

4. Молдовян Н. А. Проблематика и методы криптографии. - СПб.: Изд-во СПбГУ, 1998. - 245 с.

Рецензент: Корнейчук М.Т.

Поступила 21.12.2011

УДК 004.021

Петров А.О.

Східноукр. Націонал. Унів. Ім. В. Даля

## МОДЕЛІ ТА МЕТОДИ РОЗПІЗНАВАННЯ МОВИ

### 1. Приховані Марківські моделі

Найшвидша та ефективна взаємодія між людьми відбувається за допомогою усної мови. За допомогою мови можуть бути передані різні почуття й емоції, а головне — корисна інформація. Необхідність створення комп'ютерних інтерфейсів звукового введення-виводу не викликає сумнівів, оскільки їх ефективність заснована на практично необмежених можливостях формулювання у всіляких областях людської діяльності.

Перша електронна машина, що синтезує англійську мову, була представлена в Нью-Йорку на торговельній виставці в 1939 році та називалася voder, але звук, який вона відтворювала, був украй нечітким. Перше ж пристрій для розпізнавання мови було створено у 1952 році та був здатен розпізнавати цифри [3].

У процесі розпізнавання мови можна виділити наступні складності: довільний, найвний користувач; спонтанна мова, супроводжувана аграматизмами й мовним «сміттям»; наявність акустичних перешкод і викривлень; наявність мовних перешкод [20].

Із усього різноманіття методів у даній статті ми розглянемо можливість створення статистичної моделі за допомогою прихованих Марківських моделей (ПММ) [1].

Прихована Марківська модель — статистична модель, що імітує роботу процесу, схожого на марківський процес з невідомими параметрами. Завданням ПММ ставиться визначення невідомих параметрів на основі спостережуваних. Отримані параметри може бути використано в подальшому аналізі, наприклад, для розпізнавання образів. ПММ може розглядатися як найпростіша Байєсова мережа довіри [1].

При аналізі природньої мови першим кроком необхідно визначити: до якої частини мови ставиться кожне зі слів у пропозиції. В англійській мові завдання на цьому етапі називається Part-Of-Speech tagging. Яким образом ми можемо визначити частину мови окремого члена пропозиції? Розглянемо речення англійською мовою: «The can will rust». Отже, the — певний артикль або частка «тем»; can — може одночасно бути й модальним дієсловом, і іменником, і дієсловом; will — модальне дієслово, іменник і дієслово; rust — іменник або дієслово. У статистичному підході необхідно побудувати таблицю ймовірностей використання слів у кожному граматичному значенні. Це завдання можна розв'язати на основі тестових текстів, проаналізованих вручну. І відразу можна виділити одну із проблем: слово «can» у більшості випадків використовується як дієслова, але іноді воно може бути й іменником. Враховуючи цей недолік, була створена модель, що ухвалює в увагу той факт, що після артикля піде прикметник або іменник:

$$\operatorname{argmax}_{t_1..t_n} \prod_{t=1}^n p(w_t | t_t) p(t_t | t_{t-1}), (1)$$

де  $t$  – таг (іменник, прикметник і т.д.);  $w$  – слово в тексті ( $\text{rust} \rightarrow \text{tag} = w$ ) – імовірність того, що слово  $w$  відповідає тагу  $t$ ;  $p(t_1|t_2)$  – імовірність того, що  $t_1$  після  $t_2$ .

Із запропонованої формули видно, що ми намагаємося підібрати таку таг, щоб слово підходило тагу, і таг підходив попередньому тагу. Даний метод дозволяє вважати, що «can» виступає в ролі іменника, а не як модального дієслова [5].

Ця статистична модель може бути описана як ергодична ПММ:

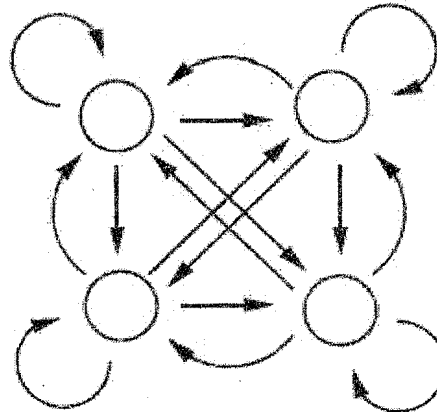


Рис. 1. Ергодична Марківська модель

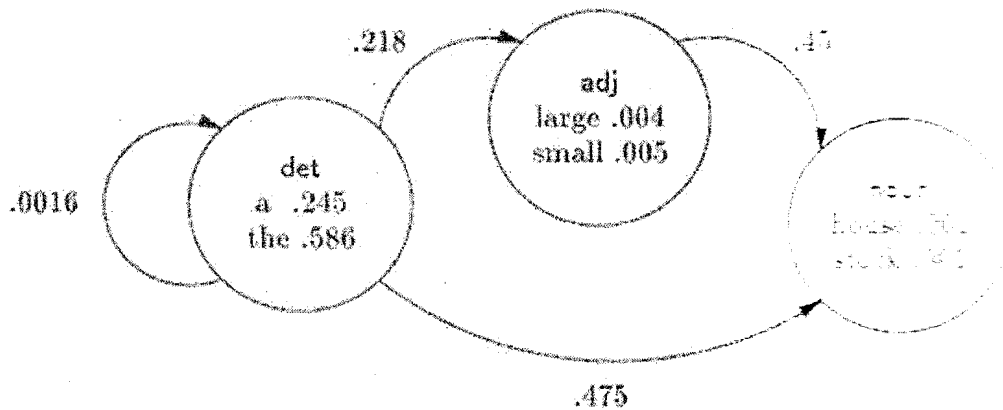


Рис. 2. Ергодична Марківська модель

Кожна вершина в даній схемі позначає окрему частину мови, у якій записуються пари (слово; імовірність, що слово ставиться саме до цієї частини мови). Переходи показують можливу ймовірність проходження однієї частини мови за іншою. Так, наприклад, імовірність того, що підряд будуть іти 2 артикля, за умови, що зустрінеться артикль, буде рівна 0,0016. Даний етап розпізнавання мови дуже важливий, тому що правильне визначення граматичної структури пропозиції дозволяє підібрати вірну граматичну конструкцію для експресивного фарбування відтвореного пропозиції.

Також існують  $n$ -грамні моделі розпізнавання мовного потоку. Вони засновані на припущенні, що ймовірність уживання чергового слова в пропозиції залежить тільки від  $n-1$  слів. Сьогодні найбільш популярні біграмні та триграмні моделі мови. Пошук у таких моделях відбувається по великій таблиці (корпусу). Незважаючи на швидко працюючий алгоритм, такі моделі не здатні вловити семантичні й синтаксичні зв'язки, якщо залежні слова перебувають на відстані 5 слів друг від друга. Використання ж  $n$ -грамних моделей, де  $n$  більше чому 5, вимагає величезних потужностей [24].

Як ми вже відзначали вище, самою популярною моделлю на сьогоднішній день є триграмна модель. Умовна імовірність спостереження пропозиції  $w_1, \dots, w_n$  наближена до:

$$P(w_1, \dots, w_n) = \prod P(w_i | w_1, \dots, w_{i-1}) \approx \prod P(w_i | w_{i-2}, \dots, w_{i-1})$$

Наприклад, розглянемо пропозицію «I want to go home». Імовірність цієї пропозиції можна обчислити від рахунку частоти n-грама (у цьому прикладі оберемо  $n=3$ ):

$$P(I, \text{ want, to, go, home}) \approx P(I) * P(\text{want}|i) * P(\text{to}|i, \text{ want}) * P(\text{go}|\text{want, to}) * P(\text{home}|\text{ to, go})$$

Варто відзначити далекодіючу триграмну модель, в якій аналіз ведеться не тільки по двом попереднім словам, а по будь-якій парі слів, що перебувають поруч. Така триграмна модель може пропускати малоінформативні слова, тим самим поліпшуючи передбачуваність сполучуваності в моделі

## 2. Методи динамічного програмування

У системах розпізнавання мови, що містять слова, розпізнавання вимагає порівняння між вхідним словом і різними словами в словнику. Ефективний розв'язок проблеми лежить у динамічних алгоритмах порівняння, метою якого є введення тимчасових масштабів двох слів в оптимальну відповідність. Алгоритми такого типу є динамічними алгоритмами трансформації часової шкали. У даній статті представлено два варіанти реалізації алгоритму призначені для розпізнавання окремих слів.

Дослідження в області розпізнавання мови, так само як і в інших областях, впливають по двом напрямкам: фундаментальні дослідження, метою яких є розробка й тестування нових методів, алгоритмів і концепцій на некомерційній основі; і прикладних досліджень, метою яких є поліпшення існуючих методів, дотримуючись певних критеріїв. У цій статті розглядається розпізнавання окремих слів у тенденції прикладних досліджень.

Фундаментальні дослідження спрямовані на одержання середньострокової або довгострокової вигоди, у той час як прикладні дослідження спрямовані на швидке поліпшення існуючих методів або розширення їх використання в областях, де такі методи ще практично не використовуються.

Поліпшити швидкість розпізнавання мови можна при обліку наступних критеріїв:

Розмір пізнаваної лексики;

- Ступінь спонтанності мови, яку необхідно розпізнати;
- Залежність/незалежність від диктора;
- Час, необхідне для приведення системи в рух;
- Час пристосування системи для нових користувачів;
- Час вибору й розпізнавання;
- Ступінь розпізнавання (виражена словом або пропозицією).

Сьогодні системи розпізнавання мови будуються на основі принципів визнання форм розпізнавання. Методи й алгоритми, які використовувалися дотепер, можуть бути розділені на чотири більші класи: Методи дискримінантного аналізу, засновані на Байєсовської дискримінації; приховані моделі Маркова; динамічне програмування – часові динамічні алгоритми (DTW); Нейронні мережі [4].

Алгоритм динамічного трансформування часу (DTW) обчислює оптимальну послідовність трансформації (деформації) часу між двома тимчасовими рядами. Алгоритм обчислює обоє значення деформації між двома рядами й відстанню між ними.

Припустимо, що в нас є дві числові послідовності  $(a_1, a_2, \dots, a_n)$  і  $(b_1, b_2, \dots, b_m)$ . Як бачимо, довжина двох послідовностей може бути різною. Алгоритм починається з

розрахунків локальних відхилень між елементами двох послідовностей, що використовують різні типи відхилень. Найпоширеніший спосіб для обчислення відхилень є метод, що розраховує абсолютне відхилення між значеннями двох елементів (Евклідова відстань). У результаті отримуємо матрицю відхилень, що має  $n$  рядків і  $m$  стовпців загальних членів:

$$d_{ij} = |a_i - b_j|, \quad i = \overline{1, n}, \quad j = \overline{1, m}. \quad (2)$$

Мінімальна відстань у матриці між послідовностями визначається за допомогою алгоритму динамічного програмування й наступного критерію оптимізації:

$$a_{ij} = d_{ij} + \min(a_{i-1, j-1}, a_{i-1, j}, a_{i, j-1}). \quad (3)$$

де:  $a_{ij}$  — мінімальна відстань між послідовностями  $(a_1, a_2, \dots, a_n)$  і  $(b_1, b_2, \dots, b_m)$ . Шлях деформації – це мінімальна відстань у матриці між елементами  $a_{11}$  і  $a_{nm}$ , що полягають із тих  $a_{ij}$  елементами, які виражають відстань до  $a_{nm}$

Глобальні деформації складаються із двох послідовностей і визначаються по наступній формулі:

$$GC = \frac{1}{p} \sum_{i=1}^p w_i, \quad (4)$$

де:  $w_i$  – елементи, які належать шляху деформації;  $p$  – їх кількість. Розрахунки проводилися для двох коротких послідовностей і зазначені в таблиці, у якій виділена послідовність деформації.

	-2	10	-10	15	-13	20	-5	14	2
3	5	12	25	37	53	70	78	89	90
-13	16	28	15	43	37	70	78	105	104
14	32	20	39	16	43	43	62	62	74
-7	37	37	23	38	22	49	45	66	71
9	48	38	42	29	44	33	47	50	57
-2	48	50	46	46	40	55	36	52	54

Існує три умови, що накладають на DTW алгоритм для забезпечення швидкої конвергенції:

1. Монотонність – шлях ніколи не вертається, тобто: обоє індексу,  $i$  і  $j$ , які використовуються в послідовності, ніколи не зменшуються.

2. Безперервність – послідовність просувається поступово: за один крок індекси,  $i$  і  $j$ , збільшуються не більше ніж на 1.3.

3. Граничність – послідовність починається в лівому нижньому куті й закінчується в правому верхньому.

Приклад деформації послідовності з використанням мови програмування Java наведений нижче:

```
void dtwalg (double a[], double b[], Stack<Double> _w, double _dw[][])
```

```

int n=a.length, m=b.length;
double d[][]=new double[n][m];
for(int i=0;i<n;i++)
    for(int j=0;j<m;j++)d[i][j]=Math.abs(a[i]-b[j]);
_dw[0][0]=d[0][0];
for(int i=1;i<n;i++)_dw[i][0]=d[i][0]+_dw[i-1][0];
for(int j=1;j<m;j++)_dw[0][j]=d[0][j]+_dw[0][j-1];
for(int i=1;i<n;i++)
    for(int j=1;j<m;j++)
        if(_dw[i-1][j-1]<=_dw[i-1][j])
            if(_dw[i-1][j-1]<=_dw[i][j-1])_dw[i][j]=d[i][j]+_dw[i-1][j-1];
        else _dw[i][j]=d[i][j]+_dw[i][j-1];
        else
            if(_dw[i-1][j]<=_dw[i][j-1])_dw[i][j]=d[i][j]+_dw[i-1][j];
            else _dw[i][j]=d[i][j]+_dw[i][j-1];
int i=n-1,j=m-1;
double element=_dw[i][j];
_w.push(new Double(_dw[i][j]));
do
{
    if(i>0&&j>0)
        if(_dw[i-1][j-1]<=_dw[i-1][j])
            if(_dw[i-1][j-1]<=_dw[i][j-1]){i--;j--;} else j--;
        else
            if(_dw[i-1][j]<=_dw[i][j-1])i--; else j--;
        else if(i==0)j--; else i--;
        _w.push(new Double(_dw[i][j]));
}
while(i!=0||j!=0);
}

```

Тому що для визначення основи послідовності в динамічному програмуванні оптимальним є використання метод зворотного програмування, необхідно використовувати певний динамічний тип структури, який називається «стек». Подібно будь-якому динамічному алгоритму програмування, DWT має поліноміальну складність. Коли ми маємо справу з більшими послідовностями, виникають дві незручності: запам'ятовування більших числових матриць; виконання великої кількості розрахунків відхилень.

Існує поліпшена версія алгоритму, Fastdwt, яка вирішує дві вищевказані проблеми. Розв'язок полягає в розбивці матриці станів на 2, 4, 8, 16 і т.д. менших по розміру матриць, за допомогою повторюваного процесу розбивки послідовності введення на дві частини. Таким чином, розрахунки відхилення проводяться тільки на цих невеликих матрицях, і шляхах деформації, розрахованих для невеликих матриць.

*Розглянемо більш докладно використання DWT алгоритму в розпізнаванні мови. Звук проходить через середовище, як позовжня хвиля зі швидкістю, що залежить від щільності середовища. Найпростіший спосіб вистави звуків – синусоїдальний графік. Графічна вистава вібрацій повітря під тиском протягом деякого часу.*

Форма звукової хвилі залежить від трьох факторів: амплітуди, частоти й фази.

Амплітуда — переміщення синусоїдальних графів вище й нижче тимчасової осі (в = 0), що відповідає енергії завантаженої звукової хвилі. Вимір амплітуди може бути зроблене в одиницях тиску (децибелах DB), які вимірюють амплітуду звичайного звуку за допомогою логарифмічних функцій. Вимір амплітуди використовуючи децибели дуже

важливо на практиці, тому що ця пряма вистава про те, як гучність звуку сприймається людьми. Частота – число циклів синусоїди за одну секунду. Швид коливань починається із середньою лінією, потім досягає максимуму й мінімуму, а після повертається до середньої лінії. Частота циклу вимірюється за одну секунду або в герцах (Гц). Величина зворотна частоті називається періодом – час необхідний звуковій хвилі для завершення циклу.

Останній фактор – фаза. Вона вимірює положення відносно початку синусоїдальної кривій. Фаза не може бути почута людиною, однак її можна визначити щодо положення між двома сигналами. Проте, слуховий апарат сприймає положення звуку на різних фазах.

Для того щоб розібрати звукові хвилі на синусоїдальній кривій скористаємося теоремою Фур'є. Вона говорить, що будь-яка комплексна періодична хвиля може бути розібрана за допомогою синусоїдальної кривої з різними частотами, амплітудами й фазами. Цей процес називається аналіз Фур'є, і його результатом є набір амплітуд, фаз і частот для кожного синусоїдального компонента хвилі. Складаючи ці синусоїдальні криві разом, одержуємо оригінальну звукову хвилю. Крапка частоти або фази, узятая разом з амплітудою, називається спектром. Будь-який періодичний сигнал показує, рекурсивну модель часу, яка відповідає першій частоті коливань сигналу й називається основною частотою. Вона може бути обмірювана з мовного сигналу, за допомогою перевірки періоду коливань близько 0 осі. Спектр показує частоту короткої послідовності звуків, і якщо ми прагнемо проаналізувати її розвиток протягом часу, необхідно знайти спосіб, що дозволяє продемонструвати це. Це можна показати на спектрограмі. Спектрограма — це діаграма у двох вимірах: частота й час, — у якій колір крапки (темний – сильний, світлий – слабкий) визначає амплітуду інтенсивності. Метод відіграє важливу роль у розпізнаванні мови, і професіонал може розкрити багато подробиць, дивлячись тільки на звукову спектрограму.

Сучасні методи виявлення можуть точно визначити початкову й кінцеву крапку вимовленого слова у звуковому потоці, на основі обробки сигналів мінливих протягом часу. Дані методи оцінюють енергію й середню величину в короткому відрізку часу, а також обчислюють середній рівень перетинання нуля.

Створення початкової й кінцевої крапки – просте завдань, якщо аудіо-запис зроблено в ідеальних умовах. У цьому випадку відношення сигнал-шум великий, тому що визначити дійсний сигнал у потоці шляхом аналізу образів не представляє праці. У реальних умовах усі не так просто: фоновий шум має величезну інтенсивність і може порушити процес відділення слів у потоці мови

Кращий алгоритм відділення слів — алгоритм Рабинел-Ламель. Якщо розглядати строб-імпульсів  $\{s_1, s_2, \dots, s_n\}$ , де  $n$  – число образів строб-імпульсів, а  $s_i, i=1, n$  – чисельне вираження зразків, загальна енергія строб-імпульсів обчислюється:

$$E(n) = \frac{1}{n} \sum_{i=1}^n s_i^2. \quad (5)$$

Середній рівень перетинання нульового рівня:

$$ZCR(n) = \sum_{i=1}^{n-1} \text{sign}(s_i) \cdot \text{sign}(s_{i+1}), \quad (6)$$

$$\text{sign}(s_i) = \begin{cases} 1 & \text{if } s_i > 0 \\ 0 & \text{if } s_i < 0 \end{cases}$$

Метод використовує три числові рівні: два для енергії (верхній, нижній) і один для середнього перетинання нульового рівня. Крапка, починаючи з якої енергія перекриває верхній рівень і рівень позитивних і негативних значень, не скасовує встановлений рівень, який вважається відправною крапкою голосового звучання (не тиші). Пошук першої такої крапки проводиться шляхом схрещування імпульсів від початку й до кінця, і це визначить першу область із мовою. Зворотний перехід, з кінця в початок, дозволяє визначити кінцеву крапку останньої області з мовою. Визначення усередині області може бути зроблене шляхом схрещування імпульсів між двома цими крапками. Початок глухої області починається в крапці, у якій енергія стає менше значення нижнього рівня.

Звернемо увагу на рис. 3, на якому до й послу видалення глухої області:

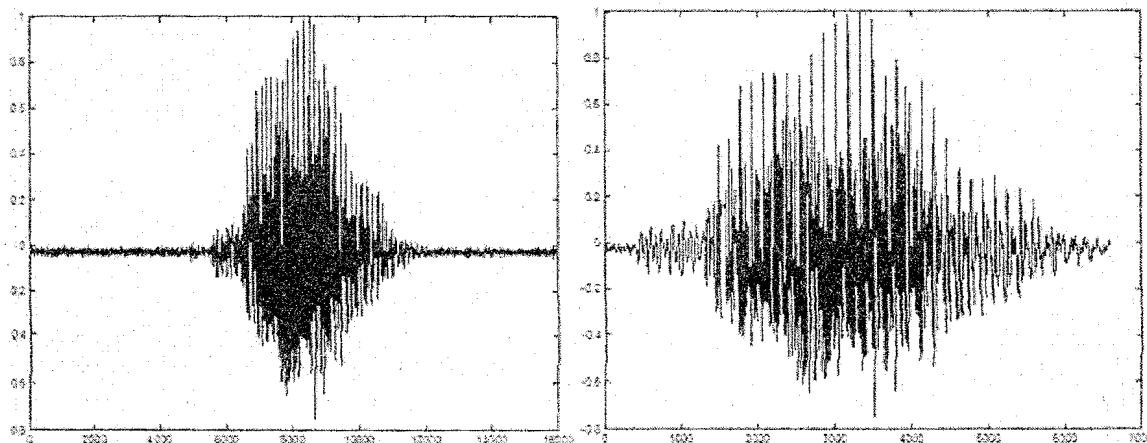


Рис. 3. Звуковий сигнал слова «pine»

Визначення слова може здійснюватися шляхом порівняння числових форм сигналів або шляхом порівняння спектрограми сигналів. Процес порівняння в обох випадках повинен компенсувати різні довжини послідовності й нелінійний характер звуку. DWT алгоритму вдається розібрати ці проблеми шляхом знаходження деформації, відповідної до оптимальної відстані між двома рядами різної довжини.

Існують дві особливості застосування алгоритму:

1. Пряме порівняння числових форм сигналів. У цьому випадку, для кожної числової послідовності створюється нова послідовність, розміри якої значно менше. Алгоритм має справу із цими послідовностями. Числова послідовність може мати кілька тисяч числових значень, у той час як підпослідовність може мати кілька сотень значень. Зменшення кількості числових значень може бути виконане шляхом їхнього видалення між кутовими крапками. Цей процес скорочення довжини числової послідовності не повинен змінювати своєї вистави. Безсумнівно, процес приводить до зменшення точності розпізнавання. Однак, беручи до уваги збільшення швидкості, точність, по суті, підвищується за рахунок збільшення слів у словнику.

2. Вистава сигналів спектрограм і застосування алгоритму DTW для порівняння двох спектрограм. Метод полягає в поділі цифрового сигналу на деяка кількість інтервалів, які будуть перекриватися. Для кожного імпульсу інтервали дійсних чисел (звукових частот), буде розраховувати Швидким перетворення Фур'є і буде зберігатися в матриці звукової спектрограми. Параметри будуть однаковими для всіх обчислювальних операцій: довжин імпульсу, довжини перетворення Фур'є, довжини перекриття для двох послідовних імпульсів. Перетворення Фур'є є симетрично пов'язаним із центром, а комплексні числа з однієї сторони пов'язані із числами з іншої сторони. У зв'язку із цим, тільки значення з першої частини симетрії можна зберегти. Таким чином, спектрограма буде представляти матрицю комплексних чисел, кількість рядів у такій матриці є рівній половині довжини перетворення Фур'є, а кількість стовпців буде визначатися залежно від довжини звуку. DTW буде застосовуватися на матриці речовинних чисел у результаті сполучення спектрограми значень, така матриця називається матрицею енергії.

### Висновки

DTW алгоритми є дуже корисними для розпізнавання окремих слів в обмеженому словнику. Для розпізнавання швидкої мови використовуються прикладні моделі Маркова. Використання динамічного програмування забезпечує подинамічну складність алгоритму:  $O(n^2v)$ , де  $n$  – довжина послідовності, а  $v$  кількість слів у словнику. DTW мають кілька слабких сторін. По-перше,  $O(n^2v)$  складність не дозволяє більшим словникам, які збільшують успішність процесу розпізнавання. По-друге, важко обчислити два елементи у двох різних послідовностях, якщо взяти до уваги, що існують безліч каналів з різними характеристиками. Проте, DTW залишається простим у реалізації алгоритмом, відкритим для поліпшень і підходящим для додатків, яким потрібне просте розпізнавання слів: телефони, автомобільні комп'ютери, системи безпеки і т.д.

### Література

- 1) *Cenni Marкова. StatSoft: [www.statsoft.ru/home/portal/taskboards/mark.htm](http://www.statsoft.ru/home/portal/taskboards/mark.htm)*
- 2) *Benoit Legrand, C.S. Chang, S.H. Ong, Soek-Ying Neo, Nallasivam Palanisamy. Chromosome classification using dynamic time warping, Sciencedirect Pattern Recognition Letters 29 (2008) 215–222*
- 3) *Методы математической статистики и моделирования в сравнительно-историческом языко-знании. Языкознание.ру: [yazykoznanie.ru/content/view/27/215/](http://yazykoznanie.ru/content/view/27/215/)*
- 4) *Cory Myers, Lawrence R. Rabiner, Aaron E. Rosenberg, Performance Tradeoffs in Dynamic Time Warping Algorithms for Isolated Word Recognition, Ieee Transactions On Acoustics, Speech, And Signal Processing, Vol. Assp-28, No. 6, December 1980*
- 5) *F. Jelinek. «Continuous Speech Recognition by Statistical Methods.» IEEE Proceedings 64:4(1976): 532-556*
- 6) *Rabiner, L. R., A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition, Proc. of IEEE, Feb. 1989*
- 7) *Куневич Я.Г., Кушнарев Д.А. Статистическая модель языка. Сборник работ 66-ой научной конференции студентов и аспирантов Белорусского государственного университета (т. 3). Май, 2009.*
- 8) *Rabiner, L. R., Schafer, R.W., Digital Processing of Speech Signals, Prentice Hall, 1978.*
- 9) *Stan Salvador, Chan, Fastdtw: Toward Accurate Dynamic Time Warping in Linear Time and Space, IEEE Transactions on Biomedical. Engineering, vol. 43, no. 4*
- 10) *Young, S., A Review of Large-Vocabulary Continuous Speech Recognition, IEEE Signal Processing Magazine, pp. 45-57, Sep. 1996*
- 11) *Sakoe, H. & S. Chiba. (1978) Dynamic programming algorithm optimization for spoken word recognition. IEEE, Trans. Acoustics, Speech, and Signal Proc., Vol. ASSP-26.*



- 12) Furtună, F.; Dârdală, M., *Using Discriminant Analysis in Speech Recognition, The Proceedings Of The Fourth National Conference Human Computer Interaction Rochi 2007, Universitatea Ovidius Constanța, 2007, Matrixrom, Bucharest, 2007*
- 13) *Speech Separation by Humans and Machines, Kluwer Academic Publishers, 2005.*
- 14) Холоденко А.Б. О построении статистических языковых моделей для систем распознавания текста. Интеллектуальные системы: [intsys.msu.ru/invest/speech/articles/rus\\_lm.htm](http://intsys.msu.ru/invest/speech/articles/rus_lm.htm)
- 15) Davies, K.H., Biddulph, R. and Balashek, S. (1952) *Automatic Speech Recognition of Spoken Digits, J. Acoust. Soc. Am.* 24(6) pp.637 — 642
- 16) Кофман А. Понятие нечеткого подмножества. В К. А., *Введение в теорию нечетких множеств. Москва: Радио и связь. 1982.*
- 17) Fink, G. A. *Markov Models for Pattern Recognition From Theory to Applications. Berlin Heidelberg: Springer-Verlag. 2008.*
- 18) Медведев М.С. Использование вейвлет-преобразования для построения моделей фонем русского языка. *Вестник КрасГУ. Серия физ.-мат. науки (9). 2006.*
- 19) *Now a Machine That Talks With the Voice of Man. (14 January 1939 r.). Science News Letter.*
- 20) Бабин Д.Н., Мазуренко И.Л... Холоденко А.Б. О перспективах создания системы автоматического распознавания слитной устной русской речи. *Интеллектуальные системы(т.8). 2004.*
- 21) Беседин И.Ю. Анализ проблем автоматического распознавания речи. *Вестник Ставропольского государственного университета (70). 2010.*
- 22) Галунов В.И., Соловьев А. *Современные проблемы в области распознавания речи. Информационные технологии и вычислительные системы, №2, 2004.*
- 23) Никитин А., Райков П. *Вопросно ответные системы.*
- 24) Протасов С.В. Вывод и оценка параметров действующей триграммной модели языка. *Труды международной конференции «Диалог 2008».*

Рецензент: Рибальский О.В.

Поступила 30.11.2011

УДК 004.621.

Піскун С.Ж., Хорошко В.О.  
ДУІКТ

## МЕТОД РАНЖИРУВАННЯ ПОКАЗНИКІВ ЯКОСТІ ФУНКЦІОНУВАННЯ СИСТЕМИ ЗАХИСТУ ІНФОРМАЦІЙНОГО СЕРЕДОВИЩА

### Вступ

Системи захисту інформаційної сфери (СЗІС) держави являють собою складні технічні системи. При вирішенні задач аналізу та синтезу СЗІС, прийняття рішення в умовах невизначеності функціонування систем використовуючи методи непарних множин для обробки інертної вихідної інформації. Експертне оцінювання багатовимірних об'єктів реального світу представляє важливу та актуальну проблему, особливо значущу для тих практично важливих випадків, коли окрема властивість, що визначає якість інформаційної сфери не піддається фізичним вимірюванням, а оцінюється експертами за їх значимістю і прояву. Цей напрямок є одним з пріоритетних у розвитку інтелектуальних систем в яких бази знань формуються шляхом обробки інформації отриманої від експертів-фахівців в заданій області предметної діяльності [1,2]. Так само ще одним важливим додатком експертного оцінювання багатовимірних об'єктів, яким є інформаційна сфера є скаляризацією векторних критеріїв у задачах агрегування і дезагрегування інформації в інтелектуальних системах [3,4,5], а також в задачах векторної оптимізації.