Міністерство освіти і науки України
Тернопільський національний економічний університет
Факультет комп'ютерних інформаційних технологій
Кафедра комп'ютерної інженерії

Паздрій Леонід Ігорович.

Алгоритми евристичного методу побудови байєсівських мереж за навчальними даними / Euristic algorithms of Bayesian sets building by learning data

Спеціальність 8.091501 – Комп'ютерні системи та мережі

Дипломна робота за освітньо-кваліфікаційним рівнем «магістр»

Науковий керівник
д.т.н., професор Теслюк В.М.
_____

Дипломну роботу допущено до захисту
«___»_____ 20 __ р.
Зав. кафедри КІ
Березький О.М. _____

Тернопіль – 2017

RESUME

Diploma work on a theme "Algorithm of heuristic method of construction of Bayesian networks from educational data" on the Master Diploma Defence from specialist „Computer systems and networks" written on the 94 pages and contains 53 illustrations, 6 tables, 3 applications and 50 sources for references.

The aim of work is development of algorithms of construction of Bayesian network (BN) and forming of probabilistic conclusion structure in Bayesian networks, and also creation, on their basis, systems of support of making decision.

Research methods are base on theory of chances, mathematical statistics, theory of the graphs, theory of forming of probabilistic conclusion and other methods of intellectual analysis of data.

It is conducted analysis of current status of development of intellectual analysis of data and reasonable efficiency of application of vehicle of Bayesian networks. It is shown that the basic lack of construction of topology of Bayesian networks is exponential complication. For the construction of structure of BN the algorithm of determination of measure of connection between tops, the values of mutual information are used in that, is offered, and as a merit function of model structure is description minimum length. The algorithm of construction of exact probabilistic conclusion is worked out in BN from educational data. For the calculation of values of probabilities of the states of tops instead of tables of conditional probabilities there is the used matrix of empiric values of general probability of all network distribution. For an evaluation to quality of the built structure new modification of function of Cooper-Herskovycha is got. The programmatic realized system of support of making decision is worked out for the intellectual analysis of data on the basis of BN. Theoretical and practical job performances are useful to application in an educational process.

Keywords: ALGORITHM, BAYESIAN NETWORK, INTELLECTUAL ANALYSIS.

ЗМІСТ

ВСТУП

Актуальність теми. Технологічний розвиток комп’ютерної техніки останнього півсторіччя характеризується широким використанням засобів обробки цифрової інформації, в різних сферах людської діяльності. При цьому з'явився побічний ефект цієї активності – це надвеликі бази даних зібраної інформації. Існує багато фірм, компаній та організацій, що роками накопичують інформацію, сподіваючись на те, що вона допоможе їм у прийнятті рішень.

Для опису різних концепцій і методів, які поліпшують бізнесові рішення шляхом використання систем підтримки прийняття рішень (СППР), у 1980-х роках запропоновано термін “Business Intelligence” (BI), діловий інтелект, бізнес-аналітика або бізнес-інтелект. Це складається з алгоритмічно-програмних продуктів наступних класів:

- засоби побудови сховищ даних;
- системи оперативної аналітичної обробки;
- інформаційно-аналітичні системи;
- засоби інтелектуального аналізу даних (ІАД);
- інструментарій для формулювання запитів і створення звітів.

Технологія ІАД (Data Mining) з’явилася для розв’язування задач виявлення прихованих взаємозв'язків всередині великих баз даних. Більшість інструментів інтелектуального аналізу даних (ІАД) ґрунтується на двох технологіях: машинне навчання (machine learning) і візуалізація (візуальне подання інформації). Байєсівські мережі (БМ) якраз і поєднують у собі ці дві технології. Це досить молодий напрям розвитку науки, що з’явився на стику теорії ймовірностей та теорії графів (розділ дискретної математики). Ідея впровадження БМ полягає у представленні причинно-наслідкових зв’язків процесу у вигляді графа.

Аналіз існуючих методів ІАД показав, що байєсівські мережі, у порівнянні з популярними моделями “чорних скриньок”, надають більш

зрозуміле тлумачення своїх висновків, припускають логічну інтерпретацію і модифікацію структури відношень між змінними задачі. БМ також дозволяють в явній формі враховувати попередній апріорний досвід експертів. Завдяки вдалому представленню у вигляді графів БМ досить зручні для розв'язання прикладних задач користувачів. Вони ґрунтуються на фундаментальних положеннях і результатах теорії ймовірностей, які розроблялися на протязі декількох сотень років, що забезпечує їм успіх в розв'язанні практичних задач.

Актуальність застосування БМ як інструменту ІАД визначило тему і напрям дипломної роботи.

Мета роботи. Метою роботи є підвищення швидкості та якості інтелектуального аналізу даних шляхом розробки алгоритму формування ймовірнісного висновку і створення на його основі оригінальної системи підтримки прийняття рішень.

Досягнення поставленої мети вимагає постановки і розв'язання таких основних завдань:

- порівняльний аналіз існуючих алгоритмів побудови топології та ймовірнісного висновку в байєсівській мережі.

- розробка алгоритму побудови топології БМ за навчальними даними із застосуванням функції опису мінімальної довжини (ОМД).

- розробка оригінальної архітектури та реалізація комп'ютерної системи підтримки прийняття рішень для інтелектуального аналізу даних на основі мереж Байєса.

Об'єктом дослідження є експериментальні та статистичні дані щодо протікання процесів, які потребують ефективної аналітичної обробки з метою виявлення невідомих, нетривіальних взаємозв'язків між атрибутами, необхідних для прийняття рішень.

Предмет дослідження – ймовірнісні мережі Байєса, методи формування топології та ймовірнісного висновку у байєсівських мережах.

Методи дослідження ґрунтуються на теорії ймовірностей, математичній статистиці, теорії графів, теорії формування ймовірнісного висновку та інших

методах інтелектуального аналізу даних.

Наукова новизна одержаних результатів визначається такими теоретичними і практичними результатами:

1. Розроблено новий алгоритм ітераційно евристичного методу побудови БМ, який базується на використанні оцінки взаємної інформації між вершинами та функції опису мінімальної довжиною. Це дає можливість збільшити розмір БМ та значно прискорити швидкодію обчислень. Він відрізняється тим, що на першому етапі обчислюється значення взаємної інформації між усіма вершинами, а на другому виконується цілеспрямований пошук з використанням в якості оціночної функції мережевих структур функцію ОМД.

2. За оціночну функцію для евристичної побудови БМ запропонована модифікація функції Купера-Герсковича. Її перевага у порівнянні з функцією Купера-Герсковича полягає у переході від обчислення факторіалів до сум, що суттєво прискорює процес обчислення та знімає обмеження на розмір вибірок навчальних даних.

3. Запропоновано алгоритм побудови точного ймовірнісного висновку в БМ за навчальними даними, швидкодія якого, на відміну від інших точних методів, залежить не від кількості дуг мережі, а від розміру навчальної вибірки. Як наслідок, в два рази прискорюється процес побудови ймовірнісного висновку з одночасним збереженням високої точності обчислень. Найбільшою перевагою запропонованого алгоритму є його простота використання (прикладного програмування) у порівнянні з більшістю існуючих.

4. На основі запропонованих алгоритмів розроблена та програмно реалізована оригінальна система підтримки прийняття рішень для інтелектуального аналізу даних на основі БМ.

Практичне значення одержаних результатів. Розроблена та реалізована оригінальна архітектура системи підтримки прийняття рішень для ІАД на основі БМ, яка може використовуватись для розв’язання багатьох практичних задач.

Теоретичні і практичні результати роботи корисні для застосування у

навчальному процесі. Зокрема, вони можуть бути використані в курсах “Експертні системи”, “Системний аналіз”, “Теорія прийняття рішень”, “Проектування систем підтримки прийняття рішень” та “Проектування комп’ютерних інформаційних систем”.

Результати роботи впроваджені в навчальний процес і використовуються в курсі “Проектування комп’ютерних систем та мереж”.

Публікації та апробація ДР. За результатами проведених наукових досліджень підготовлено тези доповіді «Інтелектуальний аналіз даних на основі баєсівських мереж» на Всеукраїнській школі-семінарі «Сучасні комп’ютерні інформаційні технології» (ACIT-2016) [9].

1 БАЙЄСІВСЬКІ МЕРЕЖІ В ТЕХНОЛОГІЇ ІНТЕЛЕКТУАЛЬНОГО АНАЛІЗУ ДАНИХ

1.1 Технологія інтелектуального аналізу даних

За останні роки широкого застосування набули засоби обробки цифрової інформації. Це дає можливість підвищити ефективність створення баз даних (БД) в різних сферах діяльності людини. Але результатом такої активної діяльності стали надзвичайно великі масиви зібраних даних. В результаті, з'явилась думка, що ці масиви містять великий об'єм корисної інформації.

Агентство Gartner Group, основною діяльністю якого є аналіз ринків інформаційних технологій, ввело термін "Business Intelligence" (BI), діловий інтелект, бізнес-аналітика або бізнес-інтелект. Дане визначення використовується для опису різноманітних концепцій та методів для покращення бізнес-рішення за допомогою систем підтримки прийняття рішень (СППР).

BI – це програмні засоби, які використовуються для доступу та аналізу інформації, що знаходиться в сховищі даних. "Business Intelligence" також сприяють прийняттю вірних і обґрунтованих управлінських рішень. Цей термін поєднує різні засоби та технології аналізу та обробки інформації підприємства. На основі даних засобів синтезуються BI-системи, мета яких – поліпшення якості інформації, що сприяє прийняттю управлінських рішень.

BI – системи також відомі під назвами СППР або Decision Support System (DSS). Такі системи, дані перетворюють в інформацію, на основі якої можна приймати рішення.

Gartner Group визначає специфіку ринку систем Business Intelligence як набір програм наступних класів:

1 – основи створення сховищ даних (data warehousing, СД);

2 – технології оперативної аналітичної обробки даних (OLAP – online analytical processing);

3 – інформаційно-аналітичні системи (enterprise information systems);

4 – ІАД засоби;

5 – інструменти для виконання запитів і побудови звітів.

ІАД засоби в BI-системах займають окрему важливу позицію та відіграють роль однієї з п’яти опор.

1.1.1 Методи, задачі, та алгоритми інтелектуального аналізу даних

Методами та алгоритмами ІАД являються: байєсівські мережі, лінійна регресія, метод опорних векторів, методи найближчого сусіда і k – найближчого сусіда, штучні нейронні мережі, кореляційно-регресійний аналіз, еволюційне програмування і генетичні алгоритми, ієрархічні методи кластерного аналізу, дерева рішень, методи пошуку асоціативних правил, у тому числі алгоритм Apriori, символьні правила, метод обмеженого перебору, неієрархічні методи кластерного аналізу, включаючи алгоритми k – середніх і k – медіани, різні методи візуалізації даних та інші методи.

Метод представляє собою спосіб або правило, певний шлях прийому рішень теоретичного, практичного, пізнавального, управлінського характеру.

Алгоритм – точна послідовність дій (кроків), які перетворюють вхідні дані в потрібний результат.

На конференції IEEE-ICDM (International Conference on Data Mining), яка проходила у Гонконгу в грудні 2006 року, були визначені 10 найкращих алгоритмів ІАД. Такі алгоритми визнані кращими та відзначені преміями і нагородами: (1) ACMKDD Innovation Award, (2) IEEE ICDM Research Contributions Award та (3) Google Scholar. В таблиці 1.1 наведено перелік, а в роботі [1] представлено розширений опис цих алгоритмів.

Таблиця 1.1 – Алгоритми для ІАД

№	Назва алгоритму
1	2
1	C4.5, відноситься до дерев рішень
2	k – середніх, відноситься до неієрархічного кластерного аналізу

Продовження таблиці 1.1

1	2
3	SVM або метод опорних векторів
4	Apriori, відноситься до методів пошуку асоціативних правил
5	EM (expectation maximization) або максимізації сподівання, відноситься до методів кластерного аналізу
6	PageRank, відноситься до пошукових Web-алгоритмів
7	AdaBoost або посилення слабких класифікаторів
8	kNN (k – nearest neighbor classification) або k – найближчого сусіда, відноситься до методів класифікації
9	Naive Bayes, відноситься до байєсових мереж
10	CART (Classification and Regression Trees), відноситься до дерев рішень

До задач ІАД можна віднести наступне.

1. Класифікація. Являється найпростішим і дуже поширеним завданням ІАД. Метою класифікації є визначення ознак, які характеризують групи об'єктів досліджуваного набору даних – класи. На основі таких ознак даний об'єкт можна віднести до того чи іншого класу. За допомогою методів найближчого сусіда; k – найближчого сусіда; байєсівських мереж; дерева рішень; нейронних мереж розв’язуються задачі класифікації.

2. Кластеризація. Логічне продовження класифікації являється кластеризація. Особливість даного завдання полягає в тому, що на початку класи об'єктів не визначені. Поділ об'єктів на групи являється результатом кластеризації. Для розв’язання задачі використовують карти Кохонена та байєсівські мережі.

3. Асоціація. Виконує пошук закономірностей між взаємопов'язаними подіями в наборі даних. На відміну від двох попередніх завдань ІАД, пошук закономірностей здійснюється не на основі властивостей аналізованого об'єкта, а між декількома подіями, які відбуваються одночасно. Найбільш відомим алгоритмом розв’язання задачі пошуку асоціативних правил – алгоритм Apriori.

4. Послідовна асоціація або просто послідовність. Послідовність дозволяє знайти тимчасові закономірності між транзакціями. Завдання послідовності подібне асоціації, але її метою є встановлення закономірностей не між одночасно наступаючими подіями, а між подіями, зв'язаними в часі (тобто в часі, що відбуваються з деяким певним інтервалом). Інакше кажучи, послідовність визначається високою ймовірністю ланцюжка зв'язаних у часі подій Наприклад, після придбання житла мешканці в 60% випадків протягом двох тижнів купляють холодильник, а впродовж двох місяців 50% з них купляють телевізор.

5. Прогнозування. В результаті розв’язання задачі прогнозування на основі особливостей історичних даних оцінюють пропущені або майбутні значення цільових чисельних показників. Для розв’язання таких задач широко використовують способи математичної статистики та нейронні мережі.

6. Визначення відхилень або викидів. Мета розв’язання цієї задачі – виявлення та аналіз даних, які щонайбільше відрізняються від всієї множини даних, виявлення так званих нехарактерних шаблонів.

7. Оцінювання. Дана задача виконує прогнозування постійних значень ознак.

8. Аналіз зв'язків. Полягає в знаходженні залежностей в наборах даних.

9. Візуалізація. Результатом візуалізації являється створення графічного образу даних, які аналізуються. Для розв’язання цієї задачі використовують графічні методи, що показують наявність закономірностей у даних.

10. Підведення підсумків. Мета такої задачі – опис конкретних груп об'єктів з аналізованого набору даних.

1.1.2 Проблеми використання технології ІАД

Для максимального використання можливості масштабованих інструментів ІАД потрібно вибрати, очистити та перетворити дані, деколи інтегрувати інформацію, отриману із зовнішніх джерел, а також встановити спеціальне середовище для роботи ІАД алгоритмів. Результати ІАД в значній

мірі залежать від рівня підготовки даних, а не від можливостей будь-якого алгоритму чи набору алгоритмів. Близько 75% роботи з технологією ІАД виконується на збір даних.

Відомий у світі знавець в області СД, CRM та ІАД, керівник консалтингової компанії Two Crows Херб Эдельштайн (Herb Edelstein) висловлює думку, що ІАД знаходиться на ранній стадії розвитку, а процес використання ІАД на практиці показує себе складнішим, ніж очікується. Чимало організацій зацікавлені у використанні цієї технології, але тільки деякі динамічно впроваджують дані проекти. ІТ- команди захопилися домислом, що засоби ІАД легкі у використанні. Вважається, що варта лише запустити такий інструмент на терабайтній базі даних і зразу появиться корисна інформація. Насправді, успішний ІАД-проект вимагає розуміння суті діяльності, знання даних та інструментів, а також процесу аналізу даних.

Ще до початку використанням технологій ІАД необхідно детально проаналізувати її проблеми, а також пов'язані з нею обмеження та критичні питання. Також потрібно зрозуміти, чого саме дана технологія не може дати. ІАД не в змозі замістити аналітика. Технологія не може дати відповіді на ті питання, які не були задані. Вона не може замінити аналітика, а всього лише надає для нього сильний інструмент для полегшення і покращення його роботи.

Технологія ІАД мультидисциплінарна область, для розробки програмного забезпечення. Необхідно залучати фахівців з різних областей, а також забезпечити їхню якісну взаємодію. Отримати корисну інформацію можливо тільки тоді, коли є розуміння суті даних. Використання ІАД повинне бути постійно пов'язане з підвищенням рівня кваліфікації користувачів. Але спеціалістів в області ІАД, які б добре розбиралися в бізнесі вкрай мало.

1.2 Байєсівська мережа – інструмент інтелектуального аналізу даних

1.2.1 Виникнення байєсівських мереж

БМ з’явилися на стику двох наук теорії ймовірностей та теорії графів

(рисунок 1.1).

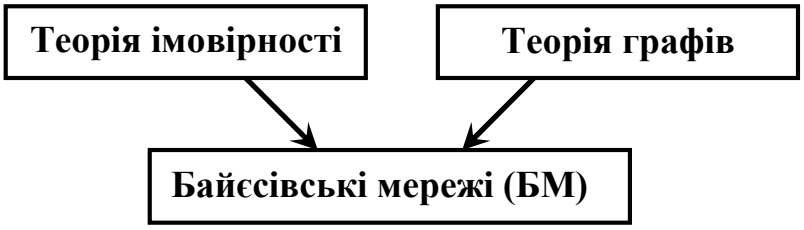


Рисунок 1.1 – БМ на стику двох наук

Термін “байєсівська мережа” (Bayesian Network) був запропонований Джуді Перлом в 1985 році, з метою акцентування трьох аспектів [2]:

- природи вхідних даних;

- отримання достовірної інформації при застосуванні теореми Байєса;

- застосування причин та наслідків – запропонована в 1763 році в посмертній роботі Томаса Байєса.

Ідея впровадження БМ полягає в представленні причинно-наслідкових зв’язків процесу у вигляді графа. Треба зауважити, що ідея представлення причинно-наслідкових зв’язків у вигляді графів розглядалася набагато раніше. Біолог–генетик Сьюел Райт (Sewall Wright, 1889-1988 р.) запропонував пат-аналіз (path analysis) – статистичний метод, який дозволяє оцінити ступінь взаємовпливу змінних у причинно-наслідковій моделі. Для цього він об’єднав лінійні регресійні моделі та спрямовані графи, пізніше ця ідея була розвинута Гербертом Сімоном (Herbert Simon, 1916-2001) та Хьюбертом Балоком (Hubert Blalock).

До впровадження терміну “байєсівська мережа” Джуді Перлом в 1985 році, БМ застосовувалися під назвою каузальних мереж (causal network), тобто мережі з причинно-наслідковими зв’язками. Байєсівськими вони стали завдяки застосуванню в каузальних мережах теореми Байєса.

Байєсівські мережі (БМ) представляють собою графічні моделі подій і процесів на основі об'єднання деяких результатів теорії ймовірностей і теорії графів. Вони тісно пов'язані з діаграмами впливу, які можна використати для

прийняття рішень. Незважаючи на свою назву, ці мережі не обов'язково мають на увазі тісний зв'язок з байєсівськими методами. Назва пов'язана, насамперед, з байєсівським правилом ймовірнісного висновку [3]. В літературі байєсівські мережі (BN – Bayesian networks) іноді зустрічаються під назвами байєсівські мережі довіри (BBN – Bayesian belief networks), причинні мережі (causal networks) або ймовірнісні мережі (probabilistic networks). БМ представляють собою зручний інструмент для опису досить складних процесів і подій з невизначеностями. Вони виявилися особливо корисними при розробці та аналізі машинних алгоритмів навчання. Головною ідеєю побудови графічної моделі є поняття модульності, шляхом розкладу складної системи на прості складові частини. Задля об'єднання окремих елементів у систему використаються результати теорії ймовірностей, які забезпечують моделі практичну дієздатність у цілому, а також дають можливість поєднувати графічні моделі з базами даних. Такий граф-теоретичний підхід до побудови моделі дає досліднику можливість будувати моделі процесів з множини сильно взаємодіючих змінних, а також створювати структури даних для наступної розробки ефективних алгоритмів їхньої обробки та прийняття рішень.

Формалізм побудови узагальнених графічних моделей поєднує в собі багато методів статистичного моделювання, таких як факторний аналіз, аналіз розподілів, моделі сумішей розподілів, приховані марковські моделі, фільтри Калмана, моделі Айзинга та деякі інших. Всі зазначені моделі можна розглянути в рамках графічних моделей байєсівського типу як окремі приклади загального формалізму. Перевагою такого підходу є те, що методи дослідження процесів та обробки даних, розроблені в одній області, можуть бути успішно перенесені в інші.

Незважаючи на те, що байєсівським мережам приділяється багато уваги в зарубіжній літературі, принципи їхньої побудови, навчання та використання ще недостатньо освітлені у вітчизняних публікаціях, що істотно утрудняє їхнє розуміння й застосування.

1.2.2 Переваги застосування байєсівських мереж

На відміну від інших методів ІАД, застосування байєсівських мереж для аналізу процесів різної природи, діяльності людини та функціонування технічних систем дозволяє враховувати та використовувати будь-які вхідні дані – експертні оцінки і статистичну інформацію. В свою чергу змінні можуть бути дискретними і неперервними, а характер їх надходження при аналізі та прийнятті рішення може бути як в режимі реального часу так і у вигляді статистичних масивів інформації і баз даних. При цьому, завдяки використанню представлення взаємодії між факторами процесу у вигляді причино-наслідкових зв'язків в мережі, у порівнянні з іншими методами ІАД, досягаються максимально високий рівень візуалізації та, як наслідок, чітке розуміння суті взаємодії факторів процесу між собою. Іншими перевагами БМ є можливості врахування невизначеностей статистичного, структурного і параметричного характеру, а також формування висновку за допомогою різних методів – наближених і точних. Загалом, можна сказати, що БМ – це високоресурсний метод ймовірнісного моделювання процесів довільної природи з невизначеностями різних типів, який забезпечує можливість достатньо точного опису їх функціонування, оцінювати прогнози та будувати системи управління.

1.2.3 Граф-теоретичне представлення байєсівської мережі

Теорія графів, як інструмент дискретної математики, досить широко застосовується для моделювання процесів різної природи. Одним із можливих варіантів застосування є моделювання ієрархічних структур в теорії організації.

БМ – це графи із деякими спеціальними властивостями. Розглянемо основи побудови БМ у вигляді графів.

Граф складається з вузлів (змінних, вершин), з'єднаних дугами. Дуга, яка з'єднує змінні, вказує на існування причинного зв'язку між ними. Якщо дуга не спрямована, то вона свідчить тільки про наявність зв'язку між змінними. Граф, що містить тільки неспрямовані дуги, називають неспрямованим. Якщо дуга

має стрілку, вона вказує на напрям залежності – від причини до наслідку. Граф, у якому всі дуги спрямовані, називають спрямованим. Всі байєсівські мережі – спрямовані графи.

1.2.3.1Типи графічних структур байєсівської мережі

1. Дерево – така структура БМ, у якій будь-яка вершина може мати не більш ніж одну батьківську (рисунок 1.2).

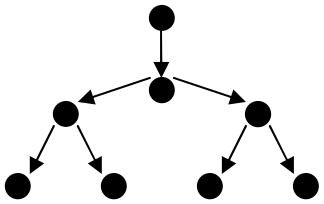


Рисунок 1.2 – Структура БМ у вигляді дерева

2. Полі-дерево – така структура БМ, у якій будь-яка вершина може мати більш ніж одну батьківську вершину, але при цьому між будь-якими двома вершинами повинно бути не більше одного з’єднуючого їх шляху (рисунок 1.3).

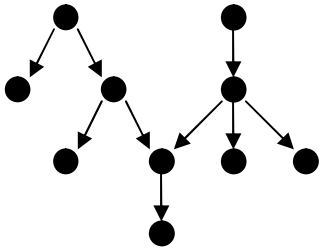


Рисунок 1.3 – Структура БМ у вигляді полі-дерева

3. Мережа – це структура у якій будь-яка вершина може мати більш ніж одну батьківську вершину; при цьому між будь-якими двома вершинами може бути більше одного з’єднуючого їх шляху(рисунок 1.4).

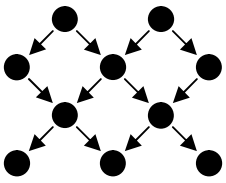
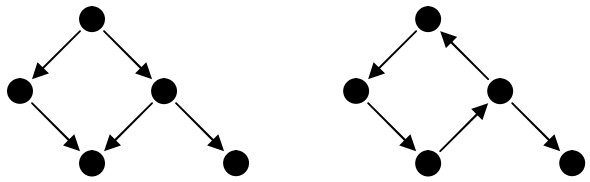


Рисунок 1.4 – Структура БМ у вигляді мережі

Дерева та полі-дерева називають також однозв'язними мережами, а мережі – багатозв’язаними мережами. Якщо всі дуги графа спрямовані, то його називають спрямованим ациклічним графом. Якщо спрямований граф містить хоча б один замкнений цикл, то його називають спрямованим циклічним графом. Приклади спрямованого ациклічного та циклічного графів показані на рисунок 1.5.



а б

Рисунок 1.5 – Спрямовані ациклічний (а) та циклічний (б) графи.

Спрямований ациклічний граф називають одиночно з’єднаним, якщо між будь-якими його двома змінними існує тільки один шлях, що їх з’єднує. Графи, показані на рисунку 1.5, не відносяться до такого типу. Очевидно, що графи з одиночними з’єднаннями (дерева та полі-дерева) є простішими для аналізу, але вони рідко зустрічаються при описанні реальних процесів та подій.

Деякі змінні БМ мають спеціальні імена. Змінну (вузол, вершину) називають дитячою (нащадком), якщо вона залежить від однієї або більше інших змінних. Батьківська змінна має одну або більше дитячих змінних. До множини нащадків відносять дитячі змінні, які відносяться до однієї батьківської змінної, а також дитячі змінні дитячих змінних вищого рівня. Одна і та ж змінна може бути батьківською і дитячою одночасно. Якщо змінна не має жодної батьківської, то її називають кореневою. Кореневі змінні – самі важливі змінні мережі; як правило, це атрибут, який нас цікавить. Змінну називають листком, якщо вона не має дитячих змінних. Однозв’язаний граф із змінними, що мають спеціальні назви, наведено на рисунку 1.6.

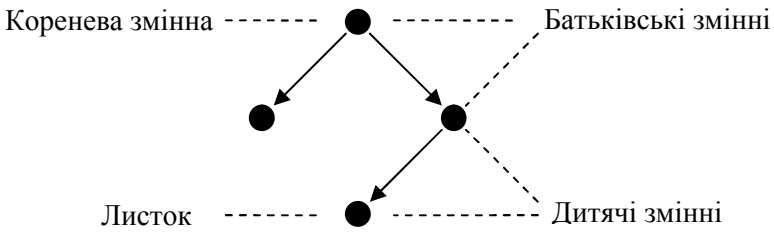


Рисунок 1.6 – Однозв’язана мережа.

Неоднозв’язаність – це важлива властивість мережі. Якщо мережі однозв’язані, то висновок (рішення) можна знайти досить легко і прямолінійно. Але всі практичні задачі приводять, як правило, до неоднозв’язаних графів рис. 1.5.б. Такі задачі значно ускладнюють і уповільнюють формування висновку.

1.2.4 Стандартний математичний запис байесівської мережі

БМ представляє собою пару $< G, B >$, у якій перша компонента G – це спрямований нециклічний граф, що відповідає випадковим змінним і записується як набір умов незалежності: кожна змінна незалежна від її батьків в G. Друга компонента пари B – це множина параметрів, що визначають мережу. Компонента містить параметри $\Theta_{X^{(i)}\left| pa\left(X^{(i)}\right)\right.} = P\left(X^{(i)}\left| pa\left(X^{(i)}\right)\right.\right)$ для кожного можливого значення $x^{(i)} \in X^{(i)}$ та $pa(X^{(i)}) \in Pa(X^{(i)})$, де $Pa(X^{(i)})$ позначає набір батьків змінної $X^{(i)} \in G$. Кожна змінна $X^{(i)} \in G$ представляється у вигляді вершини. Якщо розглядають більше одного графа, то для визначення батьків змінної $X^{(i)}$ в графі G використають позначення $Pa^{G}(X^{(i)})$. Повна спільна ймовірність БМ обчислюється за формулою:

$$P_B(X^{(1)},...,X^{(N)}) = \prod_{i=1}^{N} P_B(X^{(i)}\left| Pa(X^{(i)})\right.)$$

З математичної точки зору БМ – це модель подання існуючих і відсутніх імовірнісних залежностей. При цьому зв'язок $A \to B$ являється причинним, коли подія A – причина виникнення B, тобто коли існує механізм, відповідно

до якого значення, прийняте A, впливає на значення, прийняте B. БМ називають причинною (каузальною), коли всі її зв'язки являються причинними.

1.3 Формула Байєса

Позначимо через Ω вибірковий простір подій (або множина подій) випадкових експериментів. Цей вибірковий простір містить всі можливі значення випадкової змінної. По аналогії із словом “стан” слово “змінна” неоднозначне. “Змінна” в імовірнісному контексті дійсно означає змінну, а в граф-теоретичному контексті “змінна” означає атрибут. Якщо змінна приймає конкретне значення, то говорять, що має місце конкретна подія. В теорії оцінювання сигналів, аналізі часових рядів та багатьох інших випадках події називають спостереженнями.

Припустимо, що є дві змінні E і H, які деяким чином пов’язані між собою. Якщо ми отримуємо конкретне значення H, тобто має місце конкретна подія, то цікаво знати, якою буде при цьому ймовірність події E. Тобто необхідно знайти умовну ймовірність події E. Умовна ймовірність події визначається за формулою:

$$p(E \mid H_k) = \frac{p(E \cap H_k)}{p(H_k)} \tag{1.1}$$

де $\cap -$ операція перетину множин.

Якщо $E_1, E_2, ..., E_n -$ взаємновиключні події такі, що $\cup_{i=1}^{n} E_i = \Omega$, то кажуть, що події E_i формують повну (вичерпну) множину. Дві змінні не *зв’язані* (не перетинаються), якщо вони не мають однакових значень. Якщо дві змінні є вичерпними і незв’язаними, то можна записати, що

$$E = \bigcup_i (E \cap H_i), \quad (E \cap H_i) \cap (E \cap H_j) = \varnothing, \; i \neq j \tag{1.2}$$

Формулу Байєса широко застосовують в системах підтримки прийняття рішень, біометриці, теорії оцінювання і зв'язку та багатьох інших галузях науки і техніки.

Теорія побудови БМ ґрунтується на припущенні, що події є вичерпними і не перетинаються. Якщо ця умова не виконується, то результати застосування мережі будуть неконсистентними (тобто неточними). У випадку коли події вичерпні і не перетинаються, то ймовірність події E можна обчислити за допомогою умовних ймовірностей

$$p(E)=\sum_{i=1}^{n} p(E\cap H_i)=\sum_{i=1}^{n} p(E|H_i)\cdot p(H_i). \qquad (1.3)$$

Використовуючи рівняння (1.3), суму перетинів події E з H можна виразити так:

$$p(E\cap H_k)=p(E\mid H_k)\cdot p(H_k)=p(H_k\mid E)\cdot p(E). \qquad (1.4)$$

З рівності $p(E\mid H_k)\cdot p(H_k)=p(H_k\mid E)\cdot p(E)$ знайдемо, що

$$p(H_k\mid E)=\frac{p(E\mid H_k)\cdot p(H_k)}{p(E)},$$

а із врахуванням (1.3) отримаємо вираз

$$p(H_k\mid E)=\frac{p(E\mid H_k)\cdot p(H_k)}{\sum_{i=1}^{n} p(E\mid H_i)\cdot p(H_i)}, \qquad (1.5)$$

який представляє собою формулу Байєса. На основі цієї формули будуються БМ.

В (1.5) H_k означає будь-яку гіпотезу з n можливих. Ймовірності $p(E|H_k)$ задаються експертами апріорно, або розраховують по навчальним даним [4, 5]. Тобто, їх можна розглядати як відповідь на запитання: "Якою буде ймовірність деякого виміру, якщо відомо яка гіпотеза була реалізована?". Ймовірності $p(E|H_k)$ є дуже корисними, тому що, як правило, легше знайти ймовірність послідовності подій типу причина-наслідок, ніж навпаки. Значення $p(H_k)$ називають апріорними ймовірностями, вони визначають початкові ймовірності для всіх гіпотез. Перевага байєсівського методу полягає в тому, що апріорні ймовірності можна уточнювати (оновлювати) у відповідності до реалій протікання процесу, який досліджується. Це дозволяє уточнювати ймовірності подій при надходженні додаткової інформації. Знаменник виразу (1.5) можна розглядати як нормуючий член, який встановлює значення ймовірності між 0 та 1.

При аналізі та моделюванні випадкових величин найчастіше використовують нормальний або Гауссів розподіл, який має декілька властивостей, що сприяють прискоренню та спрощенню аналізу процесів. Він описується виразом

$$N(x;\,\mu_x,\sigma_x)=\frac{1}{\sqrt{(2\pi\sigma_x)}}\cdot\exp\left(-\frac{1}{2}\,\frac{(x-\mu_x)^2}{\sigma_x}\right), \qquad (1.6)$$

де μ_x – середнє значення процесу; σ_x – стандартне відхилення. Очевидно, що для описання випадкових процесів можна скористатись будь-яким іншим розподілом, який краще відповідає реальності.

Іншими типами розподілів можуть бути такі: бета-розподіл, біноміальний, розподіл χ – квадрат, нецентральний розподіл χ – квадрат, дискретний рівномірний розподіл, експоненціальний, F – розподіл, нецентральний F – розподіл, гамма-розподіл, геометричний, гіпергеометричний, логнормальний, негативний біноміальний та розподіли

Пуассона, Рейлі, t–розподіл Стьюдента, нецентрований t–розподіл, рівномірний неперервний розподіл та розподіл Вейбулла.

1.4 Типи байєсівських мереж

1.5.1. Дискретні БМ

Дискретні БМ – мережі, у яких змінні вузлів представлені дискретними величинами.

Дискретні БМ мають такі властивості:

– кожна вершина представляє собою подію, яка описується випадковою величиною, яка може мати кілька станів;

– всі вершини, пов'язані з "батьківськими", визначаються таблицею умовних ймовірностей (ТУЙ) або функцією умовних ймовірностей;

– для вершин без "батьків" ймовірності її станів є безумовними (маргінальними).

Інакше кажучи, у байєсівських мережах довіри вершини представляють собою випадкові змінні, а дуги – ймовірнісні залежності, які визначаються через таблиці умовних ймовірностей. ТУЙ кожної вершини містить ймовірності станів цієї вершини за умови станів її "батьків". На рис. 2.6 наведено приклад дискретної БМ, кожна з вершин може приймати одне з двох станів F або T (скорочення від false та true). Запис $P(A|B)$ означає ймовірність настання події A при умові, що подія B вже відбулась.

1.5.2. Динамічні БМ

Динамічні БМ – мережі, у яких значення вузлів змінюються з часом.

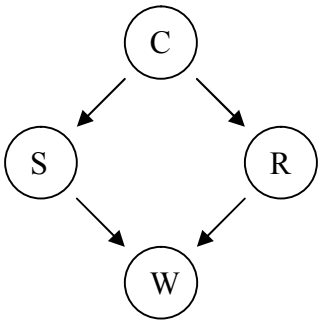
Динамічні БМ ідеально підходять для моделювання часових процесів. Їхня перевага полягає в тому, що вони використають табличне представлення умовних ймовірностей що полегшує представлення різних нелінійних явищ [6]. Треба підкреслити, що термін "часова байєсівська мережа" (temporal Bayesian network) краще відображає суть ніж "динамічна байєсівська мережа" (dynamic

Bayesian network), тому що передбачається, що структура моделі не змінюється. Зазвичай параметри моделі не змінюються з часом, але до структури мережі завжди можна додати додаткові приховані вузли для опису поточного стану процесу [3].

P(C=F)	P(C=T)
0,5	0,5

C	P(S=F | C)	P(S=T | C)
F	0,5	0,5
T	0,9	0,1

C	P(R=F | C)	P(R=T | C)
F	0,8	0,2
T	0,2	0,8



S	R	P(W=F | S,R)	P(W=T | S,R)
F	F	1,0	0
T	F	0,1	0,9
F	T	0,1	0,9

Рисунок 1.7 – Дискретна БМ з ТУЙ

Найпростіший тип динамічної БМ – це прихована модель Маркова (Hidden Markov Model), у якій в кожному шарі наявний один дискретний прихований вузол та один дискретний або безперервний спостережуваний вузол. Ілюстрація моделі наведена на рисунку 1.8. Круглі вершини позначають неперервні вузли, квадратні позначають дискретні; X – приховані вузли, а Y – спостережувані. Для визначення динамічної БМ потрібно задати початковий розподіл $P(X(t))$, топологію усередині шару та міжшарову топологію (між двома шарами) $P(Y(t)|X(t))$ [7].

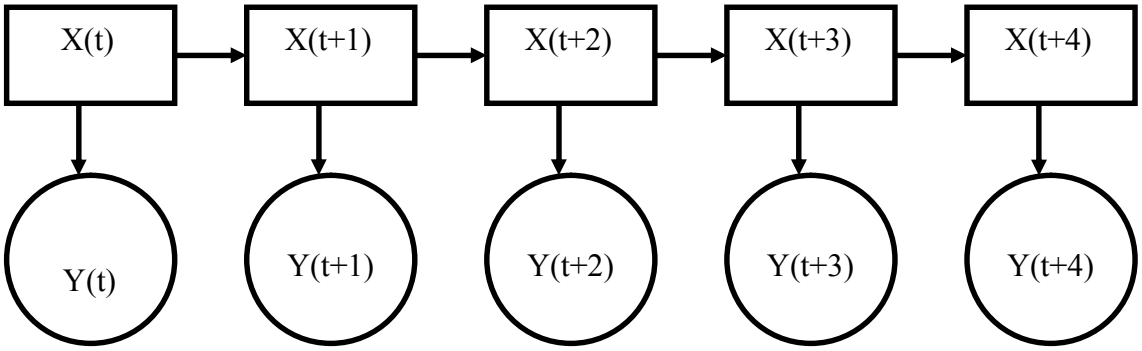


Рисунок 1.8 – Скрита двохшарова модель Маркова, в якій X – приховані дискретні, а Y – дискретні або неперервні спостережувані вузли

Мережі такого типу використовують для розпізнавання мови. У цьому випадку вузли $Y(t), Y(t+1), Y(t+2),...$ представляють собою фонеми слів, а вузли $X(t), X(t+1), X(t+2),...$ – це букви, з яких складається слово. Така модель є динамічною в тому сенсі, що дана мережа буде представляти собою множину блоків, які повторюються у різні моменти часу [8].

1.5.3. Неперервні БМ

Неперервні БМ – мережі в яких змінні вузлів – це неперервні величини. У багатьох випадках події можуть приймати будь-які стани з деякого діапазону. Тобто змінна X буде неперервною випадковою величиною, простором можливих станів якої буде весь діапазон її припустимих значень $X = \{x | a \le x \le b\}$, яке містить нескінченну множину точок. В цьому випадку некоректно говорити про ймовірності окремого стану, тому що при їх нескінченно великій кількості вага кожного буде дорівнювати нулю. Тому розподіл ймовірностей для неперервної випадкової величини визначається інакше, ніж у дискретному випадку; для їх опису використовуються функції розподілу ймовірностей і щільності розподілу ймовірностей.

Неперервні БМ використовуються для моделювання стохастичних процесів у просторі станів з неперервним часом. На рисунку 1.9 наведено приклад неперервної БМ; у цьому випадку використовується розподіл Гауса.

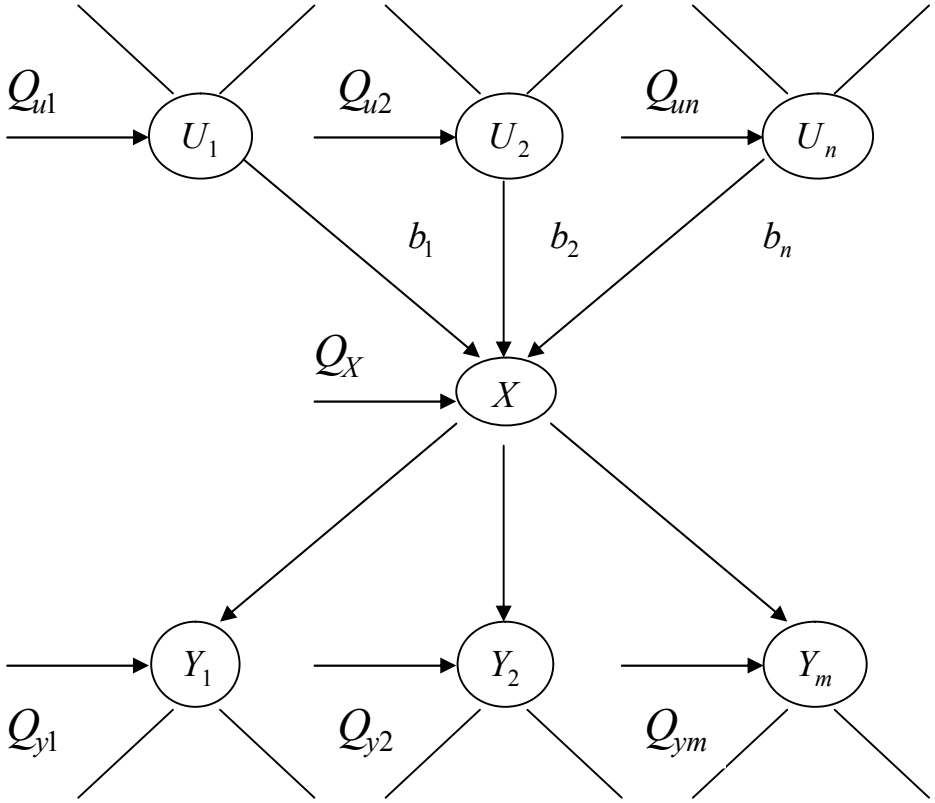


Рисунок 1.9 – Приклад неперервної БМ

Тут $U_1, U_2, ..., U_n$ – батьківські вузли, X, $Y_1, Y_2, ..., Y_m$ – дитячі вузли змінної X, $b_1, b_2, ..., b_n$ – вагові коефіцієнти, $Q_x, Q_{u1}, ..., Q_{un}, Q_{y1}, .., Q_{ym}$ – шумові коефіцієнти.

Нехай вузол X має безліч батьків $U = \{U_1, U_2, ..., U_n\}$, тоді умовний розподіл для X задається за формулою $f(X|U_i) = N(x; \mu_x + b_i \cdot \mu_i, \sigma_x)$, де коефіцієнт b_i показує зв'язок між X та його i-м батьком (його ще називають ваговим коефіцієнтом) [9, 10]:

$$N(x; \mu_x + b_i \cdot \mu_i, \sigma_x) = \frac{1}{\sqrt{2 \cdot \pi \cdot \sigma_x}} \cdot \exp\left(\frac{1}{2} \cdot \frac{(x - (\mu_x + b_i \cdot \mu_i))^2}{\sigma_x} \right),$$

де μ – математичне очікування, σ – дисперсія, а коефіцієнт $\frac{1}{\sqrt{2 \cdot \pi \cdot \sigma_x}}$ називають нормуючою константою, яка гарантує, що $\int_x N(x; \mu_x + b_i \cdot x_i, \sigma_x) = 1$.

Зв'язок між змінною X і її батьками $(U_1, U_2, ..., U_n)$ можна представити за допомогою звичайної лінійної регресійної моделі:

$$X = b_1 \cdot U_1 + b_2 \cdot U_2 + ... + b_n \cdot U_n + Q_x,$$

де Q_x – шумова компонента, що може бути записана у вигляді розподілу Гауса з нульовим математичним очікуванням, а $b_1, b_2, ..., b_n$ регресійні коефіцієнти, що показують зв'язок між змінною X і $U_1, U_2, ..., U_n$ її батьками.

1.5.4 Гібридні БМ

Гібридні БМ – мережі, які містять вузли з дискретними і неперервними змінними.

При використанні БМ, що містять неперервні і дискретні змінні, існує ряд обмежень:

1 – дискретні змінні не можуть мати неперервних батьків;

2 – неперервні змінні повинні мати нормальний закон розподілу, умовний на значеннях батьків;

3 – розподіл неперервної змінної X з дискретними батьками Y та неперервними батьками Z є нормальним розподілом:

$$P(X|Y = y, Z = z) = N(\mu_x(\mu_y, \mu_z), \sqrt{\sigma_x(\sqrt{\sigma_y})}),$$

де μ_x, μ_y, μ_z – математичні очікування, σ_x, σ_y – дисперсії, $\sqrt{\sigma_x}, \sqrt{\sigma_y}$ – середньоквадратичне відхилення; μ_x лінійно залежить від неперервних батьків, а σ_x взагалі не залежить від неперервних батьків. Однак, μ_x та σ_x залежать від дискретних батьків. Це обмеження гарантує можливість формування точного висновку.

Розглянемо приклад на найпростішій дворівневій мережі для випадку прямого поширення розподілу ймовірностей.

Нехай незалежні дискретні випадкові величини $X_1,...,X_S$ та неперервні випадкові величини $Z_1,...,Z_r$ впливають на результуючу випадкову величину Y, як показано на рисунку 1.10.

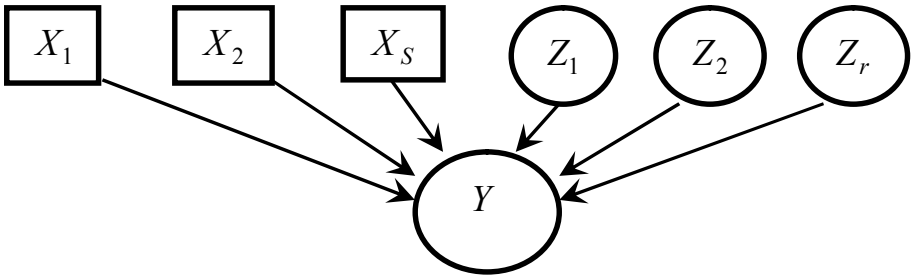


Рисунок 1.10 – Приклад дворівневої гібридної БМ з неперервними та дискретними змінними

Квадратні вершини відповідають дискретним подіям, а овальні – неперервним подіям.

Кожна з дискретних випадкових величин $X_j (j = 1,...,s)$ має своїми результатами значення $X_{i_i} (i = 1,...,n_j)$ з ймовірностями P_{ij}, для яких $\sum_{i=1}^{n_j} P_{ij} = 1$. Спільний вплив дискретних випадкових величин на Y характеризується математичним очікуванням $(\mu_{i1},...,\mu_{is})$ та дисперсіями $(\sigma_{i1},...,\sigma_{is})$. Кожна з неперервних випадкових величин Z_l має неперервний нормальний розподіл з параметрами (μ_l, σ_l), де $l = 1,...,r$. Спільний вплив неперервної випадкової величини Z_l та результатів дискретних величин на результуючу випадкову величину Y характеризується ваговими коефіцієнтами $k_{l,i1,...is}$ для $l = 1,...,r$.

Тоді характеристики результуючої величини Y можуть бути обчислені за такими формулами:

$$\mu = \sum_{i1}^{n1} \ldots \sum_{is}^{ns} p_{1,i1} \ldots p_{s,is} \left(\mu_{i1,\ldots,is} + \sum_{l=1}^{r} K_{l,i1,\ldots,is} \cdot \mu_l \right),$$

$$\sigma = \sum_{i1}^{n1} \ldots \sum_{is}^{ns} p_{1,i1} \ldots p_{s,is} \cdot \left(\left(\mu_{i1,\ldots,is} + \sum_{l=1}^{r} K_{l,i1,\ldots,is} \right)^2 + \sigma_{i1,\ldots,is} + \sum_{l=1}^{r} K_{l,i1,\ldots,is}^2 \cdot \sigma_l \right) - \mu^2 .$$

Стосовно до нашого прикладу, який містить дві вихідні неперервні ($r = 2$) змінні та одну дискретну ($s = 1$) змінну, що має три результати ($n1 = 3$), числові характеристики випадкової змінної будуть дорівнювати:

$$\begin{aligned}
\mu &= \sum_{i=1}^{3} p_i \left(\mu_i + \sum_{l=1}^{2} k_{li} \cdot \mu_i \right) = \\
&= 0{,}333 \cdot (3000 + 50000 \cdot 0{,}075 + 0{,}6 \cdot 2500) + \\
&+ 0{,}333 \cdot (3200 + 40000 \cdot 0{,}075 + 0{,}5 \cdot 2500) + \\
&+ 0{,}333 \cdot (3500 + 3000 \cdot 0{,}075 + 0{,}4 \cdot 2500) = \\
&= 0{,}333 \cdot (8250 + 7450 + 6750) = 7483{,}33
\end{aligned}$$

$$\sigma = \sum_{i=1}^{3} p_i \left(\left(\mu_i + \sum_{l=1}^{2} k_{li} \cdot \mu_l \right)^2 + \sigma_i + \sum_{l=1}^{r} k_{li}^2 \cdot \sigma_l \right) - \mu^2 = 1459505{,}6,$$

$$\sqrt{\sigma} = 1208{,}1.$$

2 ПОБУДОВА СТРУКТУРИ БАЙЄСІВСЬКОЇ МЕРЕЖІ

Інтелектуальний аналіз даних (Data Mining) – це технологія виявлення прихованих взаємозв'язків всередині великих баз даних. Більшість інструментів інтелектуального аналізу даних ґрунтується на двох технологіях: машинне навчання (machine learning) і візуалізація (візуальне подання інформації). Байєсівські мережі якраз і поєднують у собі ці дві технології.

Більшість існуючих методів побудови структури БМ можна умовно розділити на дві категорії: на основі оціночних функцій (search & scoring) та на основі застосування тесту на умовну незалежність (dependency analysis) [11,12]. Більшість із існуючих методів зустрічаються з наступними проблемами:

1. Наявність впорядкованої множини вершин (ВМВ). У більшості методів, особливо старих, вважається, що ВМВ задана, але на практиці при роботі з реальними даними і це дуже часто не відповідає дійсності.

2. Низька обчислювальна ефективність. Деякі сучасні методи працюють без використання ВМВ, замість неї використовується тест на умовну незалежність (ТУН). Але в цьому випадку необхідно виконати експоненціальну кількість таких тестів, що призводить до зменшення ефективності роботи методу у зв’язку із зростанням об’єму обчислювань.

3. Проблема побудови великих БМ. Існує декілька методів здатних побудувати структуру БМ з декілька сотень вершин, використовуючи навчальну вибірку з мільйонів записів. До таких методів відносяться Tetrad II [13] та SopLeq [14].

2.1 Алгоритми побудови байєсівських мереж

2.1.1 Алгоритми на основі оціночних функцій (search & scoring)

Чу і Ліу (Chow and Liu) запропонували алгоритм для побудови БМ у вигляді дерева, заснований на використанні значень взаємної інформації між

вершинами [15]. В якості рішення видається структура зі значенням спільного розподілу ймовірностей мережі, найбільш відповідного навчальним даним Побудова структури БМ здійснюється за $O(N^2)$ кроків, де N – кількість вершин мережі, однак цей алгоритм не працює для багатозв'язних БМ.

Рібан і Перл (Rebane and Pearl) запропонували вдосконалений змінений алгоритм Чу і Ліу для побудови БМ у вигляді полі-дерева [16].

У 1988 році Купер і Герскович (Cooper and Herskovits) розробили алгоритм Кутато (Kutato) [17]. На етапі ініціалізації алгоритму вважається, що всі вершини БМ незалежні, після чого обчислюється ентропія цієї мережі. Потім виконується додавання дуг між вершинами в мережі таким чином, щоб мінімізувати ентропію БМ. Для роботи алгоритму потрібна наявність ВМВ.

У 1992 році Купер і Герскович запропонували широко відомий алгоритм К2 [18], який виконує пошук структури з максимальним значення функції Купера-Герсковича (КГ). Для роботи алгоритму потрібна наявність ВМВ.

В 1994 році був запропонований алгоритм HGC [19]. Цей алгоритм суттєво відрізняється від інших алгоритмів тим, що вперше саме в ньому були використані два нових поняття: параметричної модульності (parametric modularity) та рівнозначності подій (event equivalence). Інші дослідники досить довго не використовували одночасно цих понять. Але одночасне застосування цих понять дозволяє об'єднувати статистичну інформацію та експертних знань для побудові БМ.

Вонг і Ксіанг (Wong and Xiang) запропонували році алгоритм для побудови Марковських мереж з використанням значення ентропії та I-map [20]. Граф G ймовірнісної моделі M називають незалежною картою (independency map, скорочено I-map), якщо з незалежності вершин графа G випливає незалежність моделі M. Цей алгоритм дозволяє представити модельований процес у вигляді I-map і у випадку, коли мережа являється однозв'язною, гарантовано будується БМ. Разом із Чу (Chu) Ксіанг розробив у 1997 році більш швидкий варіант запропонованого алгоритму [21].

Алгоритм Лема-Бахуса (Lam-Bacchus) [22], запропонований в 1996 році, виконує евристичну побудову структури мережі, використовуючи значення взаємної інформації між вершинами, а в якості функції оцінки використається функція опису мінімальною довжиною (minimum description length).

Алгоритм Бенедикта (Benedict) [23], запропонований в 1996 році, виконує евристичний пошук на основі ВМВ, аналізуючи умовні незалежності в структурі мережі на основі *d*-розділення, а в якості функції оцінки використається ентропія.

СВ алгоритм [24] запропонований в 1995 році. Він використає ТУН між вершинами мережі, для побудови ВМВ. Для побудови структури мережі використається функція КГ.

Алгоритм Фрідмана-Голдшмідта (Friedman-Goldszmidt) [25] запропонований в 1996 році. Для побудови мережі використається аналіз її локальних підструктур, а в якості функції оцінки використається функція опису мінімальною довжиною (ОМД) та оцінка Байєса.

В алгоритмі WKD, запропонованому в 1996 році, в якості функції оцінки при побудові мережі використається функція повідомлення мінімальної довжини (minimum message length), яка схожа на ОМД [26].

Алгоритм Сузукі (Suzuki) [27], запропонований в 1999 році, заснований на методі гілок та границь (branch and bound method) для завдання послідовності побудови структури мережі, а в якості функції оцінки використається ОМД.

Також існує ціла низка різноманітних поглинаючих алгоритмів (greedy algorithm) в яких для оцінювання можуть використовуватися різноманітні функції, наприклад максимальної правдоподібності або байєсівський інформаційний критерій.

2.1.2 Алгоритми з використанням тестів на умовну незалежність (dependency analysis)

Вермут і Лоуренс (Wermuth and Lauritzen) запропонували алгоритм для побудови структури БМ застосовуючи ТУН [28]. Цей алгоритм виконує послідовний перебір ВМВ. Для кожної пари вершин X_k та X_t, таких що $X_t < X_k$ (тобто X_k це предок відносно X_t), виконується обчислення значення умовної незалежності. Цей алгоритм гарантує побудову БМ по навчальним даним, але при цьому буде потрібно виконати велику кількість ТУН між вершинами, що можливо лише у випадку, коли мережа складається з невеликої кількості вершин.

В 1988 році Перл (Pearl) запропонував алгоритм побудови скінченого спрямованого ациклічного графа (boundary DAG algorithm) [29]. Цей алгоритм будує БМ маючи ВМВ та функцію спільного розподілу (або достатньо велику навчальну вибірку даних). Разом із будь-яким, не досить складним методом пошуку, цей алгоритм позбавлений проблеми, яка полягає в необхідності розрахунку великої кількості ТУН застосовуючи алгоритм Вермута і Лоуренса . Однак проблема в необхідності обчислення великої кількості ТУН з'являється при використанні цього алгоритму для побудови Марковських мереж, тобто мереж зі скритими вузлами (hidden node).

В 1990 році був запропонований SRA алгоритм [30], який являється модифікацією алгоритму скінченого спрямованого ациклічного графа. Цей алгоритм має менш жорсткі вимоги до ВМВ, для побудови БМ достатньо мати частково впорядковану множину вершин та ще деякі обмеження. Побудова БМ виконується послідовним додаванням дуг між вершинами, для цього використовується евристичний пошук. Але алгоритм виконує експоненціальну кількість розрахунків ТУН.

Алгоритм "Конструктор" (constructor algorithm) [31] запропонований в 1990 році, дуже схожий на алгоритм побудови скінченого спрямованого ациклічного графа. Він намагається замість БМ побудувати Марковську мережу. Відмінність цього метода від інших методів які використовують ТУН в тому що він не виконую надлишкові ТУН та йому не потрібна ВМВ.

Алгоритму SGS [32], запропонованому в 1990 році, для побудови структури не потрібна наявність ВМВ, але замість неї йому доводиться виконувати експоненціальну кількість тестів на умовну незалежність між вершинами.

PC алгоритм розроблений в 1991 році представляє собою вдосконалений варіант SGS алгоритму. Цей алгоритм спеціально розроблявся для побудови розріджених БМ (sparse BN), тобто для мереж з невеликою кількістю дуг між вершинами [33].

Алгоритм KDB, запропонований в 1996 році, для визначення напряму побудови мережі використовує ЗВІ. Як оціночна функція використовується функціонал, що мінімізує значення мережі.

Алгоритм FBC (full Bayesian network), запропонований в 2006 році, представляє собою удосконалений алгоритм KDB, який в якості функції оцінки при побудові мережі використовує функцію сумарних значень ЗВІ вершин.

2.1.3 Інші алгоритми

Не завжди побудована структура БМ відповідає процесу який моделюється, інколи це пов’язано з неповнотою даних спостережень або недостатньою визначеністю предметної області. Замість побудови однієї найкращої структури БМ деякі алгоритми в якості результату видають кілька мережевих структур.

Іноді дослідник може не мати всієї інформації про процес, який моделюється. Тобто деякі змінні, які впливають на процес, відсутні, їх називають прихованими змінними (hidden variables) або латентним змінними (latent variables). Існують алгоритми евристичного пошуку, які намагаються враховувати такі скриті змінні при моделюванні.

Для випадку, коли навчальні дані неповні, або частина з них невірна (missing data) було запропоновано декілька алгоритмів стиснення границь (bound and collapse) та група алгоритмів, які використовують значення

максимального математичного очікування (expectation maximization, скорочено EM).

Метод стиснення границь [34] моделює відсутність даних, припускаючи що ймовірність даних, які відсутні, приймає значення в інтервалі від 0 до 1. Тобто виконується обчислення цього інтервалу на відсутність даних, за тією інформацією яка мається. Після цього робиться стиснення границь інтервалу в точку за допомогою використання опуклої комбінації з точок екстремумів, використовуючи інформацію про неповні дані.

Алгоритм максимізації математичного очікування був запропонований в 1977 році в роботі [35]. Алгоритм намагається знайти локальні оптимальні оцінки максимальної правдоподібності параметрів. Головна ідея алгоритму полягає в тому що, якби ми знали значення всіх вузлів, то навчання (на кроці M) було б простим, оскільки ми мали би всю необхідну інформацію. Тому на кроці E робиться обчислення значення очікуваної правдоподібності (expectation of the likelihood) включаючи латентні змінні, так ніби ми мали можливість їх спостерігати. На кроці M робиться обчислення значення максимальної правдоподібності (maximum likelihood estimates) параметрів використовуючи максимізацію значень очікуваної правдоподібності отриманих на кроці E. Далі алгоритм знову виконує крок E з використанням параметрів отриманих на кроці M і так далі.

На основі алгоритму максимізації математичного очікування розроблено цілу серію подібних алгоритмів [36]. Так, наприклад, структурний алгоритм максимізації математичного очікування (structural EM algorithm) поєднує в собі стандартний алгоритм максимізації математичного очікування, що оптимізує параметри, та алгоритм структурного пошуку моделі відбору. Цей алгоритм будує мережі, ґрунтуючись на штрафних ймовірнісних значеннях, які включають значення, отримані за допомогою байєсівського інформаційного критерію, принципу мінімальної довжини опису, а також значеннях інших критеріїв.

2.2 Нелінійна поліноміальна складність задачі побудови байєсівської мережі та зменшення розрахункової складності при побудові байєсівських мереж

Побудову БМ можна виконати "у чоло", простим перебором (exhaustive search) множини всіх можливих нециклічних моделей, з яких обирається модель найбільш адекватна навчальним даним. Але тоді ця задача набуває поліноміальної складності, тому що при повному переборі кількість всіх моделей дорівнює $3^{\frac{n\cdot(n-1)}{2}} - k_{cycle}$, де n – кількість вершин, k_{cycle} – кількість моделей з циклами. Кількість всіх можливих нециклічних моделей можна порахувати за допомогою рекурентної формули Робінсона [37]:

$$f(n) = \sum_{i=1}^{n} (-1)^{i+1} \cdot C_n^i \cdot 2^{i\cdot(n-i)} \cdot f(n-i)$$

де n – кількість вершин, а $f(0) = 1$.

Таблиця 2.1 – Залежність кількості моделей без циклів від кількості вершин, які потрібно проаналізувати при повному переборі моделей

Кількість вершин	Кількість моделей без циклів		Кількість вершин	Кількість моделей без циклів
1	1		8	783.702.329.343
2	3		9	1.213.442.454.842.881
3	25		10	4.175.098.976.430.598.100
4	543		…	
5	29.281		15	2,38*10^41
6	3.781.503		…	…
7	1.138.779.265		20	2,34*10^72

Однак на практиці виконати повний перебір моделей можна тільки для мереж не більш ніж з 7 вершинами. При кількості вершин більше 7 виконати

простий перебір неможливо, тому що не вистачить ніяких обчислювальних ресурсів.

2.2.1 Функція Купера-Герсковича

Купер і Герскович запропонували метод із використанням функції Купера-Герсковича (КГ) для навчання БМ, що заснований на пошуку структури БМ із максимальним значенням функції КГ. Функція КГ структури $g \in G$ при заданій послідовності зі n спостережень $x^n = d_1 d_2 ... d_n$ записується як рівняння:

$$P(g, x^n) = P(g) \cdot \prod_{j \in J} \left(\prod_{s \in S(j,g)} \frac{\left(\alpha^{(j)} - 1\right)! \cdot \prod_{q \in A^{(j)}} \left(n[q, s, j, g]!\right)}{(n[s, j, g] + \alpha^{(j)} - 1)!} \right), \qquad (2.1)$$

де $P(g)$ – апріорна ймовірність структури $g \in G$, але взагалі її часто опускають при обчисленнях, вважаючи що всі структури рівноймовірні;

запис $j \in J = \{1,..., N\}$ означає перебір всіх вершин структури мережі g, а $s \in S(j, g)$ означає перебір множини всіх наборів значень які приймають батьківські вершини j-ї вершини

.

$$n(s, j, g) = \sum_{i=1}^{n} I(\pi_i^{(j)} = s); \qquad n[q, s, j, g] = \sum_{i=1}^{n} I(x_i = q, \pi_i^{(j)} = s),$$

де $\pi^{(j)} = \Pi^{(j)}$ означає $X^{(k)} = x^{(k)}, \forall k \in \phi^{(j)}$, а функція $I(E) = 1$, коли предикат $E = true$, в противному випадку $I(E) = 0$.

Алгоритм навчання БМ із використанням функції КГ заснований на циклічному переборі всіх можливих ациклічних мережевих структур. В g^* зберігається оптимальна мережна структура, тобто структура яка максимально відповідає навчальним даним. Оптимальною структурою буде та, яка має найбільше значення функції $P(g, x^n)$.

Алгоритм побудови БМ із використанням методу КГ:

1 – $g^* \leftarrow g_0 (\in G)$;

2 – для $\forall g \in G - \{g_0\}$ якщо $P(g, x^n) > P(g^*, x^n)$ то $g^* \leftarrow g$;

3 – в якості рішення на вихід подається g^*.

В якості прикладу обчислимо значення функції КГ для структури зображеної на рисунку 2.1, на основі 10 навчальних даних наведених у таблиці 2.2.

$$P(1, g, x^n) = \frac{(2-1)!\cdot 5!\cdot 5!}{(10+2-1)!} = 0.00036;$$

$$P(2, g, x^n) = \frac{(2-1)!\cdot 0!\cdot 5!}{(5+2-1)!} \cdot \frac{(2-1)!\cdot 4!\cdot 1!}{(5+2-1)!} = 0.0056;$$

$$P(3, g, x^n) = \frac{(2-1)!\cdot 3!\cdot 1!}{(4+2-1)!} \cdot \frac{(2-1)!\cdot 0!\cdot 6!}{(6+2-1)!} = 0.0071;$$

$$P(g, x^n) = \frac{1}{25} \cdot \prod_{j \in J} P(j, g, x^n) = 5.7254 \cdot 10^{-10}.$$

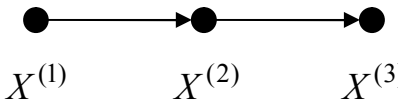


Рисунок 2.1 – Оптимальна структура БМ, що відповідає даним з таблиці 2.2

Таблиця 2.2 – Множина навчальних даних з 10 записів

n	$X^{(1)}$	$X^{(2)}$	$X^{(3)}$		n	$X^{(1)}$	$X^{(2)}$	$X^{(3)}$
1	0	1	1		6	0	1	1
2	1	0	0		7	1	0	1
3	0	1	1		8	1	0	0
4	1	0	0		9	0	1	1
5	0	1	1		10	1	1	1

Однак при використанні функції КГ необхідно враховувати обчислювальні обмеження моделюючих систем, пов'язані з кінцевою довжиною розрядної сітки. Приведемо тривіальний приклад, коли в структурі є дві вершини $X^{(1)}$ і $X^{(2)}$, а множина навчальних даних складається з мільйона записів $D = \{d^{(1)},...,d^{(1.000.000)}\}$. У такому випадку при обчисленні $P(g, x^n)$ буде потрібно обчислити факторіал виду $(n[s, j, g] + \alpha^{(j)} - 1)! = (1.000.000 + \alpha^{(j)} - 1)!$, у той час як такі солідні 32-х розрядні програми як MatLab й MathCAD здатні обчислити факторіали не більше 170!

2.2.2 Функція опису мінімальною довжиною

Згідно з теорією кодування Шеннона, при відомому розподілі $P(X)$ випадкової величини X, довжина оптимального коду для передачі конкретного значення x по каналу зв'язку прямує до $L(x) = -\log P(x)$. Ентропія джерела $S(P) = -\sum_x P(x) \cdot \log P(x)$ є мінімально очікуваною довжиною закодованого повідомлення. Будь-який інший код, заснований на невірному уявленні про джерело повідомлень, приведе до більшої довжини повідомлення. Іншими словами, чим краще побудована модель джерела, тим компактніше кодуються дані.

При навчанні джерелом даних виступає деяка невідома істинна функція розподілу $P(D|h_0)$, де $D = \{d_1,...,d_N\}$ – набір даних, а h – гіпотеза про ймовірнісне походження даних, $L(D|h) = -\log P(D|h)$ – емпіричний ризик адитивний по кількості спостережень і пропорційний емпіричній помилці. Відмінність між $P(D|h_0)$ і розподілом моделі $P(D|h)$ за мірою Кулбака-Леблера визначається як

$$\left|P(D|h)-P(D|h_0)\right|=\sum_D P(D|h_0)\cdot\log\frac{P(D|h_0)}{P(D|h)}=$$
$$=\sum_D P(D|h_0)\cdot\left|L(D|h)-L(D|h_0)\right|\geq 0\ \ , \qquad (2.2)$$

тобто, вона представляє собою різницю очікуваної довжини кодування даних на основі гіпотези про мінімально можливу довжину. Ця різниця завжди позитивна додатна і дорівнює нулю лише при повному збігу двох розподілів. Іншими словами, гіпотеза краща у тому випадку, коли середня довжина кодування даних як можна менша. Принцип ОМД у своїй спрощеній і найбільш загальній формі стверджує: серед безлічі моделей варто обрати ту, яка дозволяє описати дані найбільш коротко без втрати інформації.

У загальному випадку задача ОМД виглядає наступним чином. Спочатку задається множина навчальних даних $D=\{d_1,...,d_n\}$, $d_i=\{x_i^{(1)}x_i^{(2)}...x_i^{(N)}\}$ (нижній індекс – номер спостереження, а верхній – номер змінної), n – кількість спостережень, кожне спостереження складається з N ($N\geq 2$) змінних $X^{(1)},X^{(2)},...,X^{(N)}$, кожна j-та змінна ($j=1,...,N$) має $A^{(j)}=\{0,1,...,\alpha^{(j)}-1\}$ ($\alpha^{(j)}\geq 2$) станів, кожна структура БМ представляється N множинами предків ($\Pi^{(1)},...,\Pi^{(N)}$), тобто для кожної вершини $j=1,...,N$, $\Pi^{(j)}$ – це множина батьківських вершин, така що $\Pi^{(j)}\subseteq\{X^{(1)},...,X^{(N)}\}\setminus\{X^{(j)}\}$ (вершина не може бути батьком самої себе , тобто петлі в графі відсутні). Тоді ОМД структури $g\in G$ при заданій послідовності з n спостережень $x^n=d_1d_2...d_n$ обчислюється за формулою:

$$L(g,x^n)=H(g,x^n)+\frac{k(g)}{2}\cdot\log(n), \qquad (2.3)$$

де $k(g)$ – кількість незалежних умовних ймовірностей у структурі БМ g, а $H(g,x^n)$ – це емпірична ентропія.

$$H(g,x^n)=\sum_{j\in J}H(j,g,x^n), \tag{2.4}$$

$$k(g)=\sum_{j\in J}k(j,g), \tag{2.5}$$

де ОМД j-ї вершини обчислюється за формулою:

$$L(j,g,x^n)=H(j,g,x^n)+\frac{k(j,g)}{2}\cdot\log(n), \tag{2.6}$$

де $k(j,g)$ – кількість незалежних умовних імовірностей j-ї вершини:

$$k(j,g)=(\alpha^{(j)}-1)\cdot\prod_{k\in\phi(j)}\alpha^k, \tag{2.7}$$

де $\phi(j)\subseteq\{1,..,j-1,j+1,...,N\}$, ця така множина для якої $\Pi^{(j)}=\{X^{(k)}:k\in\phi^{(j)}\}$.

Емпірична ентропія j-ї вершини обчислюється за формулою:

$$H(j,g,x^n)=\sum_{s\in S(j,g)}\sum_{q\in A^{(j)}}-n[q,s,j,g]\cdot\log\frac{n[q,s,j,g]}{n[s,j,g]}, \tag{2.8}$$

де

$$n(s,j,g)=\sum_{i=1}^{n}I(\pi_i^{(j)}=s), \tag{2.9}$$

$$n[q,s,j,g]=\sum_{i=1}^{n}I(x_i=q,\pi_i^{(j)}=s), \tag{2.10}$$

де $\pi^{(j)} = \Pi^{(j)}$, а $X^{(k)} = x^{(k)}, \forall k \in \phi^{(j)}$, функція $I(E) = 1$, якщо предикат $E = true$, інакше $I(E) = 0$.

Простий алгоритм навчання БМ із використанням ОМД виглядає наступним чином: циклічно робиться перебір усіх можливих нециклічних структур мереж. В g^* зберігається оптимальна структура мережі. Оптимальною структурою буде та, у якої буде найменше значення функції $L(g, x^n)$.

Алгоритм побудови БМ із використанням ОМД

1. $g^* \leftarrow g_0 (\in G)$;

2. для $\forall g \in G - \{g_0\}$ якщо $L(g, x^n) < L(g^*, x^n)$ то $g^* \leftarrow g$;

3. в якості рішення на вихід подається g^*.

2.2.3 Порівняння ефективності оціночних функцій

Для побудови структури БМ найчастіше застосовують методи із використанням функцій КГ або ОМД, а також різні модифікації цих функцій, наприклад МФКГ. Результати обчислювальних експериментів показали, що на коротких навчальних вибірках, до 170 записів, і мережах, що складаються не більш ніж з 10 вершин, функція КГ працює швидше в порівнянні з функцією МФКГ й ОМД. Але функції МФКГ й ОМД, на відміну від КГ, працюють із навчальними вибірками будь-якого розміру. Також з'ясовано, що методи побудови БМ із використанням функцій КГ та її модифікацій, виконують перенавчання БМ, тобто часто такі мережі містять зайві дуги.

На рисунках 2.2 та 2.3 показано графіки витраченого часу на побудову БМ з використанням МФКГ та ОМД. Для задавання порядку додавання дуг у БМ використовуються ЗВІ. При побудові БМ в якості вибірки навчальних даних використано генетичні дані, які складаються з 600 навчальних записів. Як можна побачити з рисунків, для побудови БМ, що складаються з більш ніж з

30 вершин, алгоритм із використанням ОМД працює швидше у порівнянні із МФКГ.

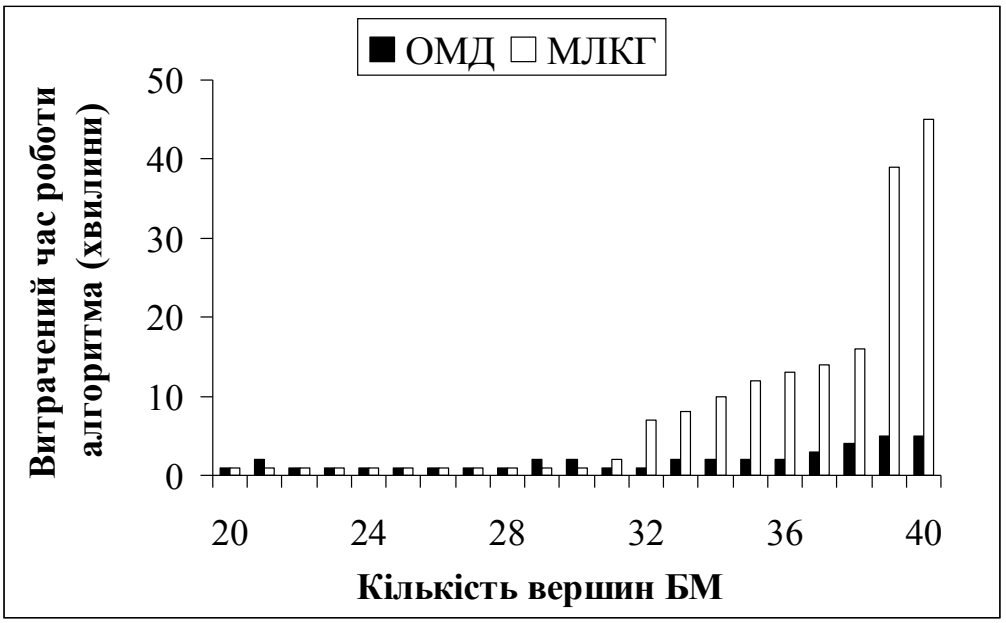


Рисунок 2.2 – Діаграма витрат часу на алгоритми з використанням функцій МФКГ та ОМД

Рисунок 2.3 – Графік витрат часу при використанні алгоритму із застосуванням функції ОМД

2.3 Алгоритм евристичного методу побудови байєсівських мереж за навчальними даними

Вхідні дані. Множина навчальних даних $D = \{d_1,...,d_n\}$, $d_i = \{x_i^{(1)} x_i^{(2)} ... x_i^{(N)}\}$ (нижній індекс – номер спостереження, а верхній – номер змінної), n – кількість спостережень; N – кількість вершин (змінних).

Перший етап.

Для всіх пар вершин обчислюються значення взаємної інформації $Set\_MI = \left\{ MI(x^i, x^j);\ \forall i, j \right\}$. Після цього елементи множини Set_MI впорядковують за зменшенням

$$Set\_MI = \{MI(x^{m_1}, x^{m_2}), MI(x^{m_3}, x^{m_4}), MI(x^{m_5}, x^{m_6}),...\}.$$

Другий етап.

Крок 1. З множини значень взаємної інформації Set_MI обирають перші два максимальних значення $MI(x^{m_1}, x^{m_2})$ та $MI(x^{m_3}, x^{m_4})$. За цими значеннями $MI(x^{m_1}, x^{m_2})$ та $MI(x^{m_3}, x^{m_4})$ будується множина моделей G:

{ $(m_1 \to m_2;\ m_3 \to m_4)$, $(m_1 \to m_2; m_3 \leftarrow m_4)$, $(m_1 \leftarrow m_2; m_3 \leftarrow m_4)$, $(m_1 \leftarrow m_2; m_3 \to m_4)$, $(m_1 \leftarrow m_2;$ m_3 не залежить від $m_4)$, $(m_1 \to m_2;$ m_3 не залежить від $m_4)$, (m_1 не залежить від $m_2;$ $m_3 \to m_4)$, (m_1 не залежить від $m_2;$ $m_3 \leftarrow m_4)$, (m_1 не залежить від $m_2;$ m_3 не залежить від $m_4)$}.

Запис виду $m_i \to m_j$ означає, що вершина x^{m_i} є предком вершини x^{m_j}.

Крок 2. Здійснюється пошук, серед всіх моделей множини G. В параметрі g^* зберігається оптимальна структура БМ. Оптимальною

структурою буде та, у якої буде найменше значення функції $L(g, x^n)$. Де $L(g, x^n)$ – ОМД структури моделі при заданій послідовності з n спостережень $x^n = d_1 d_2 ... d_n$.

1. $g^* \leftarrow g_0 (\in G)$;

2. для $\forall g \in G - \{g_0\}$ якщо $L(g, x^n) < L(g^*, x^n)$ то $g^* \leftarrow g$;

3. на вихід в якості рішення подається g^*.

Крок 3. Після того як знайдена оптимальна структура (структури) g^* з G, з множини значень взаємної інформації Set_MI обирають наступне максимальне значення $MI(x^{i\_next_i}, x^{j\_next})$. За отриманим значенням $MI(x^{i\_next_i}, x^{j\_next})$ і структурою (структурам) g^* будується множина моделей G у вигляді: { (g^*; $i\_next \rightarrow j\_next$), ($g^*$; $i\_next \leftarrow j\_next$), (g^*; i_next не залежить від j_next)}. Після цього виконується крок 2.

Умова завершення алгоритму.

Евристичний метод буде виконуватися до тих пір, поки не буде проаналізовано певне число елементів множини або всі $\frac{N \cdot (N-1)}{2}$ елементи множини Set_MI. Як показує практика, у більшості випадків нема рації виконувати аналіз більш ніж половини (тобто $\frac{N \cdot (N-1)}{4}$) елементи множини Set_MI.

Вихідні дані алгоритму – оптимальна структура (структури) g^*.

2.3.1 Приклад застосування алгоритму евристичного методу

В якості прикладу розглянемо загальновідому мережу "Азія" яка складається з восьми вершин. У таблиці 2.3 наведені ЗВІ між всіма вершинами мережі розташовані за зменшенням (перший етап алгоритму).

Таблиця 2.3 – ЗВІ між всіма вершинами БМ "Азія"

№	*MI*	*i*	*j*		№	*MI*	*i*	*j*
1	0,251	7	8		15	0,001227	3	5
2	0,136	2	4		16	0,000851	1	6
3	0,125	4	6		17	0,000508	2	7
4	0,096	2	6		18	0,000381	3	7
5	0,048	1	7		19	0,000266	4	5
6	0,036	3	4		20	0,000197	1	5
7	0,025	3	6		21	0,000128	4	7
8	0,0245	1	8		22	0,00012271	2	5
9	0,0132	4	8		23	0,00006475	5	6
10	0,0101	2	8		24	0,00003950	2	3
11	0,0051	6	8		25	0,00003249	5	7
12	0,0031	1	2		26	0,00001725	5	8
13	0,0028	3	8		27	0,00000303	1	3
14	0,0022	1	4		28	0,00000074	6	7

Побудова виконувалася вибіркою з 7000 навчальних спостережень. На рисунку 2.4 показана структура оригінальної БМ, на основі якої генерувалися значення.

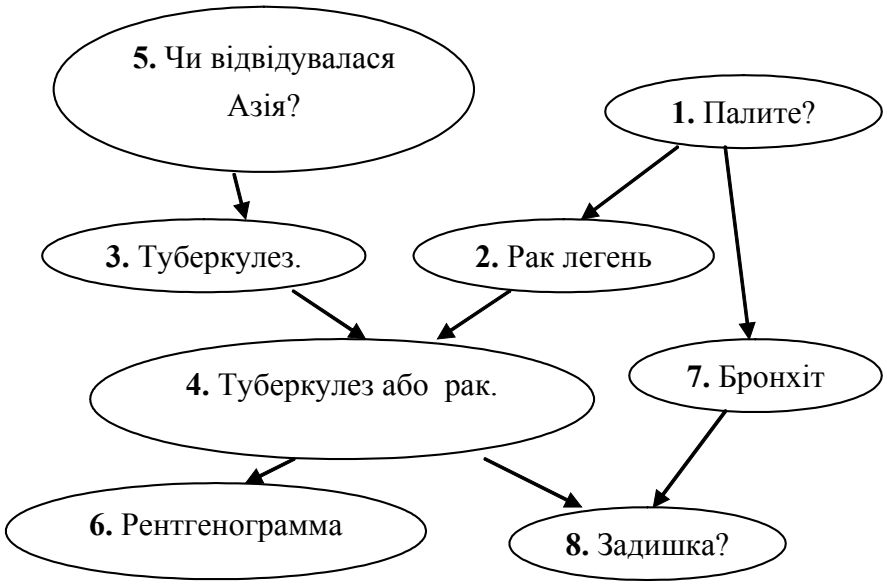


Рисунок 2.4 – Оригінальна мережа "Азія"

На 1-й ітерації по перших двох рядках *MI(7,8)* та *MI(2,4)*, відсортованої матриці *MI*, будується множина моделей з 9 структур, після перебору цієї

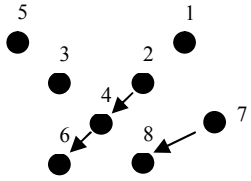
множини з’ясовується що мінімальне значення функції ОМД відповідає двом структурам (рисунок 2.5).

а *L=17820,9586* б

Рисунок 2.5 – Структури БМ отримані на 1-й ітерації

На 2-й ітерації за отриманими на 1-й ітерації структурах і *MI(4,6)* будується множина моделей з 6 структур, з яких мінімальне значення функції ОМД приймає мережа зображена на рисунку 2.6.

На 3-й ітерації за отриманою на 2-й ітерації структурою та *MI(2,6)* будується множина моделей з 3 структур. З цих трьох структур мінімальне значення функції ОМД приймає та ж сама структура зображена на рисунку 2.6.



L=16918,4958

Рисунок 2.6 – Структура БМ отримана на 2-й та 3-й ітераціях

На 4-й ітерації за отриманою на 3-й ітерації структурі (рисунок 2.6) і *MI(1,7)* будується множина моделей з 3 структур, з яких мінімальне значення функції ОМД приймають мережі зображені на рисунку 2.7.

а *L=16570,5342* б

Рисунок 2.7 – Структури БМ отримані на 4-й ітерації

На 5-й ітерації за структурами отриманими на 4-й ітерації (рисунок 2.7) і *MI(3,4)* будується множина моделей з 6 структур, з яких мінімальне значення функції ОМД приймають мережі зображені на рисунку 2.8.

На 6-й ітерації за структурами отриманими на 5-й ітерації і *MI(3,6)* будується множина моделей з 6 структур. В результаті одержані ті ж самі структури що і на 5-й ітерації (рисунок 2.8).

На 7-й ітерації за структурами отриманими на 6-й ітерації і *MI(1,8)* будується множина моделей з 6 структур. Результат збігається з результатом попередніх двох ітерацій (рисунок 2.8).

а *L=16312,9052* б

Рисунок 2.8 – Структури БМ отримані на 5-й, 6-й та 7-й ітераціях

На 8-й ітерації за структурами отриманими на 7-й ітерації (рисунок 2.8) і *MI(4,8)* будується множина моделей з 6 структур, з яких мінімальне значення функції ОМД приймають мережі зображені на рисунку 32.9.

На 9-й ітерації за структурами отриманими на 8-й ітерації і *MI(2,8)* будується множина моделей з 6 структур. В результаті одержані ті ж самі структури що і на 8-й ітерації (рисунок 2.9).

На 10-й ітерації за структурами отриманими на 9-й ітерації і *MI(6,8)* будується множина моделей з 6 структур. Результат збігається з результатом попередніх двох ітерацій (рисунок 2.9).

а L= 16153,4686 б

Рисунок 2.9 – Структури БМ отримані на 8-й, 9-й та 10-й ітераціях

На 11-й ітерації за структурами отриманими на 10-й ітерації і *MI(1,2)* будується множина моделей з 6 структур, з яких мінімальне значення функції ОМД приймають мережі зображені на рисунку 2.10.

На 12-й ітерації за структурами отриманими на 11-й ітерації і *MI(3,8)* будується множина моделей з 6 структур. В результаті одержані ті ж самі структури що і на 11-й ітерації (рисунок 2.10).

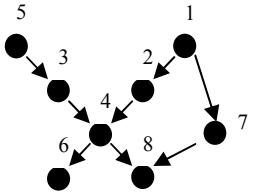
На 13-й ітерації по структурам отриманим на 12-й ітерації і *MI(1,4)* будується множина моделей з 6 структур. Результат збігається з результатом попередніх двох ітерацій.

а *L=16143,5518* б

Рисунок 2.10 – Структури БМ отримані на 11-й, 12-й та 13-й ітераціях

На 14-й ітерації за отриманими на 13-й ітерації структурами (рисунок 2.10) і *MI(3,5)* будується множина моделей з 6 структур, з яких мінімальне значення функції ОМД приймає мережа зображена на рисунку 2.11.

З 15-й по 27-й ітерації ніяких змін в структурі не відбувається тобто результат збігається з результатом 14-й ітерації (рисунок 2.11).



L=16139,1545

Рисунок 2.11 – Структура БМ отримана з 14 по 27 ітерації

Для побудови БМ "Азія" при простому аналізі всіх можливих нециклічних структур буде потрібно виконати оцінку 783 702 329 343 моделей. Тоді як евристичний метод на 27-ми ітераціях алгоритму виконує аналіз усього лише 120 структур, причому вже на 14-й ітерації, після аналізу 81 структури, алгоритм видає структуру, що повністю збігається з оригінальною мережею "Азія". Тобто наступні 13 ітерацій методу не роблять ніяких змін, тому що оптимальна структура вже знайдена на 14 ітерації.

2.3.2 Аналіз експериментальних результатів

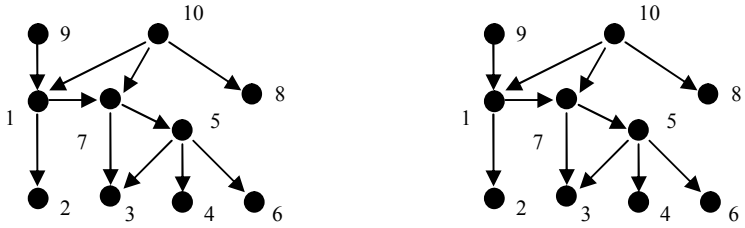
Виконано шість обчислювальних експериментів. У кожному експерименті евристичним методом проводилося навчання мережі з 10 вершин вибіркою з 2000 навчальних спостережень. Для оцінювання якості навчання використовується структурна різниця між побудованою і оригінальною БМ. У таблиці 2.4 наведені результати шести обчислювальних експериментів. Для кожного експерименту було виконано 44 ітерації навчання.

Як можна бачити з таблиці 2.4, у двох із шести обчислювальних експериментах №2 та №6 (рисунки 2.13 та 2.17) побудована мережа повністю збіглася з оригінальною БМ. У двох із шести експериментах №3 та №4 (рисунки 2.14 та 2.15) помилка навчання, тобто структурна різниця між побудованою та оригінальною моделями дорівнює 3 та 4, для мережі з 10 вершин така помилка являється прийнятною. Значні помилки навчання отримані в експериментах №1 та №5 (рисунки 2.12 та 2.16). Однак для побудови мережі був виконаний аналіз усього лише 513 та 550 моделей відповідно, на всіх 44 ітерація, у той час як при простому переборі, тобто аналіз

“в чоло”, всіх можливих нециклічних моделей потрібно було б проаналізувати 4 175 098 976 430 598 100 моделей.

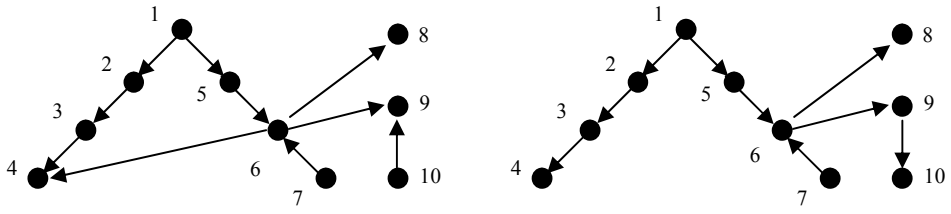
а б

Рисунок 2.12 – Експеримент №1 (а) побудована (б) оригінальна БМ



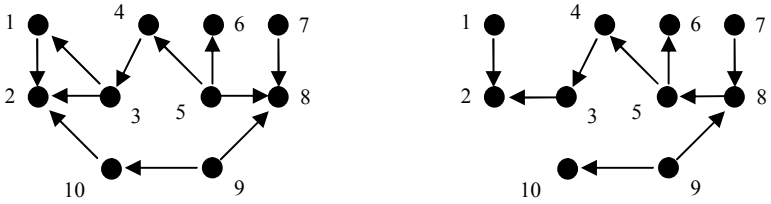
а б

Рисунок 2.13 – Експеримент №2 (а) побудована (б) оригінальна БМ



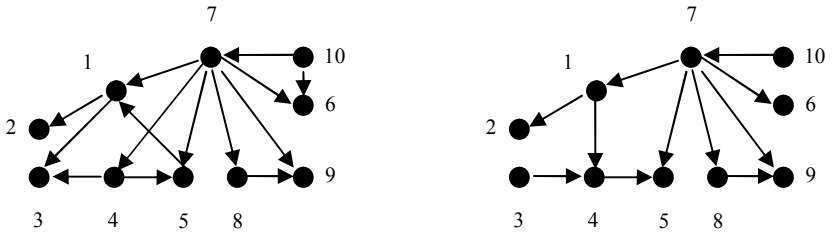
а б

Рисунок 2.14 – Експеримент №3 (а) побудована (б) оригінальна БМ



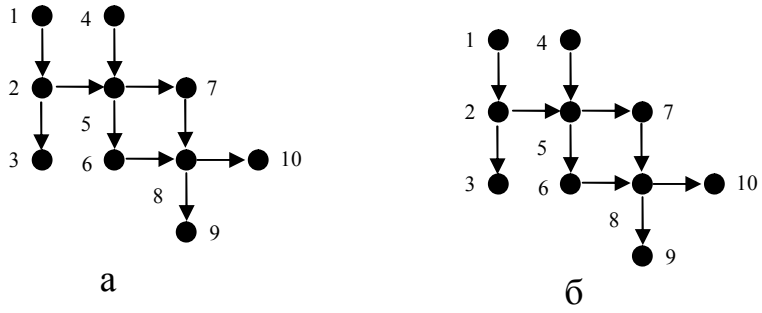
а б

Рисунок 2.15 – Експеримент №4 (а) побудована (б) оригінальна БМ



а б

Рисунок 2.16 – Експеримент №5 (а) побудована (б) оригінальна структура БМ



а б

Рисунок 2.17 – Експеримент №6 (а) побудована (б) оригінальна структура БМ

Таблиця 3.12

Таблиця 2.4 – Результати шести обчислювальних експериментів

Номер обчислювального експерименту	№1	№2	№3	№4	№5	№6
Номер рисунка який відповідає обчислювальному експерименту	Рис. 2.12	Рис. 2.13	Рис. 2.14	Рис. 2.15	Рис. 2.16	Рис. 2.17
Загальна кількість моделей, проаналізованих евристичним методом на всіх ітераціях	513	178	415	282	550	329
Зайві дуги	1	0	1	2	4	0
Відсутні дуги	0	0	0	0	1	0
Реверсовані дуги	3	0	1	1	1	0
Структурна різниця між побудованою та оригінальною БМ	8	0	3	4	7	0

2.3.3 Оцінювання якості побудови байєсівських мереж

Для оцінювання якості побудови БМ можна використовувати облік кількості зайвих, відсутніх і реверсованих дуг у побудованій БМ при порівнянні з оригінальною БМ (мережею по якій генерувалися дані). А в якості міри оцінювання помилки побудови можна використовувати структурну різницю (structure difference) або перехресну ентропію (cross entropy), між побудованою БМ і оригінальною БМ.

Для обчислення структурної різниці використовують формулу симетричної різниці структур:

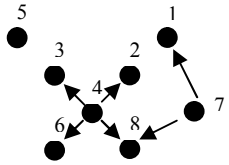
$$\begin{aligned}\delta &= \sum_{i=1}^{n} \delta_i = \sum_{i=1}^{n} card\left(\Pi^{(i)}(B) \Delta \Pi^{(i)}(A)\right) = \\ &= \sum_{i=1}^{n} card\left(\left(\Pi^{(i)}(B) \setminus \Pi^{(i)}(A)\right) \cup \left(\Pi^{(i)}(A) \setminus \Pi^{(i)}(B)\right)\right),\end{aligned} \tag{2.11}$$

де B – побудована БМ, A – оригінальна БМ, n – кількість вершин мережі, $\Pi^{(i)}(B)$ – множина предків i–ї вершини побудованої мережі B, $\Pi^{(i)}(A)$ – множина предків i–ї вершини оригінальної мережі A, $card(\xi)$ – потужність скінченої множини ξ, що визначається як кількість елементів які належать множині ξ.

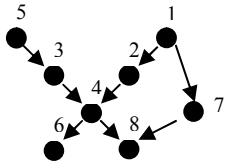
Перехресна ентропія – це відстань між розподілом побудованої БМ і оригінальною БМ. Нехай $p(v)$ – спільний розподіл оригінальної БМ, а $q(v)$ – спільний розподіл побудованої БМ. Тоді перехресна ентропія обчислюється як:

$$\begin{aligned}H(p,q) &= \sum_{v} p(v) \cdot \log \frac{p(v)}{q(v)} = \\ &= \sum_{j \in J} \sum_{s \in S(j,g)} \sum_{a \in A^{(j)}} p(X^{(j)} = a \left| \Pi^{(j)} = s \right.) \cdot \log \frac{p(X^{(j)} = a \left| \Pi^{(j)} = s \right.)}{q(X^{(j)} = a \left| \Pi^{(j)} = s \right.)} .\end{aligned} \tag{2.12}$$

Наприклад, структурна різниця між мережами, зображеними на рисунку 2.18, буде дорівнювати восьми. Проміжні значення які застосовуються для обчислення структурної різниці наведені в таблиці 2.5.



а



б

Рисунок 2.18 – Структури БМ для прикладу обчислення структурної різниці

Таблиця 3.13

Таблиця 2.5 – Проміжні розрахункові значення структурної різниці між мережами, зображеними на рисунку 2.18

Номер вершини	Множина батьків вершини структури		$\Pi^{(i)}(B)\Delta\Pi^{(i)}(A)$	$card(\Pi^{(i)}(A)\Delta \Delta\Pi^{(i)}(B))$
	рис. 3.18.а	рис. 3.18.б		
1	$\Pi^{(1)}=\{7\}$	$\Pi^{(1)}=\{\ \}$	{7}	1
2	$\Pi^{(2)}=\{4\}$	$\Pi^{(2)}=\{1\}$	{1,4}	2
3	$\Pi^{(3)}=\{4\}$	$\Pi^{(3)}=\{5\}$	{4,5}	2
4	$\Pi^{(4)}=\{\ \}$	$\Pi^{(4)}=\{2,3\}$	{2,3}	2
5	$\Pi^{(5)}=\{\ \}$	$\Pi^{(5)}=\{\ \}$	{ }	0
6	$\Pi^{(6)}=\{4\}$	$\Pi^{(6)}=\{4\}$	{ }	0
7	$\Pi^{(7)}=\{\ \}$	$\Pi^{(7)}=\{1\}$	{1}	1
8	$\Pi^{(8)}=\{4,7\}$	$\Pi^{(8)}=\{4,7\}$	{ }	0

3 СИСТЕМА ПІДТРИМКИ ПРИЙНЯТТЯ РІШЕНЬ ДЛЯ ІНТЕЛЕКТУАЛЬНОГО АНАЛІЗУ ДАНИХ НА ОСНОВІ БАЙЄСІВСЬКИХ МЕРЕЖ

Система підтримки прийняття рішень – це комп’ютерна система, що спрямована на розв’язання таких задач інтелектуального аналізу даних, як аналіз зв’язків, класифікація, кластеризація, прогнозування та візуалізація. Ці задачі розв’язуються без залучення експертів – за навчальними даними. Головне завдання цієї СППР полягає у наданні допомоги особам при прийнятті рішення (ОПР). Надалі запропоновану СППР назвемо системою підтримки прийняття рішень для інтелектуального аналізу даних на основі байєсівських мереж (СППР ІАД БМ).

СППР ІАД БМ має гнучку архітектуру, що передбачає введення нових алгоритмічних модулів для реалізації будь-яких методів побудови БМ та ймовірнісного висновку, або навіть інших методів ІАД.

За рівнем користувача СППР ІАД БМ належить до активної СППР, допомагає не тільки приймати рішення шляхом виконання складних обчислювальних операцій, але й допомагає ОПР обирати рішення, користуючись значеннями ймовірностей виникнення тієї чи іншої події, [176].

За технічним рівнем СППР ІАД БМ належить до настільної СППР, тобто ця система обслуговує лише один комп'ютер користувача (не ставилася мета створення мережевої СППР).

Згідно класифікації систем підтримки прийняття рішень, розробленої Спрагом, СППР ІАД БМ відноситься до СППР-генератора, тому що кожна БМ, яка побудована на основі навчальних даних, представляє собою причинно-наслідкову мережу, що може розглядатись як окрема СППР [178].

СППР ІАД БМ дозволяє без залучення експертів будувати моделі у вигляді БМ за навчальними даними та вирішувати задачі класифікації, кластеризації, а також прогнозування шляхом формування відповідного ймовірнісного висновку.

Архітектура будь-якої СППР у спрощеному вигляді має структуру, показану на рисунку 3.1, або близьку до цієї структури.

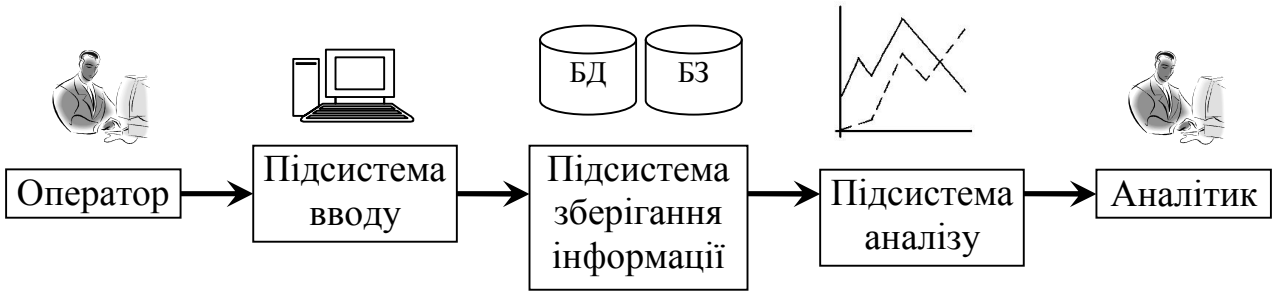


Рисунок 3.1 – Типова структура СППР

3.1 Структура та принципи функціонування системи підтримки прийняття рішень

На рисунку 3.2 наведена структурна схема розробленої СППР ІАД БМ, що призначена для накопичення даних, побудови ймовірнісних моделей у вигляді БМ та формування ймовірнісного висновку відносно заданого того чи іншого стану процесу. Система складається з трьох основних підсистем і передбачає модульно-блочну побудову.

Пристрої вводу-виводу надають користувачеві можливість завантажувати дані в СППР. Для цього підсистема вводу-виводу функціонально зв’язана з підсистемою інтерфейсу користувача.

Підсистема інтерфейсу користувача використовується для здійснення зв’язку між користувачами СППР та внутрішніми елементами системи і забезпечує ввід та вивід інформації для ОПР і експертів, а також надає доступ до зовнішніх запам’ятовуючих пристроїв.

Інтерфейс дозволяє операторові вводити інформацію, дані, команди, параметри і запити в систему та отримувати вихідну інформацію в зручному, для сприйняття, вигляді.

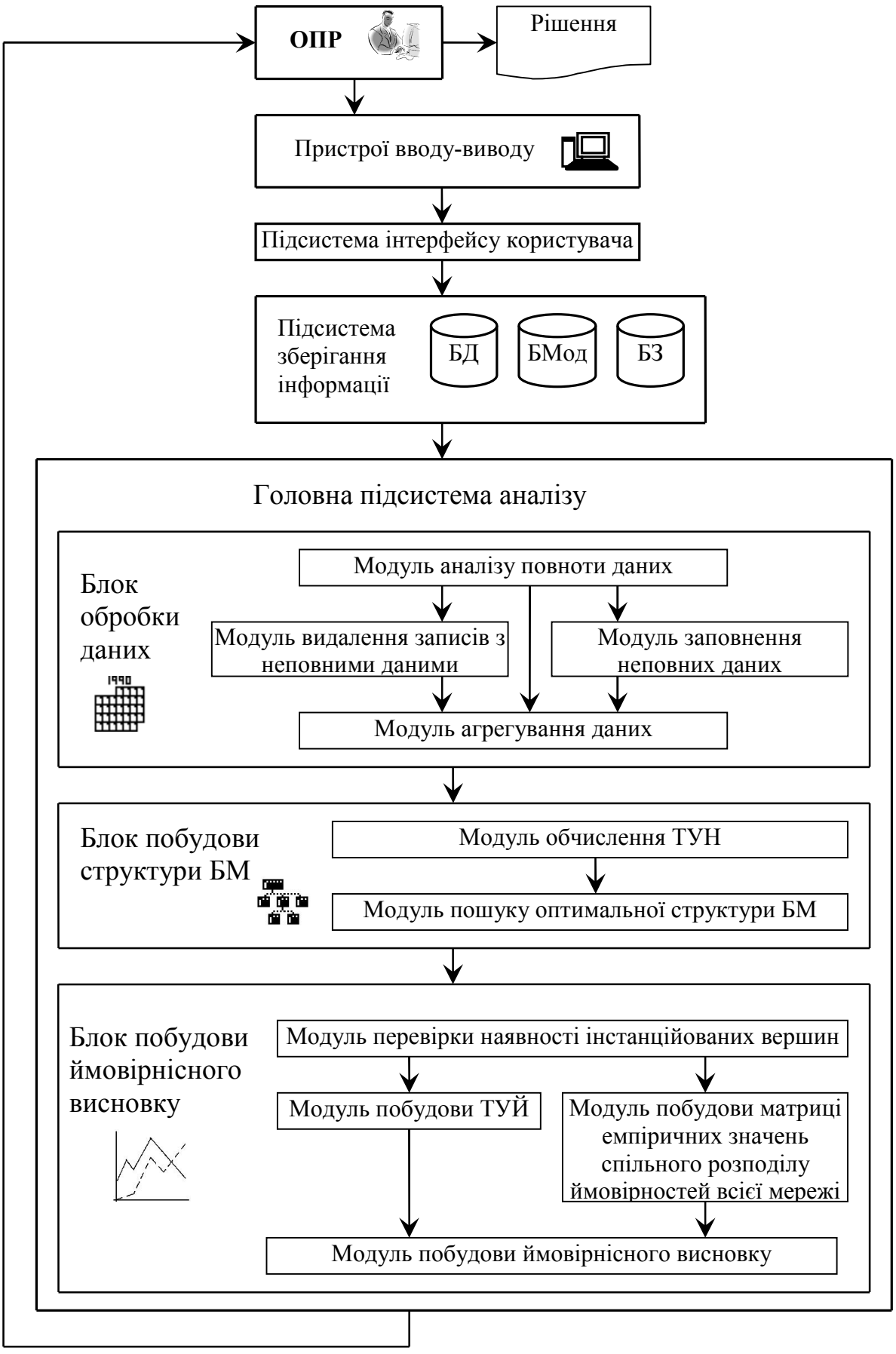


Рисунок 3.2 – Структура СППР ІАД БМ

У зв’язку з тим, що система опрацьовує досить велику кількість різного типу запитів, виникає проблема проектування та реалізації зручного інтерфейсу користувача. Для розв’язання цієї проблеми використано підхід на основі багатовіконного діалогу з використанням контекстної допомоги на кожному етапі прийняття рішення.

Для організації діалогу користувач–система в СППР реалізовані наступні запити мовної системи:

– запити на формування конкретних процедур обробки даних та прогнозування (формулювання вимог);

– запити на вибір та формулювання критеріїв розв’язку задачі;

– запити на виконання задач моделювання і прогнозування;

– запити на форму представлення результатів.

Головною з точки зору обробки даних СППР є система обробки даних та генерації результатів. Вона сприймає коректні запити користувача і виконує наступні завдання:

– читання необхідних даних у вигляді таблиць навчальних даних для побудови моделей;

– вибір методів побудови БМ;

– запуск на виконання модулів обробки даних та прогнозування;

– формування результатів обробки даних та їх зберігання в короткостроковій та довгостроковій пам’яті;

– генерування діагностичних повідомлень.

Підсистема зберігання інформації складається з бази даних (БД), бази моделей (БМод) та бази знань (БЗ), які призначені для накопичення даних, моделей у вигляді структур БМ, таблиць умовних ймовірностей та метаданих.

В рамках розробленої СППР ІАД БМ реалізовано спрощений варіант БД – текстова БД, що представляє собою сукупність навчальних даних у текстовому вигляді. Кожний такий файл включає всі навчальні дані для всіх змінних (атрибутів) процесу, тобто перший стовбець – дані для першої змінної, другий

стовбець – дані для другої змінної і т. ін. Файл даних повинен містити не менше трьох стовпців даних, тобто процес повинен описуватися не менш як трьома змінними. Дані в одному рядку (між стовпцями) повинні бути відділені один від одного пробілом, знаком табуляції або крапкою з комою. В дійсних числах для відокремлення десяткової частини числа від цілої використовується кома, наприклад 10,2.

Змінні можуть містити не тільки кількісні (числові), але і якісні (строкові) значення (наприклад, вербальні оцінки вигляду false/true або yes/no). Але в цьому випадку не допускаються пробіли усередині строкового значення, тому що значення виду "Married more 10" при завантаженні даних призведе до некоректної роботи програми. Тому такі строкові значення, які складаються з декількох слів, повинні бути замінені на такі ж строкові значення, але без пробілів, знаків табуляції та крапок з комою. Наприклад, строкове значення "Married more 10" можна замінити на "Married_more_10" (замість пробілу використається нижній знак підкреслення) або "MarriedMore10" (кожне нове слово починається з великої букви). Редагувати довгі строкові значення можна, наприклад, за допомогою програми Microsoft Excel або текстового редактора.

Головна підсистема аналізу (ГПА) складається з трьох блоків:

- обробки даних;
- побудови структури БМ;
- побудови ймовірнісного висновку.

Ця підсистема використовується для аналізу даних від користувача системою, побудови БМ та ймовірнісного висновку за цими даними. ГПА отримує дані по шині даних від підсистеми зберігання інформації.

Блок обробки даних призначений для перевірки даних, що поступають від ГПА в блок обробки даних на наявність пропусків та викидів (екстремальних значень), а також для здійснення операції агрегування над цими даними. Блок обробки даних складається з чотирьох модулів: (1) аналізу неповних даних; (2) видалення записів з неповними даними; (3) заповнення неповних даних; (4) агрегування даних.

Модуль аналізу повноти даних призначений для перевірки вхідних навчальних даних на наявність пропусків та викидів. Модуль аналізу повноти даних зв'язаний з модулями видалення записів з неповними даними, заповнення неповних даних та агрегування даних. Якщо запис навчального набору даних, який аналізується, містить малу кількість пропусків (відсутніх даних), то цей запис передається для заповнення в модуль заповнення неповних даних. В іншому випадку такий запис видаляється з множини навчальних даних в модулі видалення записів з неповними даними.

Модуль видалення записів з неповними даними призначений для видалення записів з навчальних даних, що мають велику кількість пропусків.

Модуль заповнення неповних даних призначений для заповнення записів з навчальних даних найбільш ймовірним значенням, для цього пропонується використовувати спрощену БМ.

Модуль агрегування даних служить для виконання операції агрегування даних відносно множини навчальних даних. Операція агрегування здійснюється наступним чином. Якщо якийсь вузол з множини навчальних даних приймає числові значення, то ці числові значення замінюються інтервальними оцінками.

Блок побудови структури БМ складається з модулів обчислення ТУН та пошуку оптимальної структури БМ. Даний блок призначений для побудови структури БМ за даними, які передаються з блоку обробки даних.

Модуль обчислення ТУН призначений для обчислення матриці взаємної залежності між вершинами (факторами процесу). В СППР ІАД БМ в якості ТУН використовується значення взаємної інформації.

Модуль пошуку оптимальної структури БМ реалізує цілenапрямлений пошук оптимальної структури БМ. В якості критерію оптимальності використовується функція опису мінімальною довжиною.

Блок побудови ймовірнісного висновку призначений для побудови ймовірнісного висновку по структурі БМ, що надходить з блоку побудови структури БМ. Даний блок складається з модулів перевірки наявності

інстанційованих вершин, побудови таблиці умовних ймовірностей, побудови матриці емпіричних значень спільного розподілу ймовірностей всієї мережі та модуля побудови ймовірнісного висновку.

Якщо множина інстанційованних значень порожня, то СППР передає керування в модуль побудови таблиці умовних ймовірностей. В противному випадку – модулю побудови матриці емпіричних значень спільного розподілу ймовірностей всієї мережі.

Модуль побудови таблиць умовних ймовірностей призначений для побудови ТУЙ кожного вузла БМ за множиною навчальних даних.

Модуль побудови матриці емпіричних значень спільного розподілу ймовірностей всієї мережі призначений для побудови матриці емпіричних значень сумісного розподілу ймовірностей всієї мережі Байєса за навчальними даними.

В залежності від того, від якого модуля прийшов сигнал, виконуються відповідні обчислення.

Якщо сигнал прийшов від модуля побудови таблиць умовних ймовірностей, то застосовується класичний метод прямого поширення ймовірностей по БМ. Спочатку обчислюються ймовірності значень кореневих вершин, тобто вершин, у яких відсутні батьківські вершини. Після цього визначаються ймовірності усіх інших вузлів за формулою Байєса і таблиць умовних ймовірностей.

Якщо сигнал прийшов від модуля побудови матриці емпіричних значень спільного розподілу ймовірностей всієї мережі, то розрахунки виконуються у такий спосіб. На основі інформації про структуру моделі БМ та інстанційовані значення виконується послідовний перебір всіх вузлів БМ. Якщо вузол не інстанційований, то виконується розрахунок усіх можливих станів цього вузла. Після цього виконується послідовний перебір усіх рядків матриці емпіричних значень спільного розподілу ймовірностей всієї БМ і, якщо значення вузла рядка співпадає із значенням інстанційованних вузлів і значеннями аналізованого вузла, то відповідне значення додається до значення ймовірності

відповідного стану аналізованого вузла. Після цього виконується нормування значень ймовірностей станів аналізованого вузла.

3.2 Програмна реалізація системи підтримки прийняття рішень

На основі запропонованої оригінальної архітектури СППР ІАД БМ із використанням мови програмування Delphi реалізована комп'ютерна програма для побудови мереж та ймовірнісного висновку за навчальними даними. Мова програмування Delphi є об'єктно-орієнтованим засобом програмування, що дозволяє представити розроблені алгоритми у вигляді ієрархії класів з наслідуваними властивостями та методами. Це, в свою чергу, дає можливість одержати найбільш ефективну структуру алгоритмів, забезпечує підвищену надійність розроблювального програмного продукту, та зручність його подальшого супроводу.

Для нормальної експлуатації програми необхідна наявність комп'ютера з наступними характеристиками:

– тактова частота процесора не менше 300 Мгц;

– 4 Мбайт вільного місця на жорсткому диску для програмних модулів системи та 4 Мбайт для навчальних даних і формування баз даних (останнє значення не фіксоване і може змінюватися в залежності від умов експлуатації);

– оперативна пам'ять 32 Мбайт і більше;

– операційна система Windows 98/2000/NT/XP;

– пристрій для забезпечення безперебійної роботи комп'ютера для можливості автономної роботи програми;

– клавіатура та комп'ютерна мишка;

– пристрій для запису даних і результатів на оптичні носії для резервного копіювання файлів баз даних.

Програма має можливості будувати структуру та формувати ймовірнісний висновок в байєсівській мережі.

3.2.1 Побудова структури байєсівської мережі

Програма призначена для роботи з текстовими файлами. Кожен такий файл повинен мати не менше трьох стовпців даних. Тобто програма працює не менше ніж з трьома вершинами. Для того щоб програмі вказати ім'я файлу з даними, необхідно вибрати опцію "Побудова структури БМ за даними" в головному меню та натиснути "Завантажити дані з файлу" (рисунок 3.3); після цього в діалоговому вікні необхідно вказати шлях до файлу на жорсткому диску (рисунок 3.4).

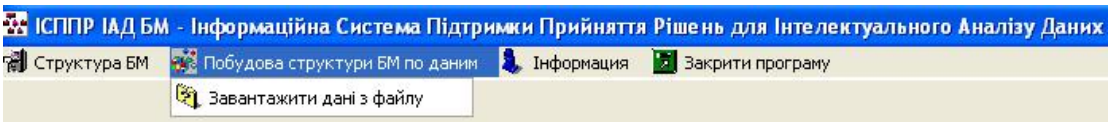


Рисунок 3.3 – Завантаження даних з файлу

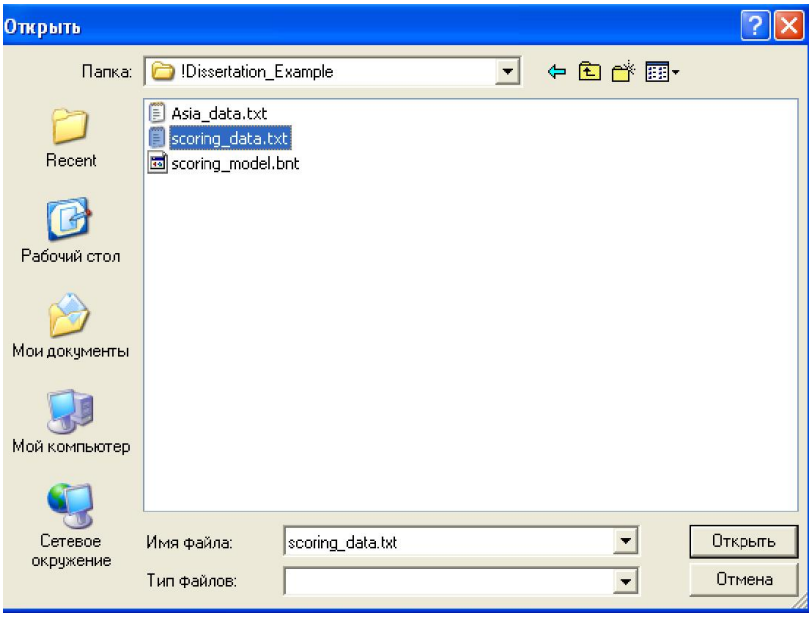
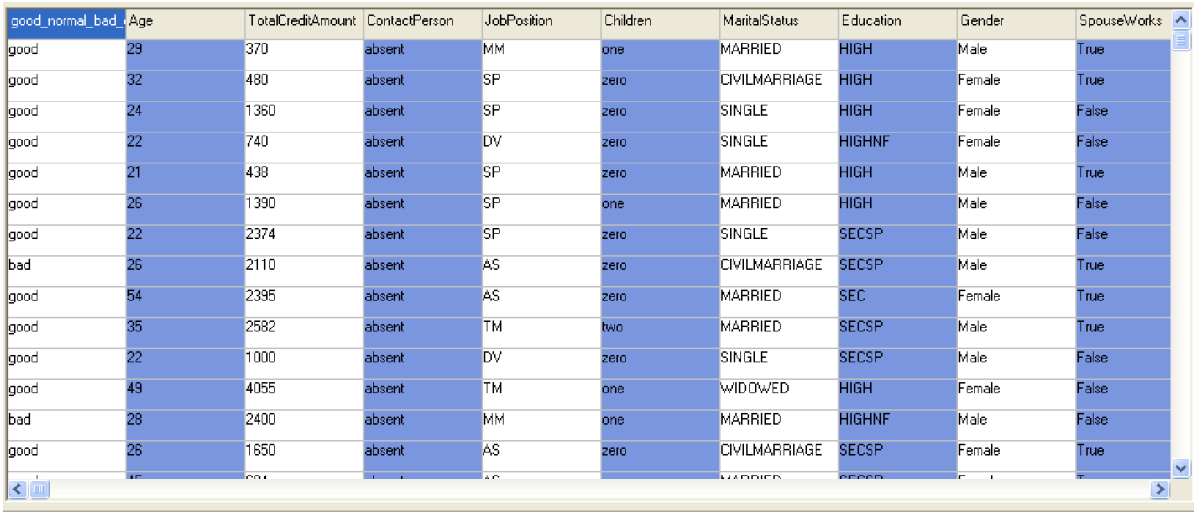


Рисунок 3.4 – Діалогове вікно завантаження даних з файлу

Після того як вказано файл, виконується завантаження даних в таблицю (рисунок 3.5). Стовпці таблиці – фактори (параметри, змінні), за якими буде будуватися топологія БМ. Надалі ці фактори будуть представлені у вигляді вершин байєсівської мережі. Рядки таблиці – це навчальні записи.



good_normal_bad_	Age	TotalCreditAmount	ContactPerson	JobPosition	Children	MaritalStatus	Education	Gender	SpouseWorks
good	29	370	absent	MM	one	MARRIED	HIGH	Male	True
good	32	480	absent	SP	zero	CIVILMARRIAGE	HIGH	Female	True
good	24	1360	absent	SP	zero	SINGLE	HIGH	Female	False
good	22	740	absent	DV	zero	SINGLE	HIGHNF	Female	False
good	21	438	absent	SP	zero	MARRIED	HIGH	Male	True
good	26	1390	absent	SP	one	MARRIED	HIGH	Male	False
good	22	2374	absent	SP	zero	SINGLE	SECSP	Male	False
bad	26	2110	absent	AS	zero	CIVILMARRIAGE	SECSP	Male	True
good	54	2395	absent	AS	zero	MARRIED	SEC	Female	True
good	35	2582	absent	TM	two	MARRIED	SECSP	Male	True
good	22	1000	absent	DV	zero	SINGLE	SECSP	Male	False
good	49	4055	absent	TM	one	WIDOWED	HIGH	Female	False
bad	28	2400	absent	MM	one	MARRIED	HIGHNF	Male	False
good	26	1650	absent	AS	zero	CIVILMARRIAGE	SECSP	Female	True

Рисунок 3.5 – Таблиця з даними

Для оцінювання моделей в програмі реалізовані наступні функції:

– опису мінімальною довжиною;

– Купер-Герсковича;

– модифікація функції Купера-Герсковича.

Для вибору бажаної функції треба поставити крапку навпроти відповідної функції, за замовчуванням використовується ОМД (рисунок 3.6).

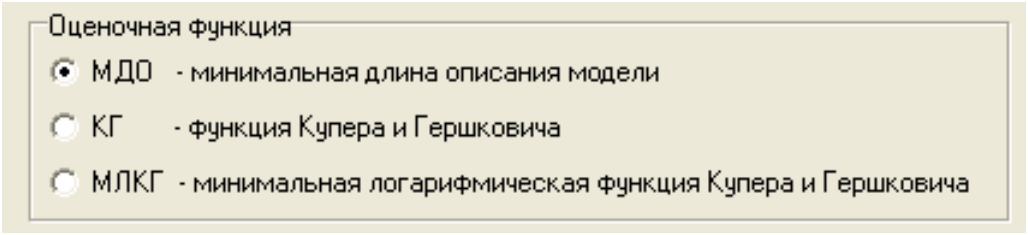


Рисунок 3.6 – Типи функцій оцінювання моделі

Натисканням лівої кнопки миші на стовпець таблиці даних (рисунок 3.5) можна дізнатися про тип фактора – ця інформація відображається зверху вікна. Програма підтримує роботу з числовими (рисунок 3.7) та рядковими (рисунок 3.8) значеннями факторів.

Рисунок 3.7 – Тип змінної – число

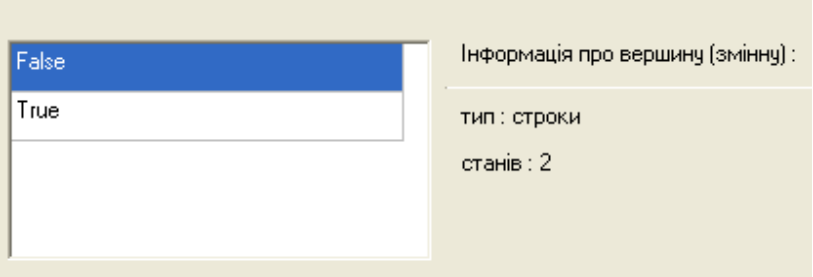


Рисунок 3.8 – Тип змінної – строкова

Для факторів типу „число” повинна бути виконана операція агрегування даних. В полі “Кількість інтервалів розбиття” вказується, на яку кількість інтервалів розбивається множина значень, що приймаються фактором (рисунок 3.9).

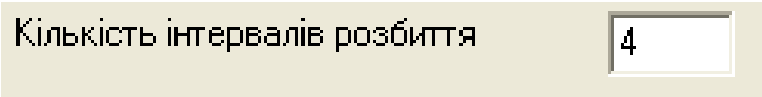


Рисунок 3.9 – Кількість інтервалів розбиття

Кількість інтервалів для числових даних можна не вказувати. У цьому випадку програма за замовчуванням автоматично виконає агрегування всіх числових даних до двох інтервалів.

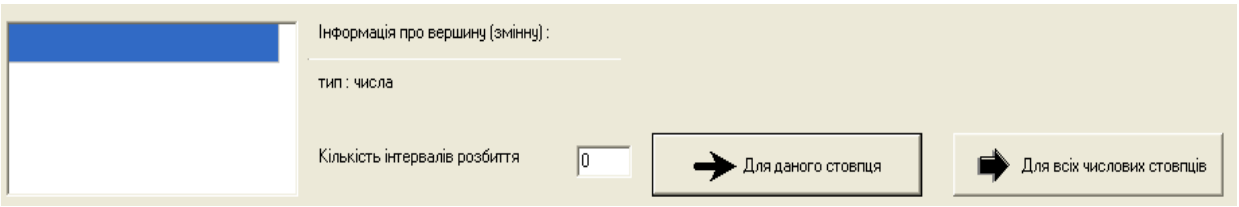


Рисунок 3.10 – Розбиття на інтервали

Вказавши кількість інтервалів розбиття і натиснувши кнопку “Для всіх числових інтервалів” буде виконано агрегування всіх числових даних до заданої кількості інтервалів для всіх числових стовпців даних з таблиці даних.

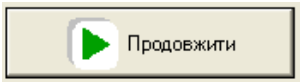


Рисунок 3.11 – Кнопка “Продовжити”

Після того як для числових факторів задана кількість інтервалів розбиття, потрібно натиснути на робочій формі “Завантаження даних” кнопку “Продовжити” (рисунок 3.11). Після завершення операції агрегування даних числові фактори вже складатимуться не з чисел, а з значень станів, які представляють собою інтервальні оцінки. Після цього знову потрібно натиснути кнопку “Продовжити”. З'явиться інформаційне вікно (рисунок 3.12), яке повідомляє яка кількість ітерацій буде виконана.

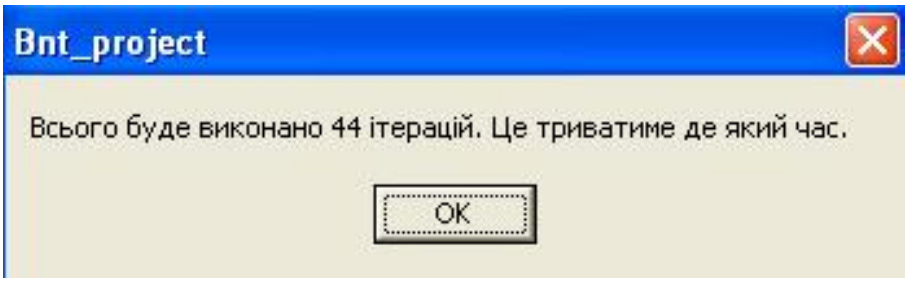
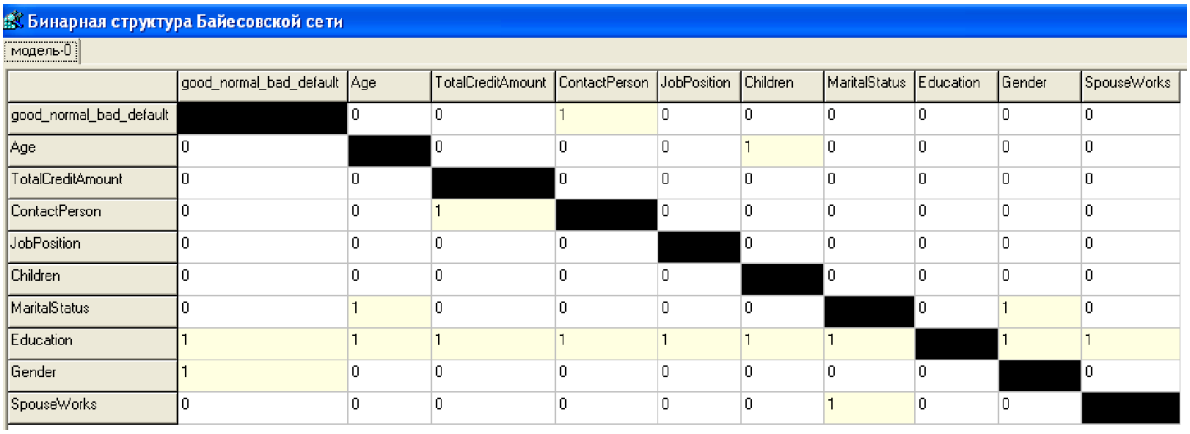


Рисунок 3.12 – Інформаційне вікно програми

Натисніть кнопку “OK”. Програма виконає побудову матриці взаємних значень факторів, а потім за алгоритмом евристичного методу побудує структуру БМ.

Побудована структура БМ відображається на робочій формі програми “Бінарна структура байєсівської мережі” у вигляді бінарної таблиці зв’язків, рисунок 3.13.

Для виведення числових значень функції ОМД, отриманих на ітераціях алгоритму побудови структури, треба натиснути на кнопку “Відкрити звіт”, (рисунок 3.14). В результаті з’явиться форма з проміжними значеннями ОМД, (рисунок 3.15).



	good_normal_bad_default	Age	TotalCreditAmount	ContactPerson	JobPosition	Children	MaritalStatus	Education	Gender	SpouseWorks
good_normal_bad_default		0	0	1	0	0	0	0	0	0
Age	0		0	0	0	1	0	0	0	0
TotalCreditAmount	0	0		0	0	0	0	0	0	0
ContactPerson	0	0	1		0	0	0	0	0	0
JobPosition	0	0	0	0		0	0	0	0	0
Children	0	0	0	0	0		0	0	0	0
MaritalStatus	0	1	0	0	0	0		0	1	0
Education	1	1	1	1	1	1	1		1	1
Gender	1	0	0	0	0	0	0	0		0
SpouseWorks	0	0	0	0	0	0	1	0	0	

Рисунок 3.13 – Бінарна таблиця зв’язків мережі

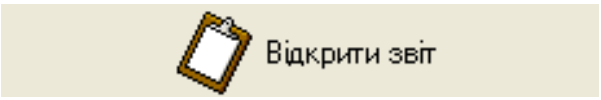


Рисунок 3.14 – Виведення значень функції ОМД

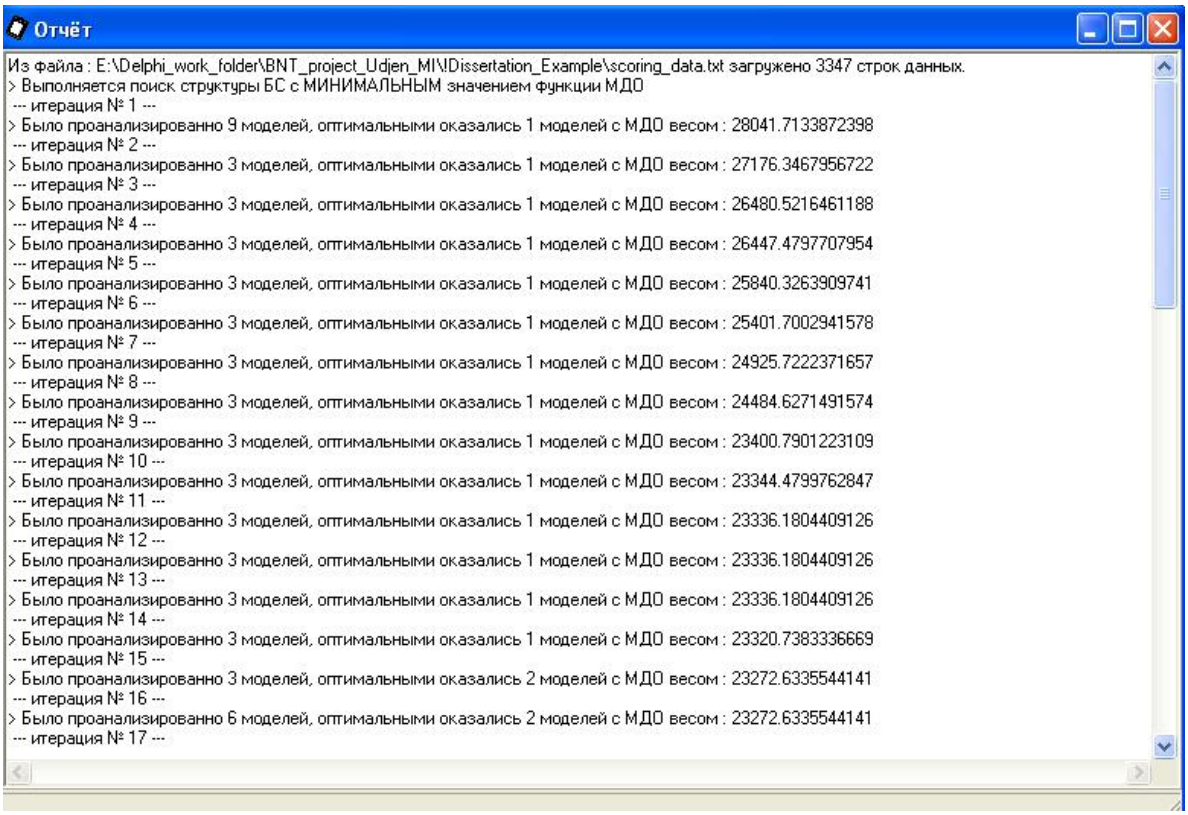


Рисунок 3.15 – Вікно звіту проміжних значень ОМД

Для виводу побудованої БМ у вигляді причинно-наслідкової мережі (рисунок 3.17) треба натиснути кнопку “Відобразити байєсівську мережу” (рисунок 3.16).

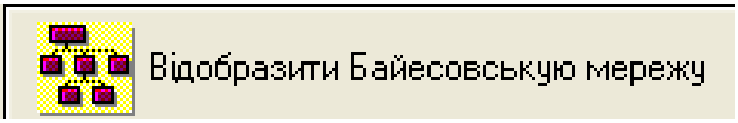


Рисунок 3.16 – Кнопка виводу БМ

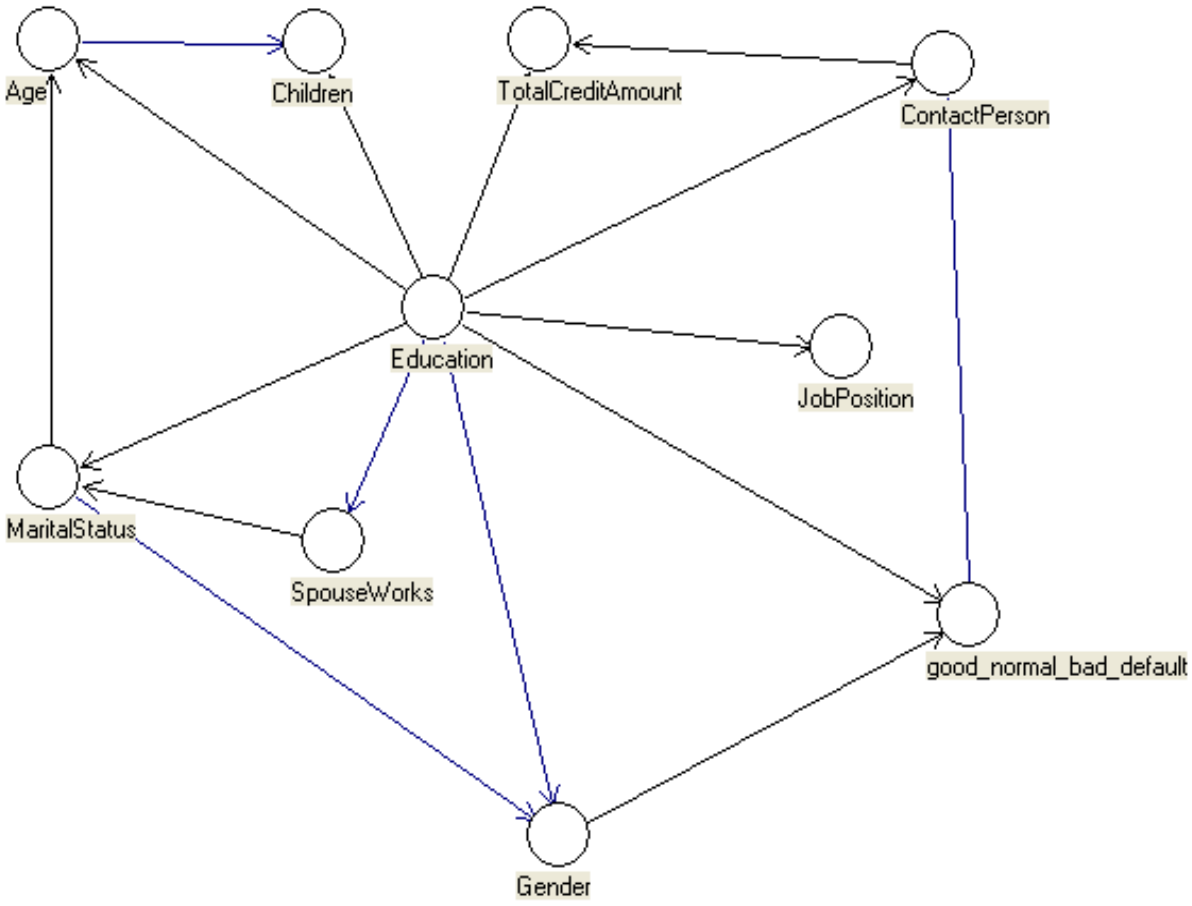


Рисунок 3.17 – Побудована БМ

Натиснувши ліву кнопку миші і утримуючи її, можна переміщати вершини мережі по формі програми. Подвійним натисненням лівої кнопки миші на вершині БМ виводиться форма з ТУЙ мережі (рисунок 3.18).

Натиснувши F5 або кнопку “Таблиця значень всієї мережі” (рисунок 3.19), з таблиці умовних ймовірностей можна вивести форму з таблицею значень спільного розподілу ймовірностей всієї БМ (рисунок 3.20).

Таблица условных вероятностей вершины JobPosition

Таблица значений вероятностей всей сети | Закрыть окно

	JobPosition					
Education	AS	DV	MM	PE	SP	TM
HIGH	0.0305	0.0776	0.1734	0.011	0.4781	0.2295
HIGHNF	0.0693	0.0601	0.1109	0.0416	0.5701	0.1479
PHD	0	0	0.2	0	0.4	0.4
SEC	0.2171	0.0448	0.0651	0.0041	0.5556	0.1133
SECSP	0.1483	0.0586	0.1284	0.0173	0.5336	0.1137

Рисунок 3.18 – Приклад таблиці умовних ймовірностей вершини БМ

Рисунок 3.19 – Кнопка “Таблиця значень всієї мережі”

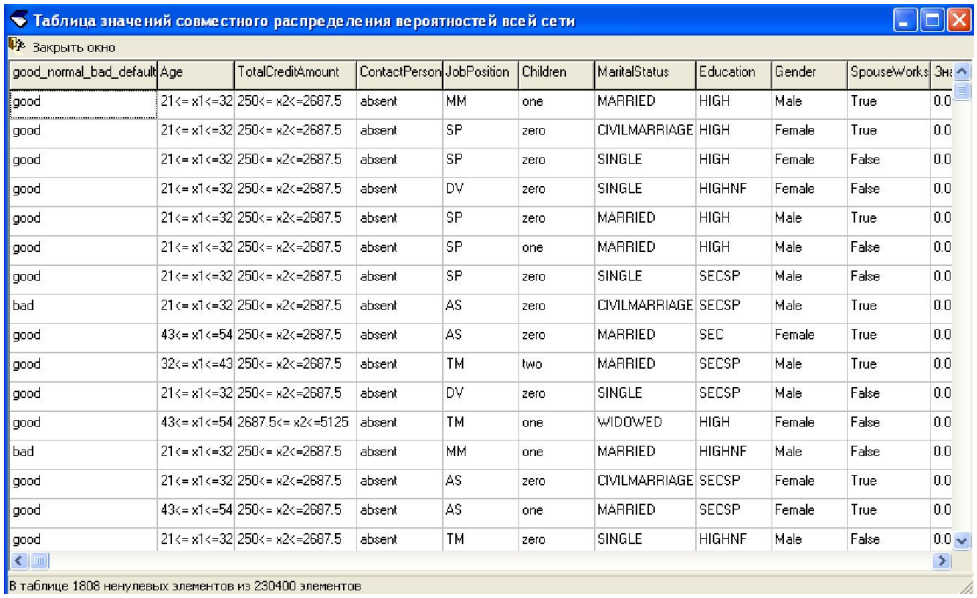


Рисунок 3.20 – Приклад таблиці значень спільного розподілу ймовірностей всієї БМ

Для збереження структури БМ в файл на диск треба натиснути кнопку “Зберегти структуру мережі”. Файл зберігається з розширенням ***.bnt***.

Для переходу в режим побудови ймовірнісного висновку потрібно або закрити програму і запустити її наново, або закрити всі вікна до самої початкової форми.

3.3 Побудова ймовірнісного висновку

Для того щоб програмі вказати ім’я файлу із збереженою структурою БМ, необхідно в головному меню програми вибрати закладку “Структура БМ” та натиснути кнопку “Завантажити файл структури БМ” (рисунок 3.21). Після цього в діалоговому вікні вказати шлях до файлу (рисунок 3.4).

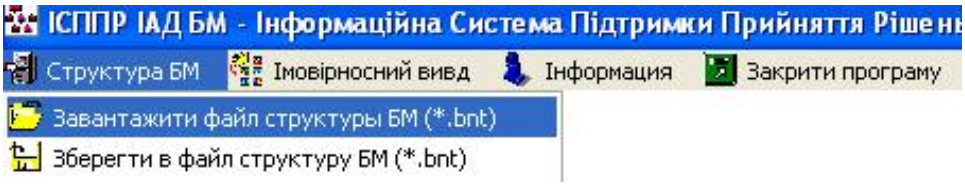


Рисунок 3.21 – Завантаження файлу з структурою БМ

Після того як структура БМ з’явилася на формі програми, можна будувати ймовірнісний висновок. Для цього в головному меню програми треба обрати закладку “Ймовірнісний висновок” та обрати тип представлення у вигляді окремих таблиць або однієї таблиці значень ймовірностей станів вершин (рисунок 3.22).

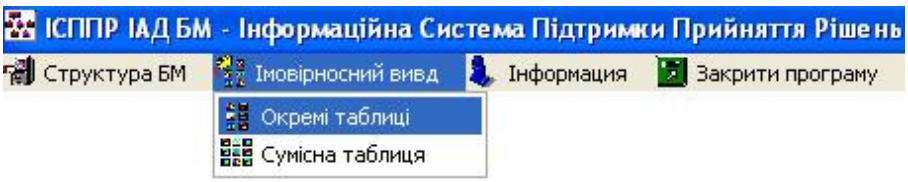


Рисунок 3.22 – Вибір типу представлення ймовірнісного висновку

Для побудови ймовірнісного виводу в БМ за навчальними даними та представлення результатів виводу у вигляді окремих таблиць, треба натиснути

на пункт меню “Окремі таблиці” (рисунок 3.23). Після цього з’явиться форма з таблицями, кожна таблиця – це вершина БМ.

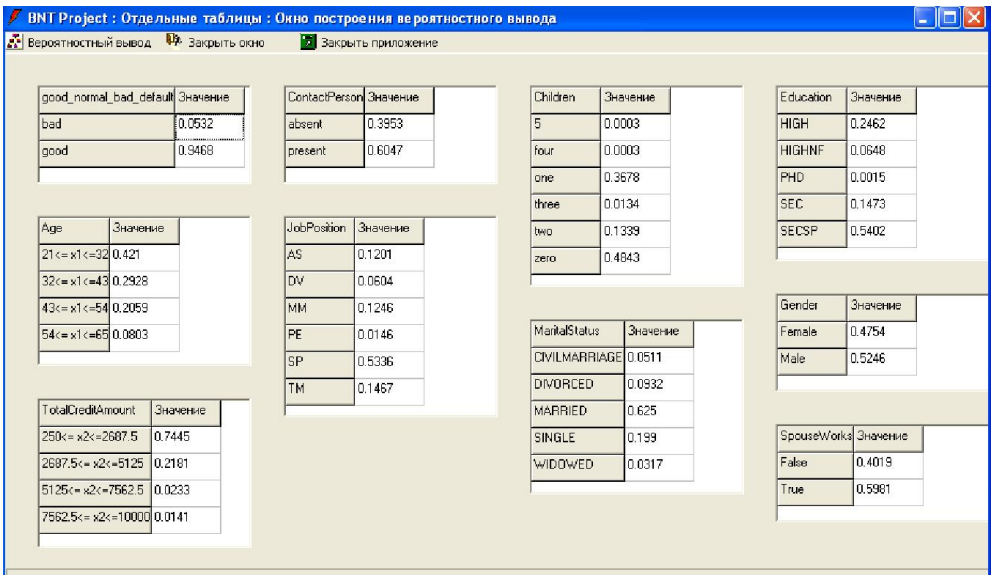


Рисунок 3.23 – Приклад представлення ймовірнісного висновку у вигляді окремих таблиць

Натиснувши ліву кнопку миші і утримуючи її, можна переміщати таблиці по екрану. Подвійним натисканням лівої кнопки миші в таблиці виконується виділення інстанційованого значення вершини (фактора). Після натискування F5 або кнопки “Точний метод” формується ймовірнісний висновок (рисунок 3.24). В програмі реалізовано алгоритм точного методу побудови ймовірнісного висновку за навчальними даними.

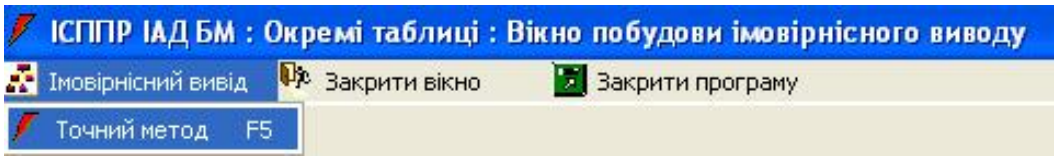
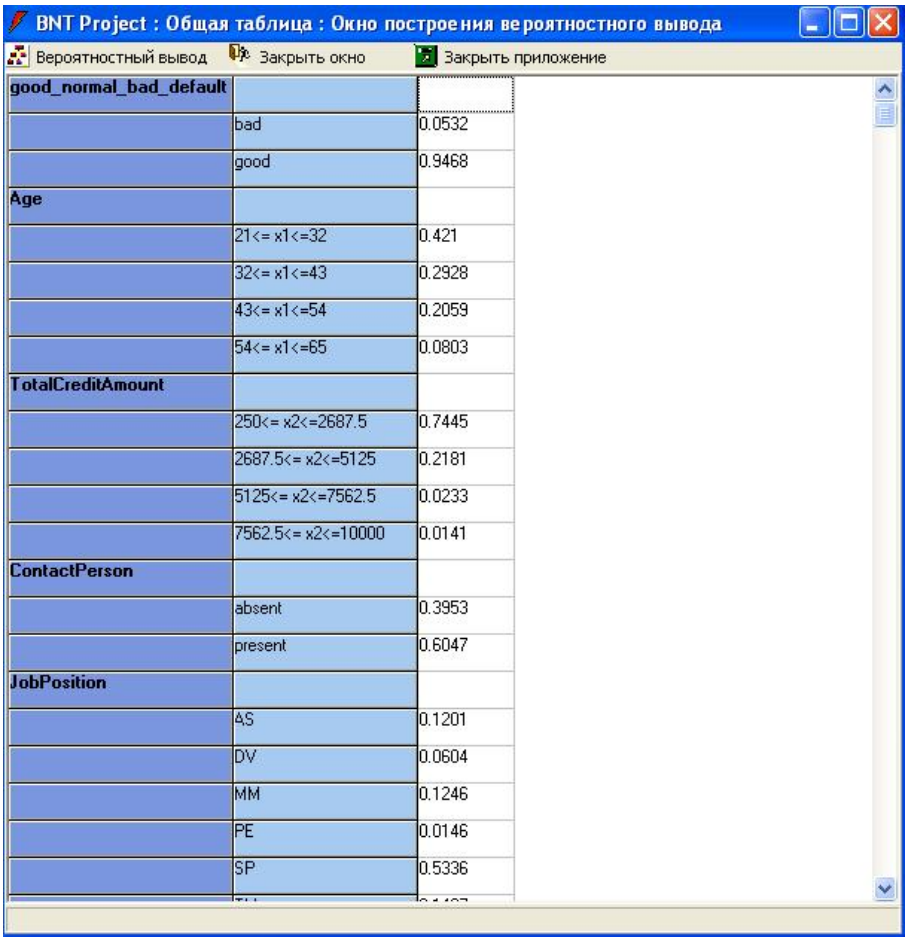


Рисунок 3.24 – Закладка “Точний метод”

Для побудови ймовірнісного висновку в БМ за навчальними даними та представлення результатів виводу у вигляді однієї загальної таблиці, треба натиснути на пункт меню “Сумісна таблиця” (рисунок 3.25).

Рисунок 3.25 – Приклад представлення ймовірнісного висновку у вигляді загальної таблиці

Після цього з’явиться форма із загальною таблицею (рисунок 3.26).



good_normal_bad_default		
	bad	0.0532
	good	0.9468
Age		
	21<= x1<=32	0.421
	32<= x1<=43	0.2928
	43<= x1<=54	0.2059
	54<= x1<=65	0.0803
TotalCreditAmount		
	250<= x2<=2687.5	0.7445
	2687.5<= x2<=5125	0.2181
	5125<= x2<=7562.5	0.0233
	7562.5<= x2<=10000	0.0141
ContactPerson		
	absent	0.3953
	present	0.6047
JobPosition		
	AS	0.1201
	DV	0.0604
	MM	0.1246
	PE	0.0146
	SP	0.5336

Рисунок 3.26 – Загальна таблиця станів та чисельних значень ймовірностей всієї побудованої мережі Байєса

Перший стовпчик таблиці – назви вершин, другий – назви станів вершин, а третій – чисельні значення ймовірностей станів вершин мережі. Подвійним натисненням лівої кнопки миші в таблиці виконується виділення інстанційованого значення вершини (фактора). Після натискування F5 або кнопки “Точний метод” здійснюється побудова ймовірнісного висновку.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Чубукова И. А. Data Mining / И. А. Чубукова – М.: Бином ЛБЗ, 2008. – 384 с.

2. Терехов С. А. Вероятностное моделирование в байесовых сетях / С. А. Терехов // Лекция для школы-семинара “Современные проблемы нейроинформатики”, 29-31 января 2003 г. – М.: МИФИ, 2003 г. – 63 с.

3. Murphy K. A brief introduction to graphical models and Bayesian networks / Technical report 2001-5-10, department of computer science, University of British Columbia, Canada, May 2001. – 19 p.

4. Терентьев А.Н. Алгоритм вероятностного вывода в байесовских сетях / А.Н. Терентьев, П.И. Бидюк, Л.А. Коршевнюк / Системный анализ и информационный технологии: сб. науч. трудов по материалам IX междунар. науч.-тех. конф., 15–19 мая 2007 г., Киев. – К.: Екмо, 2007. – С. 76.

5. Бидюк П. И. Метод вероятностного вывода в байесовских сетях по обучающим данным / П.И. Бидюк, А.Н. Терентьев // Кибернетика и системный анализ. – 2007. – № 3. – С. 93-99.

6. Zweig G.G. Speech recognition with dynamic Bayesian networks / Proceedings of the fifteenth conference on artificial intelligence, Madison (Wisconsin US). – 1998. – P. 173-180.

7. Murphy K.P. Dynamic Bayesian networks: representation, inference and learning / A PhD dissertation, University of California, Berkeley. – 2002. – 225 p.

8. Buntine W. L. A Guide to the literature on learning probabilistic networks from data // IEEE Transactions on knowledge and data engineering. – Piscataway: IEEE Educational Activities Department, 2006. – Vol. 8, № 2. – P. 195-210.

9. Hautaniemi S.K., Petri T. Korpisaari and Jukka P.P. Saarinen. Target identification with dynamic hybrid Bayesian networks / Proceedings of the forth SPIE's international symposium on sensor fusion: architectures, algorithms and applications, Orlando (Florida US), April 2000. – P. 55 – 66.

10. Nodelman U., Christian R. Shelton, and Daphne Koller. Learning continuous time Bayesian networks / Proceedings of the nineteenth international conference on Uncertainty in Artificial Intelligence (UAI'03), Acapulco (Mexico), 7–10 August, 2003. – SF.: Morgan Kaufmann, 2003. – P. 451-458.

11. Cheng J., Bell D.A. and Liu W. Learning belief networks from data: an information theory based approach / Proceedings of the sixth international conference on information and knowledge management (CIKM 1997), Las Vegas (Nevada), November 10-14. – 1997. – P.325-331.

12. Cheng J., Greiner R., Kelly J., Bell D.A. and Liu W. Learning Bayesian networks from data: an information-theory based approach // The artificial intelligence journal (AIJ). – 2002. – 137. – P. 43-90.

13. Spirtes, P., Glymour, C. and Scheines, R., Causation, prediction and search // Adaptive computation and machine learning, MIT press, January 2001. – 565 p.

14. Jouffe L. and Munteanu P. New search strategies for learning Bayesian networks / Proceedings of tenth international symposium on applied stochastic models and data analysis (ASMDA 2001). – Compiegne (France). 12–15 June 2001. – Vol. 2 – P. 591–596.

15. Chow C.K., Liu C.N. Approximating discrete probability distributions with dependence trees // IEE Transactions on information theory, May 1968. – Vol. IT-14, №3. – P. 462 – 467.

16. Rebane G., Pearl J. The recovery of causal poly-trees from statistical data. // International journal of approximate reasoning, July 1988. – Vol. 2, № 3. – P. 175-182

17. Herskovits E. and Cooper, G. Kutato: an entropy-driven system for construction of probabilistic expert systems from databases / Proceedings of the sixth international conference on Uncertainty in Artificial Intelligence (UAI'90), Cambridge, Massachusetts, USA, 27–29 July, – NY.: Elsevier science, 1991. – P 54-62.

18. Cooper G., Herskovits E. A Bayesian method for the induction of probabilistic networks from data. // Machine Learning, 1992. – 9. – P. 309-347.

19. Heckerman D., Geiger D. and Chickering D. Learning Bayesian networks: the combination of knowledge and statistical data / Technical report MSR-TR-94-09, Microsoft Research, March 1994. – 53 p.

20. Wong, S. and Xiang, Y. Construction of a Markov network from data for probabilistic inference / Third International workshop on rough sets and soft computing (RSSC'94), San Jose (California). – 1994. – P. 562-569.

21. Chu T. and Xiang Y. Exploring parallelism in learning belief networks / Proceedings of the thirteenth international conference on Uncertainty in Artificial Intelligence (UAI'97), Providence, Rhode Island, USA, 1–3 August, 1997. – SF.: Morgan Kaufmann, 1997. – P. 90-98.

22. Lam W. and Bacchus F. Learning Bayesian belief networks: an approach based on the MDL principle. // Computational Intelligence, July 1994. – Vol. 10, № 4. – P. 269-293.

23. Acid S. and Campos L. Benedict: an algorithm for learning probabilistic belief networks / Proceedings of the sixth international conference IPMU'96, Granada, Spain. – 1996. – P. 979–984.

24. Singh M. and Valtorta M. Construction of Bayesian network structures from data: a brief survey and an efficient algorithm. // International journal of approximate reasoning, 1995. –12. – P. 111-131.

25. Friedman N. and Goldszmidt M. Learning Bayesian networks with local structure. / Proceedings of the twelfth international conference on Uncertainty in Artificial Intelligence (UAI'96), Portland, Oregon, USA, 1–4 August, 1996. – SF.: Morgan Kaufmann, 1996. – P. 252-262.

26. Wallace C., Korb K. and Dai H. Causal discovery via MML / Proceedings of the thirteenth international conference on machine learning (ICML'96), Bari, Italy. – SF.: Morgan Kaufmann, 1996. – P. 516-524.

27. Suzuki J. Learning Bayesian belief networks based on the MDL principle: an efficient algorithm using the branch and bound technique. // IEICE Transaction on information and systems, February 1999. – P. 356-367.

28. Wermuth N. and Lauritzen S. Graphical and recursive models for contingency tables // Biometrika, December 1983. – Vol. 70, №3. – P. 537-552.

29. Pearl J. Probabilistic reasoning in intelligent systems: networks of plausible inference. – SF.: Morgan Kaufmann, September 1988. – 552 p.

30. Srinivas S. Russell S. and Agogino A. Automated construction of sparse Bayesian networks from unstructured probabilistic models and domain information / Proceedings of the fifth annual conference on Uncertainty in Artificial Intelligence (UAI'90), Cambridge, Massachusetts, USA, 27–29 July, 1990. – NY.: Elsevier science, 1991. – P. 295 – 308.

31. Fung R.M., Crawford S.L. Constructor: a system for the induction of probabilistic models / Proceedings of the seventh national conference on artificial intelligence (AAAI-90). – 1990. – P. 762-769.

32. Spirtes P., Glymour C. and Scheines R. Causality from probability / Proceedings of advanced computing for the social sciences, Williamsburgh. – 1990. – P. 107-121.

33. Spirtes P., Glymour C. and Scheines R. An algorithm for fast recovery of sparse causal graphs // Social science computer review (SSCORE). – 1991. – 9. – P. 62-72.

34. Sebastiani P. and Ramoni M. Bayesian inference with missing data using bound and collapse // Journal of Computational and Graphical Statistics, December 2000. – Vol. 9, № 4. – P. 779-800.

35. Dempster A.P., Laird N.M. and Rubin D. B. Maximum likelihood from incomplete data via the EM algorithm // Journal of the Royal Statistical Society. – 1977. – Vol. 39, №1. – P. 1-38.

36. Zhang.Z, Kwok. J and Yeung D. Surrogate maximization (minimization) algorithms for AdaBoost and the logistic regression model / Proceedings of the

twenty-first international conference on machine learning (ICML 2004). – 2004. – 117 p.

37. Robinson R.W. Counting unlabeled acyclic digraphs / Proceeding of fifth Australian on combinatorial mathematics. Melbourne, Australia. – 1976. – P. 28-43.