3.3. Аналіз продукції

Продукції доцільно поділити на такі типи:

- сировина;
- товар;
- послуги;
- інші (основні засоби, МШП, податки).

Аналіз може здійснюватись безпосередньо по продукції або по парах:

- (продукція, підрозділ) – у випадку складної організаційної структури підприємства (наявності філій, фірмових магазинів, складної географічної структури підприємства);
- (продукція, партнер) – у випадку складної структури ринку, на якому діє підприємство (наявності різнотипних партнерів та їх груп, кількох великих спеціальних партнерів).

Базові напрямки аналізу по продукції:

- обсяги і динаміка обігу;
- обсяги і динаміка прибутку від обігу;
- цінові та інші характеристики продукції (дизайн, функціональність, упаковка).

1. Андрейчиков А.В., Андрейчикова О.Н. Анализ, синтез, планирование решений в экономике. //М.: Финансы и статистика, 2000. – 368с. 2. Fayyad U.M., Piatetsky-Shapiro G., Smyth P., Uthurusamy R. (editors) Advances in knowledge discovery and data mining. //AAAI/MIT Press, 1996.

УДК. 681.3

В.В. Литвин
НУ "Львівська політехніка",
кафедра “Інформаційні системи та мережі”

ОСНОВНІ МЕТОДИ ФІЛЬТРАЦІЇ ІНФОРМАЦІЇ ТА ЇХ ВИКОРИСТАННЯ ПРИ ПОБУДОВІ ІНТЕЛЕКТУАЛЬНИХ ІНФОРМАЦІЙНИХ СИСТЕМ

This paper describes information filtering process in the intelligent systems. It considers filtered knowledge sets construction.

У даній статті розглядаються методи фільтрації інформації під час процесу розв’язування задачі інтелектуальною інформаційною системою. Отримана інформація використовується для побудови стратегій керування пошуком розв’язку.

Однією з основних компонент сучасних інтелектуальних інформаційних систем (ІІС) є компонента поповнення знань. Вона призначена для поповнення новими знаннями ІІС у процесі функціонування системи. Такий ріст знань, з одного боку, покращує роботу системи, видає точніші та надійніші розв'язки. З іншого боку швидкодія роботи системи значно сповільнюється. З кожним новим кроком інтерпретатора зростає кількість можливих переходів між станами системи, що у свою чергу веде до комбінаторного вибуху. Використання евристик та побудова нових метазнань, виходячи із досвіду, набутого

системою, лише акцентують увагу на більш корисних зазначених парах, але не зменшують їх кількості. Тому необхідно фільтрувати інформацію на рівні користувача.

Користувач повинен визначити, яка інформація є для нього важливою в поточний момент роботи ІІС, задати власні критерії відбору інформації, що враховують його інтереси. З іншого боку, в системі повинна бути передбачена взаємодія із користувачем. Результатом такої взаємодії є скерування роботи системи в потрібне русло з точки зору користувача, врахування інтересів клієнта щодо отриманого кінцевого результату. З погляду ІІС така взаємодія відкидає зайву інформацію, акцентуючи свою увагу на значно меншій множині знань, що веде до різкого скорочення множини зазначених пар. Такий підхід до відбору інформації задовольняє одне з основних правил реінжинірингу: індивідуальна робота з кожним клієнтом.

Акцентування уваги користувача на деякій підмножині знань A будемо називати фільтрацією інформації. Розрізняють три типи фільтрації інформації.

Відсікання. Множина знань A розбивається на 2 класи A' - прохідні і A^* - відсічені. Причому $A' \cap A^* = \varnothing$, $A' \cup A^* = A$. Відсікання здійснюється за допомогою деякої порогової величини достовірності (шкала важливості).

Агрегація. A розбивається на класи агрегації $A_1, ..., A_m$ $\left(\cup A_i = A\right)$ і вибирається спосіб агрегації, тобто перехід до макроелементів a_i, кожний з яких представляє відповідний клас A_i так, щоб по можливості зберігати характеристики, що цікавлять особу.

Типологічна вибірка. Розбивається A на класи $A'_1, ..., A'_s$ і здійснюється відбір елементів, які повинні бути представниками кожного класу.

Фільтрація інформації здійснюється на початку роботи системи або під час її функціонування.

Фільтрація інформації на початку роботи системи задається двома способами.

1. У вигляді задання обмежень на очікуваний результат системи.
2. Відбором чи вибіркою значущої інформації для користувача.

Нехай перед початком роботи системи користувач c задав множину обмежень $L^c = \{l_1, l_2, ..., l_s\}$. Розглянемо функціонал $J_2\left(L^c\right)$, який визначає, наскільки враховано обмеження користувача c. Тоді мета ІІС – якомога найкраще задовольнити ці обмеження, тобто

$$J_2\left(L^c\right) \to max.$$

Серед заданої множини знань $A = R \cup B \cup H$, де R – підмножина бази даних, B – підмножина бази знань, H – підмножина метазнань, система пропонує користувачу c зробити акцент на деякій підмножині знань $A^c(0)$. Такий акцент здійснюється переліком даних або підобластей, в яких визначені знання, що цікавлять користувача. Якщо цей перелік доволі великий, тоді задаємо ту інформацію, яка користувача не цікавить.

На початку роботи можна відмовитись від вибірки інформації, а проводити її під час функціонування системи.

Фільтрація інформації під час функціонування системи.

Визначимо функцію інтересу користувача c в момент часу t $\alpha^c(t)$ як відображення множини знань $A(t)$ в деяку підмножину знань $A^c(t)$, що цікавить

користувача c. Таку підмножину назвемо (c,t)-множиною. Формально це запишемо у вигляді:

$$\alpha^c(t): A(t) \to A^c(t)$$

Методи побудови (c,t)-множини:

1. Фільтрація інформації користувачем за допомогою повідомлень, які видає система після кожного циклу інтерпретатора;
2. Фільтрація інформації користувачем за допомогою повідомлень, які видає система після розв'язання деякої підзадачі;
3. У будь-який момент часу за допомогою діалогу між користувачем та ІІС за бажанням користувача.

$A^c(t)$ містить в собі підмножину даних $R^c(t)$, підмножину модулів $B^c(t)$ та підмножину евристик $H^c(t)$. Тобто

$$A^c(t) = R^c(t) \cup B^c(t) \cup H^c(t)$$

Побудова сумарних множин.

Побудуємо множину A^c, що містить всю інформацію, яку вибирав користувач c протягом всього часу функціонування системи:

$$A^c = \bigcup_t A^c(t)$$

Множину A^c будемо називати (c)-множиною.

Кожний користувач має власний погляд на спосіб досягнення розв'язку. Тобто множини $A^c(t)$ є різними залежно від значення c.

Залежно від методу розв'язання задачі та користувача в момент часу t, кожного разу відбуваються різні цикли інтерпретатора. Інакше кажучи, побудова узагальненої множини знань, що використовується в деякий час t, не має ніякого змісту. Тому слід будувати сумарну множину знань, що використовували різні користувачі для розв'язання деякої окремої підзадачі p. Тобто будемо розглядати множину:

$$A^p = \bigcup_c A^c(t),\ t \in [p_a^c, p_b^c]$$

де p_a^c - момент початку розв'язання підзадачі p користувачем c,

p_b^c - кінець розв'язання підзадачі p користувачем c.

Множину A^p будемо називати (p)-множиною.

Якщо t вибирається із всього проміжку розв'язку задачі, то така сумарна множина називається силовою множиною і позначаємо $\tilde{A}$. Тобто

$$\tilde{A} = \bigcup_c A^c(t),\ t \in [0, t_c]$$

де t_c – момент закінчення процесу розв'язання задачі користувачем c.

Зауважимо, що сумарна множина початкових вимог (обмежень) усіх користувачів матиме вигляд $L = \bigcup_c L^c$.

Використання (c)-множин.

Дані множини використовуються для проведення навчання системи при роботі ІІС з одним користувачем. Задача системи полягає у вивченні поведінки користувача, щоб у майбутньому, не проводячи з ним діалогу, видавати розв'язок, що цікавить користувача c.

Нехай користувач використовував ІІС N разів. На кожному етапі n експлуатації ІІС користувачем c вибираються знання. На основі цих знань побудуємо відповідні (c,t)-множини $A_n^c(t)$, $n=1,2,...,N$. Надалі для кожного n побудуємо сумарні (c)-множини $A_n^c=\left\{a_{n_1}^c,a_{n_2}^c,...,a_{n_s}^c\right\}$. Для кожного елемента $a_{n_i}^c$ із множин A_n^c визначимо кількість (c)-множин, в які входить цей елемент. Порядок елементів не є важливим. Тому індекс n_i замінимо на j. Тоді кількість (c)-множин, у які входять знання a_j^c, позначимо N_j^c.

Означення. Коефіцієнтом вибору інформації a_j^c для користувача c називається відношення кількості (c)-множин, в які входить інформація a_j^c, до загальної кількості (c)-множин:

$$\kappa_j^c=\frac{N_j^c}{N}$$

Набір елементів κ_j^c утворює вектор вибору інформації для користувача c $K_N^c=\left(\kappa_1^c,...,\kappa_{m_c}^c\right)$, де m_c – кількість різних елементів у всіх (c)-множинах разом. Інакше кажучи, m_c є потужністю множини $\bigcup\limits_n A_n^c$, $n=1,...,N$. З означення коефіцієнта вибору випливає, що $\kappa_j^c\in(0,1]$. Чим більше значення κ_j^c, тим частіше вибиралась інформація a_j^c користувачем c. Інформація, для якої коефіцієнт вибору дорівнює одиниці називається максимально вибраною.

При подальшій роботі користувача із системою збільшується N, що веде до зміни коефіцієнта вибору інформації. Однак з часом вектор вибору збігається з деяким вектором $\hat{K}^c=\left(\hat{\kappa}_1^c,...,\hat{\kappa}_{m_c}^c\right)$, який назвемо границею вибору, тобто:

$$(\forall\varepsilon>0)\left(\exists N_\varepsilon^c\right)\left(\forall N>N_\varepsilon^c\right)\Rightarrow\left\{\left\|K_N^c-\hat{K}^c\right\|<\varepsilon\right\}$$

Значення N_ε^c назвемо зміною поведінки користувача c. Чим менше N_ε^c, тим постійнішою є поведінка користувача c. Значення ε визначає точність границі вибору користувача. Чим менше значення ε, тим достовірніша границя вибору.

На основі вектора границі вибору $\hat{K}^c$ побудуємо множину відфільтрованих знань. Для цього введемо значення порогу достовірності λ (шкала важливості). Побудуємо множину $\hat{A}^c=\left\{a_j^c\,\middle|\,\kappa_j^c\geq\lambda\right\}$, яку назвемо прохідною множиною користувача c. Елементи цієї множини позначимо у вигляді $\hat{A}^c=\left\{\hat{a}_1^c,\hat{a}_2^c,...,\hat{a}_{m_\lambda}^c\right\}$.

Дану множину знань використаємо для пошуку нових стратегій процесу розв'язання задачі, такої, що модель керування розв'язанням задачі i *проходить* через ці знання

(інтерполяційна задача). Процес знаходження таких стратегій називається досвідом системи. Такий досвід базується на погляді окремого користувача, тобто має суб'єктивний характер.

Використання (p)-множин.

Задача полягає у пошуку нових стратегій пошуку розв'язку деякої підзадачі p (всієї задачі) з точки зору різних користувачів.

Розглянемо деяку підзадачу p. Вважаємо, що її розв'язують C різних користувачів. Для кожного користувача c знайдемо вектор вибору інформації щодо підзадачі p.

$$K_{N^c}^{cp} = \left(\kappa_1^{cp}, ..., \kappa_{m_c^p}^{cp}\right),$$

де N^c - кількість разів розв'язування даної задачі користувачем c,

m_c^p - використана кількість інформації користувачем c для розв'язку підзадачі p, $c = 1, 2, ..., C$.

Побудуємо вектор, елементи якого є сумою коефіцієнтів вибору однакових знань різними користувачами щодо підзадачі p, розділені на кількість разів розв'язування підзадачі p

$$\Im_{\mathrm{T}}^p = \left(\tau_1^p, \tau_2^p, ..., \tau_{m_p}^p\right),$$

де $\mathrm{T} = \sum_{c=1}^{C} N^c$, m_p - кількість різних елементів (p)-множини, $\tau_i^p = \frac{\sum_c \kappa_{j_c}^{cp}}{\mathrm{T}}$, j_c - порядковий номер вибору інформації a_i^p користувачем c.

Такий вектор назвемо об'єктивним вибором. З означення об'єктивного вибору випливає, що $\tau_i^p \in (0,1]$. Чим більше значення τ_i^p, тим частіше вибиралась інформація a_i^p щодо розв'язку підзадачі p. Інформація для якої об'єктивність вибору рівна одиниці називається максимально вибраною щодо підзадачі p.

При подальшій роботі користувачів із системою збільшується T, що веде до зміни об'єктивного вибору інформації. Однак з часом вектор об'єктивного вибору збігається до деякого вектора $\hat{\Im}^p = \left(\hat{\tau}_1^p, \hat{\tau}_2^p, ..., \hat{\tau}_{m_p}^p\right)$, який назвемо границею об'єктивного вибору, тобто:

$$(\forall \varepsilon > 0)\left(\exists \mathrm{T}_\varepsilon^p\right)\left(\forall \mathrm{T} > \mathrm{T}_\varepsilon^p\right) \Rightarrow \left\{\left\|\Im_{\mathrm{T}}^p - \hat{\Im}^p\right\| < \varepsilon\right\}$$

Значення N_ε^p назвемо зміною поведінки користувачів щодо розв'язування підзадачі p. Чим менше N_ε^p, тим більш постійною є поведінка користувачів. Значення ε визначає точність границі об'єктивного вибору для підзадачі p. Чим менше значення ε, тим достовірніша границя об'єктивного вибору.

На основі вектора границі об'єктивного вибору $\hat{\Im}^P$ фільтруємо знання, що є найбільш вибраними для підзадачі. Для цієї мети введемо значення порогу μ для вибору істотної інформації. Побудуємо множину $\hat{A}^p = \left\{a_j^p \middle| \tau_j^p \geq \mu\right\}$, яку назвемо вибраною

множиною для підзадачі p. Елементи цієї множини позначимо у вигляді $\hat{A}^p = \left\{\hat{a}_1^p, \hat{a}_2^p, ..., \hat{a}_{m_\mu}^p\right\}$.

Дану множину знань використаємо для пошуку нових стратегій процесу розв'язку підзадачі p, такої що функція керування i *проходить* через ці знання. Процес знаходження таких стратегій називається досвідом системи. Такий досвід має об'єктивний характер.

Фільтрація інформації повинна стати першим кроком на шляху побудови алгоритмів самонавчання інтелектуальних інформаційних систем. Такі алгоритми, на думку автора, нічим не поступатимуться алгоритмам розпізнавання образів, що базуються на нейронних мережах чи генетичних алгоритмах.

1. Борисов А.Н., Вилюмс Э.Р., Сукур Л.Я. Диалоговые системы принятия решений на базе мини-ЭВМ. - Рига: Зинатне, 1986. - 310 с. 2. Вилкас Э.Й., Майминас Е.З. Решения: теория, информация, моделирование. - М.: Радио и связь, 1981. 3. Литвин В.В. Модель функціонування інтелектуальної інформаційної системи. // Тези відкритої наук.-техн. конференції молодих науковців і спеціалістів Фізико-механічного інституту ім. Г.В.Карпенка НАН України. - Львів, 2000. - С. 122-123. 4. Литвин В.В. Формальна модель інтелектуальної інформаційної системи // Вісн. ДУ "Львівська політехніка" "Комп'ютерна інженерія та інформаційні технології", 1999. № 386. 5. Попов Э.В., Фоминых И.Б., Кисель Е.Б., Шапот М.Д. Статические и динамические экспертные системы. - М.: "Финансы и статистика", 1996. 6. Поспелов Г.С. Искусственный интеллект – основа новой информационной технологии. - М.: Наука, 1988. - 280 с.

УДК. 681.3

В.В. Литвин, В.В. Григор'єв
НУ "Львівська політехніка",
кафедра "Інформаційні системи та мережі"

ЗАСТОСУВАННЯ УТИЛІТАРНОГО ПІДХОДУ ДО РЕАЛІЗАЦІЇ ПРИРОДНОМОВНОГО ІНТЕРФЕЙСУ ІНТЕЛЕКТУАЛЬНИХ ІНФОРМАЦІЙНИХ СИСТЕМ

This paper considers main principles of utilitarian approach for the working out natural language interface for intellectual computer systems.

Досліджено питання застосування утилітарного підходу до реалізації природмовного інтерфейсу. Проведено аналіз інших підходів. Наведено приклад прикладної програми у якій реалізовано результати дослідження.

Як відомо, існує два основні підходи до створення систем природної мови. Перший з них, відомий як психолінгвістичний, - це осягнення через моделювання відповідних людських психологічних механізмів, що забезпечують розуміння людиною природномовних