

Салій А. С.

МЕТОДИ НАВЧАННЯ СТРУКТУРИ БАЄСІВСЬКОЇ МЕРЕЖІ

Розроблено генетичний алгоритм для навчання структури баєсівської мережі з повністю спостережуваного набору даних, який здійснює пошук по простору графа і використовує оператори мутації та кросовера. Цей алгоритм здійснює пошук по простору спрямованих ациклічних графів.

Ключові слова: баєсівська мережа, генетичний алгоритм, оператори мутації та кросовера, ациклічні графи.

Вступ

Іноді на практиці потрібно обчислити ймовірність невизначеної причини, враховуючи деякі спостережувані свідчення. Наприклад, ми хотіли б знати ймовірність конкретного захворювання, коли спостерігаємо симптоми у пацієнта. Такі проблеми часто є складними з багатьма взаємопов'язаними змінними. Можливо, існує багато симптомів і навіть більше потенційних причин. На практиці зазвичай можна отримати лише зворотну умовну ймовірність, тобто ймовірність свідчень, що дають причину, ймовірність спостереження за симптомами, якщо у пацієнта є захворювання.

Інтелектуальні системи повинні міркувати про своє оточення. Наприклад, роботів необхідно знати про можливі результати своїх дій, а система медичних експертів повинна знати, які причини призводять до яких наслідків. Інтелектуальні системи почали використовувати імовірнісні методи, щоб впоратися з невизначеністю реального світу. Замість того, щоб будувати спеціальну систему імовірнісних міркувань для кожної нової програми, хотілось б загальну основу, яка б дозволила імовірнісно міркувати в будь-якій новій програмі, не відновлюючи все з нуля. Саме цим і обґрунтована актуальність розробленого генетичного алгоритму. Баєсівські мережі, що вперше з'явилися в роботі Йуди Перла та його колег наприкінці 1980-х, пропонують саме таку незалежну основу для вірогідних міркувань.

Представлення

В основі розробленого алгоритму лежить представлення Карвальо. Індивід представлений за допомогою кортежу $\{R, A\}$, де R – перестановка, що вказує на порядок вузлів, а A – верхня трикутна матриця суміжності, рядки та стовпці якої представляють вузли в лексикографічному

порядку. Рисунок 8 ілюструє це подання. Кортеж $\{R, A\}$ кодує унікальний спрямований ациклічний граф таким чином: матриця суміжності A описує неорієнтований граф над вузлами. Тоді кожне ребро $S - T$ спрямовується у порядку R : $S \rightarrow T$, якщо $S <^R T$ (S передує T в R), або $S \leftarrow T$, якщо $T <^R S$. Зауважте, що представлення конкретного спрямованого ациклічного графа не є унікальним, оскільки може бути декілька порядків, що відповідають спрямованому ациклічному графу. Це називається надмірністю кодування. Наприклад, обидва альтернативні порядки – V, W, X, Y, Z і V, X, W, Z, Y – відповідають мережі на рисунку 1.

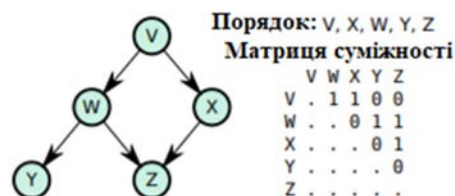


Рис. 1. Невелика баєсівська мережа (зліва) та її представлення в генетичному алгоритмі (справа)

Існує тонка, але важлива різниця між розробленим представленням та представленням Карвальо. У цьому поданні рядки та стовпці матриці суміжності A завжди однакові – вони є вузлами в лексикографічному порядку. Отже, ця матриця суміжності кодує неорієнтований граф, ребрам якого надається напрямок, використовуючи порядок R . На відміну від цього, Карвальо дозволяє порядку R визначати значення рядків і стовпців матриці суміжності A . Його матриця суміжності являє собою спрямований граф. Представлення Карвальо було б непридатним у традиційному генетичному алгоритмі, оскільки воно має погану локальність: обмін місцями двох змінних у порядку R призводить до великих змін у мережі. Наприклад, припустимо, що порядок V, X, W, Y, Z

з рисунку 1 змінився на V, Z, W, Y, X . За даним представленням це призведе до перегортання напрямку ребер $W \rightarrow Z$ і $X \rightarrow Z$, як показано на рисунку 2(а). З представленням Карвальо ті ж мутації призведуть до перегортання ребер $X \rightarrow Z$, але також видалення трьох ребер та додавання трьох нових ребер, як показано на рисунку 2(б). Наше представлення уникає цієї слабкої локальності, що перетворила б генетичний алгоритм у випадковий пошук.

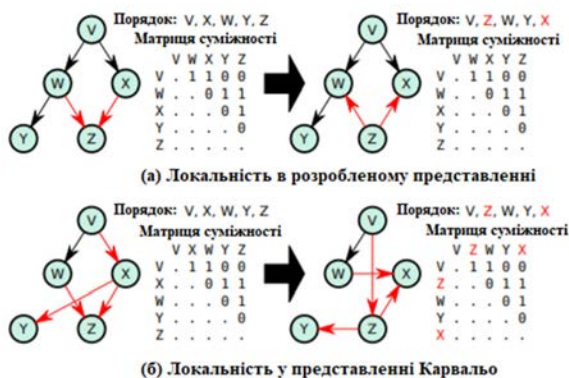


Рис. 2. Локальність представлення генетичних алгоритмів

Оператори

У розробленому алгоритмі ініціалізуємо індивід, надаючи йому довільну перестановку R для порядку та верхньотрикутну матрицю суміжності A , що має по одному значенню «1» у кожному рядку. Використовуємо процедуру турнірного відбору з розміром турніру 2. Це означає, що два індивіди вибираються випадковим чином, а той, який має більш високу пристосованість, вибирається для репродукції. Ця процедура повторюється, поки ми не відберемо достатньо екземплярів для репродукції.

Для кожної пари батьків $\{R_1, A_1\}$ та $\{R_2, A_2\}$, вибраних для репродукції, випадково застосовуємо один із таких операторів:

- Циклічний кросовер на порядках R_1 і R_2 . Раніше було показано, що цей оператор має хороші показники для структурного навчання орієнтованих ациклічних графів, які здійснюють пошук за порядками вузлів. Циклічний кросовер працює, поділяючи дві перестановки на цикли. Цикл визначається як мінімальний набір значень, які разом займають однакові позиції у двох перестановках. Рисунок 3 ілюструє циклічний кросовер.
- Двоточковий кросовер на матрицях суміжності A_1 і A_2 . Цей оператор вирівнює дві матриці суміжності в бітові рядки. Потім він обрізає обидва рядки у двох випадкових місцях та

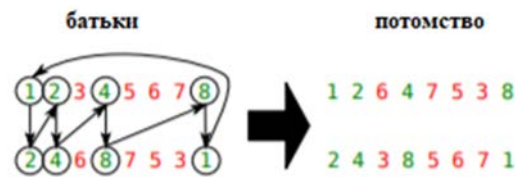


Рис. 3. Циклічний кросовер на порядках. 1, 2, 4, 8 утворюють цикл, оскільки разом вони займають однакові позиції у двох батьків. Колами та стрілками показують, як знайдено цей цикл. 3, 5, 6, 7 утворюють ще один цикл

обмінюється їхніми середніми фрагментами. Рисунок 4 ілюструє двоточковий кросовер.

- Мутація зміни місцями за порядками R_1 і R_2 . Для кожного з батьків цей оператор вибирає два випадкові вузли та обмінює їх у порядку.
- Біт-фліп мутація на матрицях суміжності A_1 і A_2 . Для кожного з батьків цей оператор перевертає кожен біт у матриці суміжності, який має невелику ймовірність.

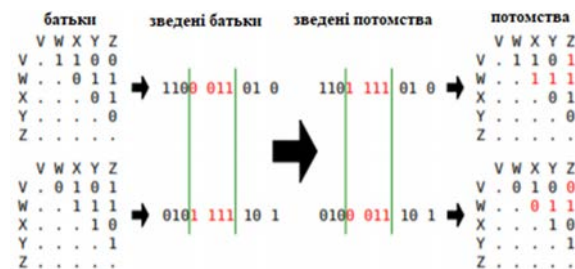


Рис. 4. Двоточковий кросовер на матрицях суміжності

Оператори кросовера поєднують інформацію обох батьків, тоді як оператори мутації цього не роблять. Два оператори на порядок перевертають ребра в мережі: кросовер перевертає багато, а мутація перевертає декілька. Два оператори на матрицях суміжності додають, видаляють або обмінюються ребрами між мережами.

Біт-фліп мутація Карвальо перевертає біти випадковим чином, незалежно від їхніх початкових значень. Оскільки більшість орієнтованих ациклічних графів, розглянутих у процесі пошуку, є рідкісними, то матриця суміжності містить більше нулів, ніж одиниць, тому цей оператор мутації швидше переверне 0 у 1, ніж зробить протилежне. Це означає, що оператор частіше додає ребра, ніж видаляє ребра. В розробленому алгоритмі зробили мутацію біт-фліпу симетричною, враховуючи початкове значення біта. Це означає, що оператор мутації з однаковою ймовірністю як додасть ребро, так і видалить ребро.

Кешування

Використовуємо два рівні кешування для прискорення обчислення басівської оцінки ме-

режі. Перший рівень кешує достатню статистику (рачує) набір даних. Кеші другого рівня встановлюють набір оцінок, які будуть визначені незабаром.

Перший рівень кешування використовує структуру даних AD-дерева Мура та Лі для прискорення обчислення таблиць спряженості. AD-дерево економить час за рахунок використання більшої кількості пам'яті. Це структура даних дерева, де кожен вузол розбиває набір даних відповідно до всіх можливих присвоєнь змінній.

Таблиці спряженості можна обчислити з дерева AD в часі, що не залежить від загального розміру набору даних. Таким чином, нам більше не потрібен один повний прохід через набір даних для обчислення таблиці спряженості.

Традиційні методи пошуку зі сходженням до вершини кешують оцінки кожної сім'ї, щоб їх можна було отримувати знову і знову, не починаючи перерахунки. Вдосконалюємо це потрохи, помічаючи, що з апіорною еквівалентною формою Баеса Діріхле сімейна оцінка розкладається на два множинні показники:

$$\begin{aligned} FamScore(X_i, Pa_{X_i}) &= \\ &= SetScore(Pa_{X_i} \cup \{X_i\}) - SetScore(Pa_{X_i}), \end{aligned}$$

де

$$SetScore(X) = \sum_{v_i \in Val(X)} \ln \ln \frac{\Gamma(\alpha_X)}{\Gamma(\alpha_X + M[v_i])}$$

для будь-якого набору змінних X , а $\alpha_X = \alpha \cdot \frac{1}{|Val(X)|}$. Для обчислення сімейної оцінки, схоже, потрібні таблиці спряженості на випадок $Pa_{X_i} \cup \{X_i\}$ та Pa_{X_i} . Але насправді останню таблицю спряженості можна отримати з попередньої, маргіналізуючи стовпчик X_i . Це означає, що для обчислення оцінки сім'ї потрібен лише один запит AD-дерева.

Другий рівень зберігає набір оцінок. Це призводить до більшої кількості кешу, оскільки кілька сімей протягом життя алгоритму можуть мати один і той самий набір батьків. Наприклад, припустимо, що хочемо обчислити $FamScore(X, \{A, B, C\})$ і пізніше $FamScore(Y, \{A, B, C\})$. У другому обчисленні $SetScore(\{A, B, C\})$ буде відновлено з кеша.

Висновки

Розробка генетичного алгоритма для навчання структури баєсівської мережі здійснює пошук по простору графа, використовує оператори мутації та кросовера. Алгоритм можна використовувати як швидкий спосіб навчити структуру баєсівської мережі з якомога меншими обмеженнями.

Список літератури

1. Peña J. M. An improved bayesian structural em algorithm for learning bayesian networks for clustering / J. M. Peña, J. A. Lozano, P. Larranaga // Pattern Recognition Letters. – 2000. – Vol. 21(9). – Pp. 779–786.
2. Singh Moninder. Construction of bayesian network structures from data: a brief survey and an efficient algorithm / Moninder Singh, Marco Valtorta. – Washington D. C., Morgan Kaufmann, 1995. – Pp. 259–265.
3. Teyssier M. Ordering-based search: A simple and effective algorithm for learning bayesian networks / M. Teyssier, D. Koller // Proceedings of the Twenty-first Conference on Uncertainty in AI (UAI). – Edinburgh, Scotland, UK, July 2005. – Pp. 584–590.

References

- Peña, J. M., Lozano, J. A., & Larranaga, P. (2000). An improved bayesian structural em algorithm for learning bayesian networks for clustering. *Pattern Recognition Letters*, 21 (9), 779–786.
- Singh, M., & Valtorta, M. (1995). *Construction of bayesian network structures from data: a brief survey and an efficient algorithm*. Washington D. C., Morgan Kaufmann.
- Teyssier, M., & Koller, D. (2005). Ordering-based search: A simple and effective algorithm for learning bayesian networks. In *Proceedings of the Twenty-first Conference on Uncertainty in AI (UAI)*. Edinburgh, Scotland, UK.

A. Saliu

METHODS OF LEARNING THE STRUCTURE OF THE BAYESIAN NETWORK

Sometimes in practice it is necessary to calculate the probability of an uncertain cause, taking into account some observed evidence. For example, we would like to know the probability of a particular disease when we observe the patient's symptoms. Such problems are often complex with many interrelated variables. There may be many symptoms and even more potential causes. In practice, it is usually possible to obtain

only the inverse conditional probability, the probability of evidence giving the cause, the probability of observing the symptoms if the patient has the disease.

Intelligent systems must think about their environment. For example, a robot needs to know about the possible outcomes of its actions, and the system of medical experts needs to know what causes what consequences. Intelligent systems began to use probabilistic methods to deal with the uncertainty of the real world. Instead of building a special system of probabilistic reasoning for each new program, we would like a common framework that would allow probabilistic reasoning in any new program without restoring everything from scratch. This justifies the relevance of the developed genetic algorithm. Bayesian networks, which first appeared in the work of Judas Pearl and his colleagues in the late 1980s, offer just such an independent basis for plausible reasoning.

This article presents the genetic algorithm for learning the structure of the Bayesian network that searches the space of the graph, uses mutation and crossover operators. The algorithm can be used as a quick way to learn the structure of a Bayesian network with as few constraints as possible.

learn the structure of a Bayesian network with as few constraints as possible.

Keywords: Bayesian network, genetic algorithm, mutation and crossover operators, acyclic graphs.

Матеріал надійшов 14.06.2021



Creative Commons Attribution 4.0 International License (CC BY 4.0)