роль у розробці та аналізі алгоритмів машинного навчання. Незважаючи на назву, мережі Байєса необов'язково тісно пов'язані з байєсовою статистикою. Їх назва більше пов'язана з використанням правила Байєса для ймовірнісного висновку [11].

Формула Байєса являє собою модель раціонального вибору в умовах неточної або неповної інформації. Процес прийняття рішень може ґрунтуватися на двох протилежних методах – дедукції й індукції. Обираючи шлях дедукції, виходять із певної гіпотези або припущення про сутність події й припускають, що відбудеться згодом, якщо ця гіпотеза достовірна. Оскільки між гіпотезою й передбачуваним розвитком подій існує стійкий зв'язок, дедукція вважається об'єктивним методом, але вона лише підтверджує або спростовує запропоновану гіпотезу й не здатна створити нові знання.

Індуктивні міркування мають протилежну спрямованість: гіпотези будуються й оцінюються на підставі даних досвіду. В основі доказу лежить процес індукції, що відображає перехід від спостережень до закономірностей, що є їх основою. Переваги індуктивного методу аналізу полягають у тому, що за допомогою даних досвіду одержують додаткову інформацію про раніше невідомі явища, будують нові гіпотези й поглиблюють свої знання про зовнішній світ. Однак при цьому неминуче виникає так звана проблема індукції: не можна бути повністю впевненим у правильності доводів. Значно складніше індуктивна диференціальна діагностика, що реалізується у визначенні ймовірності різних визначень, виходячи з наявної інформації й результатів проведених досліджень. Таким чином, дедуктивний метод надійніший і об'єктивніший, але менш продуктивний, ніж індуктивний.

Позначимо через Ω *вибірковий простір* подій (або множину подій) випадкових експериментів. Цей вибірковий простір містить всі можливі значення випадкової змінної. За аналогією 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. У теорії оцінювання сигналів, аналізі часових рядів та багатьох інших випадках події називають *спостереженнями*.

Розглянемо дві події $E \in \Omega$ і $H \in \Omega$, що відповідно є спостереженням та гіпотезою. Формула Байєса використовується для обчислення ймовірності того, що подія E відбудеться за умови, що відбулась подія H. Тобто для обчислення $P(E \mid H)$ - умовної ймовірності E при заданій події H.

Умовна ймовірність $P(E \mid H)$ дорівнює відношенню сукупної ймовірності подій E та H $P(E \cap H)$ до ймовірності події H, за умови, що вона не дорівнює нулю:

$$P(E \mid H) = \frac{P(E \cap H)}{P(H)}. \qquad (1.1)$$

Аналогічно для події H:

$$P(H \mid E) = \frac{P(H \cap E)}{P(E)}. \qquad (1.2)$$

Звідси можна ***записати правило Байєса*** (враховуючи властивість комутативності сукупної ймовірності):

$$P(H \mid E) = \frac{P(E \mid H) \cdot P(H)}{P(E)}. \tag{1.3}$$

Цей вираз показує причинно-наслідкові зв'язки між спостереженнями та гіпотезами. Ймовірності $p(H)$ та $p(E \mid H)$ є апріорними, тобто задаються до початку спостережень. Ймовірність $P(H \mid E)$ є апостеріорною. Перевага байєсівського методу полягає в тому, що апріорні ймовірності можна уточнювати (оновлювати) у відповідності до реалій протікання процесу, що досліджується. Це дозволяє уточнювати ймовірності подій при надходженні додаткової інформації.

Головне припущення теорії побудови мереж Байєса полягає в тому, що події є вичерпними ($\cup_{i=1}^{n} H_i = \Omega$) і не перетинаються. При виконанні цих умов ймовірність події E можна обчислити за допомогою умовних ймовірностей:

$$p(E) = \sum_{i=1}^{n} p(E \cap H_i) = \sum_{i=1}^{n} p(E \mid H_i) \cdot p(H_i). \tag{1.4}$$

Підставивши (1.4) у формулу (1.3), отримаємо вираз:

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

На основі цієї *формули будуються мережі Байєса.*

У формулі (1.5) H_k означає будь-яку гіпотезу з n можливих. Ймовірності $p(H_k \mid E)$ задаються експертами *апріорно* або розраховуються за навчальними даними [6, 64]. Тобто їх можна розглядати як відповідь на запитання: "*Якою буде ймовірність деякого виміру, якщо відомо, яка гіпотеза була реалізована*?". Ймовірності $p(H_k \mid E)$ 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. Сила байєсівського методу полягає в тому, що апріорні ймовірності можна уточнювати (оновлювати) у відповідності до реалій протікання процесу, що досліджується. Це дає можливість уточнювати ймовірності подій при надходженні додаткової інформації. Знаменник виразу (1.5) можна розглядати як нормуючий член, який встановлює значення ймовірності в інтервалі між 0 та 1.

При розв'язанні задач аналізу та моделювання випадкових величин найчастіше використовують нормальний або ***Гаусів розподіл***, який має декілька властивостей, що сприяють прискоренню та спрощенню аналізу процесів. Він ***описується виразом***:

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

де μ_x середнє значення процесу; σ_x - стандартне відхилення.

Очевидно, що для опису 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, не центрований t - розподіл, рівномірний неперервний розподіл та розподіл Вейбулла.

Байєсівська мережа представляє собою пару $<G,B>$, у якій перша компонента G – це спрямований нециклічний граф, що відповідає змінним процесу, що досліджується і записується у вигляді причинно-наслідкової мережі. Друга компонента пари B – це множина параметрів, що визначають мережу.

Компонента B містить параметри $\Theta_{X^{(i)}|pa(X^{(i)})} = P(X^{(i)}|pa(X^{(i)}))$ 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 використовують позначення $Pa^G(X^{(i)})$.

Повна спільна ймовірність мережі Байєса обчислюється за формулою:

$$P_B(X^{(1)},...,X^{(N)}) = \prod_{i=1}^{N} P_B(X^{(i)}|Pa(X^{(i)})) \tag{1.7}$$

З математичної точки зору мережа Байєса – це модель подання існуючих і відсутніх імовірнісних залежностей. При цьому зв'язок $A \to B$ є причинним, коли подія A – причина виникнення B, тобто коли існує механізм, відповідно до якого значення, прийняте A, впливає на значення, прийняте B. Мережу Байєса називають причинною (каузальною), коли всі її зв'язки є причинними.

Часто мережі Байєса представляють у вигляді графів із деякими специфічними властивостями. Як і будь-який граф, мережа Байєса 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. Графи, що містять тільки неспрямовані дуги, називають *неспрямованими*. Якщо дуга має стрілку, що вказує на напрям залежності – від причини до наслідку, то графи, у яких всі дуги спрямовані, називають *спрямованими*. А отже, всі байєсівські мережі – спрямовані графи.

Типи графічних структур байєсівської мережі:

1. *Дерево* – структура, у якій будь-яка вершина може мати не більше, ніж одну батьківську (рис. 1.3).

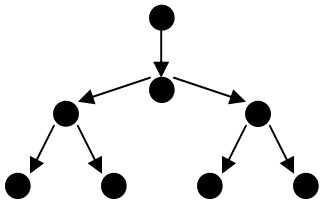


Рис. 1.3. Структура мережі Байєса у вигляді дерева

2. *Полідерево* – структура мережі Байєса, у якій будь-яка вершина може мати більше ніж одну батьківську вершину, але при цьому між будь-якими двома вершинами повинно бути не більше одного з'єднуючого їх шляху (рис. 1.4).

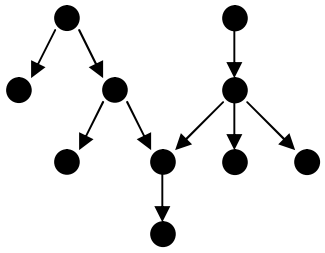


Рис. 1.4. Структура мережі Байєса у вигляді полі-дерева

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.

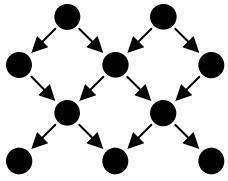
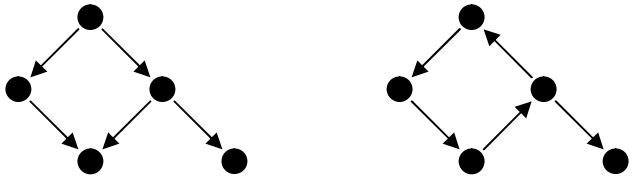


Рис. 1.5. Структура у вигляді мережі

Структури "дерево" та "полідерево" називають також ***однозв'язними мережами***, а структури типу "мережа" – ***багатозв'язними мережами***.

Як відомо з дискретної математики, 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 рис. 1.6.



а) *б)*

Рис.1.6. Спрямовані ациклічний (а) та циклічний (б) графи

Спрямований ациклічний граф називають *одиночно з'єднаним*, якщо між будь-якими двома його змінними існує тільки один шлях, що їх з'єднує. Графи, показані на рис.1.6, 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 мереж Байєса мають специфічні назви. Змінну (вузол, вершину) називають *дочірньою* (нащадком), якщо вона залежить від однієї або більше інших змінних. *Батьківська* змінна має одну або більше змінних-нащадків. До множини *нащадків* відносять похідні змінні, які відносяться до однієї батьківської змінної, а

також дочірні змінні дочірніх змінних вищого рівня. Одна і та ж змінна може бути батьківською і дочірньою одночасно. Якщо змінна не має жодної батьківської, то її називають *кореневою*. Кореневі змінні – найважливіші змінні мережі; як правило, це атрибут, який нас цікавить. Змінну називають *листом*, якщо вона не має змінних-нащадків. Однозв'язний граф із змінними, що мають спеціальні назви, представлено на рис. 1.7.

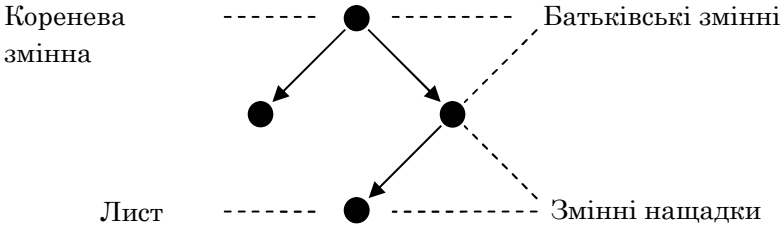


Рис. 1.7. Однозв'язна мережа

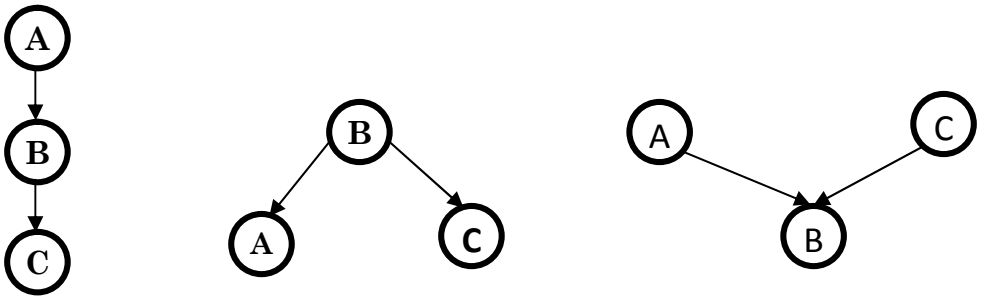
Неоднозв'язність – це важлива властивість мережі. Якщо мережі однозв'язані, то висновок (рішення) можна знайти досить легко і прямолінійно. Але при вирішенні більшості практичних задач, як правило, мають справу з неоднозв'язними графами (рис. 1.6(б)).

Такі задачі значно ускладнюють і уповільнюють формування висновку.

Розрізняють три типи зв'язків між вершинами:

- *лінійний;*
- *розбіжний* (divergent);
- *збіжний* (convergent).

На рис. 1.8 зображено відповідні типи зв'язків.



а) *б)* *в)*

Рис. 1.8. Типи зв'язків

За лінійного типу зв'язку *A* впливає на *B*, а *B*, в свою чергу, на *C* і вершини *A* і *C* є залежними між собою. Але у випадку інстанціювання вершини *B,* ці вершини стають незалежними, бо причина події вже не змінить її наслідків.

При розбіжному типі зв'язку значення вершини *A* може вплинути на значення *C* тому, що інстанціювання *A* змінює значення її батька *B*, яке, в свою чергу, впливає на значення іншого нащадка *C*. Але у випадку інстанціювання вершини *B* нащадки стають незалежними, бо значення вершини *A* вже ніяк не змінить значення вершини *B*.

При збіжному типі зв'язку інстанціювання вершини *A* впливає тільки на значення її нащадка B, а вершини A і C незалежні між собою. Але у випадку

інстанціювання вершини B значення обох її батьків змінюються, і тепер зміна значення вершини A вплине на значення вершини C. Тобто залежність між вершинами з'являється тоді, коли спільний нащадок (або подальші його нащадки) отримує якесь значення.

Узагальнюючи типи зв'язків, можна навести таке означення: дві вершини спрямованого графу A і C називаються d-незалежними, якщо для будь-якого (неспрямованого) шляху з A в C існує така проміжна вершина B, що або зв'язок у цьому шляху лінійний чи розбіжний і вершина B інстанційована, або зв'язок в шляху є збіжним і вершина B та жодний з її нащадків не інстанційовані. У протилежному випадку ці вершини будуть називатися d-залежними [14].

Ключовою властивістю мереж Байєса є те, що графічна d-незалежність і ймовірнісна незалежність вершин еквівалентні [29].

Визначають такі ***типи байєсівських мереж*** дискретні, динамічні, неперервні, гібридні.

Дискретні мережі Байєса – мережі, у яких змінні вузлів представлені дискретними величинами. ***Дискретні мережі Байєса*** мають такі ***властивості:***

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

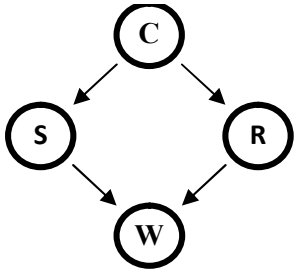
– всі вершини, пов'язані з "батьківськими", визначаються таблицею умовних ймовірностей (ТУЙ) або функцією умовних ймовірностей;

– для вершини 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. Таблиця умовних ймовірностей кожної вершини містить ймовірності станів цієї вершини за умови станів її "батьків". На рис. 1.9 наведено приклад дискретної мережі Байєса, кожна з вершин може приймати один з двох станів F або T (скорочення від false та true). Запис $P(A|B)$ означає ймовірність настання події A за умови, що подія B вже відбулась.

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

Рис. 1.9. Дискретна мережа Байєса з таблицею умовних ймовірностей

Динамічні мережі Байєса – мережі, у яких значення вузлів змінюються з часом. Динамічні мережі Байєса ідеально підходять для моделювання часових процесів. Їх перевага полягає в тому, що вони використовують табличне представлення умовних ймовірностей, що полегшує представлення різних нелінійних явищ [30].

Треба підкреслити, що термін "часова байєсівська мережа" (*temporal Bayesian network*) краще відображає суть, ніж "динамічна байєсівська мережа" (dynamic Bayesian network); він передбачає, 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 [31].

Найпростіший тип динамічної мережі Байєса – це *прихована модель Маркова* (Hidden Markov Model або скорочено HMM), у якій у кожному шарі наявний один дискретний прихований вузол та один дискретний або безперервний спостережуваний вузол. Ілюстрацію моделі наведено на рис. 1.10. Круглі вершини позначають неперервні вузли, квадратні позначають дискретні; X – приховані вузли, а Y – спостережувані. Для визначення динамічної мережі Байєса потрібно задати початковий розподіл $P(X(t))$, топологію усередині шару та міжшарову топологію (між двома шарами) $P(Y(t)|X(t))$ [32].

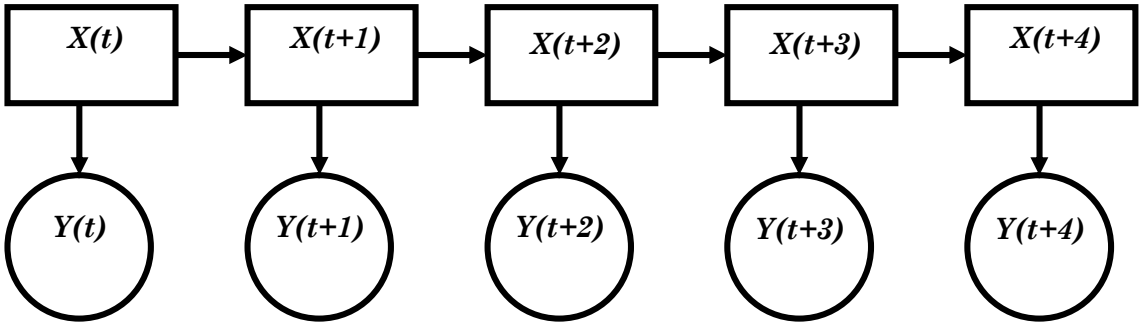


Рис. 1.10. Прихована двохшарова модель Маркова, в якій X – приховані дискретні, а Y – дискретні або неперервні спостережувані вузли

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, щоб представляти собою множину блоків, які повторюються у різні моменти часу [33].

Неперервні мережі Байєса – 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 допустимих значень $X = \{x | a \le x \le b\}$, який містить нескінченну множину точок. У 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 мережі Байєса використовують для моделювання стохастичних процесів у просторі станів з неперервним часом.

На рис. 1.11 наведено приклад неперервної мережі Байєса; в цьому випадку використовується розподіл Гауса.

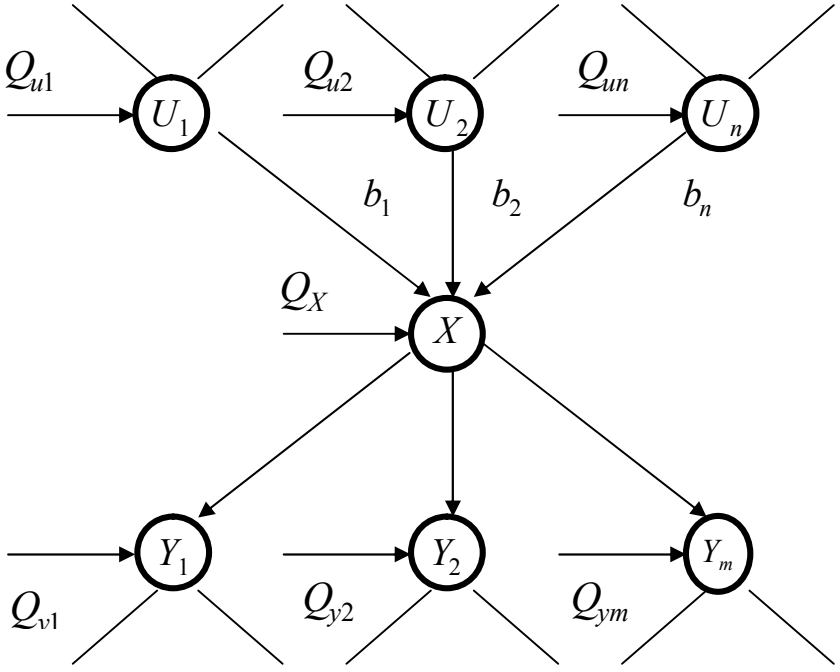


Рис. 1.11. Приклад неперервної мережі Байєса

На рис. 1.11 позначено: $U_1, U_2, ..., U_n$ – батьківські вузли для X, $Y_1, Y_2, ..., Y_m$ – дочірні 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 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) [34, 35]:

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

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:

$$X = b_1 \cdot U_1 + b_2 \cdot U_2 + ... + b_n \cdot U_n + Q_x, \tag{1.9}$$

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.

Гібридні мережі Байєса – мережі, які містять вузли з дискретними і неперервними змінними. При використанні байєсівських мереж, 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:

$$P(X|Y=y, Z=z) = N(\mu_x(\mu_y, \mu_z), \sqrt{\sigma_x(\sqrt{\sigma_y})}), \quad (1.10)$$

де μ_x, μ_y, μ_z – математичні очікування, σ_x, σ_y – дисперсії, $\sqrt{\sigma_x}, \sqrt{\sigma_y}$ – середньоквадратичні 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. Як приклад розглянемо гібридну мережу Байєса, представлену на рис. 1.12.

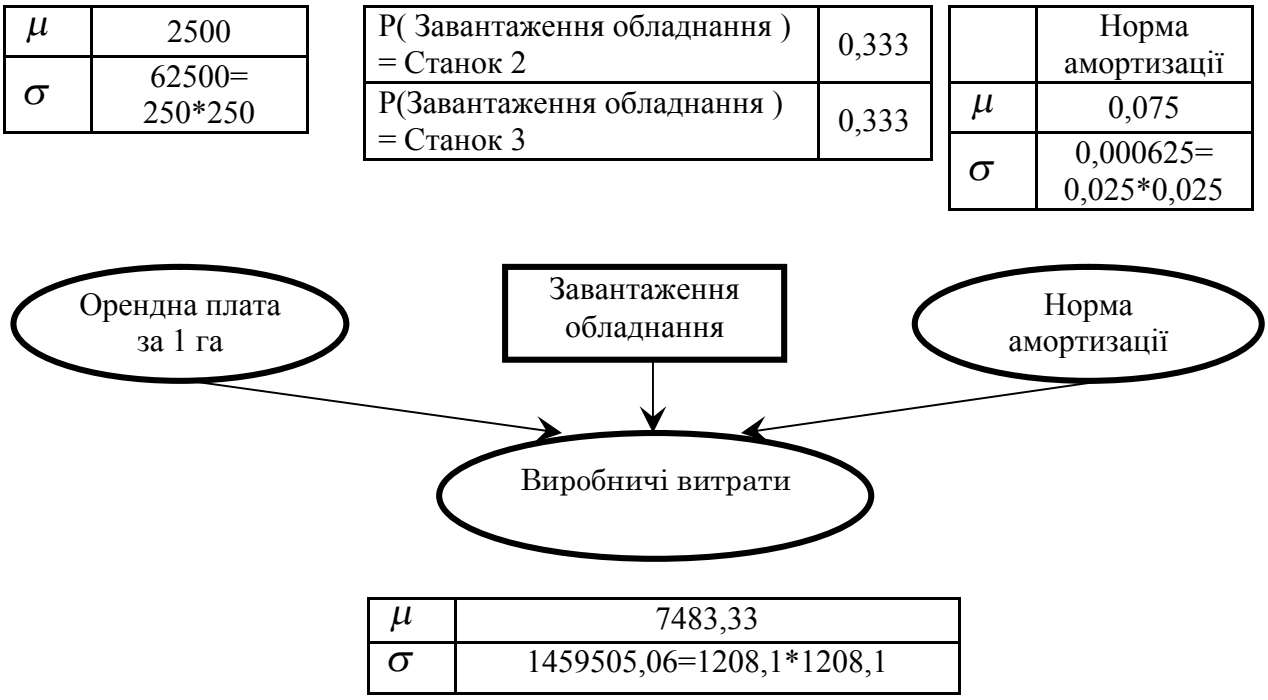


Рис. 1.12. Приклад гібридної мережі Байєса з неперервними та дискретними подіями. Квадратні вершини відповідають дискретним подіям, а овальні – неперервним подіям (гаусівським змінним)

Мережа, представлена на рис. 1.12, дозволяє оцінювати сумарні виробничі витрати залежно від використання та завантаження трьох груп устаткування (наприклад, трьох станків). Відомо, що до складу сумарних виробничих витрат (без врахування зарплати і нарахувань) входять:

– прямі виробничі витрати на кожну групу обладнання за досліджуваний календарний період, які залежать як від кількості використовуваних груп обладнання, так і від часу роботи кожної із груп протягом досліджуваного періоду часу, тобто від ступеня завантаження кожної із груп;

– витрати на амортизацію кожної із груп устаткування, що залежать як від її балансової вартості, так і встановлених норм амортизації;

– орендна плата за ділянку при кожній із груп обладнання, яка використовується для складування сировини та продукції; ця плата залежить як від площі ділянки, так і від ставок орендної плати.

Вершина "Завантаження обладнання" відповідає дискретній події, що характеризується трьома можливими станами. Ймовірність перебування в кожному з них визначається ступенем завантаження кожної із груп устаткування, за умови, що сумарне завантаження всього устаткування дорівнює одиниці. Будемо вважати, що всі групи обладнання завантажені рівномірно. Насправді можливі будь-які інші вихідні розподіли ймовірностей, які враховують різні варіанти завантаження обладнання.

Нехай ставка оренди 1 га землі в середньому становить 2500 у.о. і коливається в межах ±10%, тобто приймає значення 2500 ± 250 у.о.

Отже, цій вершині відповідають параметри $\mu = 2500$ та $\sigma = 62500 = (250)^2$. А норма амортизації може перебувати в межах 5 - 10 % від балансової вартості, тобто приймати значення 7,5 ± 2,5% (або 0,075 ± 0,025 відносних одиниць). Таким чином, цій вершині відповідають параметри $\mu = 0{,}075$ та $\sigma = 0{,}000625 = (0{,}025)^2$.

Що стосується вершини "Виробничі витрати", то вона характеризується випадковою змінною, умовно нормальною на значеннях батьків (тобто на значеннях трьох інших вершин нашого прикладу). Слід зазначити, що в загальному випадку розподіл ймовірностей для вершин, аналогічним вершині "Виробничі витрати", – не просто нормальний, а змішаний нормальний розподіл. Тобто він представляє собою зважену суміш розподілів, для кожного з яких необхідно задати його параметри:

– математичні очікування та дисперсії розподілів, що описують ступінь впливу дискретних батьків;

– вагові коефіцієнти, що враховують ступінь впливу на математичне очікування неперервних батьків.

Припустимо, що в нашому прикладі виконуються такі умови (для трьох станків):

– балансова вартість кожного станка становить 50000, 40000 й 30000 у.о. ;

– площа орендованих ділянок, закріплених за ними дорівнює 0,6; 0,5 та 0,4 га;

– оцінка прямих витрат на підтримку нормальної роботи кожного з станків у середньому становить 3000, 3200 й 3500 у.о. і отримана з 5% точністю.

Ступінь впливу батьківських вершин на "Виробничі витрати" представлено у табл. 1.2.

Таблиця 1.2

Параметри, що визначають розподіл ймовірностей для вершини "Виробничі витрати"

Завантаження обладнання	**Станок 1**	**Станок 2**	**Станок 3**
μ – математичне очікування	3000	3200	3500
σ – середньоквадратичне розсіювання	22 500 = 150*150	25 600 = 160*160	30 625 = 175*175
Норма амортизації	50 000	40 000	30 000
Ставка оренди	0,6	0,5	0,4

Ймовірнісний висновок у таких мережах полягає у поширенні ймовірностей і параметрів гаусівських законів розподілу по всій мережі залежно від отриманих свідчень. В основі процесу формування ймовірнісного висновку лежать досить складні математичні алгоритми. 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 рис. 1.13.

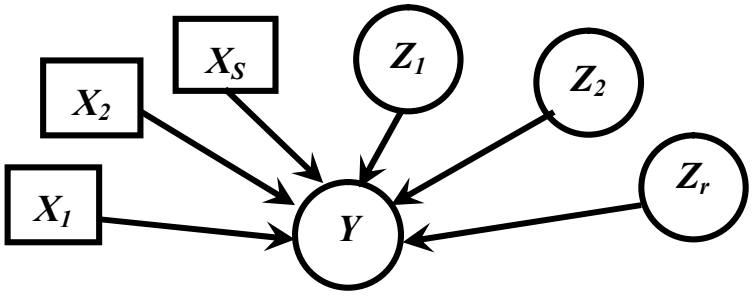


Рис. 1.13. Приклад дворівневої гібридної мережі Байєса з неперервними та дискретними змінними; квадратні вершини відповідають дискретним подіям, а овальні – неперервним подіям

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 формулами 1.11 та 1.12:

$$\mu = \sum_{i1}^{n1} \ldots \sum_{is}^{ns} p_{1,i1} \ldots p_{s,is} \left(\mu_{i1,\ldots,is} + \sum_{l=1}^{r} K_{l,i1,\ldots,is} \cdot \mu_l \right), \tag{1.11}$$

$$\sigma = \sum_{i1}^{n1} \ldots \sum_{is}^{ns} p_{1,i1} \ldots p_{s,is} \cdot \left(\left(\mu_{i1,\ldots,is} + \sum_{l=1}^{r} K_{l,i1,\ldots,is} \right)^2 + \sigma_{i1,\ldots,is} + \sum_{l=1}^{r} K^2_{l,i1,\ldots,is} \cdot \sigma_l \right) - \mu^2 \tag{1.12}$$

Стосовно розглянутого прикладу, який містить дві вихідні неперервні ($r = 2$) змінні та одну дискретну ($s = 1$) змінну, що має три результати ($n_1 = 3$), числові характеристики випадкової змінної "Виробничі витрати" складуть:

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

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

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

Контрольні питання

1 Технологія інтелектуального аналізу даних: наведіть означення та приклади практичного застосування.
2 Що означає у даному випадку слово «інтелектуальний»?
3 Дайте пояснення інтелектуальному аналізу даних як мультидисциплінарній області знань. У чому полягають особливості розвитку цієї області?
4 Наведіть означення та приклад постановки і розв'язання задачі машинного навчання.
5 Наведіть означення та приклад розв'язання задачі методами штучного інтелекту.
6 Дайте означення життєвого циклу технології, наведіть графік та приклад.
7 Наведіть приклади популярних задач і методів інтелектуального аналізу даних.
8 Наведіть приклади алгоритмів, які використовуються в інтелектуальному аналізі даних.
9 Історія виникнення мереж Байєса. Ким був Томас Байєс і в якому столітті він жив?
10 Наведіть приклади сфер людської діяльності, у яких успішно

використовуються мережі Байєса?
11 Яка головна мета побудови мережі Байєса?
12 Які типи невизначеностей враховуються у мережах Байєса?
13 Які два типи навчання необхідно виконати для побудови мереж Байєса?
14 Чим відрізняються дискретні змінні від неперервних?
15 Чи може мережа Байєса містити одночасно дискретні і неперервні змінні?
16 Наведіть недоліки і переваги використання мереж Байєса?
17 Які методи інтелектуального аналізу даних близькі за своїм призначенням до байєсівських мереж?

РОЗДІЛ 2
РОЗВ'ЯЗАННЯ СЛАБКОСТРУКТУРОВАНИХ ЗАДАЧ СИСТЕМНОГО АНАЛІЗУ ЗА ДОПОМОГОЮ БАЙЄСІВСЬКИХ МЕРЕЖ

- Застосування мереж Байєса до розв'язання слабкоструктурованих задач
- Огляд випадків навчання мереж Байєса
- Алгоритми навчання мереж Байєса з прихованими вершинами
- Побудова структури мережі Байєса з прихованими вершинами та невідомою структурою
- Методика знаходження параметрів байєсівських мереж з прихованими вершинами
- Приклади застосування мереж Байєса з прихованими вершинами

2.1. Застосування мереж Байєса до розв'язання задач системного аналізу

Сучасні тенденції соціально-економічного розвитку, суспільного прогресу і політичних процесів потребують розв'язання багатьох нагальних завдань. Однією із головних проблем, яка пов'язана з виконанням цих завдань, є те, що дуже часто задачі носять слабкоструктурований характер. Такі проблеми виникають у складних технологічних, фінансово-економічних, соціальних, екологічних і біологічних системах. Вони характеризуються умовами невизначеності різної природи, наявністю неповної, нечіткої інформації та ускладнюються впливом людського фактору. Задачі такого типу розв'язуються методами системного аналізу і теорії прийняття рішень.

Системний підхід застосовують у наукових дослідженнях і при розв'язанні практичних задач, пов'язаних з прогнозуванням, проектуванням і управлінням у технічних (технологічних) системах, біології, психології, соціально-економічній, політичній і військовій сферах. Починаючи з кінця 50-х років минулого століття, ці методи застосовують при прийнятті управлінських рішень у промисловості, фінансовій, комерційній діяльності та інших областях. Системний підхід передбачає виявлення і дослідження зв'язків та відношень між елементами об'єктів управління, встановлення ієрархічності структури та аналіз і врахування невизначеностей з метою підвищення якості управлінських рішень.

Рішенням вважається обґрунтована множина дій з боку особи, що приймає рішення (ОПР), спрямованих на об'єкт чи систему управління, який надає можливість привести даний об'єкт чи систему до бажаного стану або досягнути поставленої мети. Рішення є одним із видів розумової діяльності і проявом волі людини.

Характерними ознаками рішення є такі:

- можливість вибору з набору альтернативних варіантів: за відсутності альтернатив відсутній і вибір, отже, відсутнє й рішення;
- наявність мети: безцільний вибір не розглядається як рішення;
- необхідність вольового акту особи, що приймає рішення при виборі альтернативи, тому що вона формує рішення при боротьбі мотивів і думок.

Прийняття рішення – це процес вибору найбільш преференційного рішення з множини допустимих рішень або упорядкування множини рішень. Прийняття рішень можливе на підставі знань про об'єкт управління, процеси, що в ньому відбуваються і можуть відбутися з перебігом часу, а також за наявності множини показників, що характеризують ефективність та якість прийнятого рішення. Тобто необхідні адекватна модель об'єкту і модель прийняття та оцінювання прийнятого рішення. Під моделлю прийняття рішень мається на увазі формальне подання поставленої задачі та процесу прийняття рішень.

Переважна більшість досліджень виконується за припущення, що дані є повними, тобто значення (виміри) усіх змінних відомі для кожного варіанту моделювання. Однак це не зовсім реальні ситуації, адже приховані змінні відіграють ключову роль при розв'язанні багатьох практичних задач. Невідомий регулюючий механізм може бути ключем до підвищення якості аналізу біологічних та систем іншої природи. Зв'язки між симптомами та іншими явищами можуть дати поштовх до розгадки прихованої фундаментальної проблеми при діагностуванні та управлінні складними системами. Навмисно або ненавмисно замаскована економічна сила може бути причиною виникнення взаємно пов'язаних фінансових феноменів, які відіграють вирішальну роль для розвитку процесу [36]. Приховані змінні можуть відігравати роль деякого підсумкового механізму, що "перехоплює" інформацію із спостережуваних змінних та "спрямовує" її до іншої частини мережі. Тому введення і використання прихованих змінних при моделюванні може надати можливість значно спростити структуру моделі (байєсівської мережі) і в результаті отримати високоякісний загальний результат, тобто ймовірнісний висновок.

Байєсівські мережі з прихованими змінними – це чіткий приклад застосування системного підходу до розв'язання слабкоструктурованих задач. Знаходження причинно-наслідкових зв'язків, обробка неповних даних, отримання адекватної ймовірнісної моделі на виході – все це робить мережі Байєса ефективним інструментом системного аналізу. А головна потужність та цінність мереж Байєса у рамках системного підходу полягає у їх здатності виявляти неочевидні, нетривіальні та невідомі зв'язки між факторами, про які іноді самі експерти у відповідній предметній області не завжди можуть мати достатню уяву. Саме тому вони набули широкого застосування в найрізноманітніших сферах сучасного життя людини і продовжують активно досліджуватись та впроваджуватись у практику.

Системний аналіз – це науковий метод пізнання, який ґрунтується на визначеній послідовності дій, спрямованих на виявлення, обґрунтування та використання структурних зв'язків між змінними та/або елементами досліджуваної системи. Він спирається на комплекс загальнонаукових теоретичних, експериментальних, природничих, математичних, статистичних та інших методів дослідження. Системний аналіз має універсальний міждисциплінарний характер.

Одне з відомих означень системного аналізу характеризує його як фундаментальну теоретичну основу, яка може використовуватись для подолання складностей і труднощів дослідження об'єктів та процесів у самих різних галузях людської діяльності з подальшою метою створення моделей для прогнозування, передбачення та прийняття коректних управлінських рішень. Тому для розв'язання практичних задач розглянемо поняття саме прикладного системного аналізу.

Прикладний системний аналіз — це наукова дисципліна, яка на основі об'єктивно вибраних та системно організованих, структурно взаємопов'язаних та

функціонально взаємодіючих формалізованих і евристичних процедур, методологічних та методичних засобів, апарату математико-статистичного аналізу, програмного забезпечення і комп'ютерних систем забезпечує отримання і накопичення інформації стосовно функціонування об'єкта дослідження в умовах наявності невизначеностей структурного, параметричного і статистичного характеру з подальшою метою формування знань, достатніх для розв'язання поставленої задачі (прогнозування, передбачення та управління).

Більшість рішень у сучасних складних задачах приймаються людьми одноосібно або колегіально в умовах наявності невизначеностей різної природи та типів. ***Невизначеність*** припускає наявність факторів, при яких результати дій не є детермінованими, а ступінь можливого впливу цих факторів на результати невідома.

Аналіз умов появи невизначеностей може виконуватись на абстрактному теоретичному рівні або з точки зору конкретної ситуації, наприклад, з точки зору виявлення можливості побудови математичної моделі чи з точки зору теорії інформації, де невизначеність виступає як характеристика ситуації вибору. Категорія невизначеності характеризується деякими змінними параметрами, які описують різні види невизначеностей: глобальну невизначеність, ситуативну, політичну, соціальну і т. ін. При розв'язанні задач прийняття рішень в умовах наявності невизначеностей необхідно встановити рівень аналізу і типи невизначеностей, що розглядаються.

Необхідно зазначити, що часто невизначеність ототожнюють лише з відсутністю повної інформації про той чи інший об'єкт. Насправді недостатні знання станів об'єкту – це не єдина невизначеність. Поряд з цим іноді можна розглядати невизначеність цілей та невизначеність критеріїв вибору рішень. У багатьох реальних задачах складність прийняття рішення визначається насамперед кількістю альтернативних варіантів та кількістю і різнорідністю критеріїв оцінювання цих варіантів.

Загалом ***задачі прийняття рішень розділяють*** на три групи: структуровані, слабко структуровані та неструктуровані. Виділення добре структурованих, слабкоструктурованих і неструктурованих задач звичайно досить умовне, оскільки досягти чіткого розмежування тут неможливо. ***Добре структуровані завдання*** упорядковані і їх розв'язання – це процеси, що повторюються; крім того, вони однозначні, оскільки кожне завдання має єдиний (кращий) метод розв'язання. Для розв'язання ***слабкоструктурованих завдань*** існує більше можливих методів, і самі рішення можуть досить суттєво відрізнятися одне від одного. ***Неструктуроване завдання*** або взагалі не має відомих методів рішення, або може мати їх надто багато.

Розглядаючи докладніше ***слабкоструктуровані проблеми***, можна виділити їх такі ***особливості*:**

- задача не може бути виражена у числовій формі;
- цілі не можуть бути вираженими у термінах точно визначеної цільової функції;
- практично не існує алгоритмічного розв'язку задачі;
- якщо алгоритмічний розв'язок існує, то його не можна використати через обмеженість ресурсів (час, пам'ять і таке інше).

До слабкоструктурованих проблем відносять найбільш складні завдання стратегічного планування і розподілу ресурсів, фінансово-економічні та задачі

технічного, політичного, військово-стратегічного характеру. Розв'язання проблем, що мають "слабкоструктурований характер", і є основним завданням системного аналізу.

Слабкоструктуровані проблеми, як правило, пов'язані з розробкою стратегічних довгострокових курсів діяльності, кожен з яких охоплює багато аспектів діяльності організації і реалізується поетапно. Наприклад, визначення стратегії випуску нової продукції, технічного переозброєння виробництва, удосконалення організації управління і таке інше. Ці проблеми містять разом з добре вивченими елементами, що кількісно формалізуються, також невідомі або невимірювані компоненти, які відрізняються значною невизначеністю. Вони вирішуються за допомогою методів імітаційного математичного моделювання і системного аналізу, які поєднують у собі складні математичні розрахунки з великим об'ємом суб'єктивних суджень керівників і фахівців.

Для розв'язання слабкоструктурованих задач використовують методологію системного аналізу, яка дає можливість проектувати, реалізовувати і використовувати на практиці сучасні інформаційні системи підтримки прийняття рішень (СППР).

Необхідно відзначити, що моделювання – це один з найбільш універсальних способів вивчення різноманітних процесів та явищ. У наукових дослідженнях та інженерній практиці широко використовують чисельні методи та прийоми моделювання. При цьому розрізняють фізичне і математичне моделювання.

При фізичному моделюванні створена модель відображає поведінку об’єкта, що вивчається, із збереженням його фізичної природи та принципів функціонування. Між об’єктом, що вивчається, та його моделлю мають бути збережені деякі співвідношення подібності, що випливають із закономірностей фізичної природи явища і забезпечують можливість використання даних, які отримують за допомогою моделювання з метою оцінювання властивостей і характеристик досліджуваного об’єкта. Фізичне моделювання має досить обмежену сферу застосування внаслідок значних матеріальних та технічних труднощів, які можуть при цьому виникати.

Набагато ширші можливості має математичне моделювання.

Математичне моделювання – це спосіб дослідження об’єктів на основі вивчення явищ, що мають різний фізичний зміст, але можуть описуватись однаковими (подібними) математичними співвідношеннями. При цьому співвідношення подібності повинні зберігатись між змінними математичної моделі та найбільш важливими властивостями і характеристиками об’єкта, що вивчається.

На практиці часто використовують багато різновидів математичних моделей, що засновані на поєднанні можливостей сучасної математики, прикладної статистики і обчислювальної техніки, наприклад, графічні та імітаційні моделі. Так, графічна модель оперує системою взаємопов’язаних креслень і зображень для відображення реальних взаємозалежностей, характерних для досліджуваного об’єкту. Імітаційна модель представляє собою систему взаємопов’язаних комп’ютерних програм для імітації поведінки об’єкта.

Як сказано вище, ***об'єктом системного аналізу є слабкоструктуровані проблеми***. Тому його методи спрямовуються на "зняття" цієї слабкої структурованості об’єктів шляхом структуризації (упорядкування) задачі, тобто вибору ієрархії цілей, завдань, установлення взаємних зв'язків між елементарними

об'єктами, що визначають масштабність системи, керуючих та інформаційних зв'язків, ієрархію показників ефективності та їх динаміку.

У своєму розвитку системний аналіз традиційно спирався на методи дослідження операцій, які у своїй більшості використовують математичний апарат, який забезпечує отримання високоякісних рішень. Долаючи цю однобічність методології дослідження операцій, системний аналіз вводить у свою методологію додаткову логічну складову для поглиблення дослідження проблем управління. І, залежно від проблеми, системний аналітик приймає рішення стосовно використання тих або інших методів.

Узагальнена методика вирішення неструктурованих і слабкоструктурованих багатокритеріальних задач ґрунтується на створенні інформаційних систем підтримки прийняття рішень.

Ще одним науковим напрямом, що лежить в основі дослідження слабкоструктурованих систем, є методологія когнітивного моделювання. У рамках когнітивної моделі інформація про систему представляється у вигляді набору понять (факторів) і пов'язуючої їх причинно-наслідкової мережі, що має назву *когнітивна карта*. Така карта є віддзеркаленням суб'єктивних представлень експерта (чи групи експертів) про закони і закономірності, властиві модельованій системі. До когнітивної карти застосовують методи аналітичної обробки, орієнтовані на дослідження структури системи і отримання прогнозів її поведінки при різних управляючих подіях з метою синтезу ефективних стратегій управління.

Для розв'язання складних, у тому числі слабкоструктурованих, проблем розроблено ***процедуру прийняття рішень, що складається з таких етапів:***

1) формулювання проблемної ситуації;
2) визначення цілей;
3) визначення критеріїв досягнення цілей;
4) побудова моделей для обґрунтування рішень;
5) пошук оптимального (допустимого) варіанту рішення;
6) узгодження рішення;
7) підготовка рішення до реалізації;
8) затвердження рішення;
9) управління ходом реалізації рішення;
10) перевірка ефективності рішення.

Основними інструментами розв'язання складних проблем є:

- імітаційне моделювання;
- кореляційний, регресійний та кластерний аналіз;
- методи статистичної класифікації;
- методи аналізу ієрархій;
- теорія матриць;
- теорія експертних оцінок;
- нечітка логіка;
- нейронні мережі;
- генетичні алгоритми;
- мережі Байєса.

Отже, в багатьох ситуаціях прийняття рішень щодо слабкоструктурованих проблем, коли використовуються переважно експертні знання про предметну

область, основу яких становлять набори правил та результати суб'єктивного аналізу ситуації, доцільно використовувати саме ймовірнісні інструменти, які дозволяють розкривати невизначеності шляхом обчислення ймовірностей станів вершин мережі Байєса на основі наявної інформації про значення інших вершин, що дає змогу визначити оптимальний варіант переходу системи на наступного стану.

Тому сьогодні мережі Байєса все частіше використовуються у системах підтримки прийняття рішень, призначених для вирішення слабко структурованих задач в умовах невизначеності, коли необхідно врахувати статистичні та структурні невизначеності процесів різної природи.

2.2. Огляд випадків навчання мереж Байєса

Для опису байєсівської мережі необхідно визначити топологію графа й параметри кожного вузла. Цю інформацію можна одержати з навчальних даних, але одержання правильної топології мережі є більш складним завданням, ніж одержання параметрів вершин.

Особливий підхід необхідно застосовувати у випадку, коли деякі вершини приховані або коли маємо справу з некоректними чи неповними даними.

Виділяють чотири випадки навчання мережі Байєса, опис яких наведено в таблиці 2.1.

Таблиця 2.1

Чотири випадки навчання мережі Байєса

Структура	**Спостереження**	**Метод**
Відома	Повне	Оцінка максимальної правдоподібності
Відома	Часткове	Максимізація математичного сподівання або пошук екстремуму
Не відома	Повне	Пошук у просторі моделей
Не відома	Часткове	Структурний алгоритм максимізації математичного сподівання

Розглянемо випадок ***навчання***, ***коли відома структура, спостереження повні***, тобто ***немає прихованих вершин***.

У цьому випадку обчислюються значення параметрів кожного умовного імовірнісного розподілу, які максимізують правдоподібність навчальних даних. Нормалізоване логарифмічне рівняння правдоподібності для навчальної множини $D = \{D_1,..., D_m\}$ має вигляд:

$$L = \frac{1}{N} \cdot \sum_{i=1}^{n} \sum_{m=1}^{M} \log(\, P(X_i \mid Pa(X_i), D_m)), , \qquad (2.1)$$

де $Pa(X_i)$ означає набір батьків X_i, D – навчальна вибірка, а D_m – це 1-й елемент навчальної множини, M – кількість подій у навчальній множині.

Приклад. Дано просту мережу для фіктивного медичного прикладу, що описує наявність у пацієнта метастатичного раку мозку на основі головних болів, коми та наявності кальцію у сироватці крові [110]. Схема мережі зображена на рис. 2.1.

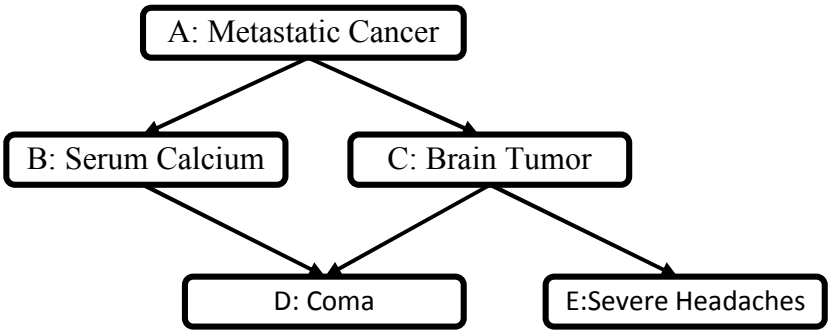


Рис. 2.1. Байєсівська мережа для задачі "Рак мозку" (випадок навчання, коли немає прихованих вершин)

Кожна вершина має по 2 стани: "наявний" і "відсутній" для вершин Metastatic Cancer, Coma, Brain Tumor, Severe Headaches; "підвищений" та "в нормі" для вершини Serum Calcium.

Нехай є дані для навчання, тобто інформація про те, скільки разів була у пацієнтів кома, а причиною цього було те, що були рак мозку і підвищений кальцій у крові $N(D=1,C=1,B=1)$, скільки разів спостерігалась кома і був підвищений кальцій в крові, але не було пухлини в мозку $N(D=1,C=0,B=1)$ и так далі. Використовуючи ці дані, оцінка максимальної правдоподібності вершини D обчислюється як:

$$P(D=d \mid B=b, C=c)=\frac{N(D=d, C=c, B=b)}{N(C=c, B=b)}=$$

$$=\frac{N(D=d, C=c, B=b)}{N(D=0, C=c, B=b)+N(D=1, C=c, B=b)}$$

Якщо вершини описані функцією Гауса, то можна підрахувати вибіркове середнє і дисперсію, а потім за допомогою лінійної регресії обчислити матрицю ваг. Для інших видів розподілу використовують більш складні процедури.

У випадку невідомої структури та повних спостережень **н**айбільш ймовірною моделлю в даному випадку є повний граф, тому що в цьому випадку буде задіяно найбільшу кількість параметрів; отже, така модель найбільше відповідатиме даним.

Формула Байєса (формула 1.1.) має вигляд:

$$P(G \mid D)=\frac{P(D \mid G) \cdot P(G)}{P(D)},$$

де G - спрямований ациклічний граф, що відповідає випадковим змінним, а $D=\{x^1,...,x^N\}$ - множина даних.

Прологарифмуємо цю формулу:

$$\log(P(G \mid D))=\log(P(D \mid G))+\log(P(G))+(-\log(P(D))), \qquad (2.2)$$

У формулі (2.2) доданок $-\log(P(D))$ грає роль штрафуючої компоненти за надмірну складність моделі. Складова $P(D \mid G)$ також може виступати штрафуючою компонентою за надмірно складні моделі (цей випадок відомий як лезо Оккама).

Для виконання точних розрахунків, пов'язаних з вибором моделі, необхідно обчислити $P(D) = \sum_G P(D \mid G)$, що є завданням експоненційної складності. Замість цього можна використовувати байєсівський інформаційний критерій, який визначається у даному випадку як:

$$\log(P(G \mid D)) \approx \log(P(D \mid G, \hat{\theta}_G)) - \frac{\log(N)}{2} \cdot \dim(G), \qquad (2.3)$$

де N - кількість моделей; $\dim(G)$ – розмірність моделі (кількість вільних параметрів); $\hat{\theta}_G$ - максимально правдоподібна оцінка параметрів, $-\frac{\log(N)}{2}\dim(G)$ - штрафуюча компонента за надмірно складні моделі.

Наступним кроком після вибору структури є навчання структури таким чином, щоб направлений ациклічний граф найкраще задовольняв даним. Це завдання є NP-складною задачею, тобто задачею з нелінійною поліноміальною оцінкою числа ітерацій. Тому зазвичай використовують локальні алгоритми пошуку, такі, як наприклад, жадібний метод пошуку екстремуму (greedy hill climbing method) або метод гілок і границь для пошуку в просторі графів.

Загалом, всі існуючі сучасні методи побудови структури мереж Байєса, як відзначають Дж. Ченг та В. Лиу [37, 38] можна умовно розділити на дві категорії:

- на основі оціночних функцій (search & scoring);
- методи із застосуванням тестів на умовну незалежність (dependency analysis).

Навчання мереж Байєса з прихованими вершинами має свої особливості. Багато реальних задач характеризується наявністю прихованих змінних (так званих латентних змінних), які не є спостережуваними і доступними для навчання [39].

Наприклад, історії хвороби часто включають описи спостережуваних симптомів, лікування, що застосовується, і, можливо, результати лікування, але рідко містять відомості про безпосереднє спостереження за розвитком самого захворювання. Очевидне питання: "Якщо хід розвитку захворювання не спостерігається, то чому б не побудувати модель без нього"? Відповідь на це питання можна знайти на рис. 2.2, де показана невелика фіктивна діагностична модель для сердечного захворювання. Існують три спостережувані фактори, що сприяють або перешкоджають розвитку захворювання, і три спостережувані симптоми. Передбачається, що кожна змінна має три можливі стани (наприклад, none (відсутній), moderate (помірний) і severe (серйозний)). Видалення прихованої змінної з мережі, наведеної на рис. 2.2а), спричиняє створення мережі, показаної на рис. 2.2б), і при цьому загальна кількість параметрів збільшується з 78 до 708. Таким чином, приховані змінні дозволяють різко зменшити кількість параметрів, потрібних для визначення байєсової мережі.

А даний факт, у свою чергу, дозволяє різко зменшити об'єм даних, необхідних для визначення в процесі навчання даних параметрів.

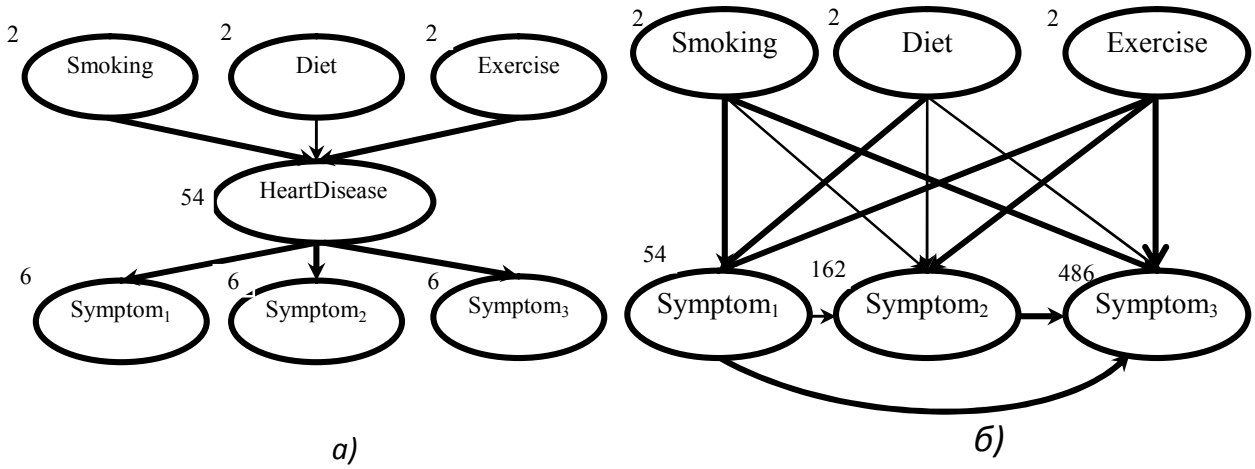


Рис. 2.2. Ілюстрація використання прихованих змінних

Приховані змінні важливі, але їх введення призводить до ускладнення задачі навчання. Наприклад, на рис. 2.2а не показано, як знайти в процесі навчання розподіл умовних ймовірностей для змінної HeartDisease, якщо задані її батьківські змінні, оскільки значення змінної HeartDisease у кожному випадку невідоме; така ж проблема виникає при пошуку в процесі навчання розподілів ймовірностей для симптомів.

Таким чином, перевагами використання прихованих вершин у мережах Байєса є:

– значне спрощення структури мережі;

– зменшення кількості параметрів, необхідних для визначення мережі Байєса, а це в свою чергу дозволяє різко зменшити об'єм даних, необхідних для визначення у процесі навчання цих параметрів.

Недоліки використання прихованих вершин в мережах Байєса:

– ускладнення задачі навчання мережі Байєса;

– у разі невідомої структури мережі невирішеною залишається проблема визначення кількості прихованих вершин, місце їх розташування та кількість станів кожної з них.

Іноді інформація щодо прихованих вершин в мережах Байєса є. Проте в загальному випадку не має ніяких даних щодо прихованої вершини. Відповідно постає дві основні проблеми навчання мереж Байєса з прихованими вершинами:

1. Скільки станів повинна мати прихована вершина?

2. Як умовні ймовірності можуть бути знайдені без даних?

Кількість станів можна визначити експериментально. Оскільки прихована вершина насправді діє як випадкова причина, то очікувано, що кількість станів буде відповідати кількості станів вершин, які вона розділяє. Насправді, кількість станів можна встановити як максимальне з числа станів предків і нащадків і потім видаляти стани, якщо вони матимуть в остаточному випадку досить низькі ймовірності.

Одними з перших дослідників проблеми знаходження умовних ймовірностей або навчання (оцінювання) параметрів прихованих вершин за умови відомої топології самої мережі, були Дж.-Л. Голмарда та А. Малет [40], які ще у 1991 описали алгоритм побудови мереж деревовидної структури. Пізніше

Д. Шпігельгалтером та Р. Коуеллом [41,42], К. Ольсеном, С. Лаурітценом, Ф. Дженсеном та ін. [42, 43] був досліджений загальний випадок для спрямованих ациклічних графів. У даних роботах описується алгоритм максимізації математичного очікування (EM-алгоритм [44,45]) для ймовірнісних мереж. Деякі дослідники зазначали, що існують певні проблеми із застосуванням EM-алгоритму для вирішення цієї проблеми, а тому запропонували, як можливу альтернативу, метод градієнтного спуску, який також дозволяє знайти локальний максимум на множині правдоподібності, що визначена параметрами множини. Активні дослідження щодо прямого порівняння обох методів тривають. Серед інших методів розв'язання проблеми навчання мереж Байєса з прихованими вершинами та відомою структурою варто виділити методи застосування вибірки Гілкса [46] та послідовного оновлення [47].

Процес навчання мережі Байєса складається із зовнішнього циклу, що є процесом пошуку структури, і внутрішнього циклу, що є процесом оптимізації параметрів. У разі повних даних внутрішній цикл виконується дуже швидко, оскільки при цьому досить лише витягнути інформацію про умовні частоти з набору даних. Якщо ж є приховані змінні, то для виконання внутрішнього циклу потрібно застосувати багато ітерацій алгоритму EM або алгоритму на основі градієнта, а на кожній ітерації будуть обчислюватись розподіли апостеріорних ймовірностей в байєсівській мережі, що є NP–складною задачею.

2.3. Алгоритми навчання мереж Байеса з прихованими вершинами

Методи на основі градієнтного підходу для навчання параметрів мереж Байєса, що містять неспостережувані дані, були запропоновані у 1995 році Binder J., Koller D., Russell S. and Kanazawa K. [48-50]. Мета методу – знаходження оцінок параметрів мережі $w_{ijk} = P(X_i = x_{ij} \mid Pa_i = pa_{ik})$, максимізуючи логарифмічну функцію правдоподібності $\ln P_w(D)$. При цьому мають виконуватись наступні обмеження $w_{ijk} \in [0,1], \sum_j w_{ijk} = 1$. Алгоритм працює з вибіркою D (набори даних $D_1 ... D_M$), структурою мережі та початковими (випадково згенерованими) значеннями для таблиці умовних ймовірностей прихованих вершин.

Даний підхід ґрунтується на розгляді $P_w(D)$ як функції від значень записів таблиць умовних ймовірностей w. Це зменшує проблему знаходження максимуму багатопараметричної нелінійної функції. Насправді, легше максимізувати функцію логарифма правдоподібності $\ln P_w(D)$. Оскільки ці дві функції монотонно зв'язані, то максимізація однієї еквівалентна максимізації іншої.

Найпростіший варіант даного підходу – це алгоритм градієнтного підйому (т. зв. «hill climbing» – пошук екстремуму). Для кожної точки w обчислюється ∇w – градієнтний вектор частинних похідних за записами таблиці умовних ймовірностей. Потім алгоритм робить невеликий крок у напрямку градієнта до точки ∇w - градієнтний вектор частинних похідних по записах таблиці умовних ймовірностей. Потім алгоритм робить невеликий крок в напрямку градієнта до точки $w + \alpha \nabla w$, де

α - розмір кроку. Цей простий алгоритм сходиться до локального оптимуму при достатньо малих α.

Єдиною проблемою є те, що даний підхід має враховувати обмеження, що w складається зі значень умовних ймовірностей, тому $w_{ijk} \in [0,1]$. Більше того, у будь-якій таблиці умовних ймовірностей значення, що відповідають частковому умовному випадку (залежно від стану предків) повинні в сумі давати 1, тобто $\sum_j w_{ijk} = 1$.

Один з методів врахування цих обмежень полягає у проектуванні ∇w на множину обмежень. У загальному випадку даний тип проекції виконується через проекцію вектора градієнта на вектор, що є нормаллю до множини обмежень, і відніманням його від вихідного вектора градієнта. У цьому випадку маємо $\sum_j w_{ijk} = 1$ для кожного i,k. Якщо для заданих i,k є декілька значень J для кожного j, вектор, ортонормальний до множини обмежень, матиме $1/\sqrt{J}$ у кожній відповідній компоненті. Таким чином, проекція градієнта на цей вектор матиме в кожній позиції ijk середнє значення компонентів градієнта, що відповідатиме $w_{i1k},...,w_{iJk}$. Віднімаючи цей вектор від вихідного вектора градієнта, отримаємо перенормований вектор як результат.

У перенормованому векторі градієнту сума компонентів, що відповідають w_{ijk} при фіксованих i,k дорівнює нулю. Таким чином, якщо взяти малий крок за напрямком цього вектора, сума $\sum_j w_{ijk}$ залишається незмінною.

Звідси, якщо почати роботу на множині обмежень, то обов'язково залишимось на цій множині, як і було задано. Алгоритм припиняє свою роботу, коли знайдений локальний максимум, тобто коли перенормований градієнт дорівнює нулю.

Ефективність використання градієнтних методів залежить від вибору ефективного методу обчислення градієнтів. Це є одним із ключових моментів успіху градієнтного спуску в нейронних мережах. Там зворотне поширення похибки (back-propagation) використовується для обчислення градієнта функції похибки по відношенню до параметрів мережі (параметрів зв'язків). Існування локального навчального алгоритму, що використовує результати виводу, спрощує проектування і реалізацію систем на основі нейронних мереж.

Схожа ситуація спостерігається і в байєсівських мережах. Фактично, для ймовірнісних мереж алгоритм виводу вже об'єднує всі необхідні обчислення, тому немає необхідності у будь-якому додатковому зворотному поширенні похибки. Градієнт може бути обчислений локально для кожної вершини з використанням інформації, що доступна з нормальних обчислень мережі Байєса.

Обчислимо вклад до градієнта кожного набору даних окремо і потім просумуємо результат:

$$\frac{\partial \ln P_W(D)}{\partial w_{ijk}} = \frac{\partial \ln \prod_{l=1}^{m} P_W(D_l)}{\partial w_{ijk}} = \sum_{l=1}^{m} \frac{\partial \ln P_W(D_l)}{\partial w_{ijk}} = \sum_{l=1}^{m} \frac{\partial P_W(D_l) / \partial w_{ijk}}{\partial P_W(D_l)}, \qquad (2.4)$$

Знайдемо вираз для обчислення $\sum_{l=1}^{m} \frac{\partial P_W(D_l)/\partial w_{ijk}}{\partial P_W(D_l)}$.

$$\frac{\partial P_W(D_l)/\partial w_{ijk}}{\partial P_W(D_l)} = \frac{\frac{\partial}{\partial w_{ijk}}\left(\sum_{j',k'} P_W\left(D_l \middle| x_{ij'}, u_{ik'}\right) P_W\left(x_{ij'}, u_{ik'}\right)\right)}{P_W(D_l)} = \frac{\frac{\partial}{\partial w_{ijk}}\left(\sum_{j',k'} P_W\left(D_l \middle| x_{ij'}, u_{ik'}\right) P_W\left(x_{ij'} \middle| u_{ik'}\right) P_W\left(u_{ik'}\right)\right)}{P_W(D_l)}, \quad (2.5)$$

Як видно з формули (2.5) w_{ijk} з'являється лише в лінійній формі. Фактично, w_{ijk} з'являється лише в одному члені суми, для якого $j' = j,\ k' = k$. Для цього випадку $P_W\left(x_{ij'} \middle| u_{ik'}\right) = w_{ijk}$. Звідси,

$$\frac{\partial P_W(D_l)/\partial w_{ijk}}{\partial P_W(D_l)} = \frac{P_W\left(D_l \middle| x_{ij}, pa_{ik}\right) P_W\left(pa_{ik}\right)}{P_W(D_l)} = \frac{P_W\left(x_{ij}, pa_{ik} \middle| D_l\right) P_W(D_l) P_W\left(pa_{ik}\right)}{P_W\left(x_{ij}, u_{ik}\right) P_W(D_l)} = \frac{P_W\left(x_{ij}, pa_{ik} \middle| D_l\right)}{P_W\left(x_{ij}, \middle| pa_{ik}\right)} = \frac{P_W\left(x_{ij}, pa_{ik} \middle| D_l\right)}{w_{ijk}} \quad (2.6)$$

Рівняння (2.5) дозволяє обчислювати градієнт разом з обчисленням апостеріорних ймовірностей з використанням звичайних операцій з мережами Байєса. По суті, будь-який стандартний алгоритм виводу для мереж Байєса для набору даних D_l обчислить значення $P_w(x_{ij}, pa_{ik} \mid D_l)$ як побічний продукт.

Вектор градієнту тепер може бути отриманий таким чином. Використовуючи алгоритм ймовірнісного висновку для кожного набору даних, окремо обчислюються $P_w(x_{ij}, pa_{ik} \mid D_l)$ для кожного ijk. Потім ці значення додаються для всіх сукупностей даних і діляться на w_{ijk}.

Алгоритм має вигляд:

repeat until $\Delta w \approx 0$

$\Delta w \leftarrow 0$

for each $D_i \in D$

ймовірнісний висновок в мережі по Dl

for each змінної i, стану j, умовного стану k

$$\Delta w_{ij} \leftarrow \Delta w_{ijk} + \frac{\partial \ln P_w(D)}{\partial w_{ijk}} = \frac{P_w(x_{ij}, pa_{ik} \middle| D_l}{w_{ijk}}$$

$\Delta w_{ijk} \leftarrow$ *проекція на простір обмежень*

$w \leftarrow w + \alpha \Delta w$

Таким чином, отримано алгоритм навчання мереж Байєса, що містять приховані змінні. Він має назву ***APN*** (adaptive probabilistic network) алгоритм. Таку мережу називають адаптивною ймовірнісною мережею (adaptive probabilistic network або скорочено APN).

Алгоритм expectation maximization (EM) застосовується, коли деякі вершини мережі Байєса приховані, а структура відома для знаходження локально оптимальних оцінок максимальної правдоподібності для параметрів. EM-алгоритм або алгоритм максимізації математичного очікування, запропонований ще у 1977 році А. П. Демпстером, Н. М.Лаердом та Д. Б. Рубіном [44] і адаптований для навчання мереж Байєса з прихованими вершинами та відомою структурою С. Л. Лауритценом [51].

Алгоритм складається з двох базових кроків E (expectation) та M (maximization). Головна ідея алгоритму полягає в тому, що якби були відомі значення всіх вершин, то навчання (на кроці M) було б простим, оскільки була б наявна вся необхідна інформація. Тому спочатку на кроці E робиться обчислення значення очікуваної правдоподібності (expectation of the likelihood) включаючи латентні змінні, так ніби є можливість їх спостерігати. Фактично робиться "доповнення" даних для прихованих змінних на основі поточних оцінок параметрів мережі $\theta^{(i)}$, тобто знаходиться розподіл апостеріорних ймовірностей по прихованих змінних за наявності отриманих даних. На кроці M обчислюються значення оцінок максимальної правдоподібності для параметрів, використовуючи максимізацію значень очікуваної правдоподібності, отриманих на кроці E. Далі алгоритм рекурсивно виконує крок E з використанням параметрів $\theta^{(i+1)}$, отриманих на кроці M, і так далі.

Нехай $X = \{X_1,..., X_n\}$ - вершини мережі Байєса; $\Theta = \{\theta_{ijk}\}, 1 \le i \le n, 1 \le j \le q_i, 1 \le k \le r_i$ - множина таблиць умовних ймовірностей $P(X_i = x_k \mid Pa(X_i) = pa_j)$ (де q_i та r_i кількість станів вершини X_i та її предків $Pa(X_i)$ відповідно); $D = \{X_i^l\}, 1 \le i \le n, 1 \le l \le N$ - множина даних, а D_m - дані прихованих змінних, які відсутні в навчальній вибірці.

Нехай $\log P(D \mid \Theta)$ - логарифмічна функція правдоподібності параметрів при заданому наборі даних. Встановивши певні початкові значення Θ^*, можна обчислити ймовірнісний розподіл для прихованих змінних $P(D_m \mid \Theta^*)$ і потім обчислити $Q(\Theta : \Theta^*)$ - математичне очікування попередньої логарифмічної функції правдоподібності:

$$Q(\Theta : \Theta^*) = E_{\Theta^*} \log P(D \mid \Theta), \tag{2.7}$$

Рівняння (2.7) може бути записане в наступному вигляді:

$$Q(\Theta : \Theta^*) = \sum_i \sum_{X_i = x_k} \sum_{P_i = pa_j} N_{ijk}^* \log P_{\theta_{ijk}}(D), \tag{2.8}$$

де $N_{ijk}^* = E_{\Theta^*}[N_{ijk}] = N \cdot P(X_i = x_k, P_i = pa_j \mid \Theta^*)$ отримується за допомогою

ймовірнісного висновку в мережі Байєса.

Схема ЕМ-алгоритму:

Крок 1: ***i=0***
вибір
Крок 2: ***repeat***
Крок 3: ***i=i+1***
Крок 4: $\Theta^i = \arg_\theta \max Q(\Theta : \Theta^{i-1})$
Крок 5: ***until*** $Q(\Theta : \Theta^{i-1}) \le Q(\Theta^{i-1} : \Theta^{i-1})$

Таким чином, на кроці E оцінюється значення N^* на основі параметрів Θ^i, а на кроці M вибирається найкраще значення параметрів Θ^{i+1}, максимізуючи Q.

У роботі [44] було доведено, що алгоритм сходиться, а також факт того, що не обов'язково буде знайдено глобальний оптимум функції $Q(\Theta : \Theta^*)$. Наявність локальних оптимумів є серйозним недоліком даного алгоритму. Одне з рішень полягає в накладенні розподілів апріорної вірогідності на параметри моделі і застосуванні версії MAP (Maximum a posteriori) алгоритму EM. Ще одне рішення полягає у перезапуску обчислень з новими, випадково вибраними параметрами. Іноді також має сенс вибирати для ініціалізації параметрів більш обґрунтовані значення.

На основі розглянутого принципу ЕМ-алгоритму можливо визначити його певні варіанти і удосконалення. Наприклад, дуже часто Е-крок (обчислення розподілів апостеріорних ймовірностей по прихованих змінних) досить трудомісткий у випадку побудови великих байєсових мереж. У цьому випадку можна застосувати наближений Е-крок, і алгоритм навчання буде залишатись ефективним. При використанні алгоритму формування вибірки за методом Монте-Карло для марковських ланцюгів (МКМЛ) процес навчання стає повністю інтуїтивно зрозумілим: кожен стан (конфігурація прихованих і спостережуваних змінних), який згенерував алгоритм за методом Монте-Карло для марковських ланцюгів, розглядається точно так, ніби він був повним спостереженням. Тому параметри можуть оновлюватися безпосередньо після кожного переходу із стану в стан в алгоритмі за методом Монте-Карло для марковських ланцюгів. При оцінюванні параметрів дуже великих мереж шляхом навчання також показали свою високу ефективність інші методи формування наближеного ймовірнісного висновку – варіаційні та циклічні.

Якщо ж доводиться враховувати приховані змінні, то завдання ускладнюється. У простому випадку приховані змінні можуть бути включені в загальний список разом із спостережуваними змінними; хоча їх значення не спостерігаються, алгоритм навчання отримує відомості про те, що вони існують, і повинен знайти для них місце в структурі мережі. Наприклад, за допомогою алгоритму може бути зроблена спроба визначити в процесі навчання структуру, показану на рис. 2.2 на основі тієї інформації, що в модель має бути включена змінна HeartDisease (тризначна змінна).

Якщо ж алгоритму навчання не надана ця інформація, то в процесі навчання виникають два варіанти: або виходити з того, що дані дійсно є повними (що змушує алгоритм визначити в процесі навчання модель з набагато більшою кількістю

параметрів, показану на рис. 2.2, б), або винайти нові приховані змінні для спрощення моделі. Останній підхід можна реалізувати шляхом включення нових варіантів операцій модифікації в процедуру пошуку структури - передбачити в алгоритмі можливість не лише модифікувати зв'язки, але і додавати або видаляти приховані змінні, або міняти їх арність.

Безумовно, в такому алгоритмі не можна передбачити можливість назвати знову винайдену змінну, допустимий HeartDisease, а також присвоювати її значенням осмислені імена. Але знову винайдені приховані змінні зазвичай зв'язуються з раніше існуючими змінними, тому людина - фахівець у цій проблемній області часто може перевіряти локальні розподіли умовної вірогідності, що стосуються нової змінної, і встановлювати її сенс.

Розглянемо випадок, зображений на рис. 2.3, коли є два пакети з цукерками, вміст яких згодом змішали. Для опису цукерок застосовано три характеристики: окрім різновиду Flavor і обгортки Wrapper, застосовано ще характеристику "з отворами" (Hole), оскільки частина цукерок – льодяники з отворами (Hole) всередині, а деякі – льодяники без отворів.

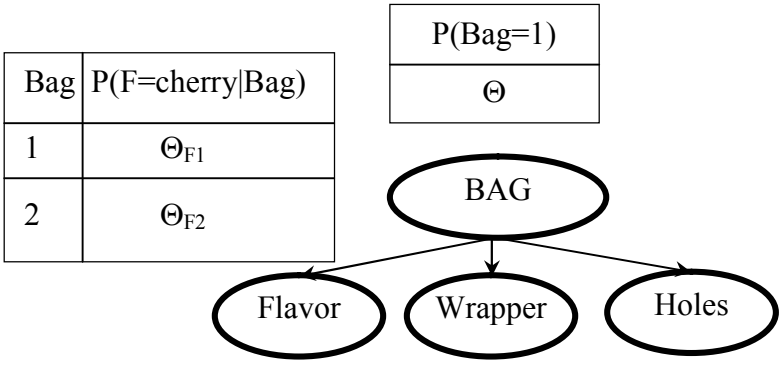


Рис. 2.3. Наївна мережа Байєса для ситуації розподілу цукерок по пакетах

Розподіл цукерок у кожному пакеті описано за допомогою наївної байєсівської моделі: у кожному конкретному пакеті характеристики незалежні між собою, але розподіл умовних ймовірностей кожної характеристики залежить від пакету. При цьому застосовано такі параметри: θ - апріорна ймовірність того, що цукерка узята з пакета Bag1; θ_{F1} і θ_{F2} - ймовірності того, що цукерка є вишневим льодяником, за умови, що вона взята з пакету Bag_1 і Bag_2 відповідно; θ_{W1} і θ_{W2} задають ймовірності того, що обгортка має червоний колір; а θ_{H1} і θ_{H2} - ймовірності того, що льодяник має отвір.

На рис. 2.3 прихована змінна відповідає просто пакету, оскільки після змішування цукерок немає можливості визначити, з якого пакету була взята кожна цукерка. Завдання полягає у відновленні опису цих двох пакетів, маючи лише характеристики цукерок, узятих з цієї суміші. Проведемо одну ітерацію алгоритму EM для розв'язання цієї задачі. Спочатку випадково згенеруємо 1000 вибірок з моделі, істинними параметрами якої є такі:

$$\theta = 0{,}5,\ \theta_{F1} = \theta_{W1} = \theta_{H1} = 0{,}8,\ \theta_{F2} = \theta_{W2} = \theta_{H2} = 0{,}2\,.$$

Це означає рівноймовірне отримання цукерок з одного або іншого пакету: у

першому пакеті в основному знаходяться вишневі льодяники в червоних обгортках і з отворами, а в другому – в основному лимонні льодяники в зелених обгортках і без отворів. Кількість цукерок восьми можливих різновидів визначена у таблиці 2.2.

Таблиця 2.2

Кількості можливих різновидів цукерок

	W = red		**W = green**	
	H = 1	**H = 0**	**H = 1**	**H = 0**
F = cherry	273	93	104	90
F = lime	79	100	94	167

Функціонування ЕМ-алгоритму розпочинається з початкової ініціалізації параметрів. Наприклад, виберемо такі значення параметрів:

$$\theta^{(0)} = 0{,}6,\ \theta_{F1}^{(0)} = \theta_{W1}^{(0)} = \theta_{W1}^{(0)} = 0{,}6,\ \theta_{F2}^{(0)} = \theta_{W2}^{(0)} = \theta_{H2}^{(0)} = 0{,}4$$

Обчислимо оцінку параметра θ . У випадку повних спостережень можна було б отримати оцінку для цього параметра безпосередньо із спостережуваних значень кількості цукерок, що відносяться до пакетів 1 і 2. Але оскільки номер пакету – це прихована змінна, розрахуємо замість цього очікувані значення кількості. Очікувана кількість N(Bag=l) є сумою ймовірностей того, що цукерка узята з пакету 1, по усіх цукерках (2.9):

$$\theta^{(1)} = \hat{N}\left(Bag = 1\right)/N = \sum_{j=1}^{N} P\left(Bag = 1 \middle| flavor_j, wrapper_j, holesj\right)/N\ . \qquad (2.9)$$

Цю ймовірність можна обчислити за допомогою будь-якого алгоритму імовірнісного виводу для байєсових мереж. А в даному випадку, наприклад, можна використати правило Байєса і застосувати вираз для умовної незалежності:

$$\theta^{(1)} = \frac{1}{N}\sum_{j=1}^{N} \frac{P\left(flavor_j \middle| Bag=1\right)P\left(wrapper_j \middle| Bag=1\right)P\left(holes_j \middle| Bag=1\right)P\left(Bag=1\right)}{\sum_i P\left(flavor_j \middle| Bag=i\right)P\left(wrapper_j \middle| Bag=i\right)P\left(holes_j \middle| Bag=i\right)P\left(Bag=i\right)}. \qquad (2.10)$$

Варто звернути увагу на те, що нормуюча константа також залежить від параметрів. Зробивши розрахунки для всіх видів цукерок, кількість яких вказана у таблиці 2.3, отримаємо, що $\theta^{(1)} = 0.6124$.

Розглянемо процес оцінювання інших параметрів, наприклад, θ_{F1}. У випадку повних спостережень, це значення можна було б оцінити безпосередньо на основі спостережуваних значень кількості вишневих і лимонних льодяників з пакету 1. Очікувана кількість вишневих льодяників з пакету 1 обчислюється за допомогою наступного виразу:

$$\sum_{j:Flavor_j = cherry} P\left(Bag = 1 \middle| Flavor_j = cherry, wrapper_j, holes_j\right).$$

Цю ймовірність також можна обчислити за допомогою будь-якого алгоритму для байєсової мережі.

Продовжуючи цей процес, знайдемо оцінки для всіх параметрів:

$\theta^{(1)} = 0{,}6124$, $\theta_{F1}^{(1)} = 0{,}6684$, $\theta_{W1}^{(1)} = 0{,}6483$, $\theta_{H1}^{(1)} = 0{,}6558$,

$\theta_{F2}^{(1)} = 0{,}3887$, $\theta_{W2}^{(1)} = 0{,}3817$, $\theta_{H2}^{(1)} = 0{,}3827$.

Логарифмічна правдоподібність даних зростає від первинного значення, приблизно рівного, -2044, приблизно до -2021 після першої ітерації, як показано на рис. 2.4.

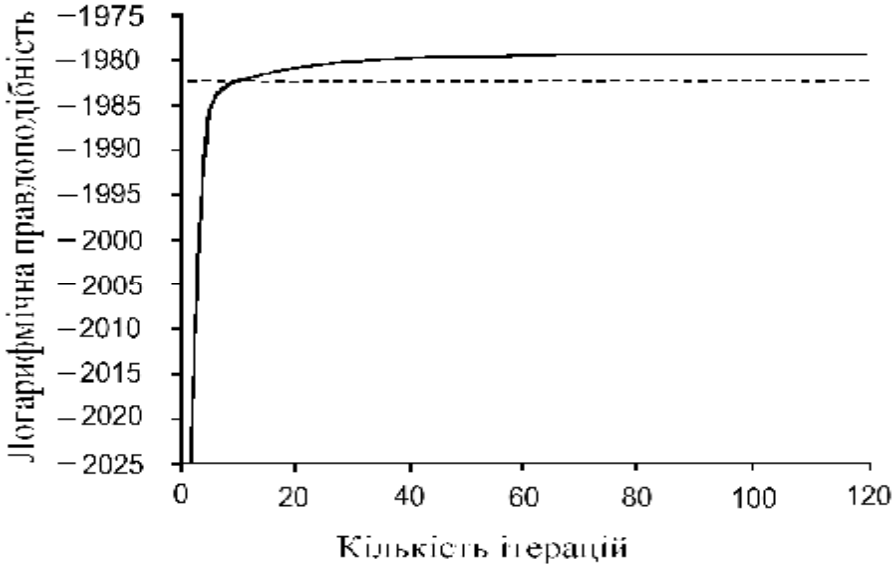


Рис. 2.4. Графік зміни значення логарифмічної правдоподібності залежно від кількості ітерацій ЕМ-алгоритму

До десятої ітерації модель, отримана у процесі навчання, краще відповідає даним, ніж первинна модель (L = -1982,214). Але подальший прогрес значно сповільнюється. Така ситуація в застосуваннях алгоритму ЕМ зустрічається дуже часто, тому в багатьох практично вживаних системах алгоритм ЕМ на останньому етапі навчання використовується у поєднанні з таким алгоритмом на основі градієнта, як алгоритм Ньютона-Рафсона.

Загальний висновок, який можна зробити на підставі цього прикладу, полягає в тому, що оновлення параметрів при навчанні байєсової мережі з прихованими змінними є безпосередньо доступним за допомогою результатів формування імовірнісного висновку для кожного прикладу. Більше того, для кожного параметра необхідно мати тільки локальні апостеріорні ймовірності. Для загального випадку, у якому в процесі навчання оцінюються параметри умовних ймовірностей для кожної змінної X_i , якщо дані її батьківські змінні, інакше кажучи, $\theta_{ijk} = P\left(X_i = x_{ij} \mid Pa_i = pa_{ik}\right)$, оновлення задається за допомогою нормалізованих очікуваних кількостей таким чином:

$$\theta_{ijk} \leftarrow \hat{N}\left(X_i = x_{ij}, Pa_i = pa_{ik}\right) / \hat{N}\left(Pa_i = pa_{ik}\right).$$

Ці очікувані величини можна отримати шляхом обчислення сум для усіх прикладів і обчислення ймовірностей $P\left(X_i = x_{ij}, Pa_i = pa_{ik}\right)$для кожного з них з використанням будь-якого алгоритму імовірнісного висновку в мережах Байєса.

2.4. Особливості побудови мережі Байєса з прихованими вершинами та невідомою структурою

Одним із найскладніших випадів є побудова мереж Байєса з прихованими вершинами та невідомою структурою. Структурне навчання полягає у знаходженні правильних зв'язків між вершинами. Цікавішою проблемою є створення прихованих вершин. Приховані вершини можуть зробити модель компактнішою. Стандартний підхід полягає в додаванні щоразу по одній прихованій вершині до певної частини мережі, виконуючи структурне навчання на кожному кроці до тих пір, поки оцінка моделі не перестане зменшуватись. Ще одна проблема – це вибір можливої кількості прихованих вершин та типу розподілу їх умовних ймовірностей. Разом з цим стоїть задача вибору місця створення прихованої вершини. Немає сенсу робити її нащадком, оскільки приховані нащадки завжди можуть бути скорочені. Тому необхідно знайти існуючу вершину, яка потребує нового предка у випадку, коли поточна множина можливих предків недостатньо адекватна.

Одним з підходів може бути дослідження умовної ентропії $H(X \mid Pa(X))$. Якщо вона дуже велика, то це значить, що поточна множина не може "пояснити" залишкову ентропію; якщо $Pa(X)$ найкраща множина предків (за значенням BIC (Bayesian Information Criteria) або хі-квадрат), яку можливо знайти в поточній моделі, то це передбачає необхідність створення нової вершини і включення її до множини $Pa(X)$.

Для розв'язання завдання ***структурного навчання мережі Байєса*** з прихованими вершинами використовують ***структурний ЕМ-алгоритм*** або алгоритм стиснення границь.

Одним з удосконалень вищеописаного принципу максимізації математичного очікування є структурний ЕМ-алгоритм [52] (SEM – structural EM algorithm), який діє багато у чому так само, як і звичайний (параметричний) алгоритм ЕМ, за винятком того, що цей алгоритм може оновлювати не лише параметри, але і структуру моделі.

Схема алгоритму SEM має вигляд:

Крок 1: ***i=0***

вибір початкової мережі (G^0, Θ^0)

Крок 2: ***repeat***

Крок 3: ***i=i+1***

Крок 4: $(G^i, \Theta^i = \arg_{G,\theta} \max Q(G, \Theta : G^{i-1}, \Theta^{i-1})$

Крок 5: ***until*** $Q(G^i, \Theta^i : G^{i-1}, \Theta^{i-1}) \le Q(G^{i-1}, \Theta^{i-1} : G^{i-1}, \Theta^{i-1})$

Цей алгоритм навчає мережі на основі штрафних ймовірнісних значень, які включають значення, отримані за допомогою байєсівського інформаційного критерію (BIC), принципу мінімальної довжини (MDL), а також інших критеріїв. Як і у звичайному алгоритмі ЕМ, тут використовуються поточні параметри для обчислення очікуваних величин на Е-кроці, а потім ці величини застосовують на М-кроці для вибору нових параметрів. В алгоритмі структурного ЕМ використовується структура для обчислення очікуваних величин, після чого ці величини застосовують на М-

кроці для оцінювання правдоподібності нових потенційних структур (у цьому полягає відмінність цього методу від методу зовнішнього циклу/внутрішнього циклу, в якому обчислюються нові очікувані величини для кожної потенційної структури). Крок 4 даного алгоритму передбачає пошук найкращої структури та найкращої множини параметрів для даної структури. На практиці ці кроки чітко відрізняються:

$$G^i = \arg_G \max Q(G, \Theta^i : G^{i-1}, \Theta^{i-1}), \tag{2.13}$$

$$\Theta^i = \arg_\theta \max Q(G^i, \Theta : G^{i-1}, \Theta^{i-1}). \tag{2.14}$$

Перший пошук (2.13) у просторі можливих графів повертає до вихідної проблеми пошуку оптимальної структури мережі Байєса. Щодо пошуку у просторі параметрів (2.14), Н. Фрідменом [45] запропоновано повторювати цю процедуру декілька разів з використанням інтелектуальної ініціалізації. Даний крок означає запуск параметричного ЕМ-алгоритму для кожної структури G^i, починаючи із структури G^0 .

Таким чином, структурний алгоритм ЕМ дає можливість вносити до мережі декілька структурних змін без повторних обчислень очікуваних величин і має здатність визначати у процесі навчання нетривіальні структури байєсівських мереж. Проте необхідно виконати ще багато дослідницької роботи, перш ніж можна буде стверджувати, що завдання визначення структури в процесі навчання вирішене остаточно.

Структурний ЕМ-алгоритм дає можливість оцінювати параметри і структури мережі Байєса з визначеною кількістю прихованих змінних. Однак проблема визначення кількості прихованих вершин, які необхідно додавати до структури мережі, на сьогодні остаточно не вирішена. Тому існує гостра необхідність у створенні ефективних додаткових механізмів для розв'язання цієї задачі.

Ще одним методом структурного навчання мереж Байєса з прихованими вершинами є ***алгоритм стиснення границь*** (bound and collapse), який запропоновано і описано в роботах М. Рамони та П. Себастіані [53, 54].

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

2.5. Методика знаходження параметрів байєсівських мереж з прихованими вершинами

На основі розглянутого вище алгоритму максимізації математичного очікування розроблено методику обчислення параметрів мережі Байєса за умови неповної вхідної інформації та відомої топології мережі. Отримана методика повністю описує весь процес оцінювання невідомих параметрів прихованих вершин і складається з таких кроків:

1) побудова мережі Байєса за навчальними даними або “вручну”;
2) генерування вибірки по заданій структурі мережі (застосовується випадку відсутності навчальних даних);
3) додавання прихованих вершин до структури мережі;
4) початкова ініціалізація невідомих параметрів мережі;
5) обчислення оцінок параметрів мережі на основі згенерованих даних з використанням алгоритму EM.

Схематичне зображення запропонованої методики показано на рис. 2.6.

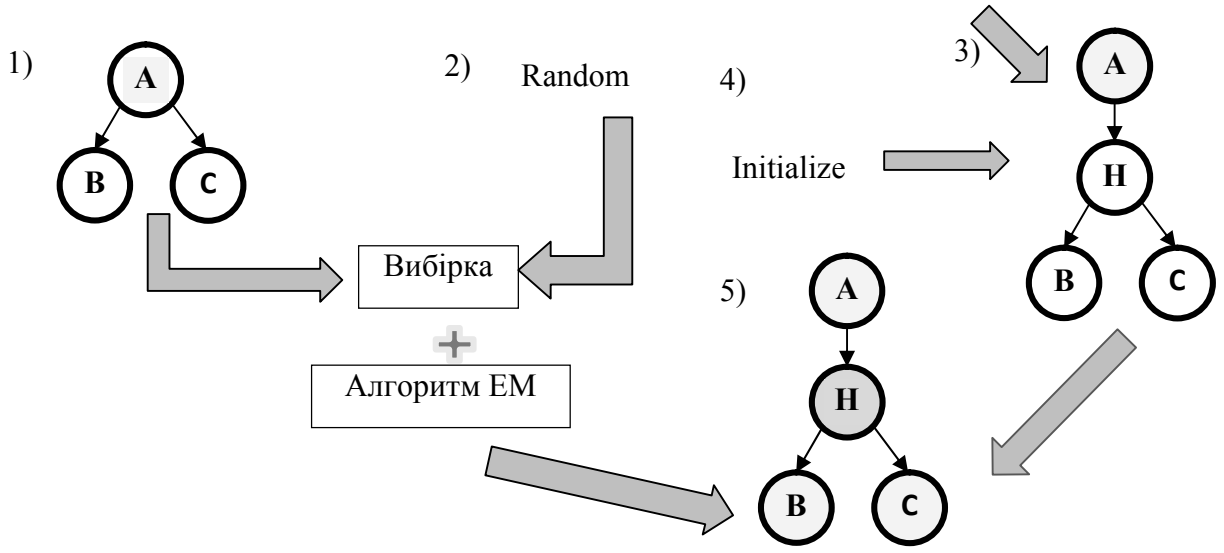


Рис. 2.6. Схема методики знаходження параметрів мережі Байєса з прихованими вершинами

На першому етапі необхідно побудувати мережу Байєса, тобто визначити топологію і знайти значення параметрів всієї мережі. На цьому етапі можливі *два варіанти*:

1) структура мережі нам відома і тоді залишається перенести її у програмне середовище і заповнити таблиці умовних ймовірностей;

2) є лише навчальні дані.

У другому випадку побудова структури здійснюється за два кроки: на першому кроці будується топологія мережі з використанням, наприклад евристичного алгоритму, а на другому обчислюють оцінки параметрів мережі, які максимально правдоподібні до навчальних даних.

На другому етапі у випадку, коли навчальні дані не задані або їх недостатньо, відбувається генерування псевдовипадкової вибірки за побудованою на першому етапі структурою мережі. Генерування відбувається таким чином. Спочатку формується ймовірнісний висновок у мережі без інстанційованих вершин $P(S_{ij})$, $i=1,...,N;\ \ j=1,...,S$. Далі обирається вершина N_{i^*} та інстанціюється один із її станів $S_{i^*j^*}$ із ймовірністю цього стану $P(S_{i^*j^*})$. Потім перераховуються ймовірності станів вершин після інстанціювання $P(S_{ij}\left|S_{i^*}=S_{i^*j^*}\right.)$, а далі обирається наступна

вершина. Ця операція повторюється доти, доки не залишиться неінстанційованних вершин. Інстанційовані стани утворюють запис у вибірці $(S_{i_1}...S_{i_N})$, $i_k \in (1,...S)$.

Алгоритм повторюється до тих пір, доки не буде створено необхідної кількості записів у вибірці.

На третьому етапі відбувається додавання прихованої вершини до мережі. Якщо ця вершина вставляється між декількома існуючими, то видаляються попередні дуги між ними і створюються нові, які пов'язані з прихованим вузлом. Якщо необхідно додати приховану батьківську вершину, то просто створюється нова вершина і відповідні дуги.

На наступному етапі параметри прихованих вершин ініціалізуються початковими значеннями. Це можуть бути як випадково згенеровані значення, так і значення, які надаються експертом.

На завершальному етапі запускається ітераційний процес алгоритму EM, який використовує попередньо згенеровану вибірку даних і результатом якого є оцінки невідомих параметрів прихованих вершин мереж Байєса.

2.6. Приклади практичного застосування мереж Байєса з прихованими вершинами

В якості прикладів практичного застосування мереж з прихованими вершинами, розглянемо класичні байєсівські мережі: Asia, Alarm, Car Starts.

Ці мережі входять до стандартного набору прикладів майже всіх відомих програмних продуктів для побудови і застосування мереж Байєса. Вони є еталонними мережами Байєса, які застосовуються дослідниками для тестування нових та існуючих алгоритмів побудови мереж та ймовірнісного висновку.

Процес тестування побудованих мереж за використання еталонних виконується наступним чином. Для тестування використовується або мережа повністю, або певна її частина (у випадку великої мережі для підвищення швидкості навчання). Видаляється, а потім знову додається вершина з деякими початковими значеннями для таблиці умовних ймовірностей тієї вершини, яка має бути прихованою і виконується алгоритм навчання параметрів прихованих вершин.

Результати роботи програми аналізуються таким чином: порівнюються істинні значення ймовірностей станів прихованої вершини, а також таблиць умовних ймовірностей зі значеннями, отриманими в результаті проведення експерименту. Також досліджується залежність логарифмічної функції правдоподібності мережі від кількості ітерацій, виконаних за алгоритмом максимізації математичного очікування.

Мережа Alarm – A Logical Alarm Reduction Mechanism. Мережа Alarm була описана і вперше використана при дослідженні мереж Байєса у 1989 році в роботі [55]. Ця мережа містить 37 вершин та 46 дуг. Вона описує систему повідомлення тривоги для контролю стану пацієнтів лікарні; при цьому обчислюються ймовірності для диференційного діагнозу на основі доступних даних. Медичні знання закодовані у графічній структурі, яка пов'язує 8 діагнозів, 16 відомостей та 13 проміжних змінних. Топологію мережі зображено на рис. 2.7.

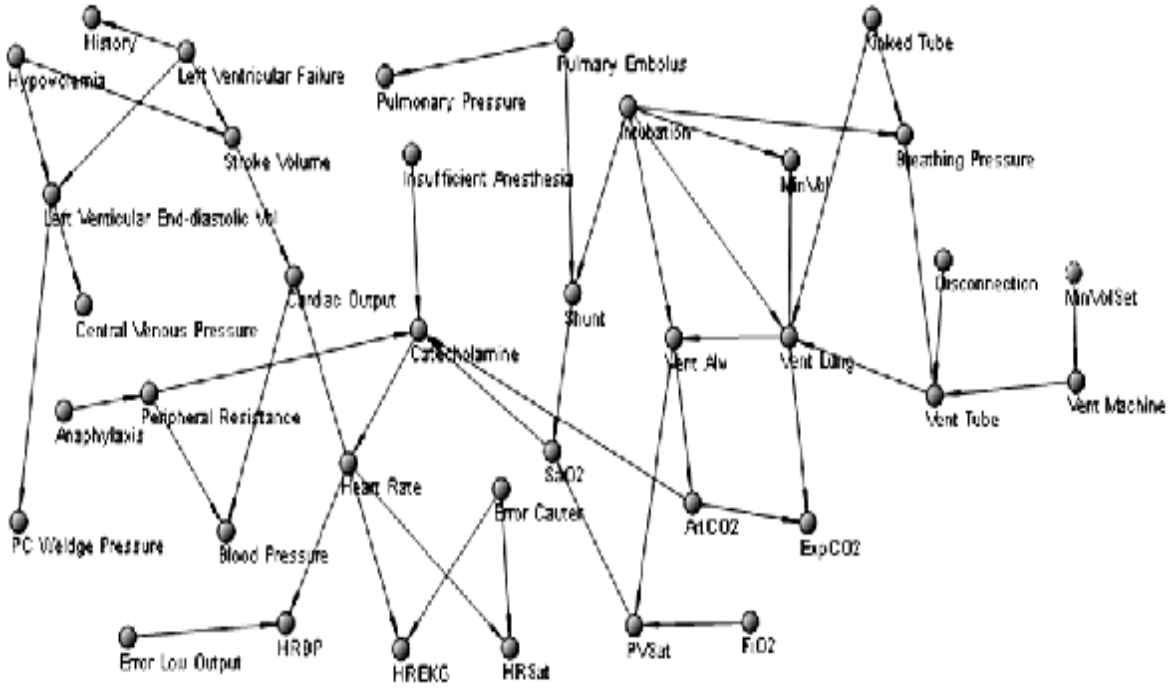


Рис. 2.7. Топологія мережі "Alarm"

Оскільки мережа Alarm є досить великою за кількістю вершин, то для використання методики знаходження невідомих параметрів прихованих вершин необхідно мати значні обчислювальні ресурси.

Для зменшення часу функціонування тесту розглянуто лише фрагмент мережі, над яким був проведено відповідний експеримент.

Цей фрагмент зображено на рис. 2.8, прихованою є вершина Left Ventricular End-diagnostic Vol.

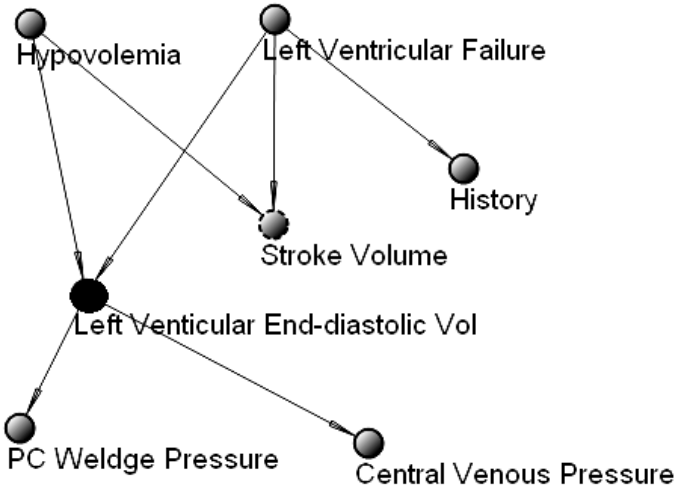


Рис. 2.8. Фрагмент мережі Alarm з введенням прихованої вершини

У таблиці 2.4 наведено істинні значення ймовірностей станів прихованої вершини та оцінки цих значень, отримані за допомогою програми BNetMaster. Як бачимо, ці значення є дуже близькими між собою.

Таблиця 2.4

Значення ймовірностей станів прихованої вершини

Стан	Початкові значення	Експериментальні оцінки
Low	0,230	0,251
Normal	0,690	0,694
High	0,076	0,055

У таблицях 2.5 та 2.6 наведено умовні ймовірності прихованої вершини: відповідно, початкова та експериментально обчислена. Ці значення є також дуже близькими між собою.

Таблиця 2.5

Істинна таблиця умовних ймовірностей прихованої вершини

Hypovolemia	LVFailure	Low	Normal	High
TRUE	TRUE	0,95	0,04	0,01
TRUE	FALSE	0,98	0,01	0,01
FALSE	TRUE	0,01	0,09	0,9
FALSE	FALSE	0,05	0,9	0,05

Таблиця 2.6

Експериментально обчислена таблиця умовних ймовірностей прихованої вершини

Hypovolemia	LVFailure	Low	Normal	High
TRUE	TRUE	0,99	0,01	0
TRUE	FALSE	0,98	0,01	0,01
FALSE	TRUE	0,01	0,23	0,76
FALSE	FALSE	0,06	0,9	0,04

На рис. 2.9 зображено залежність логарифмічної функції правдоподібності від кількості ітерацій алгоритму EM.

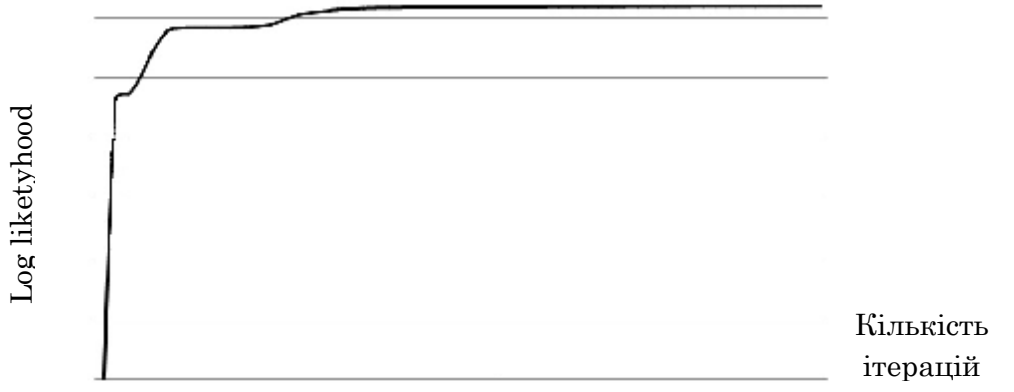


Рис. 2.9. Логарифмічна функція правдоподібності для мережі "Alarm"

З рис. 2.9 видно, що з кожною ітерацією значення функції правдоподібності зростає, що свідчить про покращення параметрів мережі. У цілому за весь процес навчання логарифм функції правдоподібності збільшився з -5,47 до -1,90.

Мережа Asia [56] - це спрощена версія мережі Байєса, що може використовуватись для діагностування пацієнта. Так визначається, чи хворіє пацієнт на туберкульоз ("Tuberculosis"), рак легень ("LungCancer") чи бронхіт ("Bronchitis") залежно від його рентгенівського знімку ("X-Ray"), наявності задишки ("Dyspnea"), перебування в Азії ("Visit to Asia") та паління ("Smoker"). Топологія мережі представлена на рис. 2.10.

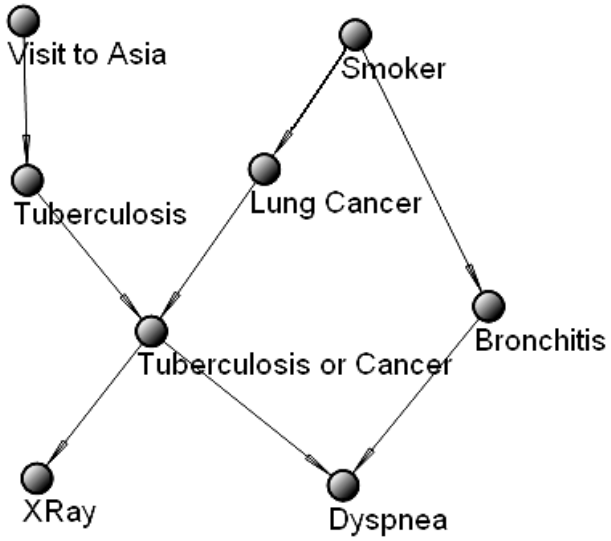


Рис. 2.10. Топологія мережі Asia

Мережа "Asia" містить 8 вершин та 8 дуг. Кожна вершина мережі відповідає певному стану пацієнта, наприклад, "Visit to Asia" показує чи відвідував пацієнт Азію останнім часом.

Ще одна назва мережі – "Chest Clinic". В якості прихованої вибиралась вершина "Tuberculosis or Cancer".

У таблиці 2.5 наведено реальні значення ймовірностей станів прихованої вершини та оцінки цих значень, отримані в результаті роботи програми BNetMaster. Похибка між значеннями є незначною.

Таблиця 2.5

Значення ймовірностей станів прихованої вершини

Стан	**Початкові значення**	**Експериментальні оцінки**
True	0,065	0,054
False	0,935	0,946

У таблицях 2.6 та 2.7 наведено умовні ймовірності прихованої вершини: відповідно, початкова та експериментально обчислена.

Отримані значення повторюють фактичні.

Таблиця 2.6

Істинна таблиця умовних ймовірностей прихованої вершини

Tuberculosis	**LungCancer**	**True**	**False**
Present	present	100	0
Present	absent	0	100
Absent	present	100	0
Absent	absent	0	100

Таблиця 2.7

Експериментально обчислена таблиця умовних ймовірностей прихованої вершини

Tuberculosis	**Lung Cancer**	**True**	**False**
Present	present	0,99	0,01
Present	absent	0,01	0,99
Absent	present	100	0
Absent	absent	0,01	0,99

На рис. 2.11 зображено залежність логарифмічної функції правдоподібності від кількості ітерацій алгоритму EM.

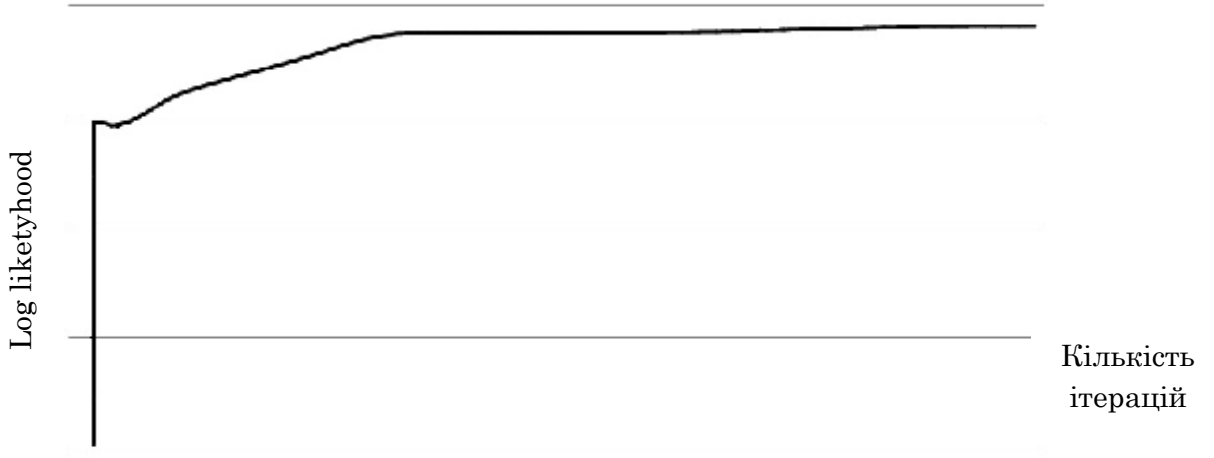


Рис. 2.11 Графік логарифмічної функції правдоподібності для мережі "Asia"

З рис. 2.11 видно, що з кожною ітерацією значення функції правдоподібності зростає, що свідчить про покращення якості оцінок параметрів мережі. У цілому за весь процес навчання логарифм функції правдоподібності збільшився з -4,99 до -2,24.

Наступним прикладом є **мережа CarStarts**, розроблена компанією Norsys, виробником програмного пакету для роботи з мережами Байеса Netica. Топологія мережі зображена на рис. 2.12.

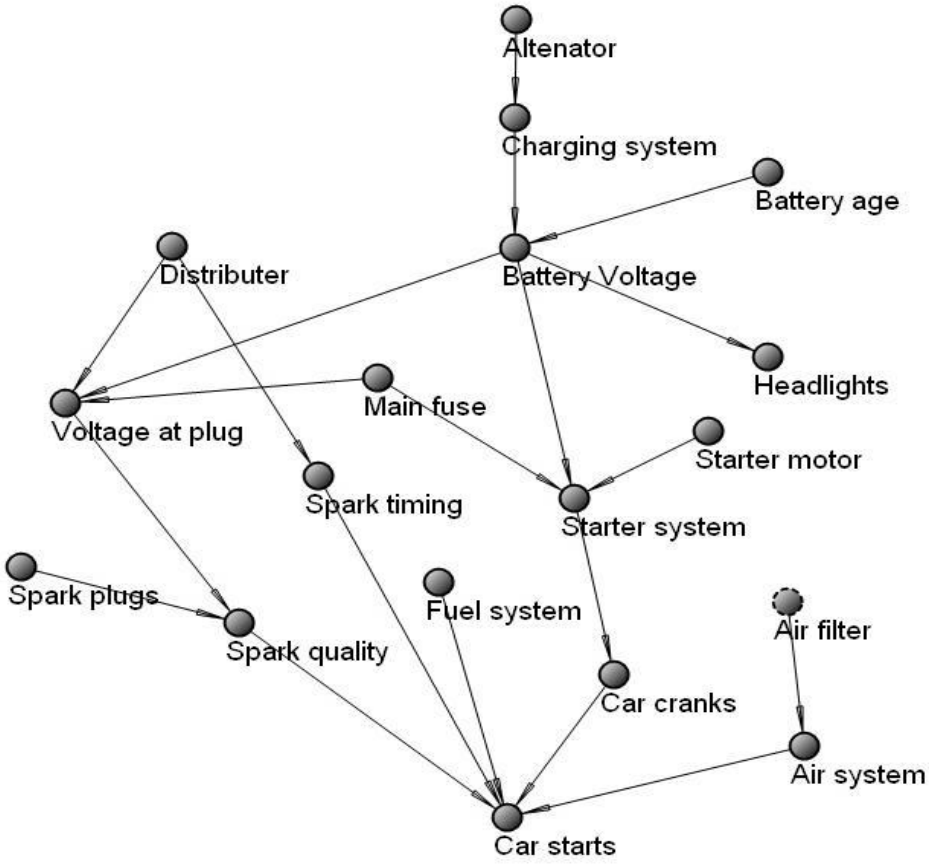


Рис.2.12. Топологія мережі CarStarts

Дана мережа є прикладом байєсівської мережі довіри – одного з найбільш поширених методів представлення знань з невизначеністю. За топологією вона є орієнтованим ациклічним графом. Вузли являють собою випадкові змінні, що можуть бути дискретними чи неперервними. Дуги відображають причинно-наслідкові зв'язки між змінними, "батьківські" вершини відображають причини, а "дочірні" – наслідки. Наприклад, ступінь зарядженості акумулятора впливає на можливість освітлення та роботу системи запалення. Основна задача такої мережі – отримати інформацію про недоступні для діагностування вузли та агрегати автомобіля через дані, що надходять у змінні та спостерігаються і відомі залежності між ними. Дані залежності змодельовані за допомогою відповідних дуг.

Мережа CarStarts містить 18 вершин та 19 дуг. В якості прихованої вершини обрано вершину "Starter system". Кожній змінній мережі ставиться у відповідність таблиця умовних ймовірностей, у якій перераховані всі можливі значення змінної за прийняття її батьківськими вершинами певних значень. У таблиці 2.8 наведено істинні значення ймовірностей станів прихованої вершини та оцінки цих значень, отримані в результаті роботи програми BNetMaster.

Таблиця 2.8

Значення ймовірностей станів прихованої вершини

Стан	**Початкові значення**	**Експериментальні оцінки**
Okay	0,596	0,550
Faulty	0,404	0,450

У таблицях 2.9 та 2.10 наведено умовні ймовірності прихованої вершини: відповідно, початкова та експериментально обчислена. Ці значення є дуже близькими, крім випадку Main fuse = "blown" і Starter motor = "Faulty", коли замість однозначного випадку, коли система запуску двигуна не спрацьовує, отримали рівноймовірні події.

Таблиця 2.9

Фактична таблиця умовних ймовірностей прихованої вершини

Main fuse	**Starter motor**	**Battery voltage**	**Okay**	**Faulty**
Okay	Okay	strong	0,98	0,02
Okay	Okay	weak	0,90	0,10
Okay	Okay	dead	0,10	0,90
Okay	Faulty	strong	0,02	0,98
Okay	Faulty	weak	0,01	0,99
Okay	Faulty	dead	0,01	0,99
Blown	Okay	strong	0,00	1,00
Blown	Okay	weak	0,00	1,00
Blown	Okay	dead	0,00	1,00
Blown	Faulty	strong	0,00	1,00
Blown	Faulty	weak	0,00	1,00
Blown	Faulty	dead	0,00	1,00

Таблиця 2.10

Експериментально обчислена таблиця умовних ймовірностей прихованої вершини

Main fuse	**Starter motor**	**Battery voltage**	**Okay**	**Faulty**
Okay	Okay	strong	0,88	0,12
Okay	Okay	weak	0,83	0,17
Okay	Okay	dead	0,10	0,90
Okay	Faulty	strong	0,00	1,00
Okay	Faulty	weak	0,00	1,00
Okay	Faulty	dead	0,00	1,00
Blown	Okay	strong	0,00	1,00
Blown	Okay	weak	0,00	1,00
Blown	Okay	dead	0,00	1,00
Blown	Faulty	strong	0,50	0,50
Blown	Faulty	weak	0,50	0,50
Blown	Faulty	dead	0,50	0,50

На рис. 2.13 зображена залежність логарифмічної функції правдоподібності від кількості ітерацій алгоритму EM.

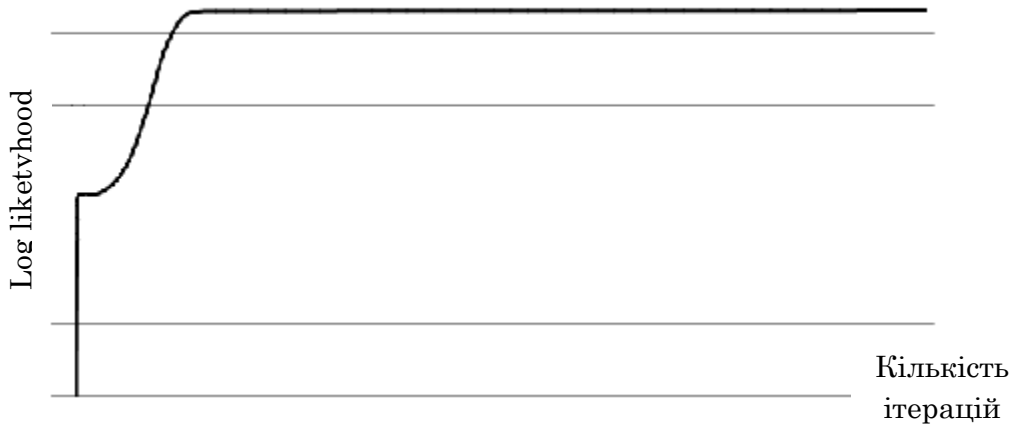


Рис. 2.13. Логарифмічна функція правдоподібності для мережі "Car Starts"

З графіку видно, що з кожною ітерацією значення функції правдоподібності зростає, що свідчить про покращення параметрів мережі. У цілому за весь процес навчання логарифм функції правдоподібності збільшився з -16,90 до -5,97. Але існує певний момент у цьому процесі, коли після деякого числа ітерацій значення функції правдоподібності перестає суттєво змінюватись. Це свідчить про те, що алгоритм має припинити своє функціонування.

Контрольні питання

1. Наведіть означення системного аналізу. Що є об'єктом вивчення для системного аналізу?
2. Наведіть особливості, ціль та сфери застосування прикладного системного аналізу.
3. Наведіть типи проблем системного аналізу. Як розв'язується проблеми кожного типу?
4. Які переваги надає застосування методології системного аналізу при розв'язанні задач прийняття рішень?
5. Що означає структурованість задачі прийняття рішень?
6. Які задачі прийняття рішень називають слабкоструктурованими?
7. Що виступає інструментом прийняття рішень при застосуванні методології системного аналізу?
8. Які особливості має прикладний системний аналіз?
9. Які існують три класи невизначеностей, пов'язаних із розв'язанням задач системного аналізу?
10. Що означає термін "інтелектуалізація процесу прийняття рішень"?
11. Що таке когнітивна карта і де можна нею скористатись?
12. Назвіть чотири основні характеристики СППР.
13. На яких принципах системного аналізу ґрунтується проектування СППР?
14. Назвіть відомі Вам критерії якості рішень.
15. Що означає ієрархічність досліджуваного об'єкта та його моделі?
16. Що означає навчання мереж Байєса?

17. Наведіть типи навчання мереж Байєса залежно від структури та наявності спостережень. Якими методами вирішуються такі задачі навчання?
18. Що таке приховані вершини в мережі Байєса? Наведіть приклади прихованих мереж. Які переваги використання прихованих вершин при моделюванні?
19. Яким чином можна перетворити приховану вершину у звичайну наявну?
20. Які параметри необхідно оцінювати при навчанні мереж Байєса?
21. Які існують методи навчання параметрів мереж Байєса з прихованими вершинами? Приклади їх використання.
22. Яку роль відіграють приховані змінні при розв'язанні слабкоструктурованих задач?

РОЗДІЛ 3
ЙМОВІРНІСНИЙ ВИСНОВОК У БАЙЄСІВСЬКИХ МЕРЕЖАХ

- Поняття ймовірнісного висновку у байєсівській мережі
- Методи оцінювання якості побудови ймовірнісного висновку
- Огляд методів формування ймовірнісного висновку у мережах Байєса
- Формування ймовірнісного висновку у байєсівській мережі на основі навчальних даних

3.1. Поняття ймовірнісного висновку у байєсівській мережі

Побудувавши структуру самостійно або залучивши експертів для її побудови, дослідник отримує мережу Байєса певної топології. Використовуючи її, можна моделювати різноманітні ситуації, тобто задавати вершинам мережі Байєса деякі значення станів і отримувати найбільш ймовірні значення станів для інших вершин мережі.

Процес обчислення оцінки стану вершини на основі апріорної ймовірності про стани інших вершин мережі Байєса називають формуванням (обчисленням, отриманням) ***ймовірнісного висновку***. Саме механізм побудови ймовірнісного висновку перетворює будь-яку мережу Байєса, яка описує відповідний процес, на повноцінну складову експертної системи.

Задача побудови ймовірнісного висновку в мережах Байєса є складною з обчислювальної точки зору та неоднозначною. Обсяг обчислень насамперед залежить від кількості вершин та дуг мережі, що з'єднують ці вершини між собою. Неоднозначність задачі полягає в тому, що за різними методами формування ймовірнісного висновку можуть бути одержані різні результати. Така ситуація характерна зокрема для великих мереж Байєса, коли необхідно застосовувати апроксимаційні алгоритми, жертвуючи при цьому точністю обчислень.

Найпростіше ймовірнісний висновок будується на простих деревах у випадках, коли задається стан кореневих вершин. Тоді для обчислення значень станів дочірніх вершин достатньо послідовно застосувати формулу Байєса.

У випадку, коли задається стан дочірньої вершини для обчислення значень станів батьківських вершин, застосовується обернена формула Байєса. В усіх інших випадках, коли структура мережі Байєса є полі-деревом або багатозв'язною мережею, доводиться застосовувати складніші алгоритми.

В якості прикладу побудови ймовірнісного висновку можна навести задачу медичної діагностики, коли за наявністю ускладнення дихання у пацієнта, йому потрібно встановити діагноз.

У табл. 3.1 наведено множину навчальних даних $<S,C,B,D>$, сформовану на підставі 10 спостережень, де S (smoker) – ймовірність паління; C (cancer) – вірогідність захворювання на рак; B (bronchitis) – можливість захворювання бронхітом; D (dyspnea) – спостереження ознак задишки. При цьому одиниця позначає позитивну відповідь, а нуль – негативну.

Таблиця 3.1

Набір з 10 навчальних спостережень для мережі Байєса

Номер спостереження	**Відповіді на навчальні дані**			
	S	**C**	**B**	**D**
1	1	1	1	1
2	1	0	1	0
3	1	0	0	0
4	0	0	1	1
5	0	0	0	0
6	1	0	0	0
7	1	1	1	1
8	0	0	0	0
9	0	0	0	1
10	0	0	0	0

Цим даним відповідає структура мережі, зображена на рис. 3.1.

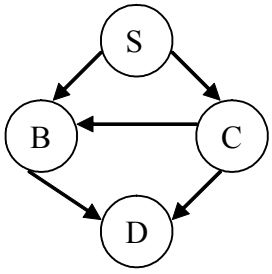


Рис. 3.1 Структура мережі Байєса, що відповідає даним табл. 3.1

Далі, на підставі даних навчальних спостережень, представлених в табл. 3.1, розраховано значення спільного розподілу ймовірностей для всієї мережі (табл. 3.2).

Таблиця 3.2

Значення спільного розподілу ймовірностей всієї мережі

Номер набору даних	**Значення вершин**				**Значення ймовірностей всієї мережі**
	S	**C**	**B**	**D**	
1	0	0	0	0	0,3
2	0	0	0	1	0,1
3	0	0	1	1	0,1
4	1	0	0	0	0,2
5	1	0	1	0	0,1
6	1	1	1	1	0,2

Виходячи з даних розрахунків, можна обчислити значення спільного розподілу ймовірностей вершин, які зображені на рис. 3.2.

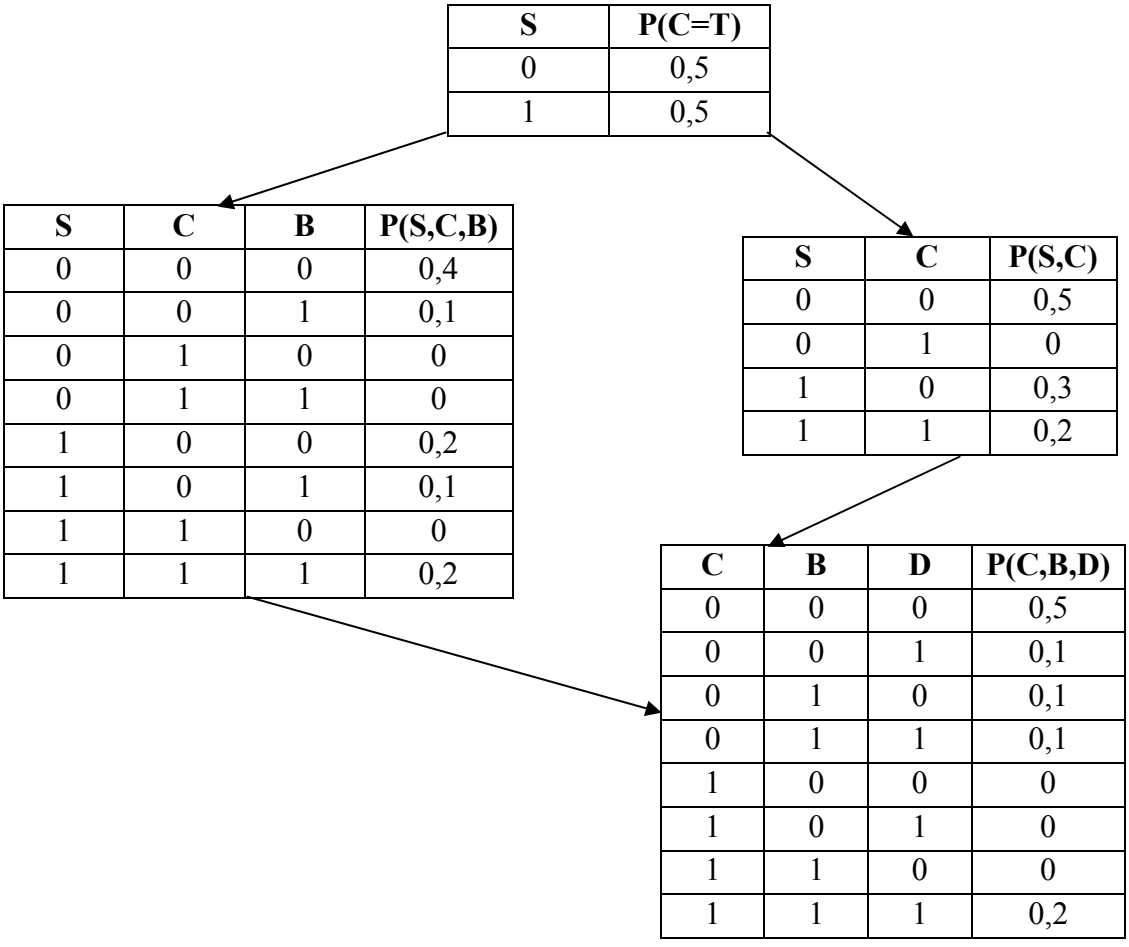


Рис. 3.2. Мережа Байєса у вигляді спільного розподілу ймовірностей вершин, що відповідають даним з табл. 3.1 і рис. 3.1

Використовуючи формулу Байєса $P(A \mid B) = \dfrac{P(A,B)}{P(B)}$ за значеннями спільного розподілу ймовірностей вершин і значеннями ймовірностей вершин, можна розрахувати значення умовних ймовірностей вершин, які, у свою чергу утворюють таблиці умовних ймовірностей.

Повна спільна ймовірність розглянутої мережі Байєса обчислюється за формулою:

$$P(S,C,B,D) = P(S) \cdot P(C|S) \cdot P(B|S,C) \cdot P(D|B,C) \qquad (3.1)$$

Розраховані значення представлено в табл. 3.3.

Емпіричні значення спільного розподілу ймовірностей всієї мережі і розраховані на основі таблиці умовних ймовірностей відрізняються між собою – це можна побачити з табл. 3.2 і рис. 3.3.

Таблиця 3.3

Значення спільного розподілу ймовірностей для всієї мережі

Номер набору даних	Значення вершин				Значення ймовірностей всієї мережі
	S	C	B	D	
1	0	0	0	0	0,332
2	0	0	0	1	0,068
3	0	0	1	0	0,05
4	0	0	1	1	0,05
5	0	1	0	0	0
6	0	1	0	1	0
7	0	1	1	0	0
8	0	1	1	1	0
9	1	0	0	0	0,164
10	1	0	0	1	0,034
11	1	0	1	0	0,051
12	1	0	1	1	0,051
13	1	1	0	0	0
14	1	1	0	1	0
15	1	1	1	0	0
16	1	1	1	1	0,2

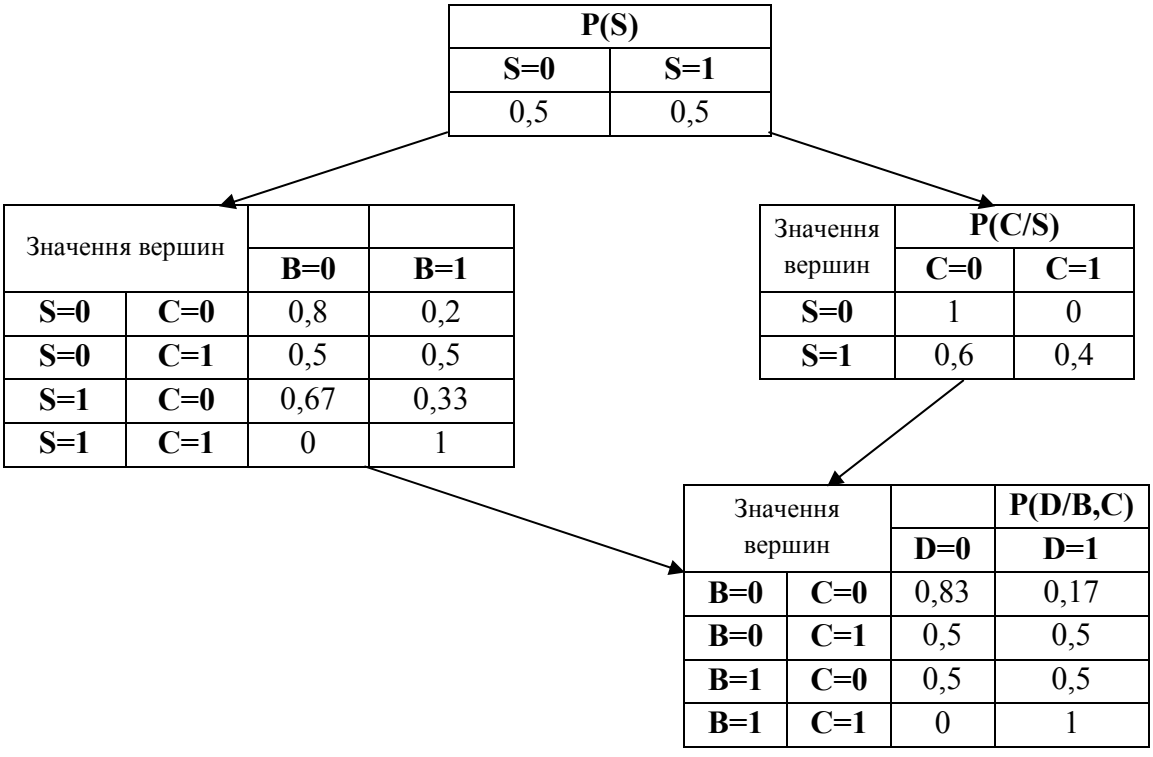


Рис. 3.3. Мережа Байєса у вигляді таблиці умовних ймовірностей вершин, що відповідають даним табл. 3.1

Зазначимо, що не тільки за числовими значеннями, але в результаті використання таблиці умовних ймовірностей, з'являються нові події, яких не було серед навчальних даних. Це пов'язано з частковою втратою інформації при переході від емпіричних навчальних даних до таблиці умовних ймовірностей.

Отже, метою ймовірнісного висновку є знаходження $P(A \mid B)$ апостеріорної ймовірності шуканих вершин А, при деякому значенні спостережуваних вершин В.

Задача побудови ймовірнісного висновку є складною з обчислювальної точки зору та неоднозначною, тобто неможливо знайти єдиний алгоритм побудови, який демонстрував би найкращі результати для всіх типів мереж.

За розміром вирішуваних задач методи формування ймовірнісного висновку в мережах Байєса можна розділити на дві групи: методи точного висновку та апроксимаційні [57, 58].

У табл. 3.4 наведено перелік методів, з яких складається кожна з груп.

Таблиця 3.4

Класифікація методів обчислення ймовірнісного висновку в мережах Байєса

<table>
<tr><th>Алгоритми точного висновку</th><th>Апроксимаційні алгоритми</th></tr>
<tr><td>1. Алгоритм Перла розповсюдження повідомлення для однозв'язних мереж (полі-дерев)</td><td>1. Часткового або неповного висновку (exact inference partially)</td></tr>
<tr><td>2. Кластеризації дерева клік (clique tree clustering)</td><td rowspan="2">2. Варіаційні методи (variational algorithms) використовуються для обчислення середніх ознак великих мереж</td></tr>
<tr><td>3. Визначаючого перетину (cutest conditioning)</td></tr>
<tr><td>4. Виключення змінних (variable elimination algorithm)</td><td>3. Методи стохастичного вибору (stochastic sampling)</td></tr>
<tr><td>5. Символьного ймовірнісного висновку (symbolic probabilistic inference)</td><td rowspan="2">4. Пошукові методи (search-based), основані на евристичних алгоритмах пошуку, які використовують при переході від задачі обчислення ймовірнісного висновку до оптимізаційної задачі</td></tr>
<tr><td>6. Диференціальний підхід (differential method)</td></tr>
</table>

3.2. Алгоритми формування точного висновку

Алгоритм Перла для розповсюдження повідомлення в однозв'язних мережах

У 1986 році Джуді Перл [59] запропонував для обчислення значень ймовірностей в мережах довіри ідею використання інформаційних повідомлень, спрямованих від дочірніх та батьківських вершин. У своїй наступній роботі 1988 року він [24] представив алгоритм розповсюдження повідомлення, інша назва якого "ітераційний алгоритм розповсюдження довіри (IBP – iterative belief propagation) для

полі-дерев". Метою цього алгоритму є "… проникнення та розповсюдження впливу нових подій або даних байєсівської мережі таким чином, щоб в результаті кожна пропозиція була безсумнівно повністю узгодженою з аксіоматикою теорії ймовірностей" [24, с. 143]. Процес розповсюдження ймовірностей у мережах Байєса є основним процесом, на якому ґрунтується висновок, тобто прийняття рішення: "Кожна нова подія або дані розглядаються як збурення, яке розповсюджується по мережі завдяки повідомленням, якими обмінюються сусідні вершини" [24, с. 143].

Ідея алгоритму полягає в тому, що при обчисленні умовної ймовірності (міри довіри) вершини *X*, використовується інформація про стан вершин-сусідів. На рис. 3.4 показано, як інформація про стан батьківської вершини передається через π-повідомлення, а через λ- повідомлення передається інформація про дочірню вершину.

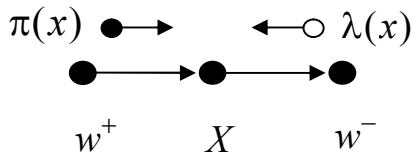


Рис. 3.4. Вершина *X* отримує повідомлення від дочірньої та батьківської вершин-сусідів

У випадку мережі Байєса, зображеної на рис. 3.4, ймовірність (значення довіри) для вершини *X* обчислюється за формулою:

$$P(x) = P(x \mid w) = P(x \mid w^+, w^-) = \frac{P(x \mid w^+) \cdot P(w^- \mid x)}{P(w^+, w^-)} = \alpha \cdot P(x \mid w^+) \cdot P(w^- \mid x) = \\ = \alpha \cdot \pi(x) \cdot \lambda(x) \tag{3.2}$$

У випадку мережі з однією батьківською вершиною та однією дочірньою вершиною застосовується формула (3.3):

$$P(x) = \alpha \cdot \pi(x) \cdot \lambda(x), \tag{3.3}$$

де α – нормуюча константа.

Нехай *W* - множина, яка містить всі змінні, для яких можна отримати поточні значення (спостереження). Таку множину називають інстанційованою. Множина інстанційованих змінних складається з двох підмножин: W_X^- - множина, яка містить всі інстанційовані змінні, що знаходяться нижче вузла X; W_X^+ - множина, яка містить всі інстанційовані змінні, що знаходяться над вузлом *X*. Для спрощення записів застосовують такі позначення: $W_B^- = \lambda$ і $W_X^+ = \pi$.

До деякої довільної вершини *X* мережі інформація може надходити від батьківських вузлів або від вузлів-нащадків. Так, $\pi(x)$ представляє інформацію, яка надходить від батьківської вершини *X*. $\pi(x)$ називають також *прямим оператором* (або *форвард-оператором*), оскільки він відображає розповсюдження інформації в

напрямку до листків мережі (зверху-вниз). У випадку, коли конкретна змінна отримує нову інформацію від батьківських вершин, вона повинна бути передана далі (вперед) до дочірніх вершин, якщо вони є. Розповсюдження інформації в напрямку “зверху-вниз” (тобто в напрямку "до листків") можна концептуалізувати за допомогою π - повідомлення. Значення π – це внутрішнє значення однієї змінної, яке використовується з метою обчислення величини довіри (умовної ймовірності) для однієї змінної.

Подібно до π, значення λ також необхідно розглядати як λ - повідомлення і λ - значення. λ - повідомлення означає, що інформація передається “знизу-вверх”, тобто – від листків до кореня, а λ - значення – це внутрішнє значення (величина) змінної. λ називають *зворотним оператором*, оскільки він свідчить про розповсюдження інформації в напрямку до кореня; він також представляє собою ретроспективне (діагностичне) підтвердження того, що від нащадків вузла X отримано інформацію, що $X = x$. Використання означення “діагностичний” є виправданим, оскільки λ- повідомлення поступають у мережу тоді, коли інстанціюється (приймає конкретне значення) один або більше атрибутів. λ - повідомлення безпосередньо впливає на кожне λ - значення змінної [60].

Наприклад, нехай для мережі, наведеної на рис. 3.5, надходять спостереження щодо змінних *B,D,F,H*, тобто $W = [B, D, F, H]$. Тоді для вузла *C* підмножина $W_C^- = [F, H]$ і $W_C^+ = [B, D]$.

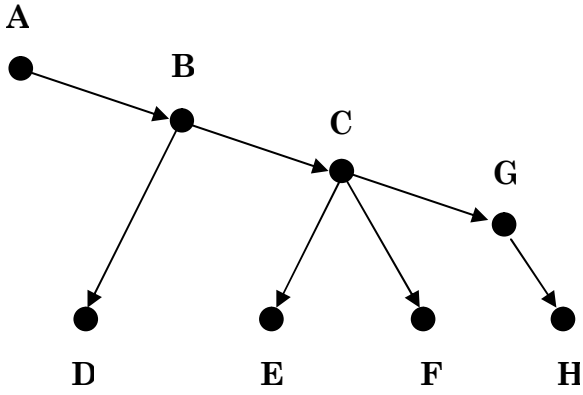


Рис.3.5. Приклад формування інстанційованих множин

Таким чином, значення λ будуть поступати від дочірніх вершин, що можна формально представити у вигляді:

$$\lambda(b_i) = \begin{cases} p(W_B^- \mid b_i), \ \textit{якщо} \quad B \notin W \\ 1, \ \textit{якщо } B \in W \textit{ і } b_i - \textit{інстанційоване значення}, \\ 0, \ \textit{якщо } B \in W \textit{ і } b_i - \textit{неінстанційоване значення}, \end{cases} \tag{3.4}$$

де b_i конкретне значення із множини *b*, які може приймати змінна (вузол) B, тобто b_i формують множину *b*.

Іншими словами, стосовно виразу (3.8) можна сказати: якщо $B \in W$ і b_i вже отримало значення від датчика чи експерта, то $\lambda(b_i) = 1$, а якщо $B \in W$, але b_i ще не отримало значення, то $\lambda(b_i) = 0$.

Значення для π формалізується простіше:

$$\pi(b_i) = p(b_i \left| W_B^+ \right.). \tag{3.5}$$

Обоє π і λ - значення використовуються для оновлення величини довіри змінної.

Довіра – це розподіл ймовірностей, який ґрунтується на всій інформації, що міститься в мережі. Таким чином, можна виразити загальний ступінь довіри до виразу "$X = x$" шляхом об'єднання і нормування π і λ - значень:

$$Bel(x) = \alpha\lambda(x)\pi(x), \tag{3.6}$$

де α - нормуюча константа.

Рівняння (3.10) характеризує розподіл ймовірностей деякого атрибута. Якщо довіра будь-якого вузла змінюється, він посилає π - повідомлення до своїх нащадків. Тобто π - повідомлення завжди передаються в прямому напрямку по мережі, коли з'являється спостереження однієї змінної. Тому π - повідомлення і π - значення називають "діагностичною" інформацією.

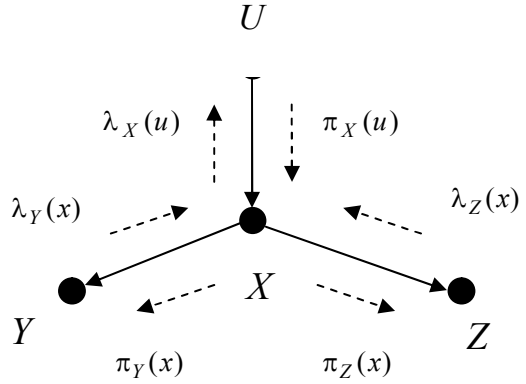


Рис. 3.6. Приклад мережі Байєса, де вершина *X* одержує повідомлення від двох дочірніх вершин та однієї батьківської

У випадку, зображеному на рис. 3.6, коли потрібно обчислити значення довіри вершини з однією батьківською вершиною і двома дочірніми, застосовується формула 3.7:

$$Bel(x) = \alpha\lambda_Y(x)\lambda_Z(x)\sum_u M_{x|u}\pi_X(u), \tag{3.7}$$

де $M_{x|u}$ - матриця, в якій зберігаються всі умовні ймовірності між змінними U і X.

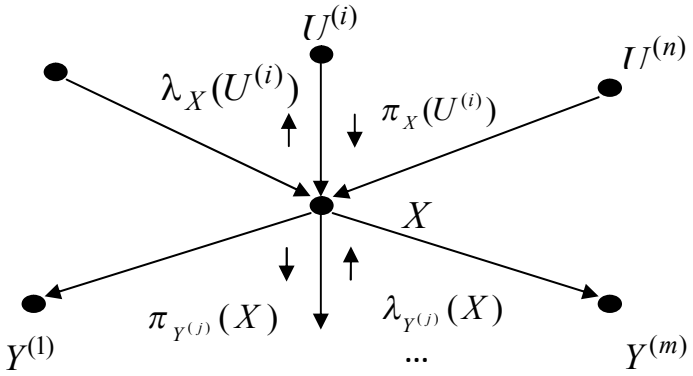


Рис. 3.7. Загальний випадок: вершина X має n батьківських та m дочірніх вершин

У загальному випадку, зображено $Y^{(j)}$му на рис. 3.7, коли вершина має m дочірніх вершин та n батьківських вершин, обчислення виконується за формулою (3.11), але з урахуванням формул (3.8) та (3.9).

$$\lambda(x)=\begin{cases}\prod_{j=1}^{m}\lambda_{Y^{(j)}}(x),\ якщо\ B\notin W,\\ 1,\ якщо\ B\in W\ і\ b_i-інстанційо ване\ значення,\\ 0,\ якщо\ B\in W\ і\ b_i-неінстанці йоване\ значення\end{cases} \tag{3.8}$$

$$\pi(x)=\sum_{c_1=1}^{k_1}\ldots\sum_{c_n=1}^{k_n}\left(p\left(x\mid u_{c_1}^{(1)},\ldots,u_{c_n}^{(n)}\right)\cdot\pi_x\left(u_{c_1}^{(1)}\right)\ldots\cdot\pi_x\left(u_{c_n}^{(n)}\right)\right). \tag{3.9}$$

При цьому вважається, що кожна i-та батьківська вершина може приймати k_i станів i=1…n, а кожна j-та дочірня вершина r_j станів: r=1…m. Тобто значення λ конкретної змінної є добутком повідомлень, які надходять до неї від її дочірніх вершин, а на π - повідомлення впливають всі π - повідомлення, які приймає одна змінна. Після перемноження комбіноване π - повідомлення надсилається дочірнім вершинам. Процес розповсюдження інформації ілюструє рис.3.8 на прикладі двох інстанційованих змінних.

На рисунку 3.8 послідовно представлено виконання кроків алгоритму Перла:

Рис. 3.8*а* – Ініціалізація дерева.

Рис. 3.8*б* – Інстанціювання двох змінних; λ - повідомлення (стрілочки без заповнення) починають розповсюджуватись в напрямку кореневої змінної.

Рис. 3.8*в* – λ - повідомлення продовжують розповсюджуватись у напрямку кореневої змінної. Дві змінні, які отримали λ - повідомлення (з рис.3.8б), змінили рівні своєї довіри, а тому посилають свої π - повідомлення (стрілочки із заповненням) у напрямку змінних, представлених листками.

Рис. 3.8*г* – λ - повідомлення досягли кореневої змінної, для якої оновлюється рівень довіри. Відразу ж після оновлення коренева змінна посилає нові π - повідомлення в напрямку змінних-листків. Разом з тим, π - повідомлення (з рис.3.8в) приходять до останніх змінних-листків.

Рис. 3.8*д* – π - повідомлення кореневої змінної досягають змінних-листків.

Рис. 3.8*е* – Ситуація така ж, як і на рис. 3.8*д*: π - повідомлення досягають змінних-листків, і мережа переходить у стан рівноваги.

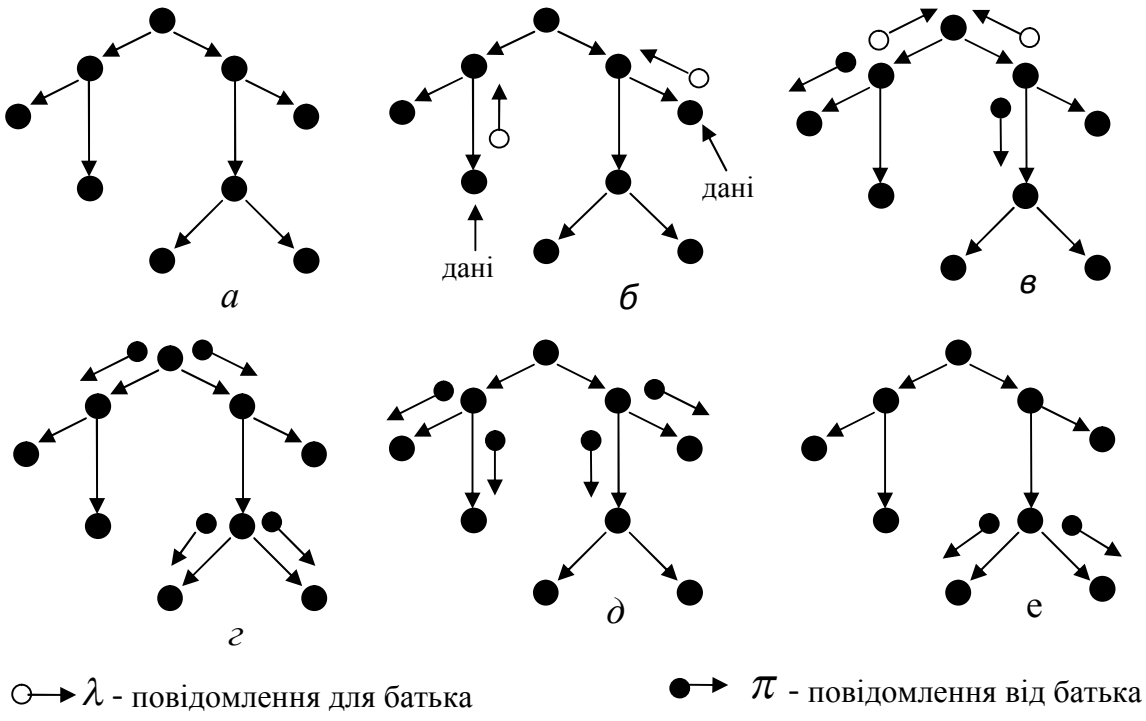


Рис. 3.8. Схематичне представлення розповсюдження інформації по полі-дереву в алгоритмі Перла

Складність цього алгоритму залежить від кількості вершин у дереві.

Алгоритми кластеризації

У 1988 році С. Лорітценом та Д. Шпігельхалтером (S. L. Lauritzen and D. J. Spiegelhalter) [56, 57, 61, 62] запропоновано ***LS-метод.***

Пізніше, у 1990 році, Ф. Дженсеном, К. Олесеном, С. Андерсеном [43] було запропоновано модифікований LS-метод, який зустрічається в літературі під назвою ***Hugin***, яку він отримав завдяки використанню у програмному комплексі Hugin Expert [43], розробленому Ф. Дженсеном та його групою.

У 1997 році С. Лорітцен і Ф. Дженсен [64] розвинули Hugin-метод настільки, що його стало можливо застосовувати в інших теоріях, у тому числі в теорії функцій довіри Демпстера-Шафера (Dempster-Shafer).

Під впливом робіт С.Лорітцена, Д. Шпігельхалтера [56] та Дж. Перла [59], у 1997 році П. Шеной (P. Shenoy) розробив новий метод, який отримав назву ***SS-метод*** (Shenoy-Shafer) [65, 66]. Як показали проведені експерименти [67], з обчислювальної точки зору SS-метод ефективніший у порівнянні з Hugin та LS-методами, а Hugin кращий за LS-метод. Але за критерієм обсягу пам'яті, необхідної для зберігання проміжної обчислювальної інформації, найкращим виявлився LS-метод, а найгіршим SS.

Пізніше, у 1998 році, А. Мадсеном та Ф. Дженсоном запропоновано ***алгоритм лінивого розповсюдження*** (lazy propagation) [68], який є комбінацією SS-алгоритму та алгоритму поглинаючого виключення [69, 70]. Крім цієї комбінації, існує ще декілька модифікацій лінивого алгоритму, зокрема, його комбінації з SS та SPI [71-74], а також з SS та AR [75] алгоритмами.

Ідея ***LS-методу ймовірнісного висновку*** є основоположною для методів кластеризації, вона полягає в тому, що для реалізації ймовірнісного висновку необхідно спочатку привести структуру мережі Байєса до вигляду об'єднаного дерева, а потім використовувати алгоритм розповсюдження повідомлень по дереву догори та донизу і послідовно перераховувати таблиці умовних ймовірностей вершин дерева. У літературі, замість терміну *"об'єднані дерева"* (junction trees), іноді застосовуються терміни *дерева клік* (clique trees), *гіпердерева* (hypertrees) та *якісні дерева Маркова* (qualitative Markov trees).

Загалом використання об'єднаних дерев дозволяє застосовувати ідею Дж. Перла обчислення ймовірнісного стану за допомогою λ і π - повідомлень. Серед подібних методів можна виділити *GDL-метод* (generalized distributive law), який фактично є симбіозом цілої низки методів. Його запропоновано С. Аджи та Р. Мак Елісом у 2000 році [76]. Суть методу полягає в передачі повідомлень по об'єднаному дереву із застосуванням локальних ядер (local kernel) вузлів об'єднаного дерева.

LS-алгоритм [56, 57, 320,77, 78] реалізується у два етапи.

На **першому етапі** виконується побудова об'єднаного дерева клік з первинної структури мережі та заповнення вершин цього дерева таблицями умовних ймовірностей мережі. Даний етап реалізується у чотири кроки:

Крок – 1. Моралізування графа.

Крок – 2. Триангуляція графа.

Крок – 3. Ідентифікація клік.

Крок – 4. Побудова об'єднаного дерева.

Другий етап передбачає обчислення значень ймовірностей станів вершин на основі алгоритмів розповсюдження значень ймовірності по об'єднаному дереву**.** Для обчислення значень ймовірностей клік використовуються λ і π - повідомлення. Після цього за значеннями ймовірностей клік обчислюють індивідуальні ймовірності вершин. Обчислювальна складність LS алгоритму дорівнює $O(p \cdot r^m)$, де p - кількість вершин мережі, а r - максимальна кількість станів, які може набувати вершина.

Перший етап: побудова об'єднаного дерева клік передбачає виконання серії перетворень ациклічного спрямованого графа G (рис. 3.9 а) для його приведення до вигляду об'єднаного дерева.

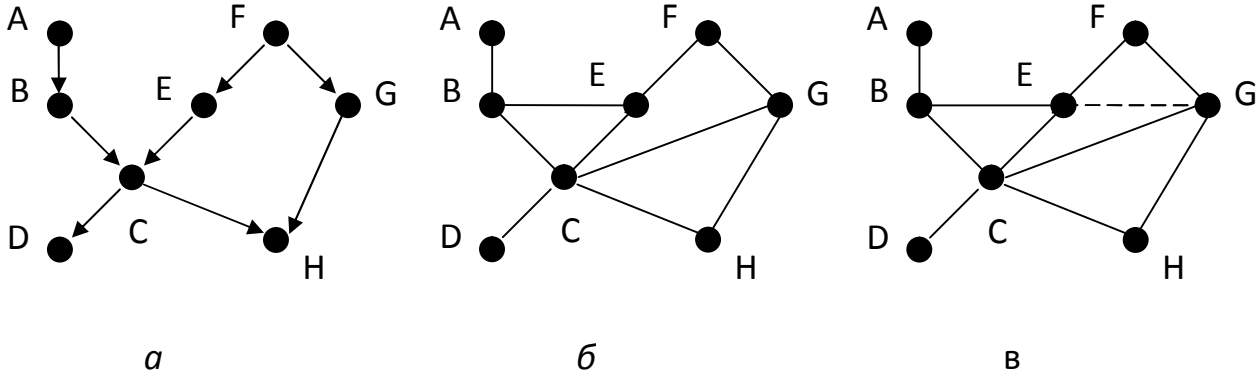


Рис. 3.9. *а*) Ациклічний спрямований граф G; *б*) моралізований граф G_m; *в*) триангульований (розбитий на трикутники) граф G_u G_u

Крок – 1. ***Моралізація графа*** G (рис. 3.9 а) до вигляду графа G_m (рис. 3.9 б): послідовно перебираються всі вершини мережі Байєса, у яких є батьки, якщо батьки вершини не зв'язані між собою – з'єднати дугами взаємно незв'язані вершини-батьки кожної вершини мережі. Якщо батьки вершини не зв'язані між собою, то між ними вводиться зв'язок ("сусід"). Таке перетворення називають ще "одруженням" вершин-батьків, після чого слід дезорієнтувати граф – спрямовані дуги графа замінити неспрямованими (для всіх вершин "батько-нащадок" замінити на зв'язок "сусід").

Крок – 2. ***Триангулювати моралізований граф*** G_m (рис. 3.9 б) до вигляду графа G_u (рис. 3.9 в), тобто представити його так, щоб довжина циклів не перевищувала трьох. Графічно ця операція означає розбиття моралізованого графа на трикутники. Для триангуляції застосовують алгоритм часової заміни [79, 80] (fill-in computation algorithm). Також на цьому кроці застосовують алгоритм пошуку максимальної потужності [81-83] (maximum cardinality search) для побудови упорядкованої множини вершин, яка позначається як α. Процес триангуляції можна описати таким чином:

1. Послідовно перебирати всі вершини мережі Байєса, доки не будуть розглянуті всі вершини.

Для цього слід виконати наступні дії:

- перевірити, чи є суміжними між собою "сусіди" аналізованої вершини, якщо так, то така вершина симпліціальна – утворює кліку разом із своїми сусідами (така вершина виключається з розгляду разом із її ребрами);

- якщо після перебору всіх вершин мережа, що залишилася до розгляду, не порожня, то перейти до наступного кроку алгоритму; у протилежному випадку, вважається, що граф триангульований;

2. У мережі, що залишилась до розгляду, послідовно перебираються вершини: шукається вершина з найбільшою кількістю сусідів (така вершина стає симпліціальною шляхом введення додаткових ребер між її несуміжними сусідами), тобто тепер вона утворює кліку з сусідами, а потім ця вершина виключається із розгляду. Після розгляду усієї мережі до початкового моралізованого ненаправленого графа додаються додаткові ребра, і такий граф буде триангульованим.

Крок – 3. За допомогою алгоритму пошуку клік [84, 85] (cliques-finding algorithm) в триангульованому моралізованому графі G_u (рис. 3.9 в) ***визначається множина клік*** (рис. 3.10), тобто підграфів, потужності яких не перевищують трьох.

Для поточної кліки слід виконати такі дії: перевірити, чи є вона підмножиною інших не перебраних клік, якщо так, то вона знищується; якщо в ній та інших кліках співпадає хоча б одна вершина, то між відповідними кліками вводиться ребро, що містить сепаратор – перетин множин вершин цих клік. Після цього виконується ранжирування множини клік, застосовуючи упорядковану множину вершин α. Ранжирування множини клік робиться таким чином: першою клікою стає та, що має вершину, яка в упорядкованій множині вершин α посідає перше місце; якщо клік, які мають цю вершину є декілька, то дивляться на наявність у кліці вершини, яка займає друге місце в упорядкованій множині вершин α, і так далі. Таким чином виконується ранжирування усієї множини клік.

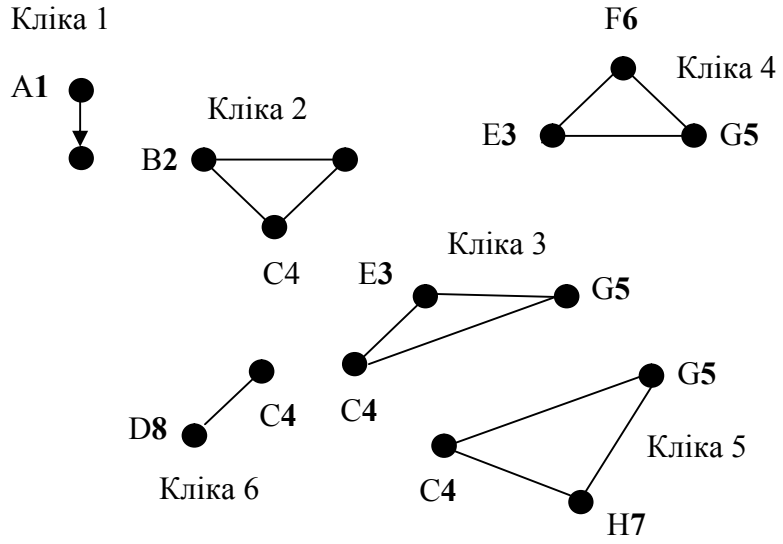


Рис. 3.10. Ранжирувана множина клік тріангульованого графа G_u

Крок – 4. ***Побудова об'єднаного дерева*** [61, 78]. Під час реалізації даного кроку всі кліки послідовно зв'язуються між собою шляхом вибору для зв'язку ребер з найбільшими сепараторами, а інші ребра видаляються. Дерево, що зв'язує всі кліки з максимальними потужностями сепараторів, буде об'єднаним деревом. Для зручності, як корінь в об'єднаному дереві, вибирається кліка з найбільшою кількістю вершин. Якщо таких вершин декілька, то перевага віддається кліці з найбільшою кількістю ребер.

Вхідними даними є ранжирувана множина клік тріангульованого графа G_u. Алгоритм побудови об'єднаного дерева можна представити наступним чином:

1. $S_i = Кліка_i \cap (Кліка_1 \cup Кліка_2 \cup ... \cup Кліка_{i-1})$

2. $R_i = Кліка_i - S_i$, тобто $R_i = Кліка_i \cap S_i$.

3. Якщо $S_i \subseteq Кліка_j$ та $j<i$ то $Кліка_j$ є батьківською для $Кліки_i$.

4. Обчислення потенційної ймовірності i - ї кліки за формулою:

$\psi_i = \psi(Кліка_i) = \prod_{\forall v \in Кліка_i} P((v \mid pa(v))$, де запис $\forall v \in Кліка_i$ означає перебір всіх вершин v i - ї кліки, а $pa(v)$ – множина батьків вершини v, які разом із v знаходяться в i - й кліці. У випадку, коли батьки є, але в i - й кліці знаходяться не всі батьки, або їх немає зовсім в цій кліці, то $P(v \mid pa(v)) = 1$. Значення ймовірностей для розрахунків беруться з оригінальної мережі Байєса (рис. 3.9 а).

5. Для кожної вершини об'єднаного дерева (кліки) обчислюються та зберігаються параметри $Кліка_1, S_i, R_i, \psi(Кліка_1)$.

У результаті буде побудоване дерево клік, тобто об'єднане дерево (рис. 3.11), побудоване за мережею Байєса, яке представлено на рис. 3.9а.

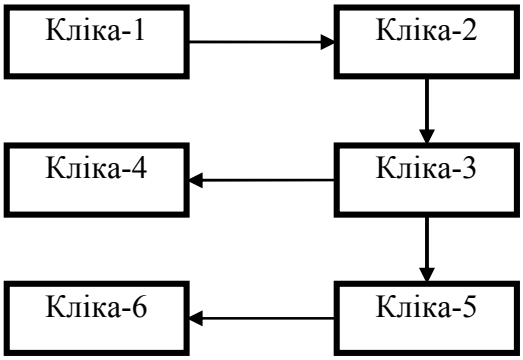


Рис. 3.11 Об'єднане дерево (дерево клік)

На завершення першого етапу об'єднане дерево заповнюється таблицями. Заповнення відбувається поетапно, в ході обходу вершин формується таблиця значень для об'єднаного дерева (табл. 3.5).

Таблиця 3.5

Таблиця значень для об'єднаного дерева, представленого на рис. 3.11

Номер кліки	Вершини кліки V	Граничні вершини S_i	Остаточні вершини R_i	Функція потенційної ймовірності кліки $\psi_i = \prod_{\forall v \in Кліка_i} P((v \mid pa(v))$
1	*A, B*	–	*A, B*	$\psi_1 = P(A \mid pa(A)) \cdot P(B \mid pa(B)) = = P(A) \cdot P(B \mid A)$
2	*B, C, E*	*B*	*C, E*	$\psi_2 = P(B \mid pa(B)) \cdot P(C \mid pa(C)) \cdot \cdot P(E \mid pa(E)) = P(B \mid \{не\ батьки\}) \cdot \cdot P(C \mid B, E) \cdot P(E \mid \{не\ всі\ батьки\}) = = 1 \cdot P(C \mid B, E) \cdot 1 = P(C \mid B, E)$
3	*C, E, G*	*C, E*	*G*	$\psi_3 = P(C \mid pa(C)) \cdot P(E \mid pa(E)) \cdot P(G \mid pa(G)) = = P(C \mid \{не\ всі\ батьки\}) \cdot P(E \mid \{не\ всі\ батьки\}) \cdot \cdot P(G \mid \{не\ всі\ батьки\}) = 1 \cdot 1 \cdot 1 = 1$
4	*E, F, G*	*E, G*	*F*	$\psi_4 = P(E \mid pa(E)) \cdot P(F \mid pa(F)) = = P(G \mid pa(G)) = = P(E \mid F) \cdot P(F) \cdot P(G \mid F)$
5	*C, G, H*	*C, G*	*H*	$\psi_5 = P(C \mid pa(C)) \cdot P(G \mid pa(G)) \cdot P(H \mid pa(H)) = = P(C \mid \{не\ всі\ батьки\}) \cdot P(G \mid \{не\ всі\ батьки\}) \cdot \cdot P(H \mid C, G) = 1 \cdot 1 \cdot P(H \mid C, G) = P(H \mid C, G)$
6	*C, D*	*C*	*D*	$\psi_6 = P(C \mid pa(C)) \cdot P(D \mid pa(D)) = = P(C \mid \{не\ всі\ батьки\}) \cdot P(D \mid C) = = 1 \cdot P(D \mid C) = P(D \mid C)$

S_i - граничні вершини i - ї кліки (вершини, що належать батьківським клікам).

R_i – остаточні вершини i - ї кліки (вершини, які не належать батьківським клікам).

$\psi(Кліки_i)$ – значення ймовірності i - ї кліки.

Заповнення починається з листів дерева і послідовно перебираються усі кліки. Розглядаються "невідмічені" вершини в необробленій кліці:

1. Якщо серед "невідмічених" вершин є така, що не зустрічається в інших необроблених кліках, то таблиця повинна мати вигляд P (this node| other nodes in clique) у такому випадку ця вершина"відмічена" і кліка вважається обробленою.

2. Якщо в первинній мережі Байєса такої таблиці немає або така вершина не одна, то використовується таблиця сукупного розподілу ймовірностей, у такому разі усі вершини кліки"відмічені", і кліка оброблена.

3. Якщо в кліці вже є "відмічені" вершини і всього одна "невідмічена" вершина, то використовується таблиця виду P (unmarked node) з первинної мережі Байєса - у такому випадку ця вершина"відмічена", і кліка вважається обробленою.

4. Якщо в кліці вже є "відмічені" вершини, а "невідмічених" вершин декілька, то використовується таблиця сукупного розподілу ймовірності "невідмічених" вершин, і тоді ці вершини "відмічені", а кліка вважається обробленою.

5. Якщо в необробленій кліці усі вершини вже"відмічені", то кліка заповнюється таблицею виду P (node any) і вважається обробленою.

За отриманими значеннями ймовірностей клік, можна обчислити значення спільної ймовірності об'єднаного дерева:

$$P(Мережі) = P(A,B,C,D,E,F,G,H) = P(Кліка_1,...,Кліка_6) = \prod_i Кліка_i = \prod_i \psi_i \qquad (3.10)$$

Другий етап – алгоритм пропагації: формування ймовірнісного висновку за об'єднаним деревом. Для кожного спостереження змінної вибирається одна таблиця, яка містить цю змінну. Задаємо нульовими усі входження, які суперечать спостереженню. В результаті вибирається одна таблиця, яка містить цю змінну. Задаємо нульовими усі входження, які суперечать спостереженню.

Ймовірнісний висновок за об'єднаним деревом за своєю ідеєю співпадає з алгоритмом Перла.

На рис. 3.12 представлено процес розповсюдження повідомлень по об'єднаному дереву.

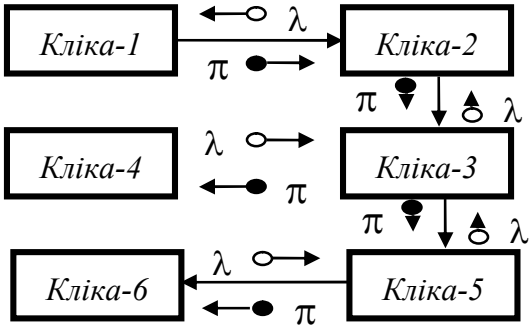


Рис. 3.12. Схематичне представлення розповсюдження інформації у вигляді λ і π– повідомлень за вузлами об'єднаного дерева (дерева клік)

Спочатку виконують *процедуру "сходження догори"* (upword). Повідомлення є результатом маргіналізації - підсумовування змінних таблиць, які не містяться в сепараторі. Після відсилання повідомлення відправник ділить свою поточну таблицю умовних ймовірностей на нього. Коли отримувачу надходить повідомлення, він множить його на свою таблицю умовних ймовірностей і виходить нова таблиця. Коли він отримує повідомлення від усіх своїх нащадків, то, у свою чергу, посилає також повідомлення своєму батьку і ділить свою таблицю на відіслане повідомлення. Процес триває доти, доки корінь зв'язного дерева не отримає повідомлення від усіх своїх нащадків.

Потім виконують *процедуру "сходження донизу"* (downward). Корінь посилає повідомлення кожному своєму нащадкові. Він ділить свою таблицю умовних ймовірностей на повідомлення, отримане від нащадка, маргіналізує таблицю по сепаратору і посилає результат. Коли нащадок отримує повідомлення від свого батька, він множить його на свою поточну таблицю умовних ймовірностей і формує таким чином свою нову таблицю. Далі він маргіналізує її (по сепаратору) і посилає своєму нащадкові. Процес триває доти, доки усе листя не отримає повідомлення. Таким чином, результатом будуть знову перераховані таблиці для кожної змінної за умови наявності спостережень, які потім необхідно нормалізувати.

Тобто знизу нагору йдуть λ - повідомлення, а потім зверху вниз йдуть π - повідомлення, у процесі проходження повідомлень відбувається перерахування значення ймовірності кліки $\psi(Кліки_i)$.

Далі для кожної вершини виконується пошук клік, у яких вона міститься. Якщо серед них є кліка, яка не є листом, то вибирається ця кліка. Інакше – будь-яка з них. Для знаходження ймовірності кожного стану вершини необхідно підсумувати усі значення ймовірностей для цього конкретного стану, які є в таблиці кліки.

За значенням ймовірності кліки виконується обчислення ймовірності кожної вершини кліки:

$$P(Кліки_i) = P(R_i, S_i) = P(R_i \mid S_i) \cdot P(S_i), \text{ то } P(R_i \mid S_i) = \frac{P(R_i, S_i)}{P(S_i)}. \qquad (3.11)$$

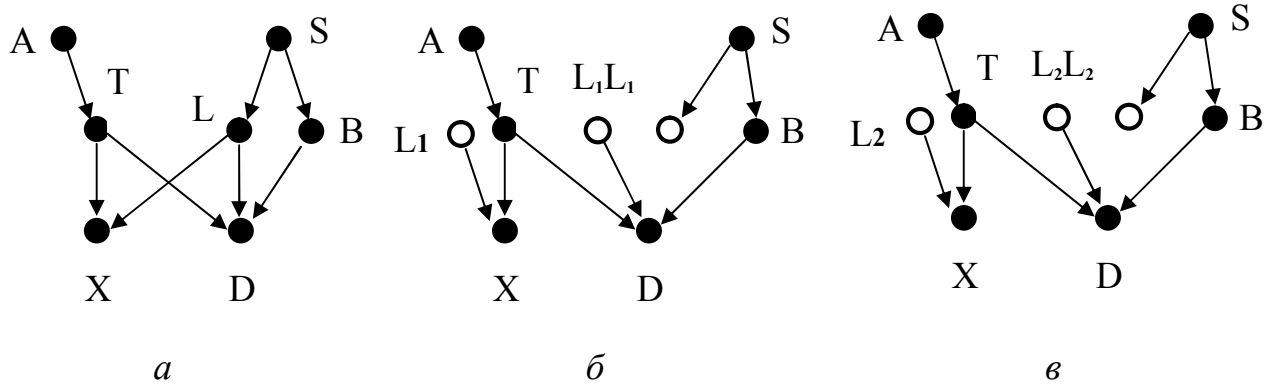
Це рівняння застосовується для обчислення значень ймовірностей інстанційованих вершин [78]. Значення $P(S_i)$ виконує роль π - повідомлення, а $P(R_i \mid S_i)\lambda$ - повідомлення у випадку, коли $S_i = \{\ \}$, то $P(S_i) = 1$. Докладнішу інформацію стосовно обчислення значень ймовірностей в об'єднаному дереві, можна знайти в роботах [65, 78, 86, 87].

Алгоритми визначаючого перетину

Дж. Перл у 1988 р. [24] запропонував **алгоритм визначаючого перетину** (*cutset conditioning*), який зручно застосовувати у випадку багатозв'язаних та занадто великих мереж Байєса, коли методам поглинаючого виключення та кластеризації об'єднаного дерева не вистачає об'єму пам'яті звичайного персонального комп'ютера.

Ідея методу полягає у перетворенні багатозв'язної мережі Байєса в однозв'язну або декілька однозв'язних, застосовуючи мінімальний визначаючий перетин. Після цього застосовується алгоритм Перла для розповсюдження повідомлення по однозв'язній мережі. Кожна отримана проста, тобто однозв'язна, мережа має одну або декілька інстанційованих вершин. На рис. 3.13а наведено приклад мережі

Байєса, в якій у випадку, коли вершина D набуває певного стану, тобто стає інстанційованою, відбувається зациклення α та π - повідомлень у підграфах *DBSL* і *DLXT*.



а *б* *в*

Рис. 3.13. Приклад перетворення багатозв'язної мережі (а) в однозв'язні (б, в), де визначаючий перетин складається із однієї вершини L

Саме ці підграфи роблять мережу багатозв'язаною, тобто потрібно позбутися циклів, які утворюють ці підграфи. Якщо вершина L може приймати один з двох станів $\{L_1, L_2\}$, тобто у випадку, коли вершина L стає інстанційованою, багатозв'язану мережу рис.3.13а можна записати у вигляді двох однозв'язних рис.3.13б та в.

У випадку мережі, представленої на рис. 3.13а, мінімальний визначаючий перетин складається лише з однієї вершини L. Значення довіри багатозв'язаної мережі рис. 3.13а є агрегованою оцінкою, яка обчислюється за значеннями довіри в однозв'язних мережах, (рис. 3.13б та 3.13в), із врахуванням апріорної ймовірності вузла L, за формулою 3.12:

$$P(A,T,X,D,B,S.L) = \sum_{L_i}(P(A,T,X,D,B,S,L=L_i)\cdot P(L=L_i)) = \\ = P(A,T,X,D,B,S,L=L_1)\cdot P(L=L_1) + P(A,T,X,D,B,S,L=L_2)\cdot P(L=L_2) \quad (3.12)$$

Однак не все в цьому методі просто. Наприклад, у випадку, коли визначаючий перетин складається з трьох вершин і кожна вершина має вісім станів, то потрібно виконати обчислення за алгоритмом Перла розповсюдження повідомлень та обчислити достовірну оцінку апріорної ймовірності для всіх 512 структур. Тобто для цього методу саме вибір мінімального визначаючого перетину є дуже актуальною задачею [77, 88]. Але ця задача, в свою чергу, має нелінійну поліноміальну складність, для розв'язання якої використовують поглинаючі евристичні методи або ВМВ.

До методів визначаючого перетину також належать алгоритми:

1. Запропонований в 1992 році ***LC-алгоритм*** (*local condition*) або *алгоритм локального визначаючого перетину* [89], фактично є модифікацією алгоритму Кіма–Перла [90].

2. Опублікований в 1994 році *алгоритм глобального визначаючого перетину* (*global conditioning*) [91] є окремим випадком LS-алгоритму [56].

3. Представлений в 1995 році *алгоритм динамічного визначаючого перетину* (*dynamic conditioning*) [94] є поліпшеною версією методу визначаючого перетину [24]. Цей алгоритм має лінійну складність, у той час, як звичайний метод визначаючого перетину [24], має експоненційну складність.

4. У 2000 році запропоновано *RC-алгоритм* (*recursive conditioning*) або *алгоритм рекурсивного визначаючого перетину* [95], який увібрав у себе кращі ідеї з алгоритмів виключення змінних [69, 70] та LS [56]. Його обчислювальна складність дорівнює $O(n \cdot \exp(w))$, де n - кількість вершин; w - довжина ВМВ виключення змінних.

Алгоритми виключення змінних

На початку дев'яностих років минулого століття *запропоновано VE-алгоритм* (*variable elimination algorithm*), тобто *алгоритм виключення змінних*. В основу алгоритму покладена відома ланцюгова формула декомпозиції спільного розподілу ймовірностей мережі, яка бере свій початок від теореми Байєса:

$$P(X) = P(X^{(N)} \mid X^{(1)},...,X^{(N-1)}) \cdot P(X^{(N-1)} \mid X^{(1)},...,X^{(N-2)}) \cdot ... \cdot P(X^{(1)}) = \\ = \prod_{i=1}^{N} P(X^{(i)} \mid X^{(1)},...,X^{(i-1)}) = \prod_{i=1}^{N} P(X^{(i)} \mid pa(X^{(i)})) \quad (3.13)$$

де $pa(X^{(i)})$ множина батьківських вершин-батьків по відношенню до вершини $X^{(i)}$.

Основна ідея алгоритму виключення змінних полягає в обчисленні ймовірності вершини мережі за формулою:

$$P(X^{(n)}, e) = \sum_{X^{(k)}} ... \sum_{X^{(3)}} \sum_{X^{(2)}} \prod_{i} P(X^{(i)} \mid pa(X^{(i)})), \quad (3.14)$$

Тобто, чим менше вершина зв'язана з $X^{(n)}$, тим віддаленіше вона знаходяться за межами внутрішньої суми; кожна внутрішня сума після обчислення перетворюється на нову змінну, яка надалі використовується як множник. Рис. 3.14 ілюструє приклад послідовного виключення вершин мережі Байєса

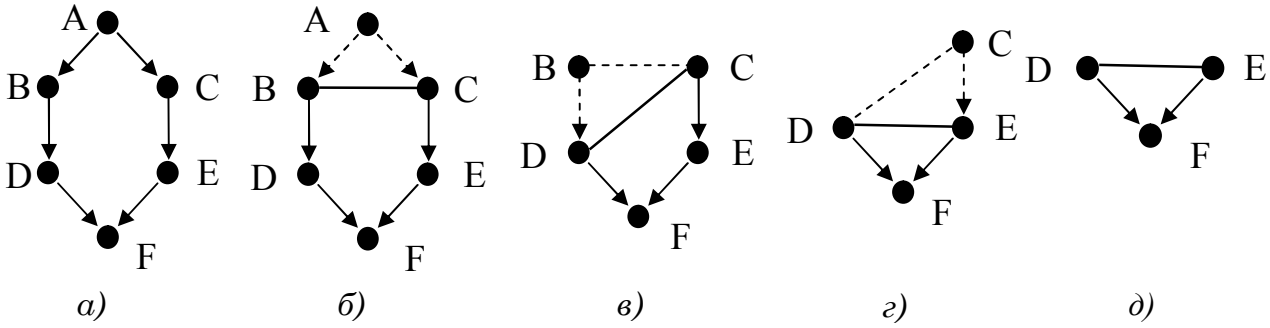


Рис. 3.14. Алгоритм виключення змінних – послідовне перетворення структури мережі Байєса: а) початкова мережа Байєса; б) виключення вершини A; в) виключення вершини B; г) виключення вершини C; д) остаточний результат – кліка з трьох вершин

Послідовність обчислень шуканої ймовірності для відповідної мережі має наступний вид.

1. Знаходження сукупного розподілу ймовірності вершин A, B і C: $P(A,B,C) = P(A) \cdot P(B \mid A) \cdot P(C \mid A)$. Виключення вершини A: $P(B,C) = \sum_A P(A,B,C)$.

2. Знаходження сукупного розподілу ймовірностей вершин B, C, і D: $P(B,C,D) = P(B,C) \cdot P(D \mid B)$. Виключення вершини B: $P(C,D) = \sum_B P(B,C,D)$.

3. Знаходження сукупного розподілу ймовірностей вершин C, D і E: $P(C,D,E) = P(C,D) \cdot P(E \mid C)$. Виключення вершини C: $P(D,E) = \sum_C P(C,D,E)$.

4. Знаходження сукупного розподілу ймовірностей вершин D, E, і F: $P(D,E,F) = P(D,E) \cdot P(F \mid D,E)$. Виключення вершини D: $P(E,F) = \sum_D P(D,E,F)$

5. Виключення вершини E: $P(F) = \sum_E P(E,F)$.

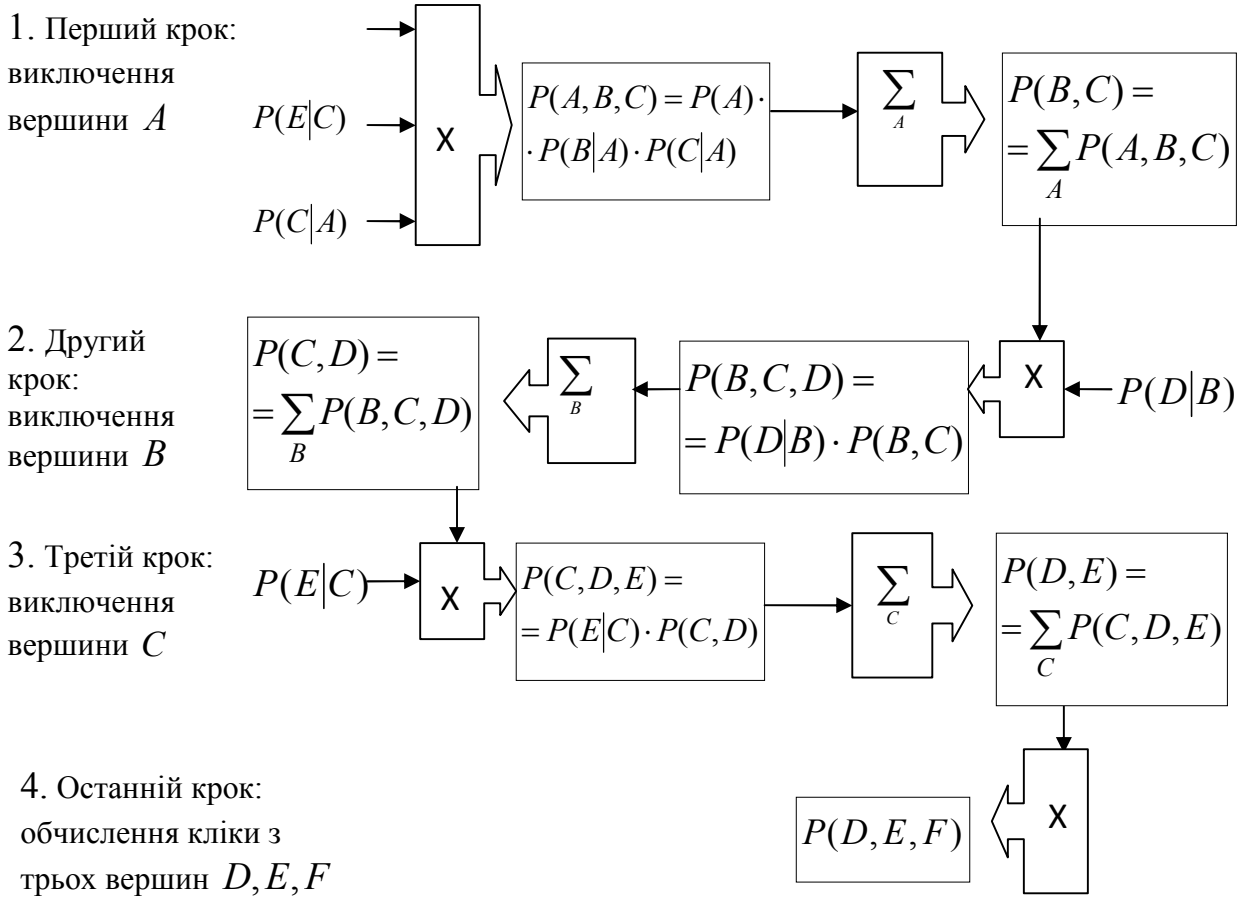


Рис. 3.15. Послідовність обчислень алгоритму виключення змінних

Докладнішу інформацію стосовно алгоритму та його практичного застосування можна знайти в літературі [69, 70, 96, 97].

Також існують різноманітні модифіковані версії цього алгоритму [98, 99], серед яких необхідно окремо відзначити ***алгоритм поглинаючого виключення*** (*bucket elimination*), запропонований Ріною Дехтер (Rina Dechter) [95]. Іноді в літературі він зустрічається під назвою алгоритм поглинаючого виключення для оцінювання довіри або алгоритм BEBA (bucket elimination for belief assessment).

Аналогічно алгоритму виключення змінних цей метод робить послідовне виключення змінних у мережі, при цьому кожна таблиця умовних ймовірностей конвертується в λ - таблицю, в якій кожне значення вершини та її батька асоціюється з відповідним λ - значенням. На основі цього підходу розроблено цілу низку алгоритмів *elim-mpe*, *elim-bel*, *elim-map*, *elim-meu* та інші [95].

Як приклад, на рис. 3.16 представлена мережа Байєса, для якої відома ВМВ $\{A,C,B,F,D,G\}$ та задано інстанційоване значення g=1, а крім того, схематично представлено приклад процесу поглинання вершин.

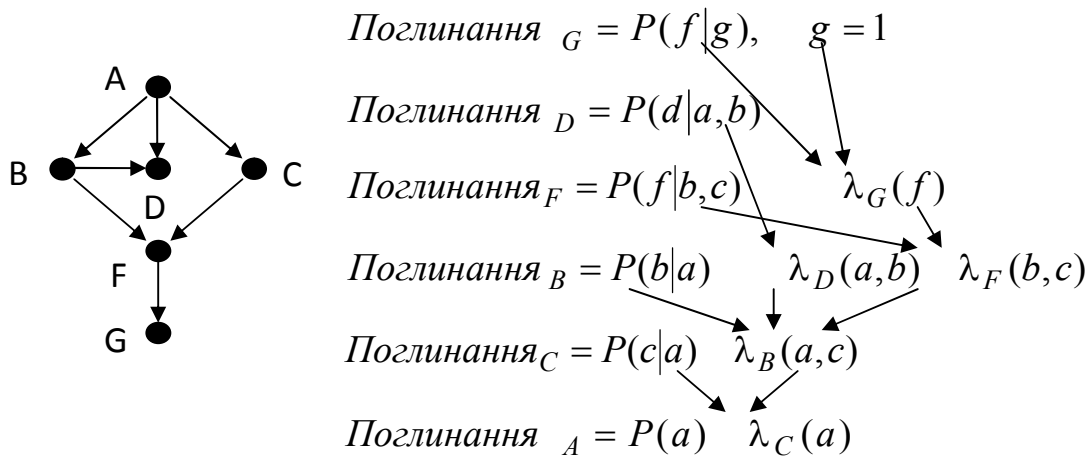


Рис. 3.16. Приклад роботи методу поглинаючого виключення

$$P(a,g=1)=P(a)\cdot\sum_f\sum_d\sum_c P(c\,|\,a)\sum_b P(b\,|\,a)\cdot P(d\,|\,a,b)\cdot P(f\,|\,b,c)\sum_{g=1} P(g\,|\,f)=$$

$$=P(a)\cdot\sum_f\lambda_G(f)\sum_d\sum_c P(c\,|\,a)\sum_b P(b\,|\,a)\cdot P(d\,|\,a,b)\cdot P(f\,|\,b,c)=$$

$$=P(a)\cdot\sum_f\lambda_G(f)\sum_d\sum_c P(c\,|\,a)\cdot\lambda_B(a,b,c,f)=P(a)\cdot\sum_f\lambda_G(f)\sum_d\lambda_C(a,d,f)=$$

$$=P(a)\cdot\sum_f\lambda_G(f)\cdot\lambda_D(a,f)=P(a)\cdot\lambda_F(a)$$

Алгоритми формування символьного ймовірнісного висновку

При розв'язанні практичних задач нерідко зустрічаються випадки, коли невідоме точне чисельне значення ймовірності, а відомий деякий інтервал значень. Найчастіше така ситуація зустрічається при роботі з експертами. Для вирішення даної проблеми застосовують ***SPI-алгоритми*** (*symbolic probabilistic inference*) або ***алгоритми символьного ймовірнісного висновку***.

Свою назву ці алгоритми отримали завдяки тому, що нечіткі значення станів вершин мережі замінюють параметрами-символами, як це робиться в табл. 5.6 для вершини C [71]. Існує два підходи для здійснення символьного ймовірнісного висновку.

Перший підхід полягає у виконанні необхідної мінімальної кількості обчислень, тобто здійсненні тільки тих обчислень, які дійсно необхідні для побудови ймовірнісного висновку.

Головна ідея полягає у переході від розв'язання задачі формування ймовірнісного висновку в мережі Байєса до комбінаторної задачі оптимізації, а саме розв'язання OFP- задачі (*optimal factoring problem*), тобто **задачі оптимальної факторизації** [100, 101], яка полягає у пошуку такої факторизації α, за якої $\mu_\alpha(s_{\{1,2,...,N\}} \to \min$, де $S_{\{i\}} \subset V = \{X^{(1)},...,X^{(N)}\}$, а μ - ***оціночна функція, яка визначається*** так:

$$\mu(S_{\{i\}} = 0, для\ i \in [1;N],\ \mu(S_{I\cup J}) = \mu(S_I) + \mu(S_J) + 2^{card(S_I \cup S_J)}, \tag{3.15}$$

де *card(S)* - потужність множини S, тобто кількість елементів множини.

Наприклад, у випадку мережі, зображеної на рис. 3.17, для обчислення значення спільної ймовірності вершин d і e можна використовувати таку факторизацію:

$$p(d,e) = \left[\sum_a \left[\sum_b \left[\sum_c \left[P(e\,|\,c)\cdot P(d\,|\,b,c)\right]\cdot P(c\,|\,a)\right]\cdot P(b\,|\,a\right]\cdot P(a)\right] \tag{3.16}$$

У цьому випадку треба виконати 72 операції множення. Однак для цього ж випадку можна скористатись іншою факторизацією:

$$P(d.e) = \left[\sum_c \left[P(e\,|\,c\left[\sum_b P(d\,|\,b,c)\left[\sum_a P(c\,|\,a\left[P(b\,|\,a)\cdot P(a)\right]\right]\right]\right]\right], \tag{3.17}$$

для реалізації якої необхідно виконати лише 28 операцій множення.

При цьому вважається, що кожна вершина мережі може перебувати в одному з двох станів.

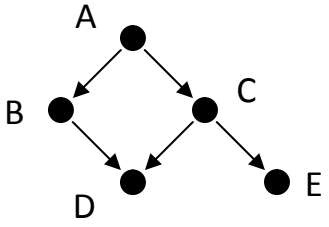


Рис. 3.17. Приклад мережі Байєса для задач *OFP* – оптимальної факторизації та *SPI* – символьного ймовірнісного висновку

Таблиця 3.6

Таблиця значень маргінальних та умовних ймовірностей вершин мережі Байєса, зображеної на рис. 3.17, із використанням параметрів-символів θ

Вершина	**Параметри-символи θ**	
	Значення вершини = 0	**Значення вершини = 1**
A	$\theta_{10} = P(A=0) = 0{,}2$	$\theta_{11} = P(A=1) = 0{,}8$
B	$\theta_{200} = P(B=0 \mid A=0) = 0{,}3$ $\theta_{201} = P(D=0 \mid A=1) = 0{,}5$	$\theta_{210} = P(B=1 \mid A=0) = 0{,}7$ $\theta_{211} = P(B=1 \mid A=1) = 0{,}5$
C	$\theta_{300} = P(C=0 \mid A=0)$ $\theta_{301} = P(C=0 \mid A=1) = 0{,}5$	$\theta_{310} = P(C=1 \mid A=0)$ $\theta_{311} = P(C=1 \mid A=1) = 0{,}5$
D	$\theta_{4000} = P(D=0 \mid B=0, C=0) = 0{,}1$ $\theta_{4001} = P(D=0 \mid B=0, C=1) = 0{,}3$ $\theta_{4010} = P(D=0 \mid B=1, C=0) = 0{,}8$ $\theta_{4011} = P(D=0 \mid B=1, C=1) = 0{,}4$	$\theta_{4100} = P(D=1 \mid B=0, C=0) = 0{,}9$ $\theta_{4101} = P(D=1 \mid B=0, C=1) = 0{,}7$ $\theta_{4110} = P(D=1 \mid B=1, C=0) = 0{,}2$ $\theta_{4111} = P(D=1 \mid B=1, C=1) = 0{,}6$
E	$\theta_{500} = P(E=0 \mid C=0) = 0{,}3$ $\theta_{501} = P(E=0 \mid C=1) = 0{,}1$	$\theta_{510} = P(E=1 \mid C=0) = 0{,}7$ $\theta_{511} = P(E=1 \mid C=1) = 0{,}9$

Другий підхід, запропонований Кастілло (Castillo), полягає у використанні символьного ймовірнісного висновку, який також працює не з чисельними даними, а з параметрами-символами [71-73]. Цей метод в якості рішення видає функції, представлені у параметрично-символьній формі (табл. 3.7).

Таблиця 3.7

Таблиця значень маргінальних та умовних ймовірностей вершин мережі Байєса, зображеної на рис. 3.17 із використанням параметра-символу θ_{300}, за умови, що вершини B та E інстанційовані (*B=1, E=1*)

Вершина	**Значення ймовірності вершини**	**Значення умовної ймовірності вершини у параметрично-символьному вигляді**
A	$P(A=0) = 0{,}2$	$P(A=0 \mid B=1, E=1) = \dfrac{0{,}126 - 0{,}028 \cdot \theta_{300}}{0{,}446 - 0{,}028 \cdot \theta_{300}}$
B	$P(B=0) = 0{,}46$	$P(B=0 \mid B=1, E=1) = 0$
C	$P(C=0) = 0{,}42 + 0{,}2 \cdot \theta_{300}$	$P(C=0 \mid B=1, E=1) = \dfrac{0{,}14 - 0{,}098 \cdot \theta_{300}}{0{,}446 - 0{,}028 \cdot \theta_{300}}$
D	$P(D=0) = 0{,}424 - 0{,}056 \cdot \theta_{300}$	$P(D=0 \mid B=1, E=1) = \dfrac{0{,}164 - 0{,}021 \cdot \theta_{300}}{0{,}446 - 0{,}028 \cdot \theta_{300}}$
E	$P(E=0) = 0{,}18 + 0{,}04 \cdot \theta_{300}$	$P(E=0 \mid B=1, E=1) = 0$

Після цього, підставляючи замість параметрів-символів фактичні чисельні значення, виконується обчислення чисельних значень ймовірнісного висновку. Так, на рис. 3.18 продемонстрована ситуація, коли робиться заміна параметра-символу θ_{300} з табл. 3.7 чисельним значенням 0,4.

Цей підхід також дозволяє досліджувати чутливість рішення, завдяки послідовному збільшенню або зменшенню чисельних значень параметрів-символів. Програмну реалізацію свого методу Кастілло виконував за допомогою засобів Mathematica та Maple.

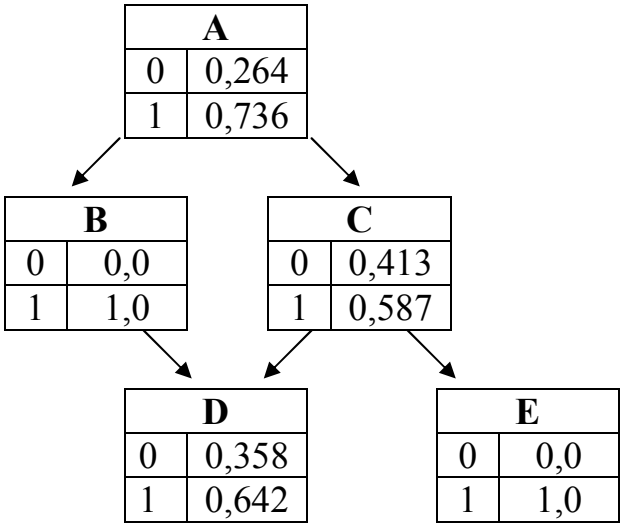


Рис. 3.18. Ймовірності станів вершин мережі Байєса, (рис. 3.17), при заміні параметра-символу в табл. 3.7 чисельним значенням $\theta_{300} = 0{,}4$.

Однак обидва підходи мають однакові проблеми: необхідність залучення спеціалізованого програмного забезпечення та велика обчислювальна складність методів, що є особливо критичним при роботі з великими мережами Байєса.

Диференціальний підхід

У 2000 році Дарвичем [102] було запропоновано диференціальний підхід (differential approach). Фактично, це єдиний відомий алгоритм для побудови ймовірнісного висновку із застосуванням диференціалів. Алгоритм складається з двох етапів: на першому етапі будується поліном, який описує ймовірнісний висновок; на другому етапі шукають перші та другі похідні поліному і будують ймовірнісний висновок, використовуючи їх властивості.

На першому етапі за допомогою упорядкованої множини вершин мережі Байєса представляють у вигляді поліноміального рівняння. Цей етап має розрахункову складність порядку $O(n \cdot \exp(w))$, де n – кількість вершин, а w - кількість станів вершин, для мережі Байєса з рис. 3.19 $n = 2, w = 2$.

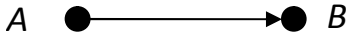


Рис. 3.19. Приклад простої мережі Байєса

Ідея використання рівнянь з параметрами-факторами за своєю суттю схожа на *SPI*-метод.

Таблиця 3.8

Таблиця значень ймовірності вершини A та параметрів θ для мережі Байєса, наведеної на рис. 3.19

A	**Значення ймовірності *P(A)***	**Параметр мережі θ**
α	0,3	$\theta_{\alpha} = 0{,}3$
$\overline{\alpha}$	0,7	$\theta_{\overline{\alpha}} = 0{,}7$

Таблиця 3.9

Таблиця умовних ймовірностей $P(A \mid B)$ та параметрів θ для мережі Байєса (рис. 3.19)

A	***B***	**Значення ймовірності** $P(A \mid B)$	**Параметр мережі θ**
α	b	0,1	$\theta_{ab} = 0{,}1$
α	$\overline{b}$	0,9	$\theta_{a\overline{b}} = 0{,}9$
$\overline{\alpha}$	b	0,8	$\theta_{\overline{a}b} = 0{,}8$
$\overline{\alpha}$	$\overline{b}$	0,2	$\theta_{\overline{a}\overline{b}} = 0{,}2$

Таблиця 3.10

Таблиця значень спільної ймовірності та параметризація вершин *A* та *B*

A	***B***	$P(A,B)$	**Параметризація**
α	b	0,03	$\lambda_a \cdot \lambda_b \cdot 0{,}03$
α	$\overline{b}$	0,27	$\lambda_a \cdot \lambda_{\overline{b}} \cdot 0{,}27$
$\overline{\alpha}$	b	0,56	$\lambda_{\overline{a}} \cdot \lambda_b \cdot 0{,}56$
$\overline{\alpha}$	$\overline{b}$	0,14	$\lambda_{\overline{a}} \cdot \lambda_{\overline{b}} \cdot 0{,}14$

Таблиця 3.11

Таблиця значень індикаторів λ залежно від наявності інформації про стан вершин A та B

Стан вершин *A* та *B*	$\lambda_{\overline{a}}$	$\lambda_{\overline{b}}$	λ_b	λ_a
a, b	0	0	1	1
$\alpha, \overline{b}$	0	1	0	1
a	0	1	1	1
Відсутня інформація про стан *A* та *B*	1	1	1	1

Для мережі Байєса, зображеної на рис. 3.19, поліноміальне рівняння у канонічному вигляді записується як:

$$F = F(\lambda,\theta) = \sum_{i=1}^{n} F(\lambda_{x_i},\theta_{f_i}) = \sum_{i=1}^{n} \left(\left(\prod \theta_{f_i} \right) \cdot \left(\prod \lambda_{x_i} \right) \right) = \theta_a \cdot \theta_{ab} \lambda_a \cdot \lambda_b + \\ + \theta_a \cdot \theta_{a\overline{b}} \cdot \lambda_a \cdot \lambda_{\overline{b}} + \theta_{\overline{a}} \cdot \theta_{\overline{a}b} \cdot \lambda_{\overline{a}} \cdot \lambda_b + \theta_{\overline{a}} \cdot \theta_{\overline{a}\overline{b}} \cdot \lambda_{\overline{a}} \cdot \lambda_{\overline{b}}, \qquad (3.18)$$

де λ - індикатор, а θ - параметри мережі, тобто значення ймовірності

$\theta_a = P(A = a)$, або умовної ймовірності $\theta_{ab} = P(B = b \mid A = a)$.

При цьому за теоремою Байєса виконується рівність:

$$P(A = a, B = b) = P(B = b \mid A = a) \cdot P(A = a) = \theta_a \cdot \theta_{ab} \tag{3.19}$$

На другому етапі виконується обчислення перших та других похідних полінома. Оскільки для перших та других похідних полінома F виконуються властивості, які застосовують при побудові ймовірнісного висновку, то підставляючи в отримані диференціальні рівняння фактичні значення змінних та інстанційованних вершин, обчислюють відповідні значення станів вершин.

Так, для перших похідних $\frac{\partial F}{\partial \lambda_{x_i}}$ та $\frac{\partial F}{\partial \theta_{f_i}}$ виконуються такі властивості:

$$\frac{\partial F(\lambda,\theta)}{\partial \lambda_x} = P(x, e - X \mid \theta) \text{ та } \frac{\partial F(e,\theta)}{\partial \theta_f}\theta(f) = P(f, e \mid \theta), \tag{3.20}$$

де f - це множина вершин мережі (family); e - множина інстанційованих станів; X - вершина мережі Байєса. Вираз e-X означає підмножину e, в якій відсутня X; наприклад, якщо $e = \{a, b, \bar{c}\}$, то виконуються умови: $e - A = \{b, \bar{c}\}$ та $e - AC = \{b\}$. Тобто чисельне значення $\frac{\partial F}{\partial \lambda_{x_i}}$ – це ймовірність для вершини X в i–му стані.

Також для кожної вершини X та множини інстанційованних станів e виконуються рівняння:

$$P(e - X) = \sum_x \frac{\partial F(e)}{\partial \lambda_x} \text{ та } P(x \mid e - X) = \frac{\partial F(e)/\partial \lambda_x}{\sum_x \partial F(e)/\partial \lambda_x}. \tag{3.21}$$

Наприклад, для мережі Байєса (рис. 3.19) у випадку, коли $e = a$ маємо:

$$P(e - A) = \frac{\partial F(e)}{\partial \lambda_a} + \frac{\partial F(e)}{\partial \lambda_{\bar{a}}} = 1,\ P(\bar{a} \mid e - A) = \frac{\partial F(e)/\partial \lambda_{\bar{a}}}{\partial F(e)/\partial \lambda_a + \partial F(e)/\partial \lambda_{\bar{a}}} = \frac{0{,}7}{1} = 0{,}7.$$

Для других похідних, за умови що $X \neq Y$ та $F_i \neq F_j$, виконуються властивості:

$$\frac{\partial^2 F(e,\theta)}{\partial \lambda_x \cdot \partial \lambda_y} = P(x, y, e - XY \mid \theta); \tag{3.22}$$

$$\frac{\partial^2 F(e,\theta)}{\partial \lambda_x \cdot \partial \theta_{f_i}} \cdot \theta(f_i) = P(x, f_i, e - X \mid \theta); \tag{3.23}$$

$$\frac{\partial^2 F(e,\theta)}{\partial \theta_{f_i} \cdot \partial \theta_{f_j}} \cdot \theta(f_i) \cdot \theta(f_j) = P(f_i, f_j, e \mid \theta). \tag{3.24}$$

Обчислювальна складність при знаходженні частинних похідних $\partial F / \partial \lambda_{x_i}$ та $\partial F / \partial \theta_{f_i}$ дорівнює $O(n \cdot \exp(w))$, а при $\frac{\partial^2 F}{\partial \lambda_{x_i} \cdot \partial \theta_{f_i}}$ дорівнює $O(n^2 \cdot \exp(w))$.

Інші алгоритми формування точного висновку

У 1989 році Р. Шехтер [75] запропонував **AR-алгоритм** від англійського *arc reversal*, тобто *реверсування дуг*. Іноді AR-алгоритм можна зустріти під назвою *зменшення кількості вершин* (*node reduction*). Цей алгоритм є одним із найперших алгоритмів точного висновку. Він виконує модифікацію мережі Байєса шляхом послідовного реверсування дуг, застосовуючи теорему Байєса та інформацію про інстанційовані вершини. Л. Гааг у 1993 році описав **алгоритм поглинання подій** (*evidence absorption*) [103], який є симбіозом *алгоритму Перла* [59] та *AR-алгоритму* [75]. Тобто, до категорії "інших" належать алгоритми, що є комбінацією декількох алгоритмів різних типів.

Апроксимаційні алгоритми формування висновку

На практиці, при аналізі задач, які складаються з сотень, тисяч а іноді і десятків тисяч факторів, з'являється потреба формування ймовірнісного висновку у великих мережах Байєса, які складаються з тисяч вершин. Для таких мереж алгоритми формування точного висновку не працюють у зв'язку з великою обчислювальною складністю, яка близька до експоненційної. Саме тому були розроблені апроксимаційні алгоритми, які, жертвуючи точністю обчислень, роблять можливим вирішення задач великої розмірності. Апроксимаційні алгоритми поділяються на:

- алгоритми стохастичної вибірки;
- алгоритми неповного висновку;
- варіаційні алгоритми;
- пошукові алгоритми.

Одним із прикладів практичного застосування апроксимаційних алгоритмів можна навести систему медичної діагностики *(QMR –швидка медична довідка* (*Quick Medical Reference*)), яка представлена на рис.3.20, її інформаційна база складається із статистичних та експертних даних [104]. QMR-система почала розроблятися у Пітсбургському університеті у 1980 році, а пізніше увійшла до складу системи Internist-I [104] як один із діагностичних інструментів лікаря-терапевта. Розробку системи Internist-I було розпочато на початку 1970-х років, а першу версію закінчено в 1974 р. Сьогодні мережа Байєса QMR-системи складається приблизно з 6000 вершин, з'єднаних більш ніж 415000 дугами. Система спроможна розпізнати близько 750 видів різних захворювань за більше ніж 5000 симптомами та результатами лабораторних аналізів.

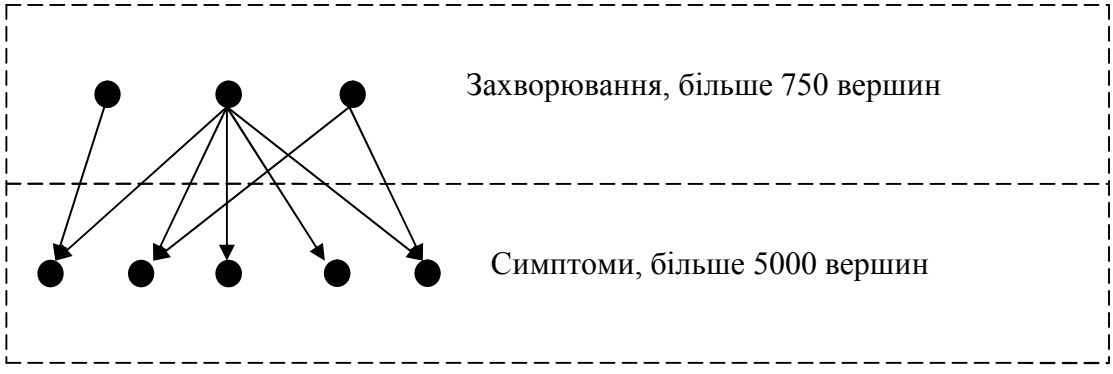


Рис. 3.20. Мережа Байєса для QMR системи (швидка медична довідка)

Аналізуючи QMR-систему, можна зрозуміти необхідність переходу від точних методів побудови ймовірнісного висновку до апроксимаційних, жертвуючи при цьому точністю обчислень. Однак цей крок дозволяє розширити межі застосування байєсівських мереж.

Алгоритми стохастичного вибору

Серед апроксимаційних алгоритмів саме алгоритми стохастичного вибору найчастіше використовують для формування ймовірнісного висновку. У літературі ця група методів зустрічається під назвами *стохастичної дискретизації* (*stochastic sampling*) або *стохастичного моделювання* (*stochastic modeling*).

Алгоритми стохастичної дискретизації поділяють на дві групи:

– дискретизації вибірки за значущістю;

– методи Монте-Карло для ланцюгів Маркова (МКМЛ).

Ідея *дискретизації вибірки за значущістю* (*importance sampling*) ґрунтується на припущенні, що деякі значення випадкової величини в процесі моделювання мають більшу значущість (ймовірність) для оцінюваної функції (параметра), ніж інші. Якщо ці "ймовірніші" значення будуть з'являтися в процесі вибору випадкової величини частіше, дисперсія оцінюваної функції зменшиться. Отже, базова методологія дискретизації вибірки за значущістю полягає у виборі розподілу, що сприяє вибору "ймовірніших" значень випадкової величини. Такий "зміщений" розподіл змінює оцінювану функцію, якщо він застосовується безпосередньо в процесі розрахунку. Однак результат розрахунку змінюється відповідно до цього зміщеного розподілу і це гарантує, що нова оціночна функція вибірки за значущістю не буде зміщеною.

Методи Монте-Карло для ланцюгів Маркова (МКЛМ) – поширений клас методів для побудови ймовірнісного висновку в байєсівській мережі; в літературі часто зустрічається під назвою *MCMC* (*Markov chain Monte Carlo*) [105]. Методи Монте-Карло – це загальна назва групи чисельних методів, які ґрунтуються на генеруванні великої кількості реалізацій випадкового процесу, що формується таким чином, щоб його ймовірнісні характеристики збігалися з аналогічними величинами розв'язуваного завдання. Однією з реалізацій *методу Монте-Карло* є *алгоритм Метрополіса* (*Metropolis algorithm*), який запропоновано у 1953 р. в роботі [106].

В свою чергу, у 1984 р. на основі *алгоритму Метрополіса-Хастінга* (*Metropolis-Hastings algorithm*), який є модифікацією *алгоритму Метрополіса,* запропоновано метод *Монте-Карло для ланцюгів Маркова* для мереж Байєса [107].

Наприклад, якщо мережа Байєса складається з множини вершин $X = \{X_1,...,X_n\}$, то математичне сподівання функції $a(X_1,...,X_n)$, яка описує розподіл множини X, записується апроксимаційним рівнянням:

$$\langle a \rangle = E(a(X)) = \sum_{\tilde{x}_1} ... \sum_{\tilde{x}_n} a(\tilde{x}_1,..., \tilde{x}_n) \cdot P(X_1 = \tilde{x}_1,..., X_n = \tilde{x}_n) \approx$$

$$\approx \frac{1}{N} \cdot \sum_{t=0}^{N-1} a(x_1^{(t)},..., x_n^{(t)}) \quad , \qquad (3.25)$$

де N - розмір вибірки; $x_1^{(t)},..., x_n^{(t)}$ – значення вершин на t - му кроці.

Серед різних варіацій алгоритмів Монте-Карло для ланцюгів Маркова найпопулярнішими є алгоритми дискретизації Гіббса (Gibbs sampler)[108] та Метрополіса-Хастінгса (Metropolis-Hastings) [109].

Окрім алгоритмів вибірки за значущістю та методів Монте-Карло для ланцюгів Маркова вирізняють наступні ***методи стохастичної дискретизації***:

- алгоритм логічного відбору (*logic sampling*), запропонований у 1986 р. Генріоном (Henrion) [111], є одним із найпростіших алгоритмів прямої вибірки. Ідея алгоритму полягає у багаторазовому моделюванні вибірки за мережею проходу за напрямком зв'язків між вершинами, відкиданні елементів вибірки, що протирічать спостереженням і оцінці ймовірностей шуканих змінних за допомогою частоти появи відповідних подій у вибірці. Якщо вирішується задача без спостережень, то алгоритм логічної вибірки працює дуже добре, але у випадку малоймовірного спостереження з'являються проблеми. Частка правильних елементів вибірки експоненційно зменшується залежно від кількості спостережень.

- ***алгоритм правдоподібного зважування*** (*LW – likelihood weighting*) – запропонований в 1989 р. Р. Фангом та К. Чангом [112], удосконалює алгоритм логічної вибірки [113]. Він працює швидше навіть у більших мережах, але проблема малоймовірних подій все одно залишається.

- алгоритм зворотної дискретизації (*backward sampling***)** запропонований Р. Фангом та Б. Фаверо в 1994 р. [114]. Ідея цього методу, як і алгоритму правдоподібного зважування [110, 112, 114], полягає в обчисленні значень ймовірностей станів вершин із застосуванням частоти появи відповідних подій, які відповідають конкретному стану вершини у навчальній множині моделей.

- адаптивний алгоритм дискретизації з відхиленням (*ARS – adaptive rejection sampling*) , запропонований у 1992 р. В. Гілксом та П. Вальдом [115, 116] і удосконалений у 2002 р. К. Кохненом [117], застосовується для мереж Байеса, в яких функції розподілу вершин описуються логарифмічними функціями.

- ***алгоритм BN-RAS*** (*Bayesian network randomized approximation scheme*) – запропонований у 1990 р. Р. Чавесом та Дж. Купером [195] демонструє прийнятні результати при роботі з великими мережами Байєса, в яких наявна велика кількість дуг між вершинами.

- ***алгоритми SIS*** (*self-importance sampling*) та ***HIS*** (*heuristic importance sampling*), розроблені Р. Шахтером та М. Питом ще в 1990 р. [118], відрізняються між собою методом генерування вибірки за значущістю: SIS використовують оціночну функцію, а HIS – модифікований алгоритм розповсюдження повідомлення за однозв'язаним деревом.

- алгоритм AIS (*adaptive importance sampling*) розробили Дж. Ченг та М. Друздзел у 2000 році [119]. У цьому алгоритмі при генеруванні навчальних даних на різних етапах навчання застосовують різні вагові коефіцієнти. У 2001 році, завдяки введенню в AIS нового правила зупинки, запропоновано ефективніший алгоритм ***AIS-BN*** [120].

Всі алгоритми стохастичної дискретизації відрізняються один від одного функціями генерування вибірки, принципом побудови вагових коефіцієнтів цих функцій та засобами генерування множини випадкових моделей для відповідного типу розподілу.

Алгоритми формування часткового або неповного висновку

Дану групу методів іноді називають по-іншому, *методами спрощення структури* (*model simplification methods*). Ідея цих методів полягає у послідовній зміні структури мережі Байєса таким чином, щоб зменшити її структурну складність настільки, щоб стало можливим застосування точних методів формування висновку. Наприклад, Ф. Дженсен та С. Андерсен [121] пропонували анігілювати маленькі (незначні) ймовірності, а Kjaerulff U. [122] та R. Engelen [123] – позбавлятися дуг, які відображають слабкий зв'язок між вершинами. Зрозуміло, що отримане таким чином значення ймовірнісного висновку, буде відрізнятися від реального. Існує багато методів формування часткового висновку, зокрема:

- ***алгоритм визначаючого обмеження*** (*bounded conditioning*) [124] – один із перших алгоритмів, запропонований у 1988 р. Horvitz E.J., Suermondt H.J. and Cooper G.F. [125], доповнений у 1994 р. Draper D. and Hanks S. [126] За своєю ідеєю він схожий на метод визначаючого перетину [24] в тому, що для зменшення обчислювальної складності необхідно позбавляються циклів всередині мережі Байєса, визначивши множину вершин, які утворюють циклічні перетини (cycle cutset);

- ***алгоритм локального часткового оцінювання*** або **LPE** (*localized partial evaluation*) [126]. При обчисленні ймовірностей станів вершини алгоритм ігнорує частину мережі, а саме ті вершини, які знаходяться достатньо далеко від неї;

- **покроковий алгоритм SPI** (*incremental SPI*), представлений Д'Амбросио в 1993 р., є модифікованою версією алгоритму точного символьного висновку для роботи з великими мережами Байєса [127];

- ідея **алгоритму часткового оцінювання ймовірності** (*probabilistic partial evaluation*) [128] полягає в тому, щоб поліпшити процедуру обчислення ймовірностей моделі завдяки ігноруванню відмінностей між схожими ймовірностями; особливістю алгоритму є використання поняття батьківського контексту (parent context), яке дозволяє робити декомпозицію ймовірнісного розподілу по аналогії з SPI-алгоритмом та застосовування VE або BEBA алгоритмів, які за упорядкованою множиною вершин роблять згортку параметрів, що, у свою чергу, дозволяє визначати та ігнорувати відмінності між ймовірностями;

- ***алгоритм mini-bucket*** запропонувала Р. Дехтер в 1997 р. [129] як модифікацію алгоритму поглинаючого виключення [95], згодом на основі даного алгоритму були розроблені алгоритми *aprox-opt* та *aprox-bel-max;*

- в алгоритмі ***Sarkar*** [130] для зменшення обчислювальної складності запропоновано апроксимацію структури мережі, а саме перехід від байєсівської мережі до байєсівського дерева;

- ***алгоритм зменшення розміру таблиці умовних ймовірностей у просторі станів*** (*state space abstraction algorithm*) [131] передбачає, що для зменшення обсягів обчислень і спрощення моделі слід зменшити розмір таблиці умовних ймовірностей кожної вершини.

Варіаційні алгоритми

Головна ідея варіаційних алгоритмів полягає в усередненні значень ймовірностей вершин, тобто при обчисленні розглядаються тільки значущі величини, а всі інші не враховуються. Для кожної вершини мережі розривають

зв'язки з іншими вершинами і вводять нові варіаційні параметри. Завдяки послідовній зміні значень варіаційних параметрів досягається мінімізація перехресної ентропії між апроксимаційним та фактичним значенням ймовірності. Мережа трансформується в підграф початкового графу, в якому деякі вершини позбавляються зв'язків, доки не буде можливо застосувати точний алгоритм ймовірнісного висновку.

Алгоритм VB (*variational Bayesian*) описано Ghahramani Z. G. and Beal M. J. [132], а докладний опис роботи різних варіаційних алгоритмів при формуванні ймовірнісного висновку для QMR-системи – у роботах Jordan M., Jaakkola T.S. та ін. [104, 133].

Пошукові алгоритми

Основою пошукових алгоритмів (search-based) є ідея переходу від задачі ймовірнісного висновку до оптимізаційної задачі пошуку найбільш ймовірного значення. Вважається, що найбільш ймовірну частку мережі складає порівняно невелика кількість вершин, і, виконавши пошук найбільш ймовірних інстанціювань, можна отримати границі апостеріорних ймовірностей значень вершин. Тобто пошукові алгоритми ґрунтуються на евристичних алгоритмах пошуку, які використовують при переході від розв'язання задачі ймовірнісного висновку до оптимізаційної задачі. За своєю метою ця група алгоритмів схожа на алгоритми стохастичної дискретизації: у даному випадку також вирішується задача пошуку найбільш ймовірного висновку [134]. Дана група алгоритмів широко використовується в діагностиці – медичній та інженерно-технічній.

На базі цього підходу Дж. Купер у 1984 р. розробив медичну інформаційну систему NESTOR [135], в 1991 р. Henrion запропонував *Top-N* метод [111], а в 1993 р. Poole розробив Conflict – метод медичної діагностики [136]. У 1994 році Драздзел теоретично обґрунтував, що асиметрія сукупного розподілу ймовірностей мережі залежить від асиметрій таблиць умовних ймовірностей. Однак емпіричні результати Lin та Druzdzel показали, що хоча невелика частина всієї кількості інстанціювань справді займає найбільш ймовірну частку мережі, вона все одно дуже велика.

У 1995 році створено алгоритм з використанням генетичного підходу для паралельних обчислень у системі OVERMIND, яка, в свою чергу, є частиною он-лайн експертної системи PESKI, призначеної для інженерно-технічної діагностики в космічній програмі Space Shuttle [137].

У 1998 році запропоновано алгоритм для мереж Байєса з великою кількістю дуг [138], який ґрунтується на IB-підході (independence-based), ідея якого полягає в агрегуванні ймовірностей з метою зменшення об'єму обчислень.

3.3. Формування ймовірнісного висновку у байєсівській мережі на основі навчальних даних

Для формування ймовірнісного висновку вхідними даними є такі:

1. Множина навчальних даних $D = \{d_1,...,d_n\}$, де $d_i = \{x_i^{(1)}, x_i^{(2)},..., x_i^{(N)}\}$ (нижній індекс – номер спостереження, а верхній – номер змінної), n – кількість спостережень. Кожне спостереження складається з $N(N \geq 2)$ змінних

$X^{(1)}, X^{(2)}, \ldots, X^{(N)}$, кожна *j*-а змінна *(j=1,…,N)* має $A^{(j)} = \{0,1,\ldots,\alpha^{(j)}-1\}$, де $(\alpha^{(j)} \geq 2)$ станів.

2. Структура мережі Байєса G представлена множиною з N предків $(\Pi^{(1)},\ldots,\Pi^{(N)})$, тобто для кожної вершини j=1,…,N; $\Pi^{(j)}$ – множина батьківських вершин, при цьому $\Pi^{(j)} \subseteq \{X^{(1)},\ldots,X^{(N)}\} \setminus X^{(j)}$ (вершина не може бути батьком для самої себе, тобто петлі в графі відсутні).

3. Множина інстанційованих вершин $\{X^{(P_1)} = x^{(P_1)},\ldots,X^{(P_v)} = x^{(P_v)}\}$, тобто вершин, що перебувають у деякому певному стані з одиничною ймовірністю. Якщо множина інстанційованих вершин порожня, то потрібно використати ймовірнісний висновок, який ґрунтується на класичній теоремі Байєса.

Формування висновку за методом ймовірнісного висновку на основі навчальних даних відбувається покроково:

Крок 1. За множиною навчальних даних обчислюється матриця емпіричних значень спільного розподілу ймовірностей всієї мережі $P(X^{(1)},\ldots,X^{(N)})$ за формулою:

$$P_{matrix}(X^{(1)} = x^{(1)},\ldots,X^{(N)} = x^{(N)}) = \frac{n\left[X^{(1)} = x^{(1)},\ldots,X^{(N)} = x^{(N)}\right]}{n}, \quad (3.26)$$

де n – кількість навчальних спостережень, $x^{(j)} \in A^{(j)}$, а

$$n[X^{(1)} = x^{(1)},\ldots,X^{(N)} = x^{(N)}] = \sum_{j=1}^{n} I(X^{(1)} = x_j^{(1)},\ldots,X^{(N)} = x_j^{(N)}, \quad (3.27)$$

де функція *I(E)=1*, якщо предикат *E=true*, інакше *I(E)=0*.

Алгоритм обчислення значень ймовірностей всіх можливих станів неінстанційованих вершин:

for *j=1* ***to*** *N if* $X^{(j)} \notin \{X^{(P_1)},\ldots,X^{(P_v)}\}$ ***then***
begin
sum*=0;*
$\forall x^{(j)} \in A^{(j)}$ *do*
begin
for *k=1* ***to*** *last_string_matrix* ***do***
begin
if $(X_{matrix}^{(P_1)} = x^{(P_1)})$ ***and ...and*** $(X_{matrix}^{(P_v)} = x^{(P_v)})$
and $(X_{matrix}^{(j)} = x^{(j)})$ ***then***
begin
$P(X^{(j)} = x^{(j)}) = P(X^{(j)} = x^{(j)}) + P_{matrix}(X_{matrix}^{(1)},\ldots,X_{matrix}^{(N)})$;
end;
end;
$sum=sum+P(X^{(j)} = x^{(j)})$;
end;

$\forall x^{(j)} \in A^{(j)}$ *do*

begin

$$P(X^{(j)} = x^{(j)}) = \frac{P(X^{(j)} = x^{(j)}}{sum};$$

end;

end;

Далі обчислюються значення ймовірностей всіх можливих станів, неінстанційованих вершин за даним за алгоритмом.

Крок 2. Перебираємо послідовно всі вершини мережі Байєса. Якщо вершина не інстанційована, то потрібно обчислити значення ймовірностей всіх можливих станів цієї вершини. Для цього виконується послідовний перебір усіх рядків матриці емпіричних значень спільного розподілу ймовірностей всієї мережі. Якщо значення вершин рядка збігаються зі значеннями інстанційованих вершин і станом аналізованої вершини, то відповідне значення $P_{matrix}(X^{(1)},...,X^{(N)})$ додається до значення ймовірності відповідного стану аналізованої вершини. Після цього нормуються значення ймовірностей станів аналізованої вершини. *Вихідними даними* є значення ймовірностей всіх можливих станів всіх неінстанційованих вершин.

Приклад роботи методу ймовірнісного висновку в мережі Байєса на основі навчальних даних.

Розглянемо побудову ймовірнісного висновку з використанням класичної теореми Байєса та на основі навчальних даних.

***Вхідні дані задачі*:**

1. Множина навчальних даних з табл. 3.1.

Таблиця 3.12

Набір з 10 навчальних спостережень для мережі Байєса

Номер спостереження	**Відповіді на навчальні дані**			
	S	**C**	**B**	**D**
1	1	1	1	1
2	1	0	1	0
3	1	0	0	0
4	0	0	1	1
5	0	0	0	0
6	1	0	0	0
7	1	1	1	1
8	0	0	0	0
9	0	0	0	1
10	0	0	0	0

2. Структура мережі Байєса, наведена на рис. 3.1.

Структура мережі, що відповідає даним з таблиці 3.1 (рис. 3.21).

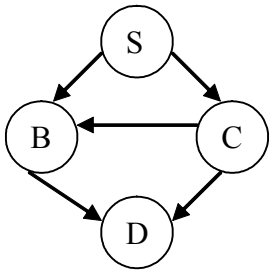


Рис. 3.21 Структура мережі Байєса, що відповідає даним табл. 3.1

3. Множина інстанційованих вершин складається з двох вершин: $\{S=0; C=0\}$, тобто пацієнт не палить і не хворий на рак: $P(S=0)=1$ та $P(C=0)=1$.

Визначити: ймовірність наявності у пацієнта задишки та бронхіту: $P(B)=?$; $P(D)=?$

Побудова ймовірнісного висновку з використанням класичної теореми Байєса.

Вхідні дані: множина навчальних даних (дані з таблиці 3.1.) При обчисленні використовуються таблиці умовних ймовірностей (рис. 3.22):

P(S)	
S=0	**S=1**
0,5	0,5

Значення вершин			
		B=0	**B=1**
S=0	**C=0**	0,8	0,2
S=0	**C=1**	0,5	0,5
S=1	**C=0**	0,67	0,33
S=1	**C=1**	0	1

Значення вершин	**P(C/S)**	
	C=0	**C=1**
S=0	1	0
S=1	0,6	0,4

Значення вершин			**P(D/B,C)**
		D=0	**D=1**
B=0	**C=0**	0,83	0,17
B=0	**C=1**	0,5	0,5
B=1	**C=0**	0,5	0,5
B=1	**C=1**	0	1

Рис. 3.22. Мережа Байєса у вигляді таблиць умовних ймовірностей вершин, що відповідають даним табл. 3.1

Обчислення. Для розрахунку ймовірності вершин "задишка" й "бронхіт" застосуємо звичайний алгоритм прямого розповсюдження інформації по мережі Байєса із використанням теореми Байєса.

$$P(B=0)=P(B=0|S=0,C=0)\cdot P(S=0,C=0)=0{,}8\cdot 1=0{,}8;$$
$$P(B=1)=P(B=1|S=0,C=0)\cdot P(S=0,C=0)=0{,}2\cdot 1=0{,}2;$$
$$P(D=0)=P(D=0,B,C=0)=P(D=0,B=0,C=0)+P(D=0,B=1,C=0)=$$
$$=P(D=0\mid B=0,C=0)\cdot P(B=0\mid C=0)\cdot P(C=0)+P(D=0\mid B=1,C=0)\cdot P(C=0)=$$
$$=0{,}83\cdot 0{,}8\cdot 1+0{,}5\cdot 0{,}2\cdot 1=0{,}764;$$
$$P(D=1)=P(D=1,B,C=0)=P(D=1,B=0,C=0)+P(D=1,B=1,C=0)=$$
$$=P(D=1\mid B=0,C=0)\cdot P(B=0\mid C=0)\cdot P(C=0)+P(D=1\mid B=1,C=0)\cdot P(C=0)\cdot$$
$$\cdot P(B=1|C=0)\cdot P(C=0)=0{,}17\cdot 0{,}8\cdot 1+0{,}5\cdot 0{,}2\cdot 1=0{,}236$$

Результат. За умови, що пацієнт не палить і не хворий на рак, ймовірність наявності у нього бронхіту дорівнює 20%, а задишки $23{,}6\%$.

Побудова ймовірнісного висновку на основі навчальних даних.

Обчислення ймовірності вершин "задишка" й "бронхіт":

$$P(B=0)=P(S=0,C=0,B=0,D=0)+P(S=0,C=0,B=0,D=1)=0{,}4;$$

$$P(B=1)=P(S=0,C=0,B=1.D=0)+P(S=0,C=0,B=1,D=1)=0{,}1;$$

$$sum=P(B=0)+P(B=1)=0{,}5;$$

$$P(B=0)=\frac{P(B=0)}{sum}=0{,}8;\ P(B=1)=\frac{P(B=1)}{sum}=0{,}2;$$

$$P(D=0)=P(S=0,C=0,B=0,.D=0)+P(S=0,C=0,B=1,D=0)=0{,}3;$$

$$P(D=1=P(S=0,C=0,B=0,.D=1)+P(S=0,C=0,B=1,D=1)=0{,}2;$$

$$sum=P(D=0)+P(D=1)=0{,}5;$$

$$P(D=0)=\frac{P(D=0)}{sum}=0{,}6;\ P(D=1)=\frac{P(D=1)}{sum}=0{,}4.$$

Результат. З отриманих обчислень випливає, що якщо пацієнт не палить і не хворий на рак, ймовірність наявності в нього бронхіту дорівнює 20%, а задишки 40%.

Аналіз отриманих результатів. Ймовірнісний висновок, який ґрунтується на прямому розповсюдженні інформації із використанням теореми Байєса щодо наявності у пацієнта задишки, дав чисельний результат $23{,}6\%$, а метод ймовірнісного висновку на основі навчальних даних - 40%. Значення ймовірності для вершини, що описує задишку для різних методів відрізняються в силу того, що при формуванні таблиць умовних ймовірностей на експериментальних даних відбувається часткова втрата інформації. Але необхідно зазначити, що запропонований метод формування ймовірнісного висновку на основі навчальних даних дає точний чисельний результат, який точно співпадає з навчальними даними з табл. 3.1.

3.4. Методи оцінювання якості побудови ймовірнісного висновку

Для оцінювання якості роботи будь-якого методу побудови ймовірнісного висновку можна скористатися такими величинами: середньоквадратична похибка, *KL*-відстань або квадратична відстань Хелліндженра (Hellinger) [88].

Середньоквадратична похибка MSE :

$$MSE = \frac{1}{\sum_{X^{(i)} \in X \setminus E} card(A^{(i)})} \cdot \sum_{X^{(i)} \in X \setminus E} \left(\sum_{\forall x^{(i)} \in X^{(i)}} (P(x^{(i)} \mid e) - \hat{P}(x^{(i)} \mid e)^2) \right), \tag{3.28}$$

KL-відстань D_K між значенням ймовірності $P(X^{(i)} \mid e)$ та оцінкою $\hat{P}(X^{(i)} \mid e)$:

$$D_K(P(X^{(i)} \mid e), \hat{P}(X^{(i)} \mid e)) = \sum_{\forall x^{(i)} \in A^{(i)}} \left(P(x^{(i)} \mid e) \cdot \log\left(\frac{P(x^{(i)} \mid e}{\hat{P}(x^{(i)} \mid e)} \right) \right). \tag{3.29}$$

KL-відстань D_K всієї мережі Байєса:

$$D_K(P, \hat{P}) = \frac{1}{card(X \setminus E)} \cdot \sum_{X^{(i)} \in X \setminus E} D_K(P(X^{(i)} \mid e), \hat{P}(X^{(i)} \mid e)) \tag{3.30}$$

Квадратична відстань Хелліндженра D_H між значенням ймовірності $P(X^{(i)} \mid e)$ та оцінкою $\hat{P}(X^{(i)} \mid e)$:

$$D_H(P(X^{(i)} \mid e), \hat{P}(X^{(i)} \mid e)) = \sum_{\forall x^{(i)} \in A^{(i)}} \left(\sqrt{P(x^{(i)} \mid e)} - \sqrt{\hat{P}(x^{(i)} \mid e)} \right)^2 . \tag{3.31}$$

Квадратична відстань Хелліндженра D_H всієї мережі Байєса:

$$D_H(P, \hat{P}) = \frac{1}{card(X \setminus E)} \cdot \sum_{X^{(i)} \in X \setminus E} D_H\left(P(X^{(i)} \mid e), \hat{P}(X^{(i)} \mid e) \right). \tag{3.32}$$

У наведених формулах $X = \left\{ X^{(1)}, ..., X^{(N)} \right\}$ - множина всіх вершин графа мережі Байєса *G*, де кожна *j*-а вершина мережі $(j = 1, ..., N)$ має $A^{(j)} = \left\{ 0, 1, ..., \alpha^{(j)} - 1 \right\}$, $(\alpha^{(j)} \geq 2)$ станів; запис $card(A^{(j)})$ означає потужність множини $A^{(j)}$ (кількість елементів, з яких складається множина); $E \subset X$, $E = e$ множина подій (інстанційовані вершини); $P(X^{(i)} \mid e)$ значення ймовірності вершини $X^{(j)}$ за умови, що мала місце подія $E = e$, $P(X^{(i)} \mid e)$ значення оцінки ймовірності.

Контрольні питання

1. Дайте означення ймовірнісного висновку у мережах Байєса. Наведіть приклади.
2. Поясніть своїми словами, що означає термін «ймовірнісний висновок».
3. В чому полягає проблема обчислювальної складності при використанні ймовірнісного висновку в мережах Байєса? Для яких мережевих структур простіше робити обчислення, а для яких складніше?

4. Наведіть методи обчислення якості побудови ймовірнісного висновку в мережах Байєса.
5. Які існують групи методів обчислення ймовірнісного висновку?
6. Які існують методи та алгоритми формування точного ймовірнісного висновку?
7. Наведіть приклади статистичних характеристик для оцінювання якості роботи будь-якого методу побудови ймовірнісного висновку.
8. Які існують методи та алгоритми формування апроксимаційного ймовірнісного висновку?
9. У чому полягає різниця між точним та апроксимаційним висновком?
10. Поясніть суть прямого і зворотного операторів, які використовують при передачі повідомлень.
11. Що означають π – і λ – повідомлення мережі? Для чого вони потрібні?
12. У чому полягає суть алгоритму Перла? Для яких структур він використовується? Вкажіть його переваги та недоліки.
13. Що означає «довіра» стосовно мережі у цілому?
14. Поясніть термін «ініціалізація дерева».
15. У чому полягає алгоритм кластеризації дерева клік (LS-метод)? Для яких структур він застосовується? Вкажіть його переваги та недоліки.
16. Поясніть різницю між термінами: *"об'єднані дерева"* (junction trees), *дерева клік* (clique trees), *гіпердерева* (hypertrees) та *якісні дерева Маркова* (*qualitative Markov trees*).
17. Поясніть терміни *моралізування* і *триангуляція* графа.
18. У чому полягає суть алгоритму визначного перетину? Для яких структур він використовується? Проаналізуйте його переваги та недоліки.
19. У чому полягає алгоритм виключення змінних? Для яких структур він застосовується? Проаналізуйте його переваги та недоліки.
20. У чому полягає суть алгоритму символьного ймовірнісного висновку? Для яких структур він використовується? Проаналізуйте його переваги та недоліки.
21. Чим можна пояснити неоднозначність ймовірнісних висновків, отриманих за допомогою різних методів?
22. В чому полягає алгоритм ймовірнісного висновку на основі диференційного підходу? Для яких структур він використовується?
23. Поясніть алгоритм формування точного ймовірнісного висновку на основі навчальних даних. Наведіть приклад його використання.
24. Що означає *нормування* значень ймовірностей?
25. Яким ще може бути тип висновку крім *ймовірнісного*?