

Интелигентни системи за мобилни устройства, базирани на Java технология

Христо Ненов, Петър Антонов

Intelligent systems for mobile devices, based on Java technologies: *The paper describes one of the methods for creating possibility to use intelligent systems from some mobile device like PDA, Smartphone, iPaq etc. This method uses the power of Java technologies and principal of creating of network enterprise applications.*

Key words: *Mobile devices, Intelligent systems, Java technologies, Model-driven architecture, KDD, Data-Mining.*

ВЪВЕДЕНИЕ

Известно е, че извличането на знания от данни е процес на откриване на смислени корелации, зависимости, повтарящи се образци (patterns), тенденции и аномалии в големи масиви, съхранявани в складове, чрез използване на техники и алгоритми от областта на машинното обучение, разпознаването на образи, статистиката, невронните мрежи и визуализацията на данни. За разлика от извличането на данни посредством заявки към база данни, извличането на знания от данните генерира имплицитно формулирана, неочевидна, предварително неизвестна, потенциално полезна информация [1]. Използването на БД е неизбежно при решаване на задачите по оптимално стохастическо управление, предвид наличието и необходимостта от множество разнородни данни. Това прави особено подходящо прилагането на решения, свързани с използване на БД, като KDD и Data Mining. Проблемите, които съществуват в тази област, са свързани със силната разнородност на данните и източниците на данни, както и с множеството различни методи на KDD, водещо до трудности при формализирането, прилагането, интерпретацията на резултатите и тяхната сравнимост и преносимост. По тази причина подобен род системи винаги се асоциират с необходимостта на налична мощна изчислителна техника. Целта на настоящата статия е да представи и анализира едно по-различно решение, а именно - реализация на такава система върху мобилно устройство от типа PDA или Smartphone.

ИНТЕЛИГЕНТНИ СИСТЕМИ И МОБИЛНИ УСТРОЙСТВА

Използването на мобилно устройство като база за приложението и необходимостта от голяма изчислителна мощ са коренно противоположни неща, дори и при съвременните сравнително развити апаратни реализации. Съществено подпомагане на това може да се постигне чрез решаването на следните въпроси:

- необходимост от унифицирани процедури за оценка;
- необходимост от лесна интерпретируемост на резултатите.

В тази посока са и следващите задачи разгледани в настоящата статия :

- сравняване на математическата и същностна основа на различни методи за индуктивно търсене,
- синтез на типова, унифицирана процедура (алгоритъм, система), която да направи възможно директното сравняване и преносимост на резултатите
- изследване на различни възможности за програмна реализация на системата.

За целта се изследват различни методи за класификация, като се търси изчислителна общност, независима от началната им същност.

Под образ се разбира множество от явления, обединени от общи свойства, а под изображение – всяко конкретно явление, което по своите свойства трябва да

бъде причислено към един от образите. Следвайки първоначалното приетата система от понятия, образът е група, която се формира като еталон в процеса на обучение, а изображението е хипотетичен представител на някои от предварително постулираните образи, чиято класификация се осъществява в процеса на разпознаване [1,2,3].

Съществен, при решаване на задачата от комплекса “обучение–разпознаване”, е изборът на пространството, в което се представят образите и изображенията, както и метрика, даваща количествена представа за формирането на образите по съответстващите и свойства. За решаването на първата част на задачата се използва координатна система (хиперпространство) на съответните признаци, описващи важните свойства на образа или изображението. Втората част е свързана с начина, по който е прието да се описва близостта или отдалечеността на обектите в пространството на признаците. Формализацията на понятието “мярка за сходство” е една от основните задачи от теорията на разпознаване на образи.

Разстоянието между две изображения S_i и S_j е някакво абстрактно свойство на пространството на признаците, отговарящо на условията:

- всяко изображение се представя като точка в пространството на признаците
- мярката на сходството е винаги неотрицателна величина: $L(S_i, S_k) \geq 0$ (1)
- мярката за близост притежава свойството симетрия: $L(S_i, S_k) = L(S_k, S_i)$ (2)
- мярката за сходство на едно изображение със самото себе си е по-голяма (или не по-малка) от мярката за сходство на това изображение с всяко друго:

$$L(S_i, S_i) = \max_R L(S_i, S_k)$$
 (3)
- при компактни образи мярката на сходство трябва да е монотонна функция на отдалечеността на точките, съответстваща на сравняваните изображения.

МНОГОМЕРНА ЗАДАЧА

Логично е да се приеме в многомерната задача, че най-простият начин за определяне на отдалечението между две точки (изображения) е т.н. Евклидова мярка. В такъв случай към гореизложените общи ограничения се добавя и известното геометрично условие:

$$L(S_i, S_k) \leq L(S_i, S_j) + L(S_j, S_k),$$
 (4)

а мярката на подобие се приема като разстояние в средно квадратичен смисъл:

$$L(S_i, S_j) = \sqrt{d^2 L(S_i, S_k)}$$
 (5)

Нека в пространството на признаците са зададени точки: $W_1, W_2 \dots W_m$, представляващи центрове на обобщения или еталонни образци, които са определени предварително в процеса на обучението.

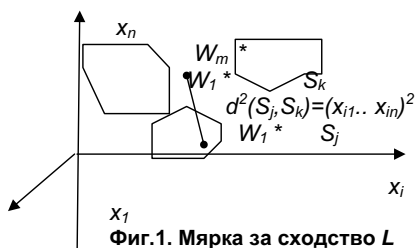
Всяко входно изображение, подлежащо на разпознаване (класификация) се представя като точка в n -мерното пространство.

Входното изображение се отнася към този образ, разстоянието до който е минимално, т.е.

$$S_k \in W_j, \text{ ако } L(S_k, W_j) = \min_j L(S_k, W_j)$$
 (6)

Горното условие е решаващото правило, в съответствие с което входното изображение S_k се отъждествява с някой от образите W_i . Като имаме предвид, че входните изображения и еталоните са зададени в координатното пространство на признаците:

$$S = S(X_1, X_2, \dots, X_n) \quad W = W(X_1^*, X_2^*, \dots, X_n^*)$$
 (7)



В тази чисто геометрична постановка разпознаването се осъществява по разстоянието между точката S_k и всяка от точките $W_1, W_2, \dots, W_m$, (Фиг.1).

Тези разстояния са мярка за сходството на разпознаването изображение с всеки от еталонните образци.

Квадратичната мярка за близост се пресмята по формулата:

$$L(S_i, W) = \sqrt{\sum_{i=1}^n (X_i - X_i^*)^2} \quad (8)$$

Изразът, известен като “правило на най-близкия съсед” (kNN), не е единствен. Понякога се прилага правило, класифициращо изображението по областта, в която то се включва:

$$S_k \in W_j, \text{ ако } L(S_k, W_j) = L_{kj} = \sqrt{(X_i - X_i^*)^2} \leq L_{j\max} \quad (9)$$

или линейно разделящо правило:

$$\sum a_i X_i - a_{m+1} = 0 \quad (10)$$

Във всички случаи обаче, приемането на квадратичното Евклидово разстояние се базира на независимост на признаците $X_i (i=1, 2, \dots, n)$, т.е. разстоянията се определят в декартова координатна система. В случай, че признаците са зависими, координатната система е афинна, а разстоянията се пресмятат като:

$$L'(S_i, W) = \sqrt{\sum_{i=1}^n a_i (X_i - X_i^*)^2} \quad (11)$$

или:

$$d^2(S_i, W) = (\bar{x} - \bar{\mu}_i)^T u_i^{-1} (\bar{x} - \bar{\mu}_i) + (\bar{x} - \bar{x}^*)^T u (\bar{x} - \bar{x}^*) \quad (12)$$

където μ_i и W_i са центри на ковариационната матрица на съответния образ.

ПРАВИЛА ЗА РАЗПОЗНАВАНЕ

За да се сравнят изчислителните процедури при геометричния и вероятностен подходи се търси възможно най-общото правило за приемане на решение, базирано на минимизиране на средния риск. Без намаляване общността на задачата се разглежда двуалтернативен случай (два класа на състояние) в n -мерно пространство на признаците. Правилото за разпознаване на изображения S_k се свежда до причисляването му към този клас (образ), до който разстоянието е най-малко. Средната стойност (риск), която трябва да бъде заплатена при многократно разпознаване на неизвестни вектори на наблюдение е (Фиг.2):

$$R = C_{11} P(W_1) p_1 + C_{12} P(W_1) q_1 + C_{22} P(W_2) p_2 + C_{21} P(W_2) q_2 \quad (13)$$

където:

$P(W_1), P(W_2)$ – априорна вероятност за поява на съответните еталонни образи (класове)

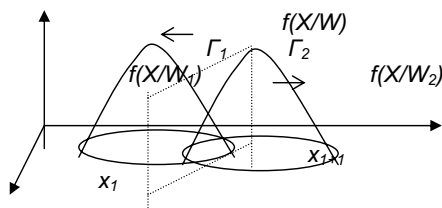
$C_{ij} (i, j=1, 2)$ – стойността на загубите (печалбите), които се изплащат при неправилна или правилна класификация

$p_1 = \int_{\Gamma_1} \dots \int f(\bar{X}/W_1) dx$ - условна вероятност за правилно класифициращо решение към клас W_1

$q_1 = \int_{\Gamma_2} \dots \int f(\bar{X}/W_1) dx$ - вероятност за неправилно класифициране на изображението от W_1 към W_2 - "лъжлива тревога"

$p_2 = \int_{\Gamma_1} \dots \int f(\bar{X}/W_2) dx$ - условна вероятност за правилно класифициращо решение към клас W_2

$q_2 = \int_{\Gamma_2} \dots \int f(\bar{X}/W_2) dx$ - вероятност за неправилно класифициране на изображението от W_2 към W_1



Фиг.2. Области и разделяща повърхност

След заместване и като се има предвид, че:

$$\int_{\Gamma_2} \dots \int f(\bar{X}/W_1) dx = 1 - \int_{\Gamma_1} \dots \int f(\bar{X}/W_2) dx \quad (14)$$

получава се математическо-то очакване на функцията на средния риск във вида:

$$R = C_{12} P(W_1) + C_{22} P(W_2) + \int_{\Gamma_2} \dots \int [-(C_{12} - C_{11})P(W_1)f(\bar{X}/W_1) + (C_{21} - C_{22})P(W_2)f(\bar{X}/W_2)] dx \quad (15)$$

Търси се такъв избор на едната от двете области (напр. Γ_1), че да се минимизира функцията на риска, т.е. интегралът да приеме своята максимална отрицателна стойност. Ако се приеме, че за даденото изображение $\bar{X}$ подинтегралната величина не е отрицателна, следва, че за да се намали риска, трябва да се отнесе това изображение към областта Γ_1 . Следователно, решаващото правило минимизиращо риска се свежда до класифициране към областта на всяко изображение, за което подинтегралният израз е отрицателен:

$$\bar{X} \in W_1, \text{ ако } (C_{12} - C_{11})P(W_1)f(\bar{X}/W_1) + (C_{21} - C_{22})P(W_2)f(\bar{X}/W_2) < 0 \quad (16)$$

От тук се получава правилото за приемане на класифициращи решения, осигуряващо минимизиране на средния риск :

$$\bar{X} \in \begin{cases} W_1, \text{ ако } \frac{f(\bar{X}/W_1)}{f(\bar{X}/W_2)} > \frac{(C_{21} - C_{22})P(W_2)}{(C_{12} - C_{11})P(W_1)} \\ W_2, \text{ ако } \frac{f(\bar{X}/W_1)}{f(\bar{X}/W_2)} < \frac{(C_{21} - C_{22})P(W_2)}{(C_{12} - C_{11})P(W_1)} \end{cases} \quad (17)$$

Представеният алгоритъм представлява завършена и оптимизирана система за анализ и оценка на информацията и класификация. Такъв тип системи работят по метода на „обучение и разпознаване“. Експериментално е установено, че за правилната работа на алгоритъма е нужна обучаваща извадка от поне десет наблюдения за даден клас. Изчислителната процедура варира по сложност само при изчисляване и определянето на разделящите функции („обучението“ на системата), останалите етапи на работа не зависят от обема на данните.

Следващата стъпка е намирането на подходящи технологии и възможни сценарии за програмна реализация.

JAVA ТЕХНОЛОГИИ ЗА ПРОГРАМНА РЕАЛИЗАЦИЯ

Java езикът за програмиране и технологиите, които са свързани с него, отговарят на няколко много важни условия от програмна гледна точка [5,7,8]:

- платформена независимост – благодарение на спецификата на изпълнение

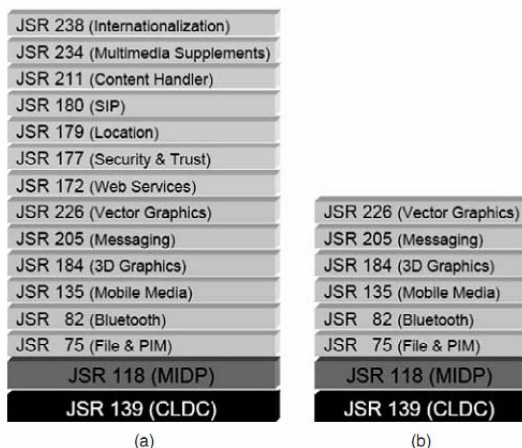
(JVM) кодът е изпълним на всякакъв тип хардуерна и операционна система (нещо повече, всяко съвременно мобилно устройство има вградена при производството си поддръжка на Java);

- сигурност – за разлика от програмите, написани на C и C++, кодът на Java не може да доведе до сривове в системата;
- многонишков интерфейс – поддръжката на многонишково изпълнение е заложена в основата на самия език.

Според стандартите на “Sun” за мобилните устройства, те се делят на два класа, най-общо казано: устройства от среден клас и устройства от висок клас [9,10]:

- Connected Limited Device Configuration (CLDC) е спецификацията за мобилни телефони двулънейни пейджъри и нисък клас PDA устройства.
- Connected Device Configuration (CDC) е спецификацията за устройства с повече памет, по добре апаратно организирана мрежова работа и по бързи процесори.

Всеки от тези два стандарта има свои изисквания от Java спецификации(JSR), които изграждат „Mobile Services Architecture (MSA)” на даденото устройство.



Фиг. 3: а) Пълна MSA; б) MSA, адаптирана към дадено устройство [9]

Табл.1[9]

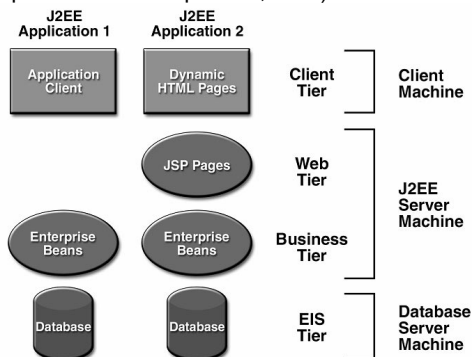
	JSR	Nokia 6,600	Nokia N70	Nokia N96
Symbian OS version		v7.0 s	v8.1a	v9.3
Release Date		16 June 2003	27 April 2005	11 Feb 2008
CLDC 1.0/1.1		1.0	1.1	1.1
MIDP 2.0/2.1		2.0	2.0	2.1
Bluetooth	82	✓		✓
Mobile Media (MMAPI)	135	✓	✓	✓
Wireless Messaging	120	✓	✓	✓
Wireless Messaging	205 ¹⁹			✓
File and PIM	75		✓	✓
Mobile 3D Graphics (M3G)	184		✓	✓
JTWI	185		✓	✓
Web Services	172		✓	✓
Location API	179			✓
Security and Trust Services	177			✓
SIP API	180			✓
Scalable 2D Vector Graphics	226			✓
Advanced Media Supplements	234			✓
MSA	248			MSA Subset

В таблица 1 е представено развитието на поддръжката на различните услуги от операционната система "Symbian", използвана върху различни модели на „Нокиа“.

В зависимост от целите поставени при създаването на системата и съобразяването с възможностите на различните мобилни устройства са реализирани три варианта, като два от тях са по модела „клиент-сървър“ архитектура (Фиг.4):

а) **Модел на слабия (thin) клиент [5]** - цялата обработка и управление на данните (бизнес логиката на системата) се извършва на отдалечена машина (сървър), а на мобилното устройство се визуализират само резултатите от изчислителните процеси. Клиентската част базирана на мобилното устройство е отговорна само за представителния слой (Реализация по този сценарий се осъществява чрез Web технологии на Java - Servlet, Jsp, EJB от страна на сървъра и Browser от страна на клиента)

б) **Модел на дебелия (Fat) [5] клиент** - сървърът е отговорен само за управлението на данните. Софтуерът на клиента осъществява логиката на приложението и взаимодействието с потребителя. (Реализация по този сценарий се осъществява чрез Java технологията MicroEdition предназначена за създаване на софтуер за мобилни устройства – Smartphones, PDA)



Фиг.4. Реализация на система по многослойна „клиент-сървър“ архитектура [5]

в) **Мобилно приложение** - реализация на „независимо“ приложение, изцяло базирано на мобилното устройство (PDA, Smartphone).

Нуждата от подобен род реализация изниква поради факта, че по своята същност предните два сценария обесмислят „мобилността“ на „клиента“, тъй като за достъп до такава система се изисква наличие на връзка към нея. Отделянето на системата като самостоятелно приложение, базирано на мобилно устройство, отново позволява автономен режим на работа, отчитайки някои съображения и ограничения. Това е реализирано благодарение на следните използвани Java технологии:

- Model-Driven архитектура – описание чрез модели (лесни гъвкави интерфейси и модули на системата (сравнително намалява необходимостта от високи апаратни изисквания)[1]
- JaxP интерфейс – комуникация между Java и XML на ниво модели [1,9,11]
- JDOM интерфейс – динамично генериране на presentation layer (отпада необходимостта от предварително зареждане в паметта на форми, като по този начин се постига гъвкавост по отношение на използваната памет).[1,9]
- JaxB интерфейс - комуникация между Java и XML на ниво съхранение и извличане на данни (в зависимост от обема на данните, с които се работи, варира и ефективността на този интерфейс) [1,11].

ЗАКЛЮЧЕНИЕ

В резултат на направените изследвания и анализ могат да бъдат формулирани следните изводи: 1. Мобилните устройства, макар и доста развили се апаратно през последните години, налагат някои ограничения по отношение на тяхното използване (работна честота на процесора, количество памет и т.н.); 2. Функционирането на една такава система, базирана на мобилно устройство, зависи от спецификата на нейното използване; 3. При работа с „необучена система“ апаратните изисквания към устройството са твърде високи, а относителността в понятието „време за отговор“ на системата още повече би намалила нейната ефективност; 4. В случай на вече „обучена система“ и извлечени и синтезирани модели на данни и функции, алгоритъмът е приложим дори за устройства с не много високи технически характеристики.

Представената система е тествана върху iPaq pocket PC и няколко модела на Nokia, което доказва нейната работоспособност и ефективност.

ЛИТЕРАТУРА

- [1] Ненов, Х. Алгоритми и системи за оценка на състоянието и управление на сложни обекти - дисертация, 2008.
- [2] Недев, А. и др. Въведение в теорията на управлението. ТУ-Варна, 1999.
- [3] Недев, А., К. Тенекеджиев. Техническа диагностика и разпознаване на образи. ТУ-Варна, 1994.
- [4] Недев, А., Х. Ненов. Възможности за управление на адаптивни и предприемчиви академични структури. ISBN 954-760-084-2, 2005.
- [5] Wesley, A. Java2 Platform Enterprise Edition. Platform and Component Specifications. ISBN 0-201-70456-0, 2000.
- [6] Michalski, R.S., K. Kaufman. Building Knowledge Scouts Using KGL Metalanguage. // Fundamental Informaticae, Vol. 40, 2000, pp. 433-447.
- [7] John, S., Design Patterns Java™ Workbook. ISBN 0-201-74397-3, 2003.
- [8] O'Donahue, J. Java Database Programming Bible. ISBN 0764549243, 2002.
- [9] Issott, A., Common Design Patterns for Symbian OS. ISBN978-0-470-51635-5.
- [10] Giguere, E., Java2 ME Professional Developer's guide. ISBN 0-471-39065-8
- [11] McLaughlin, B. Java And Xml Data Binding. ISBN 0-596-00278-5.

За контакти:

Гл.ас. д-р Христо Божидаков Ненов, Катедра “Компютърни науки и технологии”, Технически университет – Варна, тел. 383-403, e-mail: ico762001@yahoo.com

Доц. д-р Петър Цветанов Антонов, Катедра “Компютърни науки и технологии”, Технически университет – Варна, тел.383-320, e-mail: peter.antonov@ieee.org

Докладът е рецензиран.