що описує задачу

Розглянемо просту мережу, що описує парадокс Монті Холла.

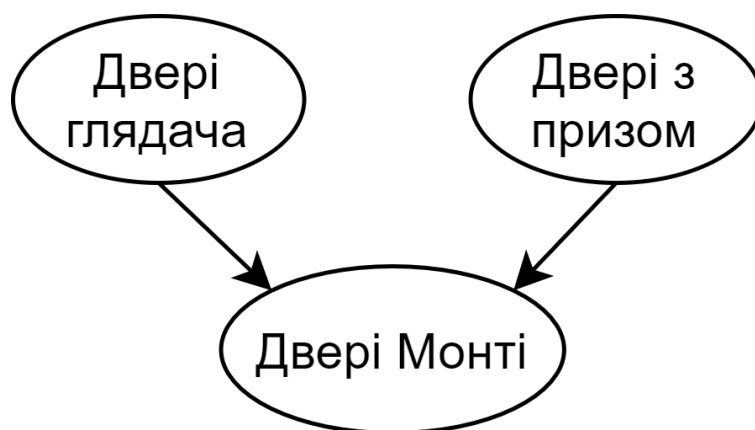


Рисунок 4.2 - направлений ациклічний граф, який описує задачу Монті Холла.

Уявимо, що гість обрав перші двері(A), потім Монті відкрив другі двері(B), де знаходиться козел. Тобто, порахуємо ймовірність, що машина знаходиться за першими дверима(A) враховуючи, що Монті відкрив другі двері(B).

$$P(\text{CarA}|B) = P(B|\text{CarA}) * P(\text{CarA}) / P(B)$$

Спочатку порахуємо P(B) тобто ймовірність, що Монті відкриє другі двері, якщо ми обрали перші.

$$P(B) = P(B|\text{CarA}) * P(\text{CarA}) + P(B|\text{CarB}) * P(\text{CarB}) + P(B|\text{CarC}) * P(\text{CarC}).$$

P(B|CarA): ½, Адже Монті може обрати будь-які двері(A чи B), якщо машина знаходиться за першими

P(B|CarB): 0

P(B|CarC): 1, адже Монті повинен обрати лише ті двері, де немає призу, і вони не обрані гостем.

$$P(\text{CarA}|B) = \frac{\frac{1}{2} * \frac{1}{3}}{\frac{1}{2} * \frac{1}{3} + 0 * \frac{1}{3} + 1 * \frac{1}{3}} = \frac{1}{3}$$

За такою же логікою порахувавши P (Car C | B), ми отримаємо ймовірність 2/3. Спробуємо вирішити дану задачу за допомогою мови Python.

4.3 Опис коду програми

Розберемося із залежностями. Двері, які обирає гість, і двері, що містять машину, є абсолютно випадковими процесами. Однак двері, які Монті вирішить відчинити, залежать від обох дверей; двері, які вибирає гість, і двері, де приз позаду. Монті повинен вибирати таким чином, щоб у дверей не було призу, і вони не можуть бути обраним гостем.

В коді нижче “First”, “Second”, “Third”, відповідають за двері, які є на вибір перед гостем.

```

import math
from pomegranate import *

#Гість обирає двері з однаковою ймовірністю
guest =DiscreteDistribution( { 'First': 1./3, 'Second': 1./3, 'Third': 1./3 } )

#Приз може знаходитися за будь-якою з дверей
prize =DiscreteDistribution( { 'First': 1./3, 'Second': 1./3, 'Third': 1./3 } )

```

Нижче можна побачити, як складається умовна ймовірність для кожного з вузлів, враховуючи всі можливі сценарії оскільки двері, вибрані Монті, залежать від інших двох дверей.

```

#Вибір Монті залежить від вибору гостя та від дверей за якою приз
monty =ConditionalProbabilityTable(
[[ 'First', 'First', 'First', 0.0 ],
[ 'First', 'First', 'Second', 0.5 ],
[ 'First', 'First', 'Third', 0.5 ],
[ 'First', 'Second', 'First', 0.0 ],
[ 'First', 'Second', 'Second', 0.0 ],
[ 'First', 'Second', 'Third', 1.0 ],
[ 'First', 'Third', 'First', 0.0 ],
[ 'First', 'Third', 'Second', 1.0 ],
[ 'First', 'Third', 'Third', 0.0 ],
[ 'Second', 'First', 'First', 0.0 ],
[ 'Second', 'First', 'Second', 0.0 ],
[ 'Second', 'First', 'Third', 1.0 ],
[ 'Second', 'Second', 'First', 0.5 ],
[ 'Second', 'Second', 'Second', 0.0 ],
[ 'Second', 'Second', 'Third', 0.5 ],
[ 'Second', 'Third', 'First', 1.0 ],
[ 'Second', 'Third', 'Second', 0.0 ],
[ 'Second', 'Third', 'Third', 0.0 ],
[ 'Third', 'First', 'First', 0.0 ],
[ 'Third', 'First', 'Second', 1.0 ],
[ 'Third', 'First', 'Third', 0.0 ],
[ 'Third', 'Second', 'First', 1.0 ],
[ 'Third', 'Second', 'Second', 0.0 ],
[ 'Third', 'Second', 'Third', 0.0 ],
[ 'Third', 'Third', 'First', 0.5 ],
[ 'Third', 'Third', 'Second', 0.5 ],
[ 'Third', 'Third', 'Third', 0.0 ]], [guest, prize] )

```

На наступному етапі створюємо нашу мережу

```

#Визначаємо стани
s1 = State( guest, name="guest" )
s2 = State( prize, name="prize" )
s3 = State( monty, name="monty" )

#Створюємо байєсівську мережу, передаючи стани в нашу модель
network = BayesianNetwork( "Solving the Monty Hall Problem" )
network.add_states(s1, s2, s3)
network.add_edge(s1, s3)
network.add_edge(s2, s3)
network.bake()

```

4.4 Результат виконання програми

```

guest    First
prize    {
  "class" : "Distribution",
  "dtype" : "str",
  "name" : "DiscreteDistribution",
  "parameters" : [
    {
      "First" : 0.3333333333333334,
      "Second" : 0.0,
      "Third" : 0.6666666666666664
    }
  ],
  "frozen" : false
}
monty    Second

```

Варто звернути увагу на вивід програми, ймовірність того, що приз буде за третіми дверима, становить приблизно 66%. Це доводить, що якщо гість змінить свій вибір дверей, у нього буде більша ймовірність виграшу.

Висновки

Високі темпи розвитку сучасної науки в цілому, потребують зручних методів опрацювання інформації, що дозволять заощадити час та інші ресурси. Баєсівські мережі сприяють такому розвитку та полегшують розв'язання багатьох задач. Дана робота аналізує різноманітні методи на засоби навчання байєсівських мереж, та їх застосунок у житті, що дозволяє наглядно побачити плюси та недоліки різних методів. По закінченню роботи було виконано усі поставлені вище завдання.

Основні результати роботи це аналіз різних пакетів, бібліотек та програмних забезпечень, що полегшують роботу із баєсівськими мережами.

Також було розроблено невелику програму на мові програмування Python, яка розв'язала відому задачу Монті Холла за допомогою використання пакету `romergranate`. Був використаний саме цей пакет, адже дана задала не потребує великих ресурсів, а цей пакет містить все, що необхідно для розв'язання. Python - мова загального призначення з читаним синтаксисом на відміну від R, яка побудована статистиками і охоплює їх конкретну мову. Тому для вирішення задач у повсякденному житті доцільніше використовувати Python. До того ж він забезпечений усіма можливими пакетами для полегшення роботи користувача.

Список використаної літератури

- [1] Эвристический метод построения байесовских сетей [Электронный ресурс] 2006 Автор А.Н. Трентьев, Р.И. Бидюк http://www.immsp.kiev.ua/publications/articles/2006/2006_3/Terentyev_03_2006.pdf
- [2] Теорія ймовірностей і математична статистика Автор Жлуктенко В. І., Наконечний С. І. [Електронний ресурс] https://www.studmed.ru/view/zhluktenko-v-nakonechniy-s-teorya-ymovrnostey-matematichna-statistika-u-2-ch-ch-teorya-ymovrnostey_0fb78eec1a0.html
- [3] Joint Probability Definition [Електронний ресурс] 13.11.19 Автор Will Kenton <https://www.investopedia.com/terms/j/jointprobability.asp>
- [4] Introduction to Probability Автор Joseph K. Blitzstein, Jessica Hwang
- [5] How To Implement Bayesian Networks In Python? – Bayesian Networks Explained With Examples [Електронний ресурс] 12.07.19 Автор Zulaikha Lateef <https://www.edureka.co/blog/bayesian-networks/>
- [6] How to define a model [Електронний ресурс] Автор Национальный исследовательский университет "Высшая школа экономики" <https://www.coursera.org/learn/bayesian-methods-in-machine-learning/lecture/9zkEL/how-to-define-a-model>
- [7] Оцінювання узагальнених лінійних моделей за байєсівським підходом в актуарному моделюванні імовірнісним висновком Автор П.І. Бідюк, С.В. Трухан
- [8] Comparison of frequentist and Bayesian inference [Електронний ресурс] 2014 Автор Jeremy Orloff and Jonathan Bloom https://ocw.mit.edu/courses/mathematics/18-05-introduction-to-probability-and-statistics-spring-2014/readings/MIT18_05S14_Reading20.pdf
- [9] Построение и методы обучения байесовских сетей [Електронний ресурс] 2004 Автор П.И. Бидюк, А.Н. Терентьев https://ela.kpi.ua/bitstream/123456789/19609/1/2004_2139_Terentiev_Article_1.pdf
- [10] Построение и методы обучения байесовских сетей [Електронний ресурс] 2004 Автор П.И. Бидюк, А.Н. Терентьев https://ela.kpi.ua/bitstream/123456789/19609/1/2004_2139_Terentiev_Article_1.pdf

- [11] A Tutorial on Learning With Bayesian Networks [Електронний ресурс] 1996
Автор David Heckerman
<https://www.cis.upenn.edu/~mkearns/papers/barbados/heckerman.pdf>
- [12] Bayesian Network Technologies: Applications and Graphical Models [Електронний ресурс] 1996 Автор Ankush Mittal, Ashraf Kassim <https://www.igi-global.com/book/bayesian-network-technologies/98>
- [13] R package for Bayesian network learning and inference [Електронний ресурс] 2020 Автор bnlearn.com <https://www.bnlearn.com/>
- [14] Введение в Scikit-learn [Електронний ресурс] 16.06.2019 Автор neurohive.io
https://neurohive.io/ru/osnovy-data-science/vvedenie-v-scikit-learn/#pll_switcher
- [15] Pomegranate package [Електронний ресурс] Автор pomegranate.io
<https://pomegranate.readthedocs.io/en/latest/>
- [16] BayesPy package [Електронний ресурс] Автор pypi.org
<https://pypi.org/project/bayespy/>
- [17] PyMC3 [Електронний ресурс] 05.06.2020 Автор Wikipedia
<https://en.wikipedia.org/wiki/PyMC3>
- [18] Інструменти побудови динамічних моделей інформаційних систем із застосуванням байєсівських мереж [Електронний ресурс] 21.03.2017 Автор Тищенко Дмитро <https://iconfs.net/w.infocom2016/%D0%86nstrumenty-pobudovy-dynamichnykh-modelej-informatsijnykh-system-iz-zastosuvannyam-baj%D1%94sivskykh-merezh>