Prohledávání dokumentů ve vektorovém modelu

Pavel Moravec[1]

Katedra informatiky, FEI, VŠB - Technická Univerzita Ostrava,
17. listopadu 15, 708 33, Ostrava-Poruba
pavel.moravec@vsb.cz

Abstrakt. *Information retrieval* se mj. zabývá ukládáním a prohledáváním dokumentů. *Vektorový model* reprezentuje dokument jako vektor mnohorozměrného prostoru. Většina jeho implementací je bohužel efektivní jen pro malé dimenze, proto jsou hledány metody, jak dimenzi vektorů dokumentů snížit. V tomto článku ukazujeme jednu z možností – využití *signatur* (bitových řetězců pevné délky) které byly v boolovském modelu velmi efektivní. Druhou významnou skupinu metod tvoří *indexování latentní sémantiky* (*LSI*), jehož výpočetní náročnost je ovšem vysoká. Proto byla navržena rychlejší metoda *náhodných projekcí*, dobře zachovávající (při dostatečně velké redukované dimenzi) vzdálenosti a úhly mezi vektory. V článku zmíníme výsledky metody ve vektorovém modelu.

Úvod

Dokumentografické informační systémy (*DIS*) ukládají textové (či multimediální) dokumenty a umožňují v nich vyhledávat. Při zpracování textových dokumentů v těchto systémech ukládáme celá slova (nebo fráze), obsažená v dokumentu – *termy*. Existuje více způsobů, jak termy v dokumentech reprezentovat. Nejčastěji používaným modelem je tzv. *boolovský* (*boolský*) *model* [17], který umožňuje vyhodnocovat dotazy na termy, obsažené v dokumentech, či jejich spojení, určená pomocí logických operátorů. Pokud však dokument přesně nesplňuje podmínku, obsaženou v dotazu, není nalezen.

Naproti tomu *vektorový model* [15], v němž je kolekce dokumentů reprezentována *maticí termů v dokumentech*, nám umožní vyhledávat i dokumenty, které obsahují dostatečné množství termů, obsažených v dotaze. Zavádí také systém vah termů, které určují významnost termu v rámci dokumentu a kolekce. Vektorový model nám také umožňuje seřadit dokumenty, získané jako odpověď na zadaný dotaz na základě míry podobnosti s dotazem; některé metody hledají zadaný počet nejbližších dokumentů.

U rozsáhlejších kolekcí narůstá čas potřebný k vyhodnocení dotazu s rostoucím počtem termů a dokumentů. Existují proto mnohé přístupy, jak omezit počet porovnávaných vektorů. Nejčastěji jde o stromové struktury jako jsou R-stromy [10], M-stromy, kd-stromy, pyramidová schémata [7] a mnohé další. U těchto struktur ovšem narážíme na problém označovaný jako *„prokletí dimensionality"*. V jeho důsledku se s rostoucí dimenzí stávají zmíněné stromové

[1] Školitel Doc. RNDr. Václav Snášel, CSc. (vaclav.snasel@vsb.cz).

struktury méně efektivní a rychlost vyhledávání bývá od určité dimenze horší než v případě *sekvenčního průchodu* celou kolekcí.

Snažíme se proto snížit alespoň celkový počet termů, určující dimenzi prostoru dokumentů. K tomuto účelu lze využít algebraických metod, souhrnně označovaných jako *indexování latentní sémantiky – Latent Semantic Indexing* (*LSI*) nebo techniky náhodných projekcí. Můžeme také tyto metody spojit a dosáhnout tak zrychlení výpočtu za cenu nižší přesnosti oproti LSI samotnému.

Druhou možností je vytvořit kratší řetězec hodnot, popisující původní vektor a použít jej k filtraci části nerelevantních dokumentů před vyhodnocením dotazu, čehož využívají signaturové metody [9,14] a VA-files [6].

Abychom mohli vyhodnotit kvalitu metod, zmenšujících dimenzi prostoru dokumentů, musíme být schopni popsat chybu, která projekcí do nižší dimenze vznikne. Toho dosahujeme buď měřením rozdílu vzdáleností mezi termy v původním a redukovaném prostoru, nebo zavedením koeficientů *přesnosti* (*precission*) a *úplnosti* (*recall*) [4]. Srovnání můžeme provést vůči původnímu prostoru, nebo vůči předem známým závislostem mezi dokumenty. Přesností rozumíme podíl *relevantních* dokumentů v nalezených, úplností poměr nalezených relevantních dokumentů a všech relevantních dokumentů. Ideální je situace, kdy by oba tyto koeficienty byly rovny 1.

V první kapitole stručně popíši vektorový model, ve druhé využití signaturových metod ve vektorovém modelu, ve třetí bude zmíněn přínos náhodných projekcí a ve čtvrté dosud provedené experimenty.

1 Vektorový model v DIS

Vektorový model reprezentuje dokumenty, obsažené v kolekci, jako vektory v n-rozměrném vektorovém prostoru $\mathbb{R}^n$, kde n je počet všech různých termů v kolekci. Dokument D_j je reprezentován jako vektor

$$d_j = (w_{j,1}, w_{j,2}, \ldots, w_{j,n}).$$

Obdobně dotaz Q je reprezentován jako vektor

$$q = (w_{q,1}, w_{q,2}, \ldots, w_{q,n}).$$

Je zřejmé, že dotazem může být i samotný dokument, je tedy možno hledat všechny dokumenty, které se dostatečně podobají vybranému.

Složky $w_{j,i}$ určují váhu termu i v dokumentu j. Termy, které nejsou v dokumentu obsaženy mají nulovou váhu. Zbývající váhy termů jsou vypočteny na základě součinu následujících tří složek [16]:

1. *Frekvence termu v dokumentu* – zohledňuje, kolikrát je daný term obsažen v konkrétním dokumentu; nejčastěji jde přímo o počet výskytů termu v dokumentu (*tf*).
2. *Frekvence termu v kolekci* – významnost termu pro rozlišení dokumentů; obvykle $\log(m/M)$ označovaná jako *inverzní frekvence dokumentu* (*idf*, kde M je počet dokumentů, obsahujících term a m počet všech dokumentů.

3. *Normalizace váhy termu* – případná normalizace složek vektorů dokumentů během výpočtu vah termů.

Při dotazování hledáme vektory dokumentů, které jsou dostatečně blízké vektoru dotazu. Euklidovská vzdálenost – s oblibou používaná při dotazování nad obrazovými daty – je v DIS většinou nahrazena koeficientem podobnosti, založeným na euklidovském skalárním součinu.

- Nebyly-li vektory předem normovány, je koeficient podobnosti definován jako

$$\delta(d_j, q) = \frac{d_j q}{||d_j||\ ||q||}$$

- V opačném případě postačí $\delta(d_j, q) = d_j q = \sum_{i=1}^{n} (w_{j,i}, w_{q,i})$

Jako relevantní označíme dokumenty, které splňují podmínku $\delta(d_j, q) \geq t$, kde t určuje minimální hodnotu koeficientu podobnosti mezi vektorem dokumentu a dotazu. Výsledné dokumenty lze na základě koeficientu podobnosti seřadit a nabídnout je uživateli od nejvíce odpovídajícího.

Vektory termů v dokumentech jsou často uloženy nebo reprezentovány ve formě řídké matice termů v dokumentech (*term-by-document matrix*). Tato matice $m \times n$ obsahuje řádkové vektory dokumentů $d_1, \ldots, d_m$, vymezující váhy termů, obsažených v dokumentu a sloupcové vektory termů $t_1, \ldots, t_n$, určující dokumenty, obsahující daný term.

2 Signatury ve vektorovém modelu

Signatura je bitovým vektorem *délky* F. Přítomnost termu v dokumentu je indikována nastavením m bitů (jejichž pozice je určena hashovací funkcí) na 1. Počet nenulových bitů v signatuře je nazýván *váhou signatury*.

Signaturu dokumentu lze vytvořit buď *zřetězením* signatur termů (vzniklá signatura je velmi dlouhá) nebo *zvrstvením* více signatur: $S_D = \bigvee_{i=1}^{k} S_{Ti}$, kde $S_{T1}, \ldots, S_{Tk}$ jsou signatury všech termů $T_1, \ldots, T_k$, obsažených v dokumentu D. Lze také tyto dva přístupy kombinovat – rozdělit text do bloků (stejné délky či váhy), vypočíst vrstvené signatury jednotlivých bloků a tyto zřetězit.

Při použití signatur v boolovském modelu jsou před vyhodnocením konjunktivního dotazu nejprve porovnány signatury termů S_{Q_j}, obsažené v dotaze (nebo zvrstvená signatura $S_Q = \bigvee_j S_{Q_j}$, má-li dokument jedinou zvrstvenou signaturu) se signaturami dokumentů S_{D_i}. Platí-li podmínka $(\forall j)[S_{D_i} \wedge S_{Q_j} = S_{Q_j}]$, může dokument zvolené termy obsahovat. Tuto situaci nazýváme *hit*. Protože hashovací funkce a vrstvení nejsou prostými funkcemi, můžeme takto nalézt i dokumenty, které hledané termy neobsahují. Tuto situaci nazveme *falešným hitem*. Proto by po porovnání signatur mělo následovat porovnání nalezených dokumentů s dotazem, které falešné hity odstraní.

Zatímco v boolovském modelu tento přístup funguje velmi dobře, u vektorového modelu selhává. Podmínka $(\forall j)[S_{D_i} \wedge S_{Q_j} = S_{Q_j}]$ ve vektorovém modelu obecně neplatí, proto musíme hledat jiné přístupy.

Očekávání, že relevantním bude dokument, v jehož signaturách bude nalezeno alespoň $k\%$ signatur dotazu nám nezajistí, že některé relevantní dokumenty nebudou vypuštěny. Navíc nejsou brány v potaz váhy termů, proto počet falešných hitů prudce stoupá s klesající hodnotou k.

Existuje však řešení - *Soubory signatur, rozdělené dle vah* (*Weight-partitioned signature files – WPSF*), navržené v [12]. Termy jsou podle frekvence (tf) resp. vah termů v dokumentu rozděleny do skupin – tzv. TF-groups. Pro každou z těchto skupin je vytvořen jeden soubor signatur a v každém souboru jsou uloženy signatury bloků jednotlivých dokumentů. Při hledání jsou pak tyto soubory procházeny v pořadí od největší hodnoty tf po nejmenší – H-L průchod (resp. naopak – L-H průchod), dokud není signatura termu pro daný dokument nalezena. Váha termu je pak spočtena na základě souboru signatur, kde byl hledaný term nalezen a hodnoty idf; na základě vah všech nalezených termů je poté spočítána hodnota $\delta(d_j, q)$. Pravděpodobnost falešného hitu je předem minimalizována – na základě vzorku dokumentů nebo očekávaných vlastností kolekce jsou pro jednotlivé soubory signatur vypočteny vhodné délky bloků a signatur a váha signatury termu. Nevadí-li nám nižší přesnost, způsobená falešnými hity, je možno použít získaných výsledků přímo, bez nutnosti dodatečného výpočtu koeficientu podobnosti mezi vektorem dotazu a vektory nalezených dokumentů.

Protože WPSF byly procházeny sekvenčním průchodem, navrhli jsme v [1] rychlejší metodu průchodu soubory signatur, založenou na S-stromu [8], modifikaci B^+-stromu, která při vkládání minimalizuje nárůst váhy signatur v uzlech celé stromové struktury. Vyhledávání pokračuje jen do uzlů, jejichž signatury S_{N_k} splňují podmínku $S_Q \wedge S_{N_k} = S_Q$.

3 Redukce dimenze náhodnou projekcí

Druhým způsobem, jak snížit čas potřebný pro porovnání je snížení dimenze prostoru dokumentů. Toho lze dosáhnout vypuštěním některých termů, avšak na úkor přesnosti a úplnosti. Proto redukujeme dimenzi jinými způsoby. Nejznámější metodou je *latentní sémantické indexování* (*LSI*), při němž v většinou počítáme *k-redukovaný singulární rozklad* matice termů v dokumentech (metoda SVD). Kromě redukce dimenze, při níž je výsledná chyba nejmenší možná, získáme navíc tzv. *latentní sémantiku* – dokumenty týkající se stejné tématické oblasti si budou blíže a vyniknou vztahy mezi významově (sémanticky) podobnými termy. Bohužel současné metody výpočtu LSI jsou výpočetně náročné – i rychlá metoda *Lanczos* [11] má časovou složitost $O(mnc)$, kde c je průměrný počet různých termů v dokumentu.

Proto byly hledány rychlejší metody, které by dosahovaly podobných výsledků za cenu horší chyby aproximace. Jednou z těchto metod je náhodná projekce [5], která při dostatečně vysoké redukované dimenzi $d, d << n$ dobře zachovává vzdálenosti a úhly mezi vektory. Náhodná projekce je založena na násobení matice termů v dokumentech maticí náhodné projekce R. Prvky R jsou náhodná čísla, jejichž distribuce má nulovou střední hodnotu a jednotkový rozptyl – nejčastěji $N(0, 1)$ a Achlioptasem navržené distribuce $\{-\sqrt{3}, 0, +\sqrt{3}\}$

s pravděpodobnostmi $\frac{2}{3}$ pro 0 a $\frac{1}{6}$ pro $\sqrt{3}$, a $-\sqrt{3}$ a $\{-1, +1\}$, obě hodnoty s pravděpodobností $\frac{1}{2}$ (poskytuje o něco horší výsledky než předchozí metody). Testy ukazují, že např. pro $m = 20000$ a $d \geq 500$ jsou přesnost i úplnost velmi dobré a pohybují se nad 90%.

Časová složitost náhodných projekcí je nižší než u LSI – $O(cdn)$, ale výsledné vektory dokumentů již nejsou řídké, nezískáme latentní sémantiku a redukovaná dimenze není natolik malá, aby překonala „prokletí dimenze". Je ale možno použít náhodné projekce jako předstupně LSI – pak dochází k rychlejšímu výpočtu singulárního rozkladu, časová složitost je $O(m(\log^2 n + c \log n))$ [13] a výsledný rozklad velmi dobře aproximuje rozklad, získaný přímým výpočtem LSI.

4 Provedené experimenty

V rámci testů signaturových metod jsem implementoval klasické signatury, S-stromy a WPSF vylepšené S-stromy. Výsledky testů s těmito strukturami, včetně testů klasických signatur a signatur termů s váhou závislou na frekvenci termu v dokumentu jsou shrnuty v [1]. Obecný popis těchto metod byl obsahem jedné z kapitol [2].

Dále bylo testováno chování VA-files při použití euklidovského skalárního součinu pro rozsahové dotazy namísto klasické euklidovské vzdálenosti. Vzhledem k neuspokojivým výsledkům nebyly tyto publikovány.

Další oblast zkoumání se týkala implementace a testů náhodných projekcí, a míry zkreslení, které přináší jejich použití vzhledem k neredukovanému prostoru. První výsledky jsou shrnuty v [3]. Další získané výsledky, srovnávající LSI a náhodné projekce následované LSI nebyly dosud publikovány.

Závěr

Oblast IR a vektorový model představují zajímavou oblast pro další výzkum. Efektivní implementace vektorového modelu pro rozsáhlé kolekce je dosud hledána a jsou stále zveřejňovány nové metody, jak vyhledávání urychlit. Zajímavými se z tohoto hlediska jeví možnost implementace dostatečně rychlého řešení singulárního rozkladu matice, které by umožnilo redukovat dimenzi velkých kolekcí a nové vylepšené indexační struktury, poskytující rychlejší odezvu, jakými by mohly být například iMinMax a iDistance [18], které v současné době implementuji.

Reference

1. P. Moravec, J. Pokorný, and V. Snášel. Vector query with signature filtering. *Proceedings of 6th BIS conference, Colorado Springs, USA*, 2003.
2. T. Skopal, P. Moravec, M. Krátký, V. Snášel, and J. Pokorný. An Efficient Implementation of the Vector Model in Information Retrieval. *Accepted at RCDL'03 Conference, St. Petersburg, Russia*, 2003.

3. P. Moravec, M. Krátký, and V. Snášel. Random Projections for Dimension Reduction in Information Retrieval Systems. *Proceedings of IMAMM'03 Conference*, 2003.

---

4. R. Baeza-Yates and B. Ribeiro-Neto. *Modern Information Retrieval.* Addison Wesley, New York, 1999.
5. E. Bingham and H. Mannila. Random projection in dimensionality reduction: applications to image and text data. In *Knowledge Discovery and Data Mining*, pages 245–250, 2001.
6. S. Blott and R. Weber. An Approximation-Based Data Structure for Similarity Search. Technical report, ESPRIT, 1999.
7. C. Böhm, S. Berchtold, and D. Keim. Searching in High-Dimensional Spaces – Index Structures for Improving the Performance of Multimedia Databases. *ACM Computing Surveys*, 33(3):322–373, 2001.
8. U. Deppisch. S-tree: A Dynamic Balanced Signature Index for Office Retrieval. In *Proc. of ACM "Research and Development in Information Retrieval"*, pages 77–87, 1986.
9. C. Faloutsos. Signature-based text retrieval methods, a survey. *IEEE Computer society Technical Committee on Data Engineering*, 13(1):25–32, 1990.
10. A. Guttman. R-Trees: A Dynamic Index Structure for Spatial Searching. In *Proceedings of ACM SIGMOD 1984, Annual Meeting, Boston, USA*, pages 47–57. ACM Press, June 1984.
11. R. M. Larsen. Lanczos bidiagonalization with partial reorthogonalization. Technical report, University of Aarhus, 1998.
12. D. L. Lee and L. Ren. Document Ranking on Weight-Partitioned Signature Files. In *ACM TOIS 14*, pages 109–137, 1996.
13. C. H. Papadimitriou, H. Tamaki, P. Raghavan, and S. Vempala. Latent semantic indexing: A probabilistic analysis. pages 159–168, 1998.
14. C. Roberts. Partial-match retrieval via method of superimposed codes. In *Proceedings of IEEE 67*, volume 12, pages 1624–1642, 1979.
15. G. Salton. *The SMART Retrieval System – Experiments in Automatic Document Processing.* Prentice Hall Inc., Englewood Clifs, 1971.
16. G. Salton and C. Buckley. Term weighting approaches in automatic text retrieval. *Information Processing and Management*, 24(5):513–523, 1988.
17. G. Salton and G. McGill. *Introduction to Modern Information Retrieval.* McGraw-ill, 1983.
18. C. Yu. *High-Dimensional Indexing.* Springer–Verlag, LNCS 2341, 2002.

Annotation. *Information retrieval* deals with storage and retrieval of documents. *Vector model* represents the document as a vector in high-dimensional space. Because the majority of vector model's implementations is efficient only for smaller dimensions, methods for document vectors reduction have been studied. We have suggested in this article one of the possibilities – the usage of *signatures* (i.e. bit strings of given length), which are known to be very efficient in Boolean model. Second important category of reduction methods is called *latent semantic indexing* (*LSI*). Since LSI is quite hard to calculate, a faster reduction method – *random projection*, well preserving distances and angles between vectors (is the reduced dimension high enough) – was proposed recently. We have studied its impact in the vector model.