Signaturové metody v textových databází

Jaroslav Pokorný
KSI MFF UK
Malostranské nám. 25 118 00 Praha 1

2. dubna 1997

Úvod

Idea signaturových metod není nikterak nová - objevuje se např. v článku Harrisona v CACM v r.1971, ale i dříve (Bourne 1963). Teprve nyní, v souvislosti s velkými databázemi na vhodných médiích nacházejí tyto metody výhodné uplatnění. Článek je dostupný v širší verzi jako kap. 4 ve skriptech J. Pokorný, I. Halaška - Databázové systémy: vybrané kapitoly a cvičení, UK Praha, Karolinum, 1994.

Signaturové metody organizace souborů používají jistý přídavný prostor, obvykle však ne větší než 10 - 20 souboru, a zároveň poskytují možnost rychlého vyhledávání požadovaných záznamů. V případě požadavků zadaných hodnotami klíčů více atributů dokonce předčí indexové metody.

Signaturové metody lze rozdělit do dvou skupin:

- vrstvené kódování
- řetězené kódování

Někdy se rozlišuje jemnější dělení podle toho, zdali jsou signatury komprimovány či ne.

Uvažujme databázi textů obsahující jednotlivé texty rozdělené do záznamů (logických bloků, stránek apod.). Princip metod obou skupin je stejný - ke každému záznamu přiřadíme signaturu, tj. jistý řetězec bitů, který vznikne transformací (kódováním, hašováním) termů záznamu, a který je podstatně menší než původní záznam. Každá signatura tedy obsahuje zakódovanou informaci obsaženou v původním záznamu. Soubor takto vzniklých záznamů - signatur se pak nazývá soubor signatur (někdy též soubor deskriptorů).

Soubor signatur má jednoduchou údržbu - vložení nového záznamu do primárního souboru (na konec) znamená připojení další signatury záznamu do souboru signatur (na konec), odstranění primárního záznamu vede k odstranění signatury ze souboru signatur, změna záznamu primárního souboru implikuje změnu odpovídajícího záznamu v souboru signatur. Ne vždy je vhodné vytvářet vazbu mezi souborem signatur a primárním souborem v závislosti na sekvenčním uspořádání, tj. fyzickém pořadí záznamů, v obou souborech, jak by vyplývalo z právě popsané aktualizace. V praxi se u každé signatury v souboru signatur připojuje jednoznačný identifikátor příslušného záznamu. Ten může být např. jeho absolutní adresou, nebo indexem do tabulky skutečných adres, případně může tvořit vstup do dynamicky řízené oblasti primárních záznamů. V aplikacích na delší dokumenty, kdy jeden dokument je rozdělen do bloků a odpovídá mu tedy více signatur, lze např. použít jeden bit identifikátoru k rozlišení, zdali jde o týž dokument nebo už o další.

Zaměříme se pouze na vrstvené kódování, které má pro textové databáze vhodnější využití, než varianta řetězeného kódování. V čl. 1 je popsána jeho základní varianta, čl. 2 se zabývá paměťovou složitostí metody, čl. 3 nabízí některá řešení implementačních otázek signatur, které vedly ke vzniku různých variací původní metody. Odstavec 3.1 ukazuje metodu transpozice jako důležitý přístup k implementaci, který podstatně redukuje časovou složitost základní varianty. Odst. 3.2 rozebírá nejznámější a také neúčinnější variantu - dvouúrovňový soubor signatur. Odst. 3.3 popisuje metody rozdělování signatur.

1 Vrstvené kódování - základní varianta

Uvažujme záznam obsahující n pro vyhledávání významných termů (klíčových slov) $t_1, \ldots, t_n$. Termy budeme (bez další diskuse) uvažovat jako víceslovné. Nejprve vytvoříme binární reprezentace jednotlivých t_i, $1 \leq i \leq n$, a ty transformujeme do posloupnosti "0" a "1" dané délky a to tak, že počet "1" v této posloupnosti bude pevně daný. Získané bitové řetězce se nazývají binární kódová slova. Počet 1-bitů v každém z nich tvoří váhu binárního kódového slova.

Signaturu záznamu získáme provedením logické operace OR na n binárních kódových slov vytvořených z n termů záznamu (obr. 1). Připojením jednoznačného identifikátoru obdržíme jeden záznam do souboru signatur.

Příklad 1

Mějme záznam

`(1992, Johannes J. Martin, Data types and data structures).`

Nejprve sestrojíme binární kódová slova (s vahou 3):

`1992`	`10001001`
`Johannes J. Martin`	`01011000`
`Data Types and Data Structures`	`10010001`
	`11011001*000010`

Ve výsledné signatuře záznamu "*" je oddělovačem mezi signaturou a identifikátorem (2. záznam).

Vyhledávání v organizaci se souborem signatur znamená k danému dotazu sestrojit nejprve tzv. signaturu dotazu. Pro zadané termy (jde vlastně o konjunktivní dotaz) sestrojíme opět binární kódová slova, "navrstvíme" je použitím logické operace OR a procházíme postupně celý soubor signatur. U každé signatury záznamu zjišťujeme porovnáním, zdali každý 1-bit v dotazu je také na odpovídající pozici v signatuře záznamu. Jestliže ano, pak přes identifikátor u dané signatury záznamu můžeme získat (ale také nemusíme) hledaný záznam (tzv. hit). Že záznam takto získaný nemusí být hitem je pochopitelné, protože jednak nemusí být transformační funkce vytvářející binární kódová slova jednoznačné, jednak se pomocí OR mohou do signatury záznamu dostat 1-bity z různých binárních kódových slov. V dalším textu nazýváme tyto záznamy chybné hity. Tento fakt dává také motivaci pro existenci váhy binárního kódového slova - je totiž třeba nějak omezit pravděpodobnost velkého množství jedniček v signatuře záznamu. Jestli je ovšem hit relevantní, závisí plně na soudu uživatele.

Příklad 2

Mějme požadavek na záznamy o knihách Johannese. J. Martina. Signatura dotazu bude `01011000`. V případě signatury záznamu `11011001` postupujeme do primárního souboru, pro signaturu záznamu např. `01010110` víme s jistotou, že odpovídající záznam není hitem.

2 Složitost vrstveného kódování

2.1 Velikost souboru signatur

Základním tématem k řešení efektivnosti signaturové organizace souboru je zřejmě velikost souboru signatur. Předpokládejme nyní, že nenastávají žádné kolize při použití transformací hodnot atributů na binární reprezentaci, resp. binární kódové slovo (ideální případ). Pro vrstvené kódování odvodil v r.1979 Roberts formuli pro optimální délku signatury záznamu d (v bitech) v závislosti na:
n ... počet termů záznamu,
Q ... průměrný počet termů specifikovaných v dotazu,
N ... počet záznamů v primárním souboru,
F ... předpokládaný průměrný počet záznamů, které nejsou hity, získaných v jednom dotazu.

Parametr F se v některých pracích také definuje vztaženě k celkovému množství záznamů N, tj. jako pravděpodobnost, že signatura byla vybrána a odpovídající záznam je chybným hitem. Označíme-li tuto pravděpodobnost F', pak F podle původní definice je rovno $N * F$'. Obě míry jsou důležité z uživatelského hlediska a jsou výhodné zejména proto, že nezávisí na hardwarové situaci, správě bufferů apod.

Formule pro výpočet velikosti signatury má tvar

$$d = \lceil \frac{n}{Q} * \frac{ln(N) - ln(F)}{ln(2)^2} \rceil \tag{1}$$

kde $ln(x)$ znamená logaritmus x o základu e. Tato délka signatury má speciální vlastnost. Odpovídající signatury jsou optimální vzhledem k F, když počet 0- a 1-bitů bude v signatuře zhruba stejný, tj. půl na půl. Odpovídající váha binárního kódového slova k je svázána s délkou signatury d přibližným vztahem

$$k\tilde{}d * \frac{ln(2)}{n} \tag{2}$$

Pro váhu k tedy získáme vzorec

$$d = \lceil \frac{1}{Q} * \frac{ln(N) - ln(F)}{ln(2)} \rceil \tag{3}$$

Při konstrukci d a k se často uplatní případ, kdy převládají dotazy na jeden term, tj. $Q = 1$.

Protože k adresování N záznamů potřebujeme $|log_2 N|$ bitů, je velikost souboru signatur pro vrstvené kódování dána výrazem

$$S = \lceil (d + |log_2 N|) * N \rceil \tag{4}$$

Příklad 3

Abychom získali představu o velikosti d a k, uvažujme situaci, kdy $N = 524$ 288 (tj. 219 záznamů), $F = 4$, n = 8 a $Q = 1$. Pak velikost signatury je 205 bitů a k=17. Jsou-li stránky velikosti B = 1024 byte, obdržíme pro soubor signatur (bez ukazatelů do primárního souboru) velikost S = 13 120 stránek. (Mají-li záznamy primárního souboru pevnou délku, lze pozici záznamu a tedy i jeho adresu v primárním souboru vypočítat.)

V praktických případech ovšem neplatí předpoklad o kolizích (neznáme např. velikost dom(A***i)) a binární reprezentace či binární kódové slovo se volí nejednoznačné. Se zvětšením signatury sice zmenšíme počet kolizí (a tím i F), ovšem současně zvětšíme i soubor signatur.

2.2 Další vlastnosti signaturových metod

Nevýhodou signatur je množství času potřebné pro jejich vytvoření a množství prostoru pro jejich uložení. Režie v čase se vyplatí v případě častého přístupu k výslednému souboru s architekturou používající technologii CD.

U vrstveného kódování se ztrácí závislost signatur na pořadí atributů, což může mít vliv na aplikaci metody.

Signatury a jejich údržbu lze provádět na pevném disku předsazeném optickému disku, kde je uložen primární soubor.

Existují i jiné možnosti. Ve víceúrovňové architektuře hardware by bylo možné zpracovávat jednotlivé požadavky paralelně na procesorech se soubory signatur na vnějších pamětech, výsledky (identifikátory) kumulovat v nadřazeném procesoru a odtud z fronty požadavků provádět vyhodnocování na bázi primárních souborů uložených na optickém disku.

Často je slyšet otázku, jak generovat binární kódová slova. V implementaci systému TLF [BK90] byla použita následující funkce pro výpočet binárních kódových slov:

1. binární reprezentace T termu t vznikne zřetězením binárních reprezentací znaků slova t,

2. je-li f generátor náhodných čísel vracející hodnoty mezi 0 a 1, označme $f_1, f_2, \ldots$ posloupnost hodnot vzniklých pomocí f, aplikujeme-li f na T,

3. generujeme k' hodnot této posloupnosti, kde $k' \geq k$ je nejmenší takové číslo, pro které množina $\{\lceil f_1 * d\rceil, \lceil f_2 * d\rceil, \ldots, \lceil f_{k'} * d\rceil\}$ konstruuje k různých hodnot. Pak tato množina dává hodnoty indexů, které v binárním kódovém slově délky d nastaví odpovídající bity na 1.

Při hledání v textech je velmi výhodné zadávat v dotazu (zvláště v češtině) pouze části termů (např. "dat*", "strukt*"), nebo obecněji podřetězce textu. Tuto alternativu základní verze signaturových metod neumožňuje. Zobrazit informace o částech termů do signatury znamená rozdělit term na překrývající se m-tice a každou vhodně zakódovat. Např. slovo " datové " pro m=3 dává trojice " da", "ato", "tov", "ové", "vé ". Každá z nich může např. jedním bitem přispívat do výsledné signatury. Tento postup ovšem nevychází z myšlenky pevně stanovené váhy k binárního slova délky d. I když lze situaci chápat tak, že k=1, analýza z odst. 2.1 se na délky d a k nevztahuje.

3 Implementace souboru signatur

Popíšeme několik technik implementace vrstveného kódování, které umožňují redukovat prohledávání souboru signatur. Původní sekvenční schéma je sice vhodné z hlediska aktualizací souboru, pro velké databáze je však čtení příliš pomalé. Výhodné může být pouze pro dávkové zpracování dotazů. Roberts popisuje techniku ukládání souboru signatur jako transponovaného souboru (odst. 3.1). V odst. 3.2 je popsáno dvouúrovňové schéma v několika variantách. V odst. ?? je uvedena další metoda, která kombinuje stromy s invertovanými seznamy. Konečně v odst. 3.3 je pro organizaci signatur využito sekvenčních struktur se společným prefixem.

3.1 Signatury jako transponovaný soubor

Nechť soubor signatur má N záznamů. Pak lze soubor N záznamů délky d bitů chápat jako **transponovaný soubor**, tj. soubor d řetězců délky N. Obsahuje-li dotaz w 1-bitů, pak lze z d řetězců kontrolovat pouze w, tj. celkově $w.N$ bitů místo $d.N$ bitů (většinou platí $w \ll d$, pro dotaz na jeden atribut dokonce $w = k$).

Příklad 4

Navážeme na příklad 4.5, kde $k = 17$, $d = 205$. Pak místo 13 120 stránek budeme v dotazu s $Q = 1$ potřebovat kontrolovat celkový objem 1088 stránek, tj. 8,3

Kontrolovat, znamená spojit w řetězců operací AND v jeden. 1-bity, které obsahuje tento řetězec, budou odpovídat záznamům, které mohou být relevantní.

Nevýhodou dané metody implementace je, že aktualizace záznamu v souboru signatur je distribuována do d řetězců, tj. vyžaduje se více přístupů na disk. Další nevýhodou je velký počet přístupů na disk jako takový. Např. pro k=4 a velikost stránky B = 1 Kbyte, N = 500 000 je počet přístupů nejméně $k * \lceil N/B \rceil$, tj. 248.

3.2 Dvouúrovňový soubor signatur

Na soubor signatur se díváme jako na soubor N_S segmentů po N_R záznamech délky d_R (obr. 3). Zřejmě $N = N_S * N_R$. Pro každý segment vytvoříme signaturu délky d_S. V literatuře se objevují dva přístupy k tvorbě signatury 2. úrovně: signatura vznikne aplikací operace OR na signatury segmentu (tato metoda je speciálním případem tzv. souboru indexovaných deskriptorů - obr. 2) a tedy má délku rovnou d_R, nebo se signatura segmentu tvoří analogicky jako signatura záznamu, tj. použijí se pro ni všechny termy ze segmentu, tj. nS = $N_R * n_R$, a tedy d_S je obecně větší než d_R. Věnujme se na chvíli prvnímu přístupu. Závažným problémem, se kterým se setkáme, je velká hustota 1-bitů v signatuře 2. úrovně. Předpokládejme, že d_R je pravděpodobnost, že bit signatury v 1. úrovni je 1-bitem, tj. vzhledem ke konstrukci signatury podle formulí z odst. 2.1 je $d_R = 1/2$. Pravděpodobnost d_S, že bit signatury v 2. úrovni je 1-bitem, se spočítá z pravděpodobnosti, kdy bit této signatury je 0-bitem. To nastane tehdy, když tento bit je u každé signatury s daném segmentu roven 0. Pravděpodobnost tohoto jevu je $(1 - d_R)^{N_R}$. Tedy $d_S = 1 - (1 - d_R)^{N_R}$ tj. Např. pro $N_R = 4$ a $d_R = 1/2$ vyjde $d_S = 0{,}938$, což je příliš velká hustota 1-bitů, než aby mohla při dotazu dostatečně vytřídit v 2. úrovni co nejvíce signatur.

Druhý problém je kombinatorický. Díky vrstvení reprezentuje signatura 2. úrovně zakódovanou informaci ne pouze o N_R záznamech, ale všech těch, které získáme jako jistý kartézský součin všech hodnot participujících v jednotlivých atributech. To může vést k tomu, že od mnoha dotazů se sice dostaneme přes 2. úroveň, avšak na 1. úrovni nedospějeme k žádnému hitu.

3.2.1 Větší signatury 2. úrovně

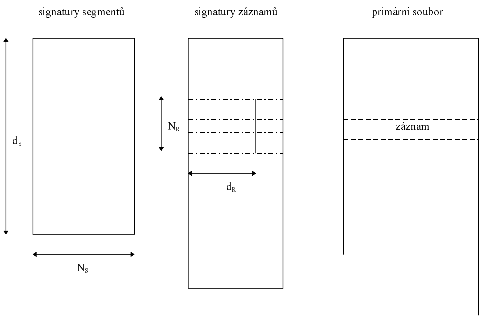
Optimální velikosti signatur a počet jedniček v nich lze zase získat ze vztahů popsaných v odst. 2.6. Vedle již daných parametrů budeme používat F_R, resp. F_S označující předpokládaný průměrný počet záznamů, které jsou chybné hity, získaných v dotazu na segment, resp. na záznam primárního souboru v segmentu. Nechť n_R označuje počet termů v primárních záznamech, tj. N_R, n_R je počet termů v segmentu. Pak pro $Q = 1$ je

$$d_R = \lceil n_R * \frac{ln(N_R) - ln(F_R)}{ln(2)^2} \rceil$$

$$k_R = \lceil \frac{ln(N_R) - ln(F_R)}{ln(2)} \rceil$$

$$d_S = \lceil N_R * n_R \frac{ln(N_S) - ln(F_S)}{ln(2)^2} \rceil$$

$$k_S = \lceil \frac{ln(N_S) - ln(F_S)}{ln(2)} \rceil$$



(5)

Příklad 5

Uvažujme N_S=64, $N_R = 8192$, $F_R = 1$, F_S=4. Pak z daných rovnic vyplývá, že $k_R = 13$, $d_R = 150$,$d_S = 756\ 000$, $k_S = 8$.

Může překvapit velikost signatury segmentu d_S. Použije- li se však transponovaného uložení pro soubor segmentů, počet přístupů na disk nebude příliš velký (viz předchozí obrázek).

Celková přídavná režie paměti potřebná pro soubor signatur je tedy

$$N_S d_S + N_S N_R d_R = N * n_R \frac{ln(N) - ln(F * F_S)}{ln(2)^2} \tag{6}$$

Vidíme, že celkový prostor je nezávislý na volbě velikosti segmentu. Také d_R nezávisí na N, nýbrž na N_R, což vede k jeho menší velikosti než v jednorozměrném případě. Celkově je paměťová režie dvouúrovňového souboru signatur podobná režii v jednorozměrném případě.

3.3 Rozdělování souboru signatur

Myšlenka rozdělení souboru signatur do skupin připomíná poněkud hašování. V každé skupině budou signatury se stejným klíčem, což je jistý podřetězec signatury, který je stejný ve všech signaturách skupiny. Stejně se získá klíč ze signatury dotazu. Na rozdíl od hašování, kde bychom testovali rovnost klíče z dotazu s klíčem v tabulce, zde stačí zjistit, zdali 1-bit v klíči dotazu je také 1-bitem v klíči v tabulce klíčů. Od těchto klíčů se již dostaneme k signaturám příslušné skupiny. Metoda je implementací vrstveného kódování. Architektura souborů v organizaci vrstveného kódování se souborem signatur rozděleným do skupin má tvar daný na obr.4.

Uchováváme-li v souboru signatur pouze (neklíčovou) část signatury, je zřejmé, že v paměťové složitosti bude tato metoda lepší než jednoúrovňový soubor signatur (klíč je uložen jen jednou), a tedy o mnoho lepší než víceúrovňový soubor signatur.

Hlavní efekt však bude v časové složitosti. Předpokládáme-li tabulku klíčů uloženou ve vnitřní paměti, prohledáme méně signatur, než u jednoúrovňového souboru signatur. Největším problémem při rozdělování souboru signatur do skupin je určení algoritmu rozdělování. V další diskusi budeme předpokládat vrstvené kódování aplikované na textové soubory, tj. signatury odpovídají logickým blokům textu (např. 1 blok = 1 řádek). Pro vyhodnocení algoritmů rozkladu souboru signatur na skupiny se používají dva koeficienty:

- koeficient redukce signatur daný jako
  (počet prohledávaných signatur)/(počet všech signatur)
- koeficient redukce skupin určený poměrem
  (počet prohledávaných skupin)/(počet všech skupin)

Protože signatury jsou náhodné bitové vektory, totéž bude platit i pro jejich podřetězce - klíče. Je tedy třeba pouze určit dva parametry - počáteční pozici klíče a jeho délku. Podle toho existují v principu čtyři algoritmy pro rozdělení souboru signatur do skupin:

1. klíče pevné délky s pevnou počáteční pozicí (obr. 4),
2. klíče proměnné délky s pevnou počáteční pozicí,
3. klíče pevné délky s nestabilní pozicí (obr. 5),
4. klíče proměnné délky s nestabilní pozicí.

Protože počáteční pozicí může být bez újmy obecnosti 1. bit signatury, hovoříme v případě 1. a 2. o pevných prefixech, resp. pohyblivých prefixech. V metodě 3. se klíče volí tak, aby obsahovaly co nejvíce 0-bitů.

Závěr

Poslední výzkumy ukazují, že dvouúrovňové vrstvené kódování má mnoho předností před ostatními metodami. Zřejmě zatím nejlepší metodou se zdá být jeho zobecnění do více úrovní (obr. 6). Je známé z implementace SŘBD Starburst. Rozdělené soubory signatur jsou snadnější v implementaci a zabírají přiměřeně malý prostor. Lepší je ovšem varianta s pohyblivým prefixem.