Івченко Л. Точні методи формування ймовірнісного висновку в мережах Байєса для задач прогнозування діяльності виробничих підприємств / Івченко Л. // Вісник ТНТУ. — 2011. — Том 16. — № 3. — С.189-195. — (приладобудування та іформаційно-вимірювальні технології).

УДК 519.226.4

Л. Івченко

Національний технічний університет України «Київський політехнічний інститут»

ТОЧНІ МЕТОДИ ФОРМУВАННЯ ЙМОВІРНІСНОГО ВИСНОВКУ В МЕРЕЖАХ БАЙЄСА ДЛЯ ЗАДАЧ ПРОГНОЗУВАННЯ ДІЯЛЬНОСТІ ВИРОБНИЧИХ ПІДПРИЄМСТВ

Резюме. *Розглянуто основні методи формування точного ймовірнісного висновку у мережах Байєса. На простих прикладах проілюстровано роботу алгоритму виключення змінної та алгоритму кластеризації даних, а також визначено випадки, коли доцільно використовувати ці алгоритми.*

Ключові слова: *мережа Байєса, ймовірнісний висновок, алгоритм виключення змінної, алгоритм кластеризації даних.*

L. Ivchenko

EXACT INFERENCE TO FORECASTING AN ENTERPRISE STATE USING BAYESIAN NETWORKS

The summary. *In this paper, we describe and analyze Bayesian network inference algorithms': clustering and value elimination. Consider a specific examples of using these algorithms in practice.*

Key words: *bayesian network, Bayesian network inference, algorithms: clustering and value elimination.*

Вступ. Мережі Байєса (МБ) – сучасний інструмент ймовірнісного моделювання та прогнозування розвитку процесів із невизначеностями різної природи і типу. Побудова моделі у вигляді МБ передбачає формування (навчання) структури мережі й параметричне навчання – формування таблиць умовних ймовірностей, оцінювання параметрів розподілів змінних і т. ін. [1, 3]. Побудовану ймовірнісну модель такого типу використовують для формування ймовірнісного висновку, тобто оцінювання ймовірностей подій (ситуацій), заданих вибраними вузлами (змінними) мережі на основі наявної апріорної та додаткової інформації стосовно інших змінних моделі.

Постановка проблеми. Існує тісний зв'язок між складністю ймовірнісного висновку в байєсівської мережі і складністю задач задоволення обмежень (Constraint Satisfaction Problem – CSP). Складність вирішення дискретної задачі CSP залежить від того, наскільки "деревоподібним" є її граф обмежень. Такі показники, як ширина гіпердерева, які встановлюють межі складності вирішення завдання CSP, можуть також застосовуватися безпосередньо до байєсівських мереж. Більше того, існує можливість узагальнити алгоритм виключення змінної таким чином, щоб він дозволяв знаходити рішення не тільки в байєсівських мережах, але й у завданнях CSP.

Аналіз досліджень і публікацій. Питання формування ймовірнісного висновку в МБ є предметом дослідження багатьох науковців. Серед них можна назвати праці Colin Dewey, Pearl Judea та ін. [4, 5]. У зазначених працях увага в основному приділена формуванню ймовірнісного висновку в мережах Байєса. Проте формування ймовірнісного висновку досі залишається надскладною задачею.

Метою даної роботи є застосування існуючих точних методів формування ймовірнісного висновку в МБ. Потрібно визначити випадки, в яких доцільно застосовувати точний висновок та проілюструвати прикладами роботу алгоритмів виключення змінної та кластеризацію даних.

Постановка завдання. Точні методи застосовують для мереж, що відносяться до класу поодинці з'єднаних мереж, також відомих як полідерева − якщо основний неспрямований граф має не більше одного шляху між двома вузлами. Коли мережа має багато з'єднань між вершинами, можна використати метод кластеризації для перетворення її у полідерево, а потім сформувати точний висновок.

Будь-яку умовну ймовірність можна розрахувати шляхом додавання елементів із повного спільного розподілу. Конкретніше слід вказати, що відповідь на запит $P(X|e)$ можна отримати з використанням такого рівняння:

$$P(X|e) = \alpha\{X, e\} = \alpha \sum_{y} P(X, e, y). \qquad (1)$$

Будь-яка байєсівська мережа створює вичерпне уявлення повного спільного розподілу. Рівняння (2) показує, що терми $P(x, e, y)$ у спільному розподілі можна записати у вигляді добутків умовних ймовірностей, узятих з мережі. Тому на будь-який запит можна знайти відповідь за допомогою байєсівської мережі, обчислюючи суми добутків умовних ймовірностей з цієї мережі:

$$P(x_1, \ldots, \cdots x_n) = \prod_{i=1}^{n} P(x_i | parents(X_i)). \qquad (2)$$

Алгоритм виключення змінної (variable elimination) є найпростішим методом. Виключення змінної відбувається шляхом обчислення виразів, подібних представленому в рівнянні (3) в порядку справа наліво. Проміжні результати зберігаються, а операції додавання по кожній змінній виконуються тільки для тих частин виразів, які залежать від цієї змінної:

$$P(b|j, m) = \alpha \sum_{e} \sum_{a} P(b) P(e) P(a|b, e) P(j|a) P(m|a). \qquad (3)$$

Подібні алгоритми виключення змінної [5, 6] та рекурсивне обумовлення [7], де виключення змінної є запитом на основі алгоритму обчислення апостеріорної ймовірності в байєсівській мережі. Його вхід – це байєсівська мережа, що містить n дискретних виключених змінних та n таблиць умовних ймовірностей (ТУЙ) зазначеного ймовірнісного розподілу кожної змінної для будь яких конкретних значень своїх родичів. Байєсівська мережа може бути повністю представлена спільним розподілом вигляду

$$\Pr(V_1 = x_1, \ldots, V_n = x_n) = \prod_{i=0}^{n} C_i(x_{i(1)}, \ldots, x_{i(k)}), \qquad (4)$$

де V_i – i-а змінна; C_i – i-та ТУЙ; $x_{i(j)}$ – значення j-ї змінної в i-й ТУЙ. Спільний розподіл змінних є добутком відповідних таблиць умовних ймовірностей.

Виключення змінної є алгоритмом відстеження, який виконує пошук у глибині дерева оцінювання змінних. Отже, виключення значень змінної – це алгоритм, який відноситься до класу алгоритмів "обумовлення". Побудова алгоритму відбувається в кілька етапів.

Перший етап не відноситься до алгоритму відстеження. Скоріше, це процедура простого генерування псевдовипадкових чисел і тестування (шляхом додавання), яка виконує пошук повного дерева із присвоюванням кожній змінній відповідного стану шляхом додавання ймовірностей, пов'язаних із кінцевими (листковими) вузлами. Нижче наведено алгоритм пошуку на псевдокоді.

```
GenAndSum ( )
1.   V = selectUnAssignedVar ( )
2.   if V = = NONE
3.     prod=1
4.     foreach CPT c
5.         prod *= eval(c)
6.     return (prod)
7.   sum = 0
8.   foreach d ∈ Dom[V]
9.     assign (V,d)
10.    sum += GenAndSum ( )
11.    unassign (V)
12.  return (sum)
```

На кожній рекурсії алгоритму вибирається ще невизначена змінна *V* і аналізується її значення (рядок 8 алгоритму). Перехід до нового вузла в дереві пошуку відповідає виконанню рядка 9, де виконується нове присвоювання. Звернемо увагу, що змінна, яка упорядковується, може бути динамічною, тобто завдяки рекурсивним повторенням алгоритму для кожного значення *V* (рядок 10) може виконуватись інстанціювання змінних, що залишилися в іншому порядку. Листковий вузол буде досягнуто тоді, коли для всіх змінних буде виконано присвоювання (рядок 2), і в кожній точці буде розраховано добуток ТУЙ. Операція *Eval* оцінює кожну ТУЙ відносно присвоєння значення поточній змінній оператором *assign* ( ). Виходячи з означення мережі Байєса, цей добуток є ймовірністю повного поточного присвоєння. Рекурсивний пошук після кожного присвоєння *V* повертає суму листкових вузлів для піддерева, що формується нижче.

Кінцевим результатом є експоненційний розрахунок одного вузла. Якщо ж маємо деякі свідчення стосовно даних *E* (тобто присвоєння значень деяким змінним) та змінну запиту *Q*, можна розрахувати присвоєння свідчення до виклику функції **GenAndSum** і вибрати *Q* як першу присвоєну змінну. Сумі, поверненій оператором **GenAndSum** після *Q*, присвоюють значення *d* з ймовірністю $(Q = d) \wedge E$, таким чином наступний розподіл *Q* можна отримати шляхом нормалізації цих ймовірностей. **GenAndSum** вказує на вивчене дерево пошуку та семантику того, що було розраховано. Решта розвитку алгоритму виключення змінної включає в себе методи оптимізації дослідження цього дерева, так що наступну *Q* можна обчислити без відвідування кожного вузла.

Перше удосконалення необхідне для того, щоб створити процедуру відстеження при пошуку. Відстеження грунтується на принципі перевірки обмежень у дереві як тільки всі змінні присвоєні й відстежені відразу, якщо обмеження порушується. У ймовірнісному контексті це виражається в оцінюванні ТУЙ як тільки вони стають присвоєними. Нижче наведено алгоритм на псевдокоді.

```
Prob-BT ( )
1.  V = selectUnAssignedVar ( )
```

```
2.  if V = = NONE
3.  return (1)
4.  sum = 0
5.  foreach d ∈ Dom[V]
6.  assign (V, d)
7.  prod = 1
8.  foreach CPT c s.t. c is newly single valued
9.  prod *= eval (c)
10. if prod != 0
11. prod *= Prob-BT()
12. sum += prod
13. unassign(V)
14. return (sum)
```

Таблиця умовних ймовірностей *c* стає однозначною функцією, коли всі її змінні істанційовані. (В даному контексті конкретна структура ТУЙ може стати однозначною, перш ніж усі її змінні будуть проілюстровані.) Оператор **Prob-BT** накопичує добуток відповідних таблиць умовних ймовірностей, які щойно стали однозначним перед пошуком піддерева нижче (рядок 9). Не важко бачити, що будь-яка з таблиць умовних ймовірностей *c*, яка стає однозначною на вузлі, буде мати вигляд як фактор у кожному листку в піддереві нижче. Таким чином, рання активації ТУЙ відповідає переміщенню загальних факторів з боку додавання. Крім того, якщо одна з ТУЙ набуває нульові значення, ми не повинні шукати піддерево нижче (рядок 10). За допомогою оператора **GenAndSum** аналізується кожен листок і оцінюється його ймовірність (яка могла бути нульовою внаслідок загального нульового коефіцієнта). Таким чином, оператор **Prob-BT** може зберегти експоненційний об'єм обчислень, виконаних оператором **GenAndSum**, але вона ще повинна відвідати кожен листковий вузол, який має ненульову ймовірність.

Встановивши важливість виключення при упорядкуванні змінних відносно виконання (навіть прийнятності) алгоритму виключення змінної при розрахунку умовної ймовірності, спробуємо вирахувати «добре» виключення змінної при упорядкуванні [8]. Обчислення кращого виключення при упорядкуванні змінних є NP-складною або надскладною задачею. Тому використовуємо алгоритм «жадібного» виключення для обчислення "напівоптимального" або "доброго" виключення.

Алгоритм «жадібного» виключення вибирає на кожній ітерації виключення вузол, що буде усунуто, з найменшою кількістю змінних. Звичайно, виключення обчислюється заздалегідь до виконання фактичного алгоритму виключення змінної. Отже, встановлюємо рівень ігрового майданчика для кожного методу, який ми використовуємо, бо всі вони попередньо обчислені виключенням змінної до фактичного запиту байєсівської мережі. Зауважимо, що алгоритм «жадібного» усунення змінних для попередніх обчислень «напівоптимального» виключення займає менше однієї секунди для всіх тестів.

Одним із обґрунтованих виборів для мінімізованої *f* є кількість дуг, необхідних для додавання до графа внаслідок його виключення. Формування ймовірнісного висновку в БМ є NP-складною задачею. Точний висновок легко формувати для багатьох реальних графічних моделей, в той час як наближений висновок дає «близьку» відповідь [9].

Алгоритм «жадібного» виключення можна описати так:

Дано: БМ представлена неспрямованим графом *G*, функція оцінки *f*.

1. Присвоюємо початкове значення усім вузлам у *G* як не маркіровані.
2. Для *k*= 1 … $|\chi|$.
3. Обираємо немаркіровану змінну $X \in \chi$, яка мінімізує $f(G, X)$.
4. $\pi(X) \leftarrow k$.
5. Додаючи дуги в *G* між усіма немаркірованими сусідами *X*.
6. *X* маркірована.

Отримано: змінна розподілу π.

Досліджуючи наведену вище послідовність етапів, можна переконатися в тому, що потрібні дві основні обчислювальні операції: отримання точкового множення пари факторів і виключення деякої змінної шляхом додавання з добутків факторів.

Зрештою, виявимо, що до запиту не належить будь-яка змінна, яка не є нащадком змінної запиту або змінною свідоцтва. Алгоритм виключення змінної дозволяє видаляти всі ці змінні, перш ніж приступати до обчислення відповіді на запит.

Описаний вище алгоритм виключення змінної є простим і ефективним засобом отримання відповідей на окремі запити [10, 11]. Якщо ж виникає необхідність обчислити апостеріорні ймовірності для всіх змінних у мережі, то цей алгоритм може виявитися менш ефективним. Наприклад, у мережі з полідеревоподібною структурою для цього буде потрібно видати *O(n)* запитів з вартістю *O(n)* кожен, що в сумі становить витрати часу $O(n^2)$. Використання алгоритмів кластеризації (відомих також під назвою алгоритмів дерева з'єднання – join tree), дозволяє скоротити цей час до *O(n)*. Із зазначеної причини дані алгоритми широко використовуються в комерційних інструментальних засобах.

Основна ідея кластеризації полягає в тому, що окремі вузли в дереві повинні бути з'єднані для формування кластерних вершин таким чином, щоб результуюча мережа стала полідеревом.

Після перетворення мережі у форму полідерева застосовується алгоритм ймовірнісного висновку спеціального призначення. По суті, цей алгоритм являє собою одну з форм алгоритму розповсюдження обмежень, де обмеження забезпечують узгодження сусідніх кластерів по апостеріорній ймовірності будь-яких змінних, які є в них загальними. За наявності ретельно продуманих засобів обліку цей алгоритм дозволяє обчислювати апостеріорні ймовірності для всіх вершин у мережі, відмінних від вершин свідоцтва, за час *O(n)*, де *n* тепер позначає розмір модифікованої мережі. Проте NP-складність розв'язуваної задачі не зникає: якщо мережа вимагає експоненціальних витрат часу і простору при виключенні змінної, то експоненціальні витрати часу і простору потрібні й для побудови таблиць CPT у кластеризованій мережі.

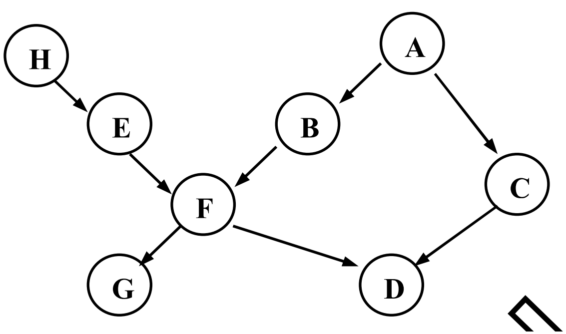

Рисунок 1. Метод кластеризації для отримання сполученого дерева

Метод кластеризації є одним із найпотужніших серед методів точного обчислення ймовірностей на МБ. Поява і швидкий розвиток також наближених методів пояснюються тим, що в загальному випадку при зростанні кількості змінних точне обчислення ймовірностей є вкрай складним з обчислювальної точки зору.

Вище було показано, що виключення змінної є ефективнішим у порівнянні з перебором, оскільки дозволяє запобігти обчисленням, що повторюються (а також забезпечити видалення змінних, що не відносяться до справи). Вимоги процесу виключення змінної до часу і простору залежать в основному від розміру найбільшого чинника, сформованого в процесі роботи даного алгоритму. А цей розмір, у свою чергу, залежить від порядку виключення змінних і від структури мережі.

Розділяють:

- однозв'язні мережі (singly connected) або полідерева (polytree), що належать до сімейства мереж, в яких є найбільше один неорієнтований шлях між будь-якими двома вершинами в мережі. Тимчасова та просторова складність точного ймовірнісного висновку в полідеревах лінійно залежить від розміру мережі;

- багатозв'язні мережі (multiply connected), в яких процедура виключення змінної в найгіршому випадку може мати експоненційну тимчасову і просторову складність, навіть якщо кількість батьківських вершин у розрахунку на кожну вершину обмежено. У цьому немає нічого дивного, якщо врахувати, що ймовірнісний висновок у байєсівських мережах є NP-важким, оскільки включає висновок у пропозиційній логіці як окремий випадок. Задача є настільки важкою, наскільки важка задача обчислення кількості виконуючих присвоювань для формули пропозиційної логіки. Це означає, що дана задача є NP-важкою, тобто строго важкою, ніж NP-повні задачі.

Висновки. Мережі Байєса – гнучкий та зручний інструмент моделювання складних процесів, які характеризуються дискретними та неперервними змінними і функціонують в умовах наявності невизначеностей різних типів. Розглянуто існуючі точні методи формування ймовірнісного висновку для МБ. Формування висновку для великих мереж Байєса є NP-складною задачею, а тому дослідники вдаються до використання точних методів формування висновку. Для простих мереж Байєса користуються такими точними методами формування висновку: (1) алгоритми виключення змінної; (2) алгоритми полідерев; (3)алгоритми кластеризації.

Література

1. Бідюк, П.І. Основні етапи побудови і приклади застосування мереж Байєса [Текст] / П.І. Бідюк, Н.В. Кузнєцова // Системні дослідження та інформаційні технології. – 2007. – № 4. – С. 26–39.
2. Murphy K. A Brief Introduction to Graphical Models and Bayesian Networks / Murphy K. // – 1998. – 19 p.
3. Би дюк, П.И. Построение и методы обучения байесовских сетей [Текст] / П.И. Бидюк, А.Н. Терентьев // Таврический вестник информатики и математики. – 2004. – № 2. – С. 139–153.
4. Colin Dewey. Inference in Bayesian Networks, – 2010. – 34 p.
5. Pearl Judea. Causality: models, reasoning, and inference, 2nd Edition, – 2009. – 22 p.
6. Dechter R., Bucket elimination: A unifying framework for reasoning. Artificial Intelligence. – 1999. – p. 41–85.
7. Zhang N.L. and Poole D. Asimple approach to bayesian network computations. In Proceedings of the Canadian Artifical Intelligence Conference. – 1994. – p. 171–178.
8. Darwiche,A. Recursiveconditioning. Artificial Intelligence. – 2001. – p. 5–41.
9. Rose F.Liu, Rusmin Soetjipto, Analysis of Three Bayesian Network Inference Algorithms: Variable Elimination, Likelihood Weighting, and Gibbs Sampling, November 4. – 2004. – p. 2–4.
10. A.K. Jain, M. N. Murty, P. J. Flynn. – Data Clustering: A Review. ACM Comput. Surv. – 1999. – p. 264–323.
11. Pearl J. Evidential reasoning using stochastic simulation / Pearl J. // Artificial Intelligence. – 1987. Artificial Intelligence. – p. 245–257.

Отримано 12.03.2011