

ІНФОРМАЦІЙНІ ТЕХНОЛОГІЇ ТА КОМП'ЮТЕРНА ТЕХНІКА

УДК 621.39

О. М. Ткаченко, к. т. н., доц.;

О. Ф. Грійо Тукало, студ.

СПРЯМОВАНИЙ ПОШУК ПРИ КВАНТУВАННІ ЛІНІЙНИХ СПЕКТРАЛЬНИХ ЧАСТОТ

Розглянуто існуючі підходи до прискорення пошуку найближчого вектора у кодовій книзі. Запропоновано математичну модель і метод спрямованого пошуку з мажоризацією. Експериментально досліджено продуктивність запропонованого методу.

Вступ

Ущільнення мовленнєвих сигналів, що його застосовують у сучасних системах цифрового зв'язку, дозволяє підвищити ефективність використання смуги пропускання каналів і тим самим сприяє зменшенню цін на послуги мобільного та Інтернет-зв'язку. Останнім часом широкого розповсюдження набули методи ущільнення на основі квантування сигналів із використанням кодових книг. Скалярне квантування (СК) лінійних спектральних пар (LSF, line spectrum frequency) не потребує ані значних обчислювальних ресурсів, ані великих витрат пам'яті для своєї реалізації, проте останнім часом у стандартах ущільнення мови частіше використовують векторне квантування (ВК). Останнє дозволяє отримати менше спектральне спотворення, порівняно із СК, за умови кодування сигналу однаковою кількістю бітів. Цей вииграш, зумовлений тим, що ВК ефективніше використовує кореляцію між окремими складовими вектора параметрів. Перевага зростає із зростанням кількості компонентів, що квантується разом, тобто зі збільшенням розмірності вектора. Проте, разом з цим зростають витрати пам'яті, а також час на пошук кодового вектора у кодовій книзі, що зумовлює суттєві обмеження на практичне застосування ВК у реальному масштабі часу. Тому в чинних стандартах застосовують субоптимальне кодування, коли 10-вимірний вектор параметрів розбивають на 2 або більше підвекторів меншої розмірності [1].

З метою зменшення часу пошуку в [2] було запропоновано декілька підходів до впорядкування векторів у кодовій книзі, названих авторами методами швидкого векторного квантування (fast vector quantization methods). Показано, що складність обчислень при застосуванні наведених методів складає порядка 25 % від складності обчислень при повному пошуку без суттєвої втрати продуктивності, що оцінювалася за спектральним спотворенням.

В [3] розроблено метод структуризації кодових книг на основі відношення мажорювання. Це дозволило зменшити складність обчислень до 16—20 % від відповідної складності при повному пошуку.

Метод «сусіднього спуску» (neighbor descent), що його запропоновано в [4], дозволяє зменшити час пошуку до 2 % від часу повного пошуку в описаному прикладі. Проте цей високий показник було досягнуто при істотному збільшенні середнього спектрального спотворення (на 0,22 дБ), що може виявитися непридатним для практичного застосування.

У статті пропонується метод, що поєднує в собі переваги методів, описаних у [3] та [4]. Заздалегідь підготовлений список сусідів для кожного кластера надалі впорядковується згідно відношення мажорювання. Це дозволяє зменшити час пошуку найближчого вектора у кодовій книзі до 2,5 %

від часу повного пошуку. При цьому спектральне спотворення в розглянутому прикладі не збільшилося і склало 0,854 дБ.

Метою роботи є зменшення складності обчислень при кодуванні мовленнєвих сигналів за рахунок спрямованого пошуку найближчого вектора у кодовій книзі.

Для досягнення поставленої мети необхідно розв'язати такі задачі:

- розробити модель і структуру даних, що зберігаються у кодових книгах;
- розробити метод пошуку найближчого вектора у створеній кодовій книзі;
- оцінити ефективність розробленого методу.

Розробка математичної моделі та методу спрямованого пошуку

Нехай кодова книга містить кінцеву множину векторів $Q = \{Y_1, Y_2, \dots, Y_N\}$, $Y_i = (y_{i1}, y_{i2}, \dots, y_{iM})$. Таким чином, з кожним вектором Y_j у кодовій книзі пов'язаний індекс, або кодове слово j , що може бути записано як N -розрядне ціле число. На вхід квантизатора поступає вектор $X = (x_1, x_2, \dots, x_M)$. В результаті кодування необхідно вибрати таке кодове слово j , що мінімізує спотворення $d(X, Y_j)$ (правило вибору найближчого сусіднього вектора). У цій роботі для вимірювання спотворення використовувалася Евклідова відстань:

$$d^2(X, Y_i) = \sum_{k=1}^M (x_k - y_{ik})^2. \quad (1)$$

Множина вхідних векторів X , що кодуються одним і тим самим індексом j , утворюють регіон Вороного:

$$v_j = \{X : d(X, Y_j) \leq d(X, Y_i); \forall i \in I\}, \quad (2)$$

де $I = \{1, 2, \dots, N\}$ — множина індексів. Альтернативним варіантом завдання регіону Вороного є

$$v_j = \{X : d(X, Y_j) \leq d(X, Y_i); \forall i \in A_j\}, \quad (3)$$

де $A_j = \{i : v_i \cap v_j \neq \emptyset\}$ — множина кодових слів, у яких регіони Вороного є суміжними (adjacent) з цим регіоном.

На рис. 1 показано графічну інтерпретацію регіонів Вороного для двовимірного випадку.

Найпростішим методом пошуку у кодовій книзі вектора, найближчого до X , є повний пошук. При цьому для кодової книги розміром N необхідно N разів обчислити відстань згідно (1), що у багатьох випадках є непридатним для практичного застосування. З іншого боку, при такому підході виконується багато зайвих обчислень, оскільки ніяк не враховуються результати попередніх вимірювань відстаней. У роботі [4] запропоновано метод «сусіднього спуску» (neighbor descent), який базується на використанні таблиць суміжності A_j , підготовлених заздалегідь для кожного регіону v_j .

Основна ідея цього методу полягає у тому, що результати обчислення відстаней використовуються для переходів до суміжних регіонів у напрямку оптимуму. Перехід від кодового слова j до кодового слова l відбувається за умови

$$d(X, Y_l) \leq d(X, Y_j); \quad l \in A_j. \quad (4)$$

Такий підхід для двовимірного випадку проілюстровано на рис. 2.

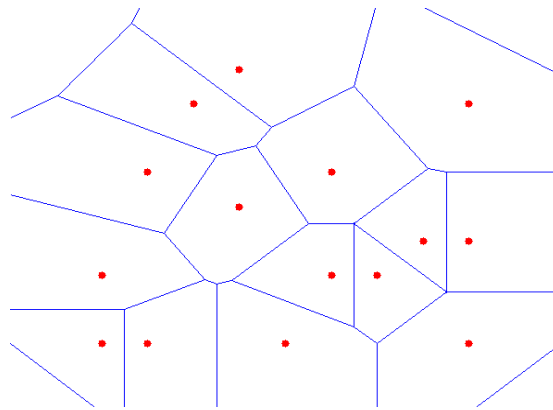


Рис. 1. Приклад діаграми Вороного для двовимірного випадку

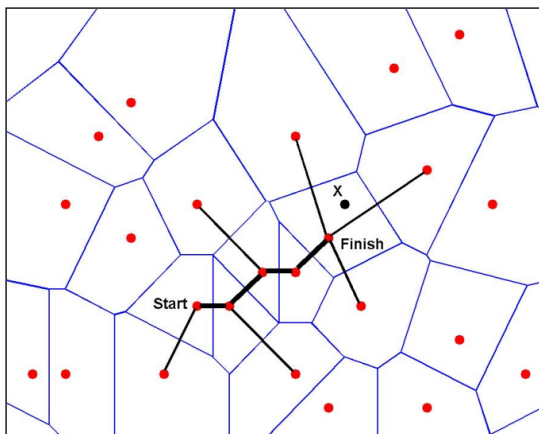


Рис. 2. Пошук найближчого вектора за методом «сусіднього спуску»

Розглядалися дві основні модифікації методу «сусіднього спуску». У першій з них (full neighbor descent, FND) спочатку обчислювалася відстань від вхідного вектора до усіх суміжних кодових слів для даного регіону Вороного, після чого приймалося рішення про перехід до кодового слова з мінімальною відстанню або про завершення пошуку (повний обхід сусідів):

$$d(X, Y_l) \leq d(X, Y_i); \quad \forall i \in A_j, \quad l \in A_j. \quad (5)$$

Друга модифікація (random neighbor descent, RND) відрізнялася тим, що сусідні кодові слова перевірялися у випадковому порядку і перехід до сусіднього кодового слова відбувався одразу, як тільки виконувалася умова (4) (випадковий обхід сусідів). Порівняння цих підходів проводилося для кодових книг, що містили $N = 4096$ векторів розмірністю $M = 5$. При цьому вищу продуктивність було отримано при випадковому обході сусідів [4]. В обох випадках за початкову гіпотезу обиралося вихідне кодове слово, знайдене для попереднього фрейму, що дозволяло використовувати кореляцію між сусідніми фрагментами мовленнєвого сигналу.

Наведені в [4] результати дозволяють зробити припущення, що ефективність пошуку буде тим вищою, чим менше часу витрачається на пошук сусіда для переходу згідно умови (4) для кожного кодового слова j . Фактично на цьому етапі виконується витратний лінійний пошук, і його негативний вплив на загальні результати буде істотнішим зі зростанням кількості сусідів. Так, для вищезгаданої кодової книги кількість сусідів коливається у межах 40...160 і складає у середньому 70 для кожного регіону. За таких умов лінійний пошук є вже доволі неефективним. Випадковий обхід сусідів зменшує час лінійного пошуку і тим самим підвищує загальну продуктивність.

Ефективність такого підходу суттєво залежить від того, у якому порядку буде здійснюватися обхід сусідніх кодових слів. При цьому логічно зробити припущення, що впорядкувавши обхід векторів для кожного регіону можна зменшити час лінійного пошуку і тим самим досягти подальшого зменшення складності обчислень.

Один з можливих підходів до впорядкування кодових книг базується на використанні теорії мажоризації [5], застосування якого для прискорення пошуку найближчого вектора описано в [3]. Нехай задано вектори $Y = (y_1, y_2, \dots, y_n)$ та $Y' = (y'_1, y'_2, \dots, y'_n)$, $Y, Y' \in Q$, компоненти яких впорядковано за незростанням. Говорять, що Y мажорується Y' або Y' мажорує Y (позначається $Y \prec Y'$), якщо виконується [5]:

$$\sum_{i=1}^k y_i \leq \sum_{i=1}^k y'_i, \quad k = 1, 2, \dots, M-1; \quad (6)$$

$$\sum_{i=1}^M y_i = \sum_{i=1}^M y'_i.$$

Впорядкування, що задаються відношенням мажорювання (6), називають мажоризацією. Якщо виконується нерівність

$$\sum_{i=1}^k y_i \leq \sum_{i=1}^k y'_i, \quad k = 1, 2, \dots, M, \quad (7)$$

говорять, що Y слабо мажоруюється Y' або Y' слабо мажоруює Y (позначається $Y \prec_w Y'$).

У роботі [3] показано, як можна застосувати впорядкування, що задається відношенням мажоруювання (7), для побудови структурованих векторних кодових книг. Автори використали (7) для впорядкування сусідів кожного кодового слова j . При цьому таблицю суміжності A_j необхідно розбити на окремі класи, що формуються відповідно до рівнів мажоризації. Рівні мажоризації формуються за таким правилом. Будемо вважати, що рівень мажоризації L_i мажоруюється рівнем мажоризації L_k , якщо для кожного вектора Y , що належить L_i , на рівні L_k знайдеться вектор Y' , що слабо мажоруює Y , або формально

$$\forall Y, Y \in L_i, \quad \exists Y', Y' \in L_k, \quad Y \prec_w Y' \implies L_i \prec L_k; \quad i, k \in A_j \quad (8)$$

Розбиття таблиці суміжності A_j на класи згідно із заданим критерієм (8) будемо називати структуризацією. Структуризація має на меті скорочення обчислювальних витрат на пошук для переходу від кодового слова j до кодового слова l згідно (4).

Спрямований пошук у рамках запропонованого підходу буде відбуватися таким чином.

1. Спочатку визначається, на який рівень мажоризації потрапляє вхідний вектор X .
2. Після цього, починаючи з цього рівня L_k , відбувається пошук вектора Y_l для переходу до кодового слова l згідно (4).

3. Якщо такий вектор не знаходиться на рівні L_k , пошук продовжується на сусідніх рівнях $L_{k-1}, L_{k+1}, L_{k-2}, L_{k+2}, \dots$.

4. Якщо для жодного вектора $Y_i, i \in A_j$ (4) не виконується, пошук завершується і вектор Y_j вважається найближчим до вхідного вектора X . Інакше відбувається перехід, і пункти 1—4 повторюються для кодового слова l .

Зазначимо, що пошук рівня мажоризації для вхідного вектора X не потребує значних витрат часу і зводиться до простої процедури перевірки виконання (7). Так само, як в [4], за початкову гіпотезу обиралося вихідне кодове слово, знайдене для попереднього фрейму. Запропонований підхід фактично є модифікацією методу «сусіднього спуску», проте ми назвали його методом спрямованого пошуку з мажоризацією (СПМ), що, на наш погляд, точніше відповідає суті методу.

Експериментальне дослідження методу спрямованого пошуку

З метою експериментальної перевірки запропонованого підходу в середовищі Visual Studio було розроблено програмне забезпечення для побудови таблиць суміжності та спрямованого пошуку найближчого вектора. Побудова векторних кодових книг виконувалася за умов, описаних у [6]. Для того, щоб забезпечити можливість порівняння з результатами, наведеними в [4], 10-вимірний вектор LSF-параметрів було розбито на дві частини 5×5 , кожна кодова книга містила по 4096 підвекторів. Таким чином, обсяг даних, необхідних для представлення короткочасної спектральної інформації, становив 24 біти.

Для тестування було відібрано текст, який містив уривок з роману І. Нечуй-Левицького і складався з 20000 векторів. Оцінювання результатів проводилося за відносною кількістю пропущених при швидкому пошуку векторів у кодовій книзі, що були найближчими до вхідного, а також за сумарним спектральним спотворенням SD [1, 6].

У таблиці 1 наведено порівняння методу СПМ з FND і RND. Як видно з таблиці, метод СПМ має перевагу за рахунок суттєвого скорочення середньої кількості обчислень відстаней (КОВ) для кожного регіону n при переході до сусіднього регіону.

КОВ та глибина пошуку для різних методів

Назва методу	КОВ C		Глибина пошуку Tr		Середня КОВ для кожного регіону \bar{n}
	Середня	Максимальна	Середня	Максимальна	
FND	140,76	432	2,47	7	56,99
RND	104,46	270	4,94	19	21,15
СПМ	89,26	214	4,60	20	19,40

На рис. 3 показана залежність кількості знайдених найближчих векторів N (у відсотках) від глибини пошуку. Глибиною пошуку будемо називати кількість переходів між сусідніми регіонами, необхідних для досягнення оптимуму. Як можна побачити з рисунку, глибина пошуку при RND і СПМ дещо зростає. Тим не менш середня складність обчислень \bar{C} , яка визначається кількістю обчислень відстаней до векторів у кодовій книзі зменшується:

$$\bar{C} = \bar{n} \cdot \bar{Tr},$$

де \bar{n} — середня кількість обчислень відстаней для кожного регіону; \bar{Tr} — середня кількість переходів між сусідами.

Як впливає з таблиці 1, максимальна та середня КОВ суттєво відрізняються. Якщо для оцінювання складності обчислень орієнтуватися на значення середньої, а не максимальної КОВ, можна зменшити час, що виділяється на пошук, у 2,5—3 рази. Проте середня КОВ не є інформативною характеристикою для ущільнення мовленнєвих сигналів у реальному часі, оскільки час, заощаджений на обчислення в одному фреймі, неможливо використати в іншому. Тому складність обчислень має бути константою для різних фреймів, що визначається часовим інтервалом (вікном), відведеним на пошук найближчого вектора у кодовій книзі. Логічно припустити, що цей інтервал є величиною, пропорційною КОВ, і оцінити для різних вікон кількість пропущених векторів і спектральне спотворення. Результати дослідження наведені у таблиці 2 та зображені на рис. 4.

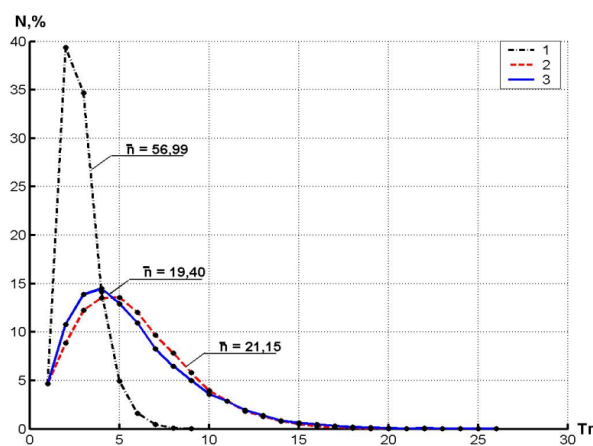


Рис. 3. Залежність кількості знайдених векторів від глибини пошуку:

1 — FND, 2 — RND, 3 — СПМ

Таблиця 2

Кількість пропущених векторів і спектральне спотворення для різних вікон

КОВ	Відсоток пропущених векторів MV		Спектральне спотворення SD , дБ	
	RND	СПМ	RND	СПМ
40	64,50 %	25,93 %	1,028	0,915
60	42,60 %	8,43 %	0,940	0,866
80	23,08 %	1,88 %	0,893	0,857
100	10,88 %	0,25 %	0,871	0,854
120	4,73 %	0,05 %	0,862	0,854
140	1,68 %	0,00 %	0,857	0,854

Як можна побачити, цілком придатні для практики результати досягаються для вікон, набагато менших за максимальні значення КОВ, наведені у таблиці 1. Це пояснюється тим, що значна КОВ припадає на перевірку останнього регіону v_j , коли фактично найближчий вектор вже знайдено і перехід не здійснюється. Зазначимо також, що метод СПМ забезпечує зменшення кількості пропущених векторів і спектрального спотворення порівняно з RND для вікон однакового розміру.

Висновки

Метод спрямованого пошуку з мажоризацією дозволяє зменшити час пошуку найближчого вектора у кодовій книзі до 2,5 % від часу повного пошуку без втрати продуктивності, що оцінювалася за спектральним спотворенням. Також він забезпечує зменшення часу пошуку до 70 %, порівняно з методом RND. Це досягається за рахунок значних обчислювальних витрат на підготовчому етапі, необхідних для створення таблиць суміжності A_j , а також збільшення обсягів пам'яті, потрібної для їх зберігання.

СПИСОК ЛІТЕРАТУРИ

1. Paliwal K. K. Efficient vector quantization of LPC parameters at 24 bits/frame / K. K. Paliwal, B. S. Atal. — IEEE Transaction on Speech and Audio Processing, 1993. — No. 2, vol. 1. — P. 3—14.
2. Zhou J. Simple Fast Vector Quantization of the Line Spectral Frequencies / J. Zhou, Y. Shoham, A. Akansu // Image Compression and Encryption Technologies. — 2001. — Vol. 4551. — P. 274—282.
3. Біліченко Н. О. Швидкий пошук при векторному квантуванні лінійних спектральних частот / Н. О. Біліченко, О. М. Ткаченко, О. Д. Феферман, С. В. Хрущак // Реєстрація, зберігання і обробка даних. — 2008. — № 2, т. 10. — С. 37—47.
4. Agrell E. Spectral coding by fast vector quantization / E. Agrell // Proc. IEEE Workshop on Speech Coding for Telecommunications, Sainte-Adèle, Québec, Canada. — 1993. — P. 61—62.
5. Маршалл А. Неравенства: теория мажоризации и ее приложения : пер. с англ. / А. Маршалл, И. Олкин. — М. : Мир, 1983. — 576 с.
6. Біліченко Н. О. LSF-вокодер на основі векторного квантування / Н. О. Біліченко, О. М. Ткаченко, О. Д. Феферман, С. В. Хрущак // Реєстрація, зберігання і обробка даних. — 2007. — № 1, т. 9. — С. 35—41.

Рекомендована кафедрою обчислювальної техніки

Надійшла до редакції 2.07.09
Рекомендована до друку 6.10.09

Ткаченко Олександр Миколайович — доцент кафедри обчислювальної техніки, **Грійо Тукало Оксана Франсисківна** — студентка Інституту інформаційних технологій та комп'ютерної інженерії.

Вінницький національний технічний університет

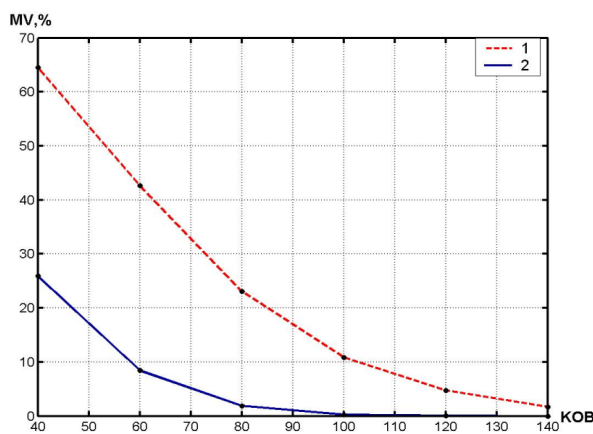


Рис. 4. Відсоток пропущених векторів для різних вікон: 1 — RND, 2 — СПМ