FAKULTA
INFORMAČNÍCH
TECHNOLOGIÍ
ČVUT V PRAZE

Evaluace algoritmů lokálně senzitivního hashování (LSH) v doporučovacích systémech

Ladislav Martínek

Ročník: 3
Obor: Znalostní inženýrství

Katedra aplikované matematiky
Vedoucí práce: Ing. Tomáš Řehořek

Klíčová slova: aproximace algoritmu nejbližších sousedů, lokálně senzitivní hashování, doporučovací systémy, implementace testovací knihovny, experimenty a analýza výsledků

14. dubna 2018

1 Úvod

V dnešní moderní společnosti, kde neustále roste objem dat nabízený na internetu, je pro uživatele velmi těžké vyhledávat informace nebo produkty. Proto se společnosti zabývají přizpůsobováním obsahu každému uživateli na internetu např. doporučením, co by se uživateli mohlo líbit. Při doporučovaní se snaží nabídnout např. film nebo článek, který by mohl uživatele zajímat. Doporučování na internetu je nutné provádět v reálném čase, což se současnými objemy dat je velice problémové. Z tohoto důvodu je výhodné, zároveň i nutné se zabývat zrychlováním doporučovacích systému.

Existuje mnoho metod, které se zrychlováním zabývají. V práci se věnuji jedné konkrétní metodě a to lokálně senzitivnímu hashování, pomocí kterého lze dosahovat několikanásobného zrychlení výpočtu při malé ztrátě přesnosti. Výstup mé práce by měl pomoci programátorům s testováním a vyhodnocováním různých parametrizací použité metody pro zrychlení vyhodnocování doporučovacích systémů.

Téma jsem si zvolil, neboť doporučovací systémy jsou využívány mnoha společnostmi na internetových stránkách, kde nám jejich přítomnost usnadňuje hledání informací. Nicméně je velmi výpočetně náročné vyhodnocovat všechna nasbíraná data v reálném čase, proto se v práci zabývám aproximací doporučovacích systémů, konkrétně pak návrhem, implementací a následným vyhodnocením knihovny na aproximaci hledání nejbližších sousedu.

2 Cíl práce

Cílem rešeršní části práce je získat přehled o doporučovacích systémech, seznámit se s algoritmy nejbližších sousedu a nastudovat algoritmy lokálně senzitivního hashování a možnosti využití těchto metod při aproximaci algoritmu nejbližších sousedu. Dalším navazující cíl rešeršní části je představení metod vyhodnocování úspěšnosti představených algoritmů a doporučovacích systému založených na těchto algoritmech.

Cílem praktické části práce je navrhnout způsob aplikace metod lokálně senzitivního hashování za účelem aproximace algoritmu hledání nejbližších sousedu, poté návrh a implementace frameworku pro testování úspěšnosti algoritmu lokálně senzitivního hashování v doporučovacích systémech. V neposlední řadě je cílem otestování implementace algoritmu s různou parametrizací na dvou datových sadách a diskuze výsledků, konkrétně poměr zrychlení výpočtu na úkor zhoršení přesnosti.

3 Analýza

Nejprve stručně vymezím pojmy a podle literatury popíši přístupy, kterým se v práci věnuji. Na to naváži analýzou současného stavu řešení aktuální problematiky.

3.1 Základní pojmy

Doporučovací systémy jsou nástroje a techniky, které generují doporučení položek, které by mohly být zajímavé nebo užitečné pro uživatele [1, pg.1]. Doporučením mohou být například produkty k prodeji, filmy nebo i slevové kupóny od obchodních řetězců.

Základním přístupem v doporučovacích systémech je filtrování obsahu a informací pomocí spolupráce mezi více uživateli (Kolaborativní filtování, angl. Collaborative filtering). V mé práci se budu věnovat filtrování pomocí spolupráce uživatelů. Klíčovou myšlenkou je, že pokud uživatelé sdílejí zájem o určité položky, budou tento zájem sdílet i nadále a budou preferovat podobné položky [2].

Algoritmy nejbližších sousedů (angl. k-nearest neighbors, k-NN) jsou přístupem kolaborativního filtrování a vždy fungují v konkrétním měřitelném prostoru, který muže být definován různými způsoby. Existence metriky je důležitá pro porovnávání podobnosti nebo vzdálenosti jednotlivých sousedů.

Lokálně senzitivní hashování (angl. Locality sensitive hashing, LSH) je metoda aproximace algoritmu k-NN. Podobně jako i jiné aproximační metody je LSH randomizované a náhodnost je typicky využita ke konstrukci datové struktury pro vytvoření modelu [3].

Základním principem LSH je rozdělení vícerozměrného prostoru na části nazývané jako kbelíky (angl. buckets). Hashovací funkce LSH transformuje mnoha dimenzionální vektory na sekvence bitů. Odlišností proti klasickým hashovacím funkcím jako jsou SHA a MD5, které se snaží kolizím co nejvíce zabránit, se hashovací přístup v LSH snaží maximalizovat pravděpodobnost kolize blízkých položek [4].

3.2 Současný stav

Metody LSH jsou různě definované pro různé metriky podobnosti a vzdáleností v prostoru. V práci se věnuji pouze metrice kosinové podobnosti. Podle [5] platí, že každá hashovací funkce $h(\mathbf{x}) \rightarrow \langle 0 \dots m)$ převede d dimenzionální vektor $\mathbf{x}$ na číslo bucketu, které je mezi 0 až $m - 1$. Hashovací funkce musí být rychlá na výpočet, jednoduše rozšiřitelná a také pravděpodobnost kolize musí odpovídat kosinové podobnosti.

V publikaci [5] představují hashovací funkce jako matice, pomocí kterých jsou jednotlivé vektory položek všech uživatelů hashovány do příslušných bucketů. Hashovací matice jsou gererovány z uniformního nebo normálního rozdělení náhodně. Nevýhodou takového řešení může být, že ne vždy může být potencionálně podobný uživatel zahashován do stejného bucketu. Problém je řešen přidáváním více a více hashovacích funkcí.

Přidávání hashovacích funkcí má ovšem svoje hranice, protože je nutné výsledky z jednotlivých funkcí agregovat. Řešení problému je navrženo v publikaci [6]. V publikaci je popsána metoda, kdy jsou procházeny kromě jednoho výsledného bucketu také podobné buckety. Při hashování se může stát, že se reprezentace bucketu bude lišit v jednom bitu. Proto při vyhodnocování dotazů jsou procházeny také blízké buckety, které jsou určeny svojí hammingovou vzdáleností, tedy počtu odlišných bitů v reprezentaci bucketu.

Cílem této metody může být snížení paměťové náročnosti oproti použití velkého množství hashovacích funkcí nebo také dosahování vetší přesnosti při aproximaci, kdy už není efektivní přidávat další hashovací funkce.

Poslední testovaný přístup LSH je založen na myšlence popsané v článku [7]. V každé reálné datové sadě jsou položky, které jsou u uživatelů více oblíbené než jiné. Tuto podobnost lze zjistit z nasbíraných dat. Základní představa je, že je mnohem více pravděpodobnější, že uživatel kladně zareaguje na populární položku než na nepopulární. V [7] je popularita položky brána jako počet relevantních hodnocení uživatelů dané položky.

Těchto hodnot popularity lze využít k úpravě projekční matice. Upravená projekční matice poté dává větší váhu na jednotlivé populární položky, proto jsou uživatele sdílející tyto položky hashováni do stejných bucketů.

Přístupy v jednotlivých publikacích jsou také testovány většinou jednorázově a pouze na datech, které jsou generovány uměle. Z tohoto důvodu jsem implementoval knihovnu, která umožní testovat různé parametrizace metod LSH na reálných datech.

4 Realizace

Pro realizaci jsem zvolil programovací jazyk Python, protože je dobře použitelný pro efektivní implementaci testovacích modulů. V rámci práce jsem vytvořil testovací framework, který se skladá z několika základních modulů. Pomocí frameworku lze testovat různé parametrizace metod LSH.

Framework se skládá z následujících částí. Základem jsou testovací scripty, ve kterých jsou systematicky provedeny testy zvolených parametrizací a metod LSH. Ve frameworku se nacházejí dva hlavní testovací skripty, které sdílejí ostatní moduly ve frameworku. Různé nezávislé skripty jsem zvolil z důvodu odlišnosti testování přesnosti algoritmu k-NN a přesnosti doporučování na základě algoritmu k-NN.

Každý ze způsobů je nutné otestovat odlišným způsobem, přičemž první jmenovaný je výpočetně méně náročný, protože při testování je každý uživatel jedním modelem testován právě jednou. Testování přesnosti k-NN je testováno na 10 nejbližších sousedů, avšak ve frameworku lze tuto hodnotu specifikovat v argumentu. Toto testování je vždy prováděno zároveň s referenčním řešením vůči kterému je testována úspěšnost.

Při testování úspěšnosti doporučování je každý uživatel testován n-krát pomocí leave-one-out křížové validace, kde n je nastaveno na hodnotu 20, ale také je možné tuto hodnotu specifikovat na jinou hodnotu. Doporučování je testováno na 5 doporučených položek (lze specifikovat hodnotu v argumentu) a následně je kontrolováno zdali se vynechaná položka nachází v doporučených 5 položkách. Z toho plyne, že při tomto testování není bezpodmínečně nutné počítat i referenční řešení, ale je to velice výhodné pro porovnání úspěšnosti, protože testování může být závislé na velikosti a volbě testovací množiny.

Prvním sdíleným modulem ve frameworku je modul předzpracování, který obstarává stažení offline dat z databáze. Je možné jej nahradit za modul, který umožní stahování z jiných datových struktur. V části předzpracování probíhá dále zpracování dat z databáze pro rychlé vyhodnocování modelu k-NN a LSH. V práci používám modul pro PostgreSQL databázi, ze které stahuji data. Modul pouze stahuje data na základe SQL příkazu. V konfiguračním souboru je specifikována konkrétní databáze s přístupovými údaji do ní. Data jsou získávána z jednotlivý tabulek jako seznam n-tic (uživatel, položka, hodnocení, časová známka). Hodnocení muže kromě samotného uživatelského hodnocení obsahovat také váhu ohodnocení určité interakce uživatele s položkou.

Dalším důležitým modulem ve frameworku je modul k-NN. Modul slouží pro vytvoření modelu k-NN a vyhodnocování dotazu na k nejbližších sousedu. Tento modul jsem do systému navrhl zejména z důvodu nutnosti referenčního modelu pro porovnávání a hodnocení metod LSH.

Metody LSH představují vlastní důležitý modul. Model slouží k rozdělení prostoru pomocí lokálně senzitivních hashovacích funkcí. Modul aproximuje dotaz na k nejbližších sousedu. Skládá se z dvou částí, první je vyhodnocení k nejbližších sousedu na principu k-NN a druhý představuje model hashovacích funkcí, která vrací množinu potencionálních uživatelů.

Poslední dva moduly jsou pomocné. Jeden slouží k zápisu dat. Data jsou do třídy posílána jako slovníky klíč–hodnota. V modulu je implementováno řazení dat a zápis do konkrétního formátu výstupu pro tvorbu grafu. Pro práci jsem zvolil formát csv představující zápis tabulky v souboru, kde jsou hodnoty oddělené čárkami. Druhý modul je nezávislý a usnadňuje tvorbu grafů z datových souborů pro přehlednou prezentaci výsledků.

5 Závěr

Cílem práce byl návrh a implementace knihovny pro testování metod lokálně senzitivního hashování, úspěšnosti těchto metod a jejich různých parametrizací.

Knihovna implementována v rámci práce umožňuje testovat různé parametrizace a modely LSH na různých databázích, které lze specifikovat v konfiguračním souboru. Z výsledných dat generovaných frameworkem je možné pomocí nezávislého modulu generovat grafy pro lepší prezentaci výsledků.

Dosažené výsledky při testování různých parametrizací jsou velice uspokojivé. Na dvou otestovaných datových sadách bylo zjištěno, že lze dosahovat času kolem 5 %. Úspěšnost doporučování se při této aproximaci pohybuje okolo 97 - 99 %. Při této úspěšnosti dosahují modely LSH přibližně o 20 % lepší catalog coverage vzhledem k referenčnímu modelu. Doporučovací systém s metodami LSH tedy doporučuje větší množství různých položek, proto mohou být metody LSH užitečnější tím, že kromě časového zrychlení, mohou doporučovat větší část položek z databáze.

Zajímavým poznatkem je zjištěná závislost mezi přesností na 10 nejbližších sousedů a úspěšnosti doporučování. Díky tomuto zjištění bylo možné otestovat velké množství různých parametrizací, protože testování přesnosti na 10 nejbližších sousedů je efektivnější. Testováním byli zjištěny některé hranice nastavení jednotlivých modelů a také nastavení pomocí kterého lze dosahovat optimálních výsledků.

V neposlední řadě je zajímavý poznatek nastavení parametrů testu pro testování metod LSH pomocí frameworku. Pro každou databázi je nutné zvolit požadované parametry při testování z ohledem na velikost databáze. Tyto parametry lze podle počtu uživatelů lehce dopočítávat a tím přizpůsobit testování pro konkrétní databázi.

V budoucnu by mohli tyto metody implementované jako součást doporučovacího systému, kde by bylo nutné se zabývat jejich paralelizací. Metody LSH lze velice efektivně paralelizovat obdobně jako samotný algoritmus k-NN. V budoucí práci by bylo také určitě nutné omezit maximální počet uživatelů v jednom bucketu.

Při využití v online doporučovacím systému by bylo nutné se zabývat automatizací nastavení modelu, tedy zvoleným množstvím hashovacích funkcí, množství bucketů a velikost okolí procházených bucketů. Zároveň na online datech není znám předem přesný objem dat ani počet uživatelů, proto by bylo zajímavé navrhnout způsob, kterým by bylo možné přizpůsobovat model při online běhu (např. přehashováním, přidávání nebo odebírání hashovacích funkcí).

Reference

[1] RICCI, F.; ROKACH, L.; SHAPIRA, B.; aj.: *Recommender systems handbook*. Springer New York Dordrecht Heidelberg London, 2011, ISBN 978-0-387-85819-7.

[2] XUE, G.-R.; LIN, C.; YANG, Q.; aj.: Scalable collaborative filtering using cluster-based smoothing. [online], August 15 - 19, 2005, [cit. 2018-03-08]. Dostupné z: `https://dl.acm.org/citation.cfm?id=1076056`

[3] SHAKHNAROVICH, G.; DARRELL, T.; INDYK, P.: Introduction. In *Nearest-neighbor methods in learning and vision: theory and practice*, Cambridge, Messachusetts, London, England: The MIT press, March 24, 2006, ISBN 0-262-19547-X, s. 1–12.

[4] WANG, J.; SHEN, H. T.; SONG, J.; aj.: Hashing for Similarity Search: A Survey. [online], Submitted on 13 Aug 2014, [cit. 2018-03-11], `1408.2927`. Dostupné z: `http://arxiv.org/abs/1408.2927`

[5] ARGERICH, L.; GOLMAR, N.: Generic LSH Families for the Angular Distance Based on Johnson-Lindenstrauss Projections and Feature Hashing LSH. [online], Submitted on 15 Apr 2017, [cit. 2018-03-17], `1704.04684`. Dostupné z: `http://arxiv.org/abs/1704.04684`

[6] LV, Q.; JOSEPHSON, W.; WANG, Z.; aj.: Multi-probe LSH: efficient indexing for high-dimensional similarity search. [online], September 23 - 27, 2007, [cit. 2018-03-17]. Dostupné z: `http://dl.acm.org/citation.cfm?id=1325851.1325958`

[7] STECK, H.: Item Popularity and Recommendation Accuracy. [online], October 23 - 27, 2011, [cit. 2018-03-19]. Dostupné z: `doi:10.1145/2043932.2043957`