

РОЗРОБКА КЛІНІЧНОЇ ЕКСПЕРТНОЇ СИСТЕМИ, ЩО ҐРУНТУЄТЬСЯ НА ПРАВИЛАХ, МЕТОДОМ ПОСЛІДОВНОГО ПОКРИТТЯ

У статті розглянуто питання програмної реалізації методу індукції правил на основі алгоритму послідовного покриття. Такий підхід дозволяє розробити систему підтримки клінічних рішень. Проект реалізовано в середовищі Netbeans на основі Java-класів.

Ключові слова: прийняття рішень, індукція правил, Java, SQL.

В статье рассмотрены вопросы программной реализации метода индукции правил на основе алгоритма последовательного покрытия. Такой подход позволяет разработать систему поддержки клинических решений. Проект реализован в среде Netbeans на основе Java-классов.

Ключевые слова: принятие решений, индукция правил, Java, SQL.

The article deals with program implementation of induction of rules based on the algorithm of sequential covering. This approach allows to develop a system to support clinical decisions. The project is implemented in Netbeans environment based on Java-classes.

Key words: decision making, rule induction, Java, SQL.

Вступ. У медицині поняття «диференціальної діагностики» означає системний підхід, що ґрунтується на доказовості, для визначення причини симптомів, що спостерігаються, у випадку, коли є кілька альтернативних пояснень, а також для зменшення переліку можливих діагнозів.

Сьогодні медичне діагностування може виконуватися автоматично з використанням комп'ютеризованих систем та алгоритмів. Такі системи переважно називаються діагностичними системами підтримки прийняття рішень або медичними діагностичними системами. Вони належать до загальнішого класу клінічних систем підтримки прийняття рішень [9-11]. Метою таких систем є системний супровід лікаря в процесі диференціальної діагностики. Багато з таких систем можуть надавати результати, навіть коли не вистачає даних, тобто в умовах невизначеності, і що найважливіше – вони не обмежені щодо кількості інформації, яку можуть зберігати та обробляти [3-8].

У статті ми представимо класифікатор, що ґрунтується на правилах і в якому модель знань представляється множиною правил IF-THEN. Спершу ми покажемо, як такі правила можуть використовуватися для класифікації. Далі представимо метод генерації таких правил на основі алгоритму послідовного покриття.

Зазначимо, що правила можуть генеруватися як з дерева рішень, так і безпосередньо з навчальних даних, використовуючи алгоритм послідовного покриття [1; 2].

Математичне означення класифікаційних правил

Традиційне означення IF-THEN-правила наведено в працях [1; 5]. Математично задача індукції класифікаційних правил формулюється таким чином. Маємо множину D , що містить N наборів навчальних даних. При цьому кожен i -й набір $(A_1^i, A_2^i, \dots, A_p^i, C^i)$ складається з вхідних даних – атрибутів A_1, \dots, A_p та вихідних даних – атрибуту класу C . Можна припустити, що атрибути A_1, \dots, A_p приймають лише категоріальні значення. Атрибут класу C приймає одне з K дискретних значень: $C \in \{1, \dots, K\}$. Метою є прогнозування класифікаційним правилом значення атрибуту класу C на основі значень атрибутів A_1, \dots, A_p .

Класифікаційним правилом R називається імплікація вигляду: $R: \bigwedge_{j \in S} (A_j \text{ is } a_j^*) \Rightarrow C = c^*$.

Тут $S \subseteq \{1, \dots, p\}$ – деяка підмножина індексів атрибутів.

При цьому слід максимізувати точність прогнозування атрибуту класу, а саме $P\{C = c\}$ для довільного $c \in \{1, \dots, K\}$. У результаті ми повинні отримати множину правил для кожного $c \in \{1, \dots, K\}$ відповідно, що в антеседенті містять умови включення для категоріальних атрибутів, а в консеквенті значення $c \in \{1, \dots, K\}$.

Метою роботи є розробити метод індукції класифікаційних правил з можливістю програмної реалізації у вигляді клінічної експертної системи.

Алгоритм послідовного покриття

Використаємо алгоритм послідовного покриття, описаний у роботі [1]. Зауважимо ще раз, що припускаємо, що всі атрибути – категоріальні.

Алгоритм послідовного покриття

Вхідні дані:

D – множина навчальних наборів даних $(A_1^i, A_2^i, \dots, A_p^i, C^i)$

Att_vals – множина всіх атрибутів A_1, \dots, A_p та їх можливих значень $A_i \in \{a_i^1, a_i^2, \dots, a_i^{K_i}\}$.

Вихідні дані: $Rule_set$ – множина класифікаційних правил.

Метод:

1. Множина класифікаційних правил $Rule_set = \{\}$
2. Для кожного класу c
3. Розпочати цикл «до»
4. Побудувати нове класифікаційне правило $Rule = \text{Добути_одне_правило}(D, Att_vals, c)$
5. Вилучити набори навчальних даних з D , що покриваються правилом $Rule$
6. Виконувати цикл з кроку 3 до настання *термінальної умови*
7. Додати нове правило до множини класифікаційних правил:

$Rule_set = Rule_set + Rule$

8. Кінець циклу з кроку 2
9. Множина навчальних правил в $Rule_set$

В основу методу *Добути_одне_правило* (D, Att_vals, c) покладена міра приросту інформації для побудови правил логіки першого порядку FOIL (First Order Inductive Learner). Метод є ітераційною процедурою по усіх атрибутах A_1, \dots, A_p .

Припустимо, що ми вже маємо класифікаційне правило:

$R: \text{IF } condition \text{ THEN } class = c.$

Метою кожного кроку $i = 1, p$ є кон'юнкція умови $condition$ за рахунок умови $condition'$ вигляду $(A_j = a_j^i)$. Тут $j \in \{1, \dots, K_i\}$. Тобто нове правило матиме вигляд:

$R': \text{IF } condition \text{ AND } condition' \text{ THEN } class = c.$

Згідно з методом FOIL $condition'$ вибирається з умови мінімізації міри:

$$FOIL_Gain = pos' \times (\log_2 \frac{pos'}{pos'+neg'} - \log_2 \frac{pos}{pos+neg}) \cdot (1)$$

Тут $pos(neg)$ – число позитивних(негативних) навчальних наборів, що покриваються правилом R , $pos'(neg')$ – число позитивних(негативних) навчальних наборів, що покриваються правилом R' . Під позитивними (негативними) навчальними наборами для певного правила маємо на увазі навчальні набори з умовою консеквенту, які задовольняють (не задовольняють) умови антеседенту правила.

Міра (1) сприяє побудові правил, що мають більшу точність і при цьому покривають якомога більше позитивних навчальних наборів.

Програмна реалізація алгоритму

База даних mysql складається з двох таблиць – таблиці `attribute`, призначеної для зберігання інформації про атрибути та таблиці `categorized_data` – для наборів навчальних даних. Структура таблиць на мові SQL для прикладу наведена нижче:

```
CREATE TABLE mysql.attribute (
  id integer not null unique,
  attribute_name varchar(25),
  attribute_field_name varchar(25),
  primary key (id)
) ENGINE=InnoDB;
CREATE TABLE mysql.categorised_data (
  id integer not null unique,
  A1 varchar(12),
  A2 varchar(8),
  .....
  A21 varchar(7),
  class varchar(68),
  primary key (id)
) ENGINE=InnoDB;
```

Програмні класи проекту включено до пакету `rule.model`. Сюди входять `beans`-класи `Attribute`, `Attribute_for_list` для роботи з даними відповідних таблиць та `Rule` – для представлення правил. SQL-запити щодо отримання відповідних даних реалізовано в класах `AttributeListPeer` та `TuplesPeer`.

У класі `Rule_set` зберігається набір навчальних правил. До того ж цей клас безпосередньо реалізує алгоритм послідовного покриття. Клас містить члени: менеджер даних `m_dataManager`, хеш-таблиці наборів навчальних даних `m_htTuples`, усіх атрибутів з їх можливими значеннями `m_htAtt_vals` та безпосередньо множину правил `m_htRule_set`.

У конструкторі класу `Rule_set` здійснюється побудова хеш-таблиць `m_htTuples` та `m_htAtt_vals`, а також застосування алгоритму послідовного покриття – через виклик методу `Sequential_covering(m_htTuples, m_htAtt_vals)`. Отримана множина правил виводиться в текстовий файл.

Клас `Rule` призначений для зберігання окремих правил. Його членами класу є дві хеш-таблиці: `m_htAntecedent` – для зберігання антеседенту правила та `m_htConsequent` – для консеквенту. За допомогою методу `public void conjunct`

Condition(Attribute_for_list attribute, String sAttribute_value) здійснюється кон'юнкція нової умови до правила. За допомогою методу public Rule copy() створюється «глибока» копія правила. При цьому використовується протокол JOS (Java Object Serialization).

Підрахунок кількості позитивних та негативних навчальних наборів здійснюється у методах класу TuplesPeer.

Повні UML-моделі класів пакету rule.model наведені у додатку.

Приклад. Для прикладу використано експериментальну базу даних біохімічних аналізів залежно від виду політравми. Навчальні набори містять 21 категоріальний атрибут та 6 різних значень атрибуту класу. Нижче наведено побудовані класифікаційні правила:

```
IF I1-10 = normal AND TNF-a = high
THEN class=craniocerebral_injury+
orthopedic_trauma_12_hours
```

```
IF Ig M = normal AND TNF-a = high
AND I1-2 = high THEN class=
craniocerebral_injury+orthopedic_tra
uma+bleeding_2_hours
```

```
IF I1-10 = normal AND TNF-a = high
THEN
class=craniocerebral_injury+orthoped
ic_trauma+bleeding_12_hours
```

```
IF Ig G = low AND TNF-a = high AND
I1-2 = normal THEN class=
```

```
craniocerebral_injury+orthopedic_
trauma_2_hours
```

```
IF I1-6 = high AND TNF-a = high
THEN class=craniocerebral_injury+
orthopedic_trauma_24_hours
```

```
IF I1-6 = high AND TNF-a = normal
THEN class=craniocerebral_injury+
orthopedic_trauma+bleeding_24_hours
```

Час побудови множини класифікаційних правил – 10207 мілісекунд. Значимо, що у випадку класів craniocerebral_injury+orthopedic_trauma_12_hours та craniocerebral_injury+orthopedic_trauma+bleeding_12_hours антеседенти правил співпадають. Це підтверджується з думкою експертів про складність діагностування цього виду травм через 12 годин. Для уточнення правил потрібні додаткові навчальні набори.

Висновки. У роботі розглянуто питання програмної реалізації методу послідовного покриття з метою побудови класифікаційних правил.

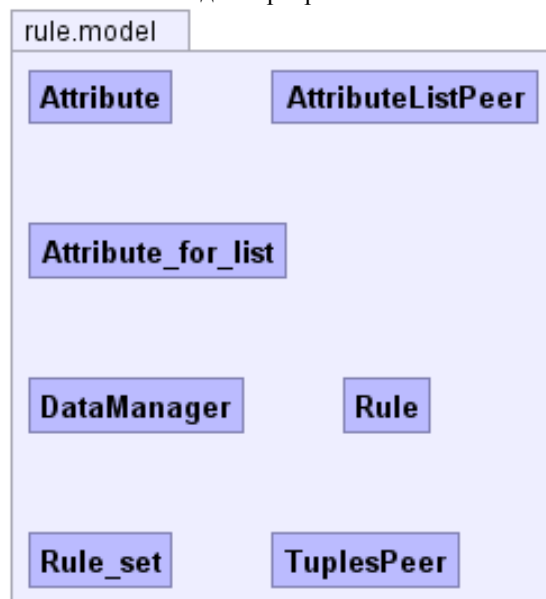
На прикладі продемонстровано, що такий підхід дозволяє розробити систему підтримки клінічних рішень.

За рахунок використання Java-класів дана реалізація методу послідовного покриття є веб-інтегрованою.

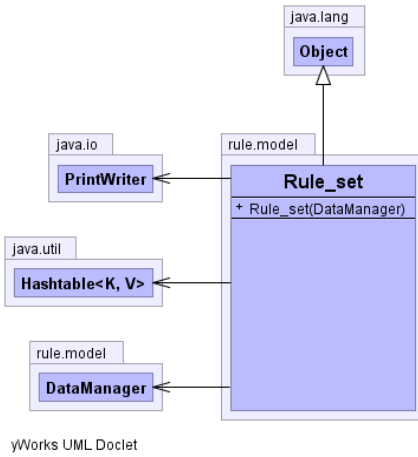
Перспективами досліджень є аналіз продуктивності програмного продукту залежно від кількості атрибутів та обсягу наборів навчальних даних.

Додаток

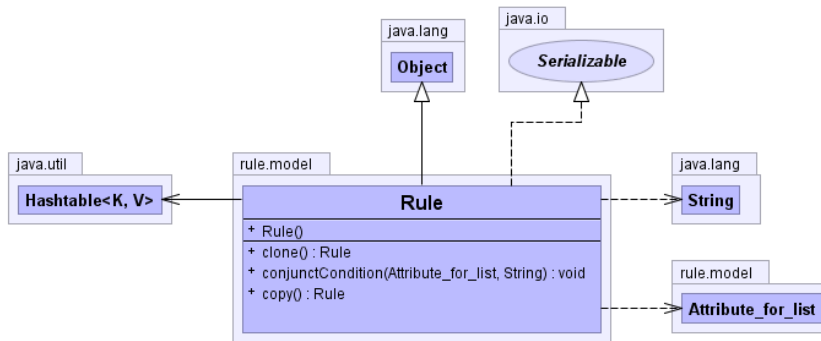
UML-моделі програмних класів



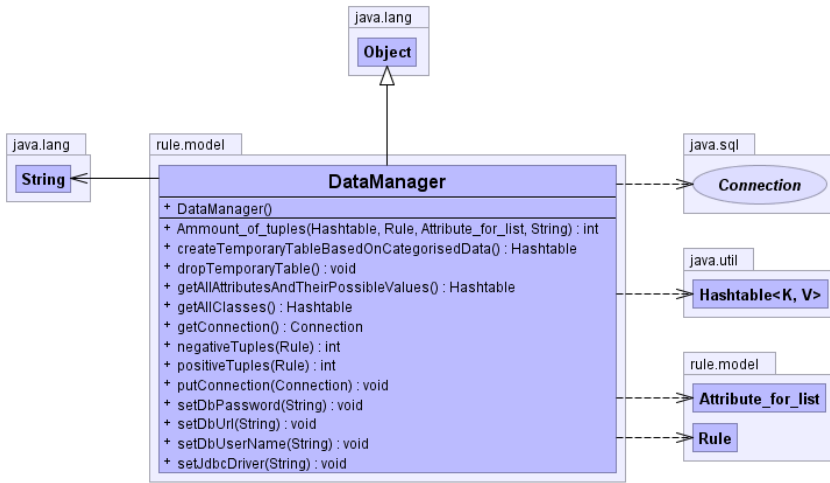
yWorks UML Doclet



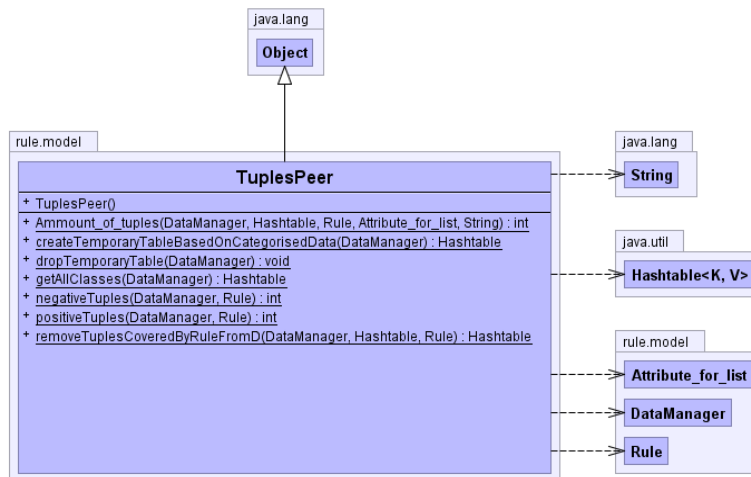
yWorks UML Doctlet



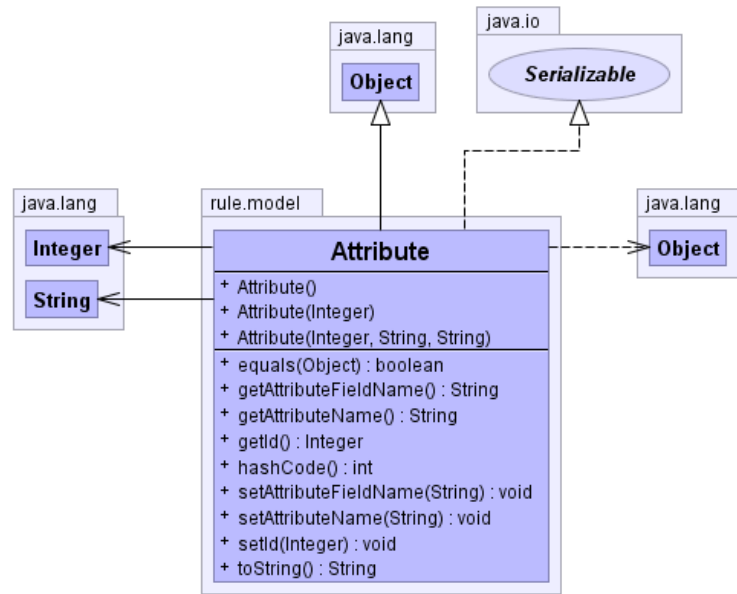
yWorks UML Doctlet



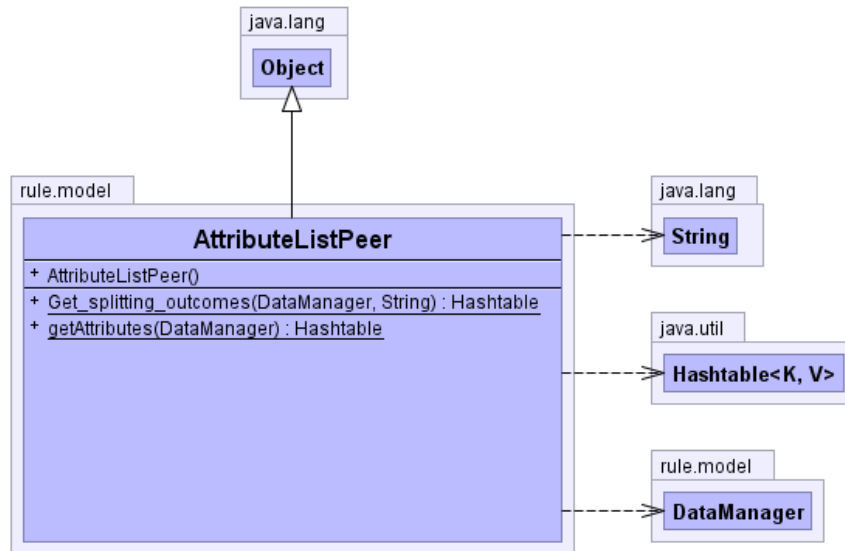
yWorks UML Doctlet



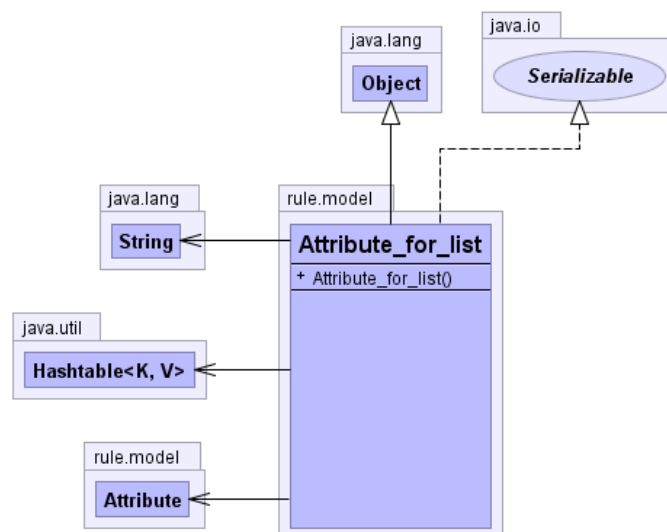
yWorks UML Doctlet



yWorks UML Doclet



yWorks UML Doclet



yWorks UML Doclet

ЛІТЕРАТУРА

1. J. Han and M. Kamber, Data Mining: Concepts and Techniques, Morgan Kaufmann, San Francisco, 1st edition, 2001.
2. T. Hastie, R. Tibshirani and J. H. Friedman, The Elements of Statistical Learning, Springer, New York, 1st edition, 2001.
3. C. Ordonez, Comparing association rules and decision trees for disease prediction, In Proc. ACM HIKM Workshop, 2006, P. 17–24.
4. C. Ordonez, Integrating K-means clustering with a relational DBMS using SQL, IEEE Transactions on Knowledge and Data Engineering (TKDE) 18(2) (2006), 188-201.
5. C. Ordonez, Models for association rules based on clustering and correlation, Intelligent Data Analysis 13(2) (2009), 337–358.
6. J. R. Quinlan. Induction of decision trees. Machine Learning, 1: 81-106, 1986.
7. J. R. Quinlan. C4.5: Programs for Machine Learning. Morgan Kaufmann, 1993.
8. L. Breiman, J. Friedman, R. Olshen, and C. Stone. Classification and Regression Trees. Wadsworth International Group, 1984.
9. Марценюк В. П., Кравец Н. О. О программной среде проектирования интеллектуальных баз данных / В. П. Марценюк, Н. О. Кравец // Клиническая информатика и телемедицина – 2004. – № 1. – С. 47–53.
10. Марценюк В. П. Математичні моделі в системі підтримки прийняття рішень страхового забезпечення лікування онкологічних захворювань: підхід на основі динаміки Гомперца / В. П. Марценюк, І. С. Андрущак, І. С. Гвоздецька, Н. Я. Климук // Доповіді Національної академії наук України. – 2012. – №1 0. – С. 34–39.
11. Марценюк В. П. О модели онкологического заболевания со временем пребывания на стадии в соответствии с распределением Гомперца / В. П. Марценюк, Н. Я. Климук // Проблемы управления и информатики. Международный научно-технический журнал. – 2012. – № 6. – С. 137–143.

© Марценюк В. П., Андрущак І. С., 2014

Дата надходження статті до редколегії 10.06.2014 р.

МАРЦЕНЮК Василь Петрович – доктор технічних наук, професор, проректор з науково-педагогічної роботи, інноваційних та комп'ютерних технологій, завідувач кафедри медичної інформатики Тернопільського державного медичного університету імені І. Я. Горбачевського.

Коло наукових інтересів: програмні системи в медичних дослідженнях, медична інформатика.

АНДРУЩАК Ігор Євгенович – кандидат технічних наук, доцент, Луцький національний технічний університет, доцент кафедри комп'ютерних технологій професійного навчання.

Коло наукових інтересів: програмні системи в медичних дослідженнях, медична інформатика.