

Framework pro podporu online evaluace doporučovacích systémů

Úvod do problematiky

Co je doporučovací systém

Nacházíme se v prostředí, kde máme množinu uživatelů a množinu položek. Uživatelé navíc nějakým způsobem vyjadřují svou preferenci pro dané položky (například tím, že jí ohodnotí, že si položku prohlídí, porovnávají ji s ostatními, ...). Úkolem doporučovacího systému (DS) je předpovědět, které uživateli neznámé položky mu budou připadat zajímavé. Mají tak uživateli představit položky, o jejichž existenci by se nemusel dovědět. Pokud uživatel hledá nějakou položku, ale neví konkrétně kterou, DS mu může s vyhledáváním pomoci (tak funguje např. search engine).

DS mají širokou škálu využití: lze je využít v e-shopech, na platformách určených k streamování hudby, filmů. Na sociálních sítích mohou doporučovat uživateli zajímavé události, osobnosti, hashtagy, apod.

S širší využití souvisí rozmanitost ve funkčnosti samotných DS. Algoritmy mohou využívat jenom informace o známé preferenci (např. jak uživatel hodnotil položky), informace o dalších uživateli (např. snažit se najít uživatele, kteří hodnotili stejné filmy podobně), doplňující informace o položkách (např. hledat položky ze stejné kategorie (elektronika, akční filmy, ...)).

Evaluace

Uživatelské studie

Malé skupině uživatelů jsou při používání platformy poskytnuta doporučení. Poté se vyhodnotí schopnosti DS pomocí otázek, které uživatelé zodpoví.

Offline evaluace

Máme data o tom, jak uživatelé hodnotili položky, data se rozdělí na trénovací a testovací. Systém se podle trénovacích dat snaží předpovědět další hodnocení, ta se porovnají s testovacími daty.

- + jednoduché na provedení, rychlé
- + vhodné pro redukování mnoha systémů jen na ty nejslibnější
- to, co uživatel preferuje může být ovlivněno tím, co je mu doporučeno

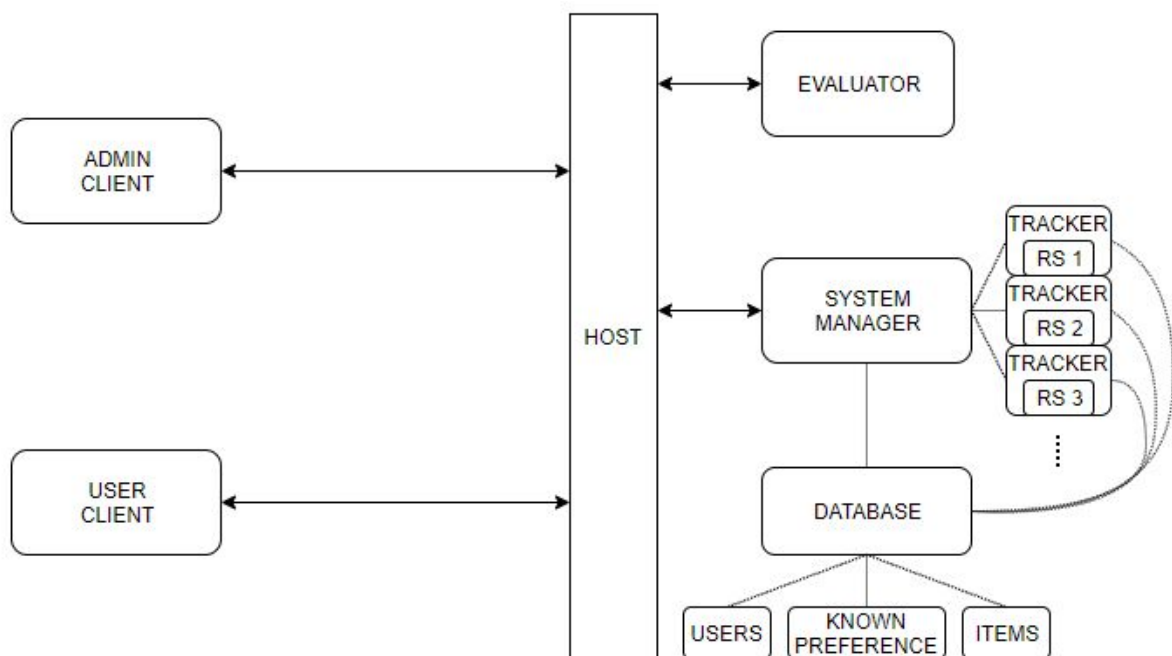
Online evaluace

Systémy doporučují reálným uživatelům za normálního provozu. Obvykle po dobu v řádu týdnů až měsíců.

Dává lepší představu o skutečné reakci uživatelů na DS.

- + lze reagovat na uživatelské požadavky v dané relaci
- + lze měřit metriky jako je click-through rate
- jejich reakce může být negativní a může ovlivnit důvěru uživatele v systém

Přehled systému



Database

Databáze tvoří množinu informací se kterými doporučovací systém pracuje. Obsahuje 3 hlavní části. Všechny uživatele (**USERS**), všechny položky (**ITEMS**) a nakonec množinu (**KNOWN PREFERENCE**) všech známých trojic (u, i, p), kde u je uživatel, i je položka a p je preference uživatele u pro položku i .

Users, Items

Seznam uživatelů a položek indexovaných jejich ID číslem. Kvůli různorodosti prostředí, kde se doporučovací systémy využívají, třídy **User** a **Item** obsahují jen jednoznačnou identifikaci ve formě **intu**. Od provozovatele platformy se očekává, že si třídy **User** a **Item** rozšíří o jemu potřebné informace.

KnownPreference

Uložená ve formě řídké matice (viz `SparseMatrix<T>`).

RS 1, 2, ...

Představuje samotný doporučovací algoritmus. Je implementován provozovatelem platformy. Musí implementovat **IManagedRecommenderSystem**.

RecommenderSystemTracker

Wrapper pro **RS** (implementuje stejný interface **IRecommnderSystem**). Úkolem trackeru je monitorovat odpovědi systému na požadavky uživatele a ukládat si uživatelův feedback.

Tyto informace si ukládá do **SafeLinkedList**, protože zde očekáváme velkou aktivitu a paralelní zpracovávání požadavků. Uložené informace se použijí pro evaluaci.

Tracker také poskytuje možnost uložení a nahrání poskytnutých doporučení do specifikovaného souboru.

SystemManager

Spravuje všechny doporučovací algoritmy a jím odpovídající trackery. Poskytuje uživatelům doporučovací systémy (zabalené v trackeru) a udržuje informace o tom, který uživatel využívá jaký tracker.

Evaluator

Obsahuje statické metody určené pro evaluaci.

Rating based

- mean absolute error (MAE)
- root mean square error (RMSE)

Ranking based

- average reciprocal hit rank (ARHR)
- average discounted cumulative gain (AverageDCG)
- average normalized discounted cumulative gain (AverageNDCG)

Speed of response

- mean response time
- median response time

Other

- click-through rate (CTR)
- user coverage
- item coverage
- average list diversity
- recall
- accuracy
- precision
- mean average precision (MAP)

Host

Dodán vlastníkem platformy. Implementuje **RS**. Poskytuje uživatelům a administrátorům doporučení a možnost evaluace.

UserClient

Aplikace skrz kterou interaguje uživatel s **Hostem**.

AdminClient

Aplikace skrz kterou interaguje administrátor s **Hostem**.

Detailnější pohled

Interface

IRecommenderSystem

Představuje základní funkčnost očekávanou od doporučovacího systému. Implementuje ho **RS** a jeho **Tracker**.

IManagedRecommenderSystem

Rozšiřuje **IRecommenderSystem** o možnost manipulace s DS. Ukládání a nahrávání DS.

Objekty

RecommenderSystemTracker

V trackeru se také nachází definice **AggregationFunction**. Jedná se o funkcionalitu, kterou do proměnné **Aggregator** dodá ten, kdo implementuje framework. Je použita pokud feedback v trackeru obsahuje i neexplicitní feedback. Získá seznam známého feedbacku který jeden uživatel dal jedné položce. Z toho má odvodit uživatelskou preferenci pro danou položku, pokud odvodit nejde vrátí null a uživatelská preference se považuje za neznámou. Dále se funkcionalita **Trackeru** dělí na 3 hlavní části.

Recommendation

Tato část se vypořádává s požadavky uživatelů týkajících se doporučení. Vesměs pouze provolává ekvivalentní metody DS. Pokud se jedná o navrácení feedbacku, uloží si ho. Pokud chce uživatel doporučení nebo seřazení položek, změří dobu odezvy a uloží si odpověď DS společně s dalšími informacemi jako **RecommendationList**.

Manipulation

Slouží k uložení a nahrání doporučení, které **Tracker** uchovává.

Evaluation

Většina funkcí jsou ekvivalenty stejně pojmenovaných funkcí v evaluation. Tyto funkce pak volá s doporučeními a feedbackem uloženým v daném **Trackeru**.

Funkce **GetAllPreferences** slouží ke konverzi feedbacku na číselnou preferenci. Pokud je pouze explicitní hodnocení, vezme to, pokud obsahuje i implicitní, odvodí se pomocí poskytnutého **Aggregator-u**.

SystemManager

SystemManager obsahuje také funkci **SystemAssignment**, která dostane ID uživatele a jejím úkolem je vrátit název RS, který mu bude poskytovat doporučení. Tuto funkcionalitu dodá vlastník platformy do proměnné **Assigner**.

Při přidávání nebo odebrání systému / jeho trackeru ze **SystemManageru** si datovou strukturu Dictionary uzamkneme. Stejně tak si uzamkneme Set obsahující množinu uživatelů, kterým současně poskytuje daný **Tracker**. To si můžeme dovolit, protože změny v těchto kolekcích očekáváme “zřídka” a tak nám nebudou operace zbytečně zpomalovat.

Evaluator

Datové struktury

SparseMatrix<T>

Představuje datovou strukturu pro matici s nízkým faktorem hustoty zaplnění. Např. v naší ukázce máme 138 493 uživatelů, 27 278 položek, ale “jenom” 20 000 263 známých hodnocení, což nám dává celkovou velikost matice 3 777 812 054 s faktorem zaplnění pouze 0,5294 % . Potřebujeme rychlý náhodný přístup, data tedy ukládáme do hašovací tabulky indexované dvojicí (userId, itemId).

SafeLinkedList<T>

Thread-safe linked list. Používáme ho na místech, kde očekáváme časté ukládání dat. Tedy v Trackeru pro ukládání systémem poskytnutých doporučení a uživatelova feedbacku.

RecommendedItem

Je to dvojice položky a preference, kterou očekává doporučovací systém.

RecommendationList

Struktura, do které si Tracker ukládá systémem poskytnutá doporučení. K seznamu doporučených položek si přidá kterému uživateli patří, čas doporučení a dobu odezvy doporučovacího systému v ms.

Feedback

Abstraktní třída. Obsahuje dvojici (userId, itemId), kterých se týká a také timestamp, kdy byl feedback podán. Má dva hlavní potomky.

ExplicitFeedback

Rozšiřuje Feedback o hodnocení, jaké dal uživatel položce.

ImplicitFeedback

Ne vždy se nám podaří získat explicitní hodnocení položky uživatelem. Někdy musíme uživatelovu preferenci uhodnout. K tomu slouží implicitní feedback. Rozšiřuje Feedback o typ implicitního ohodnocení (ve formě pojmenování) a hodnotu.

Ukázková implementace frameworku

Data

Používáme MovieLens 20M dataset (<https://grouplens.org/datasets/movielens/>). Ten obsahuje 138 000 uživatelů, 27 000 filmů a 20 milionů hodnocení (od 0.5 do 5). Také obsahuje mapování filmu na jejich IMDB stránku (<https://www.imdb.com/>). Odtud byly staženy dodatečné informace o filmech.

Pro potřeby zobrazení si vytvoříme třídu **Movie** jako potomka **Item**. Nahrajeme pak informace všech filmů do databáze ve formě **Movie**. Jedinou položku kterou nahrávat nebudeme je plakát filmu a to z důvodu úspory paměti. Až budeme chtít objekt **Movie** přenášet, vytvoříme si jeho kopii (metodou **Clone()**), a do kopie nahrajeme plakát (zakódovaný ve stringu) a přeneseme kopii.

Doporučovací systémy

DS jsou implementovány z knihovny MyMediaLite (<http://www.mymedialite.net/>). Pro demonstraci jsou použity SVDPlusPlus (třída **SVDImplemented**) a UserItemBaseline (třída **UIBImplemented**). Pro ty implementujeme interface **IManagedRecommenderSystem**.

Parametry

Administrátor by měl mít možnost spravovat a měnit hodnoty parametrů doporučovacích systémů. Toho dosáhneme pomocí reflexe. V implementaci DS stačí označit parametry podle jejich typu jako správného potomka **ParameterAttribute**. Ten má potomky pro všechny primitivní typy a umožňuje omezit, jakých hodnot může parametr nabývat. A to buď pomocí u všech typů zabudovaného delegáta **Condition**, nebo pomocí vlastností **MaxValue**, **MinValue** a **ExcludeRange** u číselných typů. Max/MinValue určují maximální/minimální hodnotu parametru a ExcludeRange říká, zda krajní hodnoty intervalu jsou povoleny.

Pokud chceme pak hodnotu parametru nastavit, zkontrolujeme, zda splňuje **Condition** popř. I rozsah (u číselných typů).