FAKULTA INFORMAČNÍCH TECHNOLOGIÍ ČVUT V PRAZE

Zadání diplomové práce

Název:	Algoritmy pro lepší porozumění doporučovaného obsahu a segmentů uživatelů
Student:	Bc. Pavel Hlubík
Vedoucí:	doc. Ing. Pavel Kordík, Ph.D.
Studijní program:	Informatika
Obor / specializace:	Znalostní inženýrství
Katedra:	Katedra aplikované matematiky
Platnost zadání:	do konce letního semestru 2022/2023

Pokyny pro vypracování

Nastudujte metody pro analýzu obsahu a uživatelů v prostředí doporučovacích systémů. Soustřeďte se na přístupy, které umožní pochopit segmenty obsahu, uživatelů a jejich vzájemnou interakci. Implementujte prototyp, který pro několik různých databází provede analýzu a vygeneruje reporty popisující stav prostředí a umožní lépe porozumět probíhajícím interakcím v čase. Alternativně se můžete soustředit na algoritmy, které umožní efektivně promítnout obsah a uživatele do stejného latentního prostoru, který následně vizualizujete a vhodně anotujete.

Elektronicky schválil/a Ing. Karel Klouda, Ph.D. dne 10. února 2021 v Praze.

FAKULTA
INFORMAČNÍCH
TECHNOLOGIÍ
ČVUT V PRAZE

Diplomová práce

Algoritmy pro lepší porozumění doporučovaného obsahu a segmentů uživatelů

Bc. Pavel Hlubík

Katedra aplikované matematiky
Vedoucí práce: doc. Ing. Pavel Kordík, Ph.D.

3. května 2021

Poděkování

Děkuji vedoucímu práce za aktivní vedení, pomoc při návrhu experimentů a časté konzultace. Děkuji také své rodině za podporu v průběhu celého studia a za vytvoření zázemí, ve kterém jsem mohl tuto práci dokončit.

Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval samostatně a že jsem uvedl veškeré použité informační zdroje v souladu s Metodickým pokynem o dodržování etických principů při přípravě vysokoškolských závěrečných prací.

Beru na vědomí, že se na moji práci vztahují práva a povinnosti vyplývající ze zákona č. 121/2000 Sb., autorského zákona, ve znění pozdějších předpisů. V souladu s ust. § 2373 odst. 2 zákona č. 89/2012 Sb., občanský zákoník, ve znění pozdějších předpisů, tímto uděluji nevýhradní oprávnění (licenci) k užití této mojí práce, a to včetně všech počítačových programů, jež jsou její součástí či přílohou a veškeré jejich dokumentace (dále souhrnně jen „Dílo"), a to všem osobám, které si přejí Dílo užít. Tyto osoby jsou oprávněny Dílo užít jakýmkoli způsobem, který nesnižuje hodnotu Díla a za jakýmkoli účelem (včetně užití k výdělečným účelům). Toto oprávnění je časově, teritoriálně i množstevně neomezené. Každá osoba, která využije výše uvedenou licenci, se však zavazuje udělit ke každému dílu, které vznikne (byť jen zčásti) na základě Díla, úpravou Díla, spojením Díla s jiným dílem, zařazením Díla do díla souborného či zpracováním Díla (včetně překladu) licenci alespoň ve výše uvedeném rozsahu a zároveň zpřístupnit zdrojový kód takového díla alespoň srovnatelným způsobem a ve srovnatelném rozsahu, jako je zpřístupněn zdrojový kód Díla.

V Praze dne 3. května 2021 . . . . . .

České vysoké učení technické v Praze
Fakulta informačních technologií

Odkaz na tuto práci

Hlubík, Pavel. *Algoritmy pro lepší porozumění doporučovaného obsahu a segmentů uživatelů.* Diplomová práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2021.

Abstrakt

Studium vztahu mezi publikem a obsahem, který konzumuje, je pro tvůrce obsahu zásadní. V této práci navrhujeme a experimentálně vyhodnocujeme různé metody vizualizace aspektů publika. Zavádíme novou techniku vkládání uživatelů a položek do stejného latentního vektorového prostoru, která dosahuje slibných výsledků na známé datové sadě Movielens. K vizualizaci publika dále využíváme samoorganizační mapy, jejichž použití pro tento typ úlohy je podle našich nejlepších znalostí novým přístupem. Poslední metodou tohoto druhu je nově publikovaný framework MDE, který překonává mnohé nevýhody t-SNE, a přitom neohrožuje kvalitu, což ukážeme experimentálně.

Zabýváme se také dalšími pohledy na interakce uživatelů. Uživatelé a položky jsou propojeni pomocí Sankeyho diagramů, které nabízejí komplexní pohled na to, které skupiny uživatelů interagují s jakým obsahem, a jsou informativnější než prosté zařazení uživatelů do jedné kategorie. Navrhujeme také přístup k vizualizaci interakcí uživatelů v čase, který může pomoci analyzovat časové závislosti.

Klíčová slova Závěrečná práce, segmentace uživatelů, samoorganizační mapy, embedding.

Abstract

Studying the relationship between audience and content they consume is vital for content creators. In this thesis, we propose and experimentally evaluate various methods to visualize aspects of the audience. We introduce a new technique of embedding users and items into the same latent vector space, which achieves promising results on the well-known Movielens dataset. To visualize the audience we further utilize self-organizing maps, usage of which for this type of task is according to our best knowledge a novel approach. The last method of this kind is a newly published framework MDE, which overcomes many drawbacks of t-SNE, yet it does not compromise quality, which we show experimentally.

We also address other views of user interactions. Users and items are connected with the help of Sankey diagrams which offer a complex insight into which groups of users interact with what content and are more informative than simply classifying users into a single category. We also propose an approach to visualize user's interactions over time which may help to analyze time-dependent behavior.

Keywords Thesis, user segmentation, self-organizing maps, embedding.

Obsah

Seznam obrázků

Seznam tabulek

Úvod

Mediální domy a všichni další tvůrci obsahu soutěží o pozornost svých uživatelů ve velmi proměnlivém prostředí a nikdo se nemůže spoléhat na to, že přízeň publika přetrvá delší dobu. Nutností se stává systematická analýza dat o chování návštěvníků, tedy o tom, jak uživatelé interagují s obsahem. V západním světě byly určitými pionýry tohoto přístupu v mediálním prostředí servery jako BuzzFeed nebo Huffington Post, v posledních letech se ale na tyto analýzy začínají čím dál tím více spoléhat i velké organizace jako Guardian, Financial Times nebo BBC. [1]

Propojení uživatele s novým, pro něj vhodným obsahem, je tradičně úlohou doporučovacích systémů. V případě, že obsah není tvořen komunitou, potřebují jeho tvůrci pochopit, jaké segmenty publika preferují jaký typ obsahu, aby mohli vůbec rozhodnout o jeho vzniku. V odborné literatuře je možné nalézt práce věnující se analýze interakcí uživatelů s uživatelským prostředím, kde jsou následně hledány opakující se vzory pomocí metody LDA [2], nebo vizualizaci chování v komunitě pomocí glyfů [3], tyto přístupy ale bohužel neberou v potaz vlastní obsah.

Většina výzkumných prací, které se zabývají reprezentací uživatele a které berou v potaz i obsah, se vydává cestou reprezentace uživatelů v latentním prostoru - embeddingu. V případě, že uživatelé obsah generují (fóra, sociální sítě, ...), je možné vytvořit efektivní reprezentaci na základě časového sledu vytvořených příspěvků, jak je ukázáno v práci Author2Vec [4]. Taková reprezentace se prokazuje užitečná nejen při vizualizaci, ale i při dalších úkolech, jako např. identifikace uživatelů trpících duševní chorobou [4].

Často se pohybujeme v prostředí, kde je obsah uživateli pouze konzumován, díky čemuž musíme hledat jiné způsoby, jak reprezentaci tvořit. Můžeme obrátit pozornost k doporučovacím systémům, které pro svůj úkol taktéž potřebují rozpoznat podobné uživatele. V této oblasti je velmi slibná práce výzkumníků z čínské společnosti Alibaba [5], kteří dokáží reprezentovat uživatele i položky v jednom společném vektorovém prostoru.

V této práci prověříme různé metody pro získání latentní reprezentace uživatelů a položek za účelem jejich zobrazení za účelem nalezení shluků uživatelů a pochopení jejich vztahů. Pro vizualizaci využijeme nejen obvyklou metodu t-SNE [6], ale i Samoorganizační mapy (SOM, [7]) a nově publikovaný velmi slibný framework Minimum-distortion Embedding [8]. Dále navrhneme vlastní metodu inspirovanou výše zmíněným přístupem společnosti Alibaba, která dokáže reprezentovat uživatele i položky v jednom společném prostoru, a tu experimentálně ověříme.

Tato práce je členěna do rešeršní a praktické části. V rešerši zmapujeme odbornou literaturu týkající se našeho problému, identifikujeme přístupy, které bude možné využít pro řešení v praxi a položíme teoretické základy všech vizualizačních technik, které budou využity. V praktické části pak experimentálně ověříme zvolené metody na různých datasetech s důrazem pro vysvětlitelnost a interpretovatelnost výsledků.

Kapitola 1

Rešerše

V rešeršní části práce analyzujeme komerční produkty a odbornou literaturu, která se týká problému pochopení segmentů uživatelů a jejich chování. Dále pak uvedeme problém doporučování a vysvětlíme základní přístup pomocí maticové faktorizace. Rozebereme i pokročile *state-of-the-art* přístupy jako model (N)EASE [9], [10], *Hypercuboids* [5] společnosti Alibaba a zhodnotíme jejich potenciál pro využití ne při doporučování, ale při segmentaci uživatelů.

Ve druhé části rešerše důkladně popíšeme teoretické základy využitých vizualizačních technik. Nejedná se pouze o obvykle využívané t-SNE [6], ale i Samoorganizační mapy a nově publikovaný (březen 2021) framework *Minimum-Distortion Embedding*.

1.1 Komerční nástroje pro analýzu chování uživatelů

Některé komerční nástroje se snaží uživatelům (v tomto případě hlavně mediálním editorům) prezentovat agregované údaje v reálném čase. Produkt společnosti IO Technologies [1]. Po publikování článku tak může editor sledovat, kolik lidí článek vidělo, jak dlouhý čas průměrně strávili na stránce (dočetli až do konce?), počet interakcí na sociálních médiích, procenta recirkulace - kolik uživatelů po dočtení pokračuje na jiný článek a další ukazatele. [11]

Podobně fungujícím nástrojem je i ten od firmy Datapine [2]. Opět nabízí množství dashboardů, které v zásadě agregují všechna možná data o provozu na webu. Zde dashboardy opět nabízí základní informace typu nejčtenějších článků, počtu návštěv, informací o čtenářích jako pohlaví a věk. Další monitorovanou oblastí je dosah článků na sociálních sítích - sdílení, komentáře, „líbí se mi“ a nově sledující.

[1] https://www.public.iotechnologies.com/
[2] https://www.datapine.com/

Výše zmíněné nástroje nabízejí více méně velké množství různých koláčových grafů a histogramů. Zde nechceme v nejmenším rozporovat jejich užitečnost, ta je v praxi rozhodně prokázána [1], je ale snad možné říct, že neposkytují celostní vhled a odpověď na otázku po tom, kdo jsou uživatelé toho či onoho webu a na jaké podskupiny se dělí.

Segmentace uživatelů

Přístupem, který se snaží odpovídat na otázky o publiku ve více celostní podobě, je *segmentace publika*. Typicky je publikum možné segmentovat na základě demografických údajů (vzdělání, povolání, velikost rodiny, pohlaví, ...), behaviorálních (loajalita, jak často se uživatel vrací, status uživatele) a samozřejmě geografických, tedy odkud se uživatelé připojují či pochází. [12]

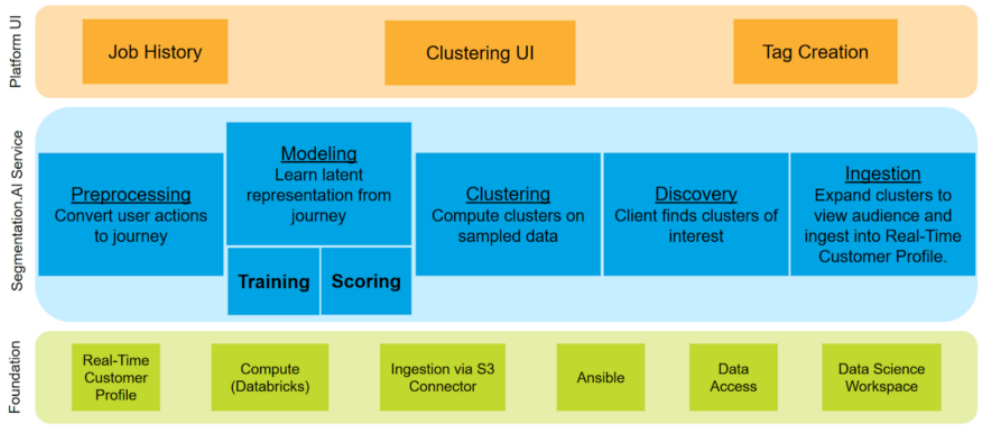
Nalezneme-li segmenty publika podle výše zmíněných kritérií, můžeme se dále zabývat tím, s jakým typem obsahu uživatelé z jednotlivých segmentů (při pohledu z úhlu strojového učení můžeme říci shluků) interagují. Pokud jsou segmenty dostatečně vyprofilované i obsahově, můžeme následně pro nový článek predikovat, jaký dopad bude mít na uživatele různých segmentů. Konkrétně je možné se ptát např. na to, jaký bude dopad článku na skupinu mužů ve věku okolo 35 let žijících v USA, kteří často čtou články z kategorií Byznys a Technologie oproti ženám ve věku okolo 20 let žijících ve Velké Británii, které zajímají kategorie Psychologie a Životní styl. Typicky chceme znát, kolik procent uživatelů z takto popsaného segmentu by mohl článek zaujmout. [12]

Nevýhodou předcházejícího přístupu je nutnost ručního nastavení pravidel pro segmentaci, tedy věku, lokace a zájmů z předchozího příkladu. Nástroj Segmentation.ai (Crowd.ai) [3] od společnosti Adobe se ji pokouší překonat zajímavým způsobem. Ten nejdříve vytvoří časovou osu interakcí uživatele, kterou následně zakóduje do jednoho 16rozměrného vektoru reálných čísel. Z těch vytvoří shluky, které uživateli zobrazí a umožní je interaktivně procházet. Architekturu systému můžeme vidět na obrázku 1.1.

V případě systému Segmentation.ai je asi nejzajímavějším prvkem právě modelování uživatelů v podobě vektorů fixní délky. Vzhledem k tomu, že se jedná o komerční produkt, není samozřejmě mnoho dostupných informací o tom, jak přesně tento proces probíhá. V popularizačním článku se dočteme pouze to, že je využívána nějaká forma hlubokého učení spolu s knihovnou Tensorflow [13]. Blíže specifikovaný není ani shlukovací algoritmus, který je používán pro odhalování segmentů. Užitečnou informací však je, že pro vizualizaci shluků ve 2D je využívána technika ShapeVis [14].

Systémem, který k analýze obsahu a uživatelů přistupuje z trochu odlišného hlediska, je nástroj *Conversation Clusters* [15]. Jedná se v zásadě o analýzu (shlukování) kolekce textových dokumentů, které jsou v tomto případě no-

[3] https://crowdai.com/



Obrázek 1.1: Architektura systému Segmentation.ai [13]

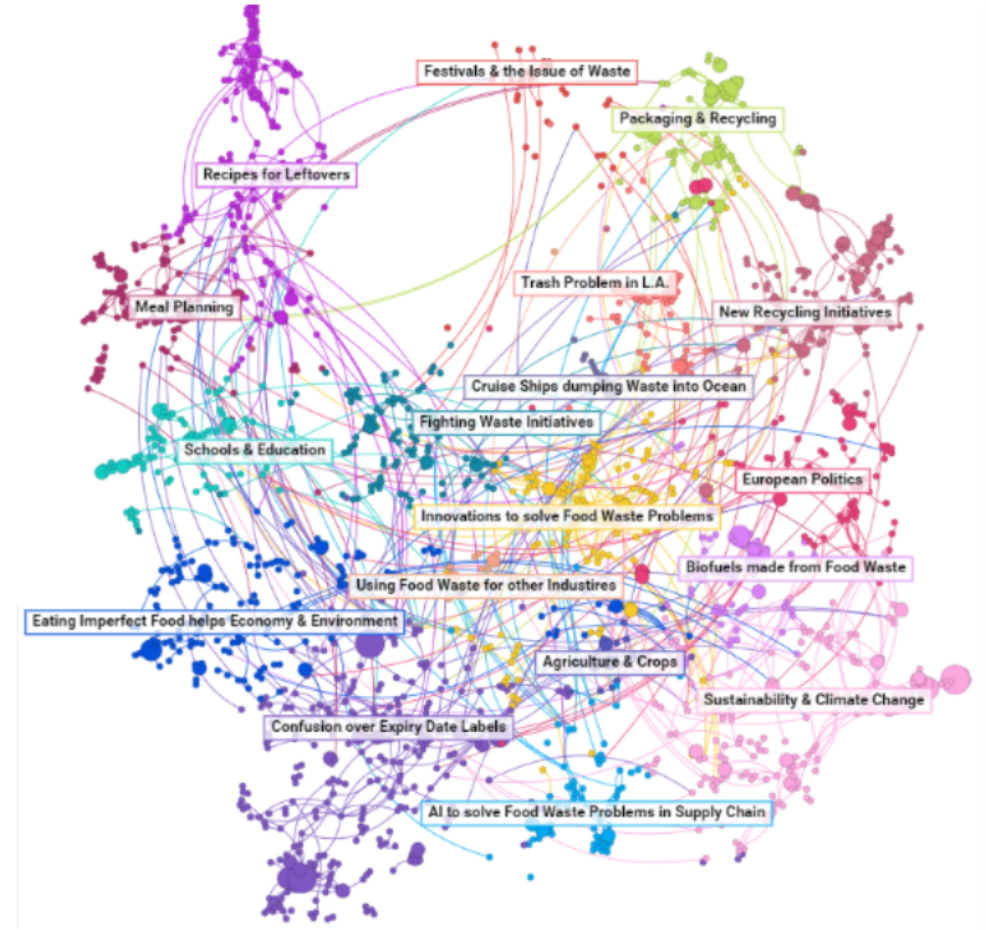
vinové články z nějaké oblasti, a tím mapovat „konverzaci“, která aktuálně probíhá na toto téma (tedy např. plýtvání jídlem a redukce odpadu). Pro tvůrce obsahu je tento vhled zajímavý, protože jim nabízí globální perspektivu a lepší porozumění toho, jaký obsah má šanci na úspěch. Nástroj opět umožňuje interaktivní procházení jednotlivých shluků. Na obrázku 1.2 můžeme vidět výstup systému při analýze zmiňovaného tématu. Je vidět, že systém klade důraz na zobrazení vztahů mezi podtématy.

1.2 Analýza chování uživatelů v odborné literatuře

Ve vědecké literatuře je možné nalézt větší rozmanitost přístupů k analýze chování uživatelů. Práce „PeopleGarden: Creating Data Portraits for Users“ [3] se zabývá uživateli internetových fór a diskusních skupin. Autoři identifikují 4 základní otázky, které zajímají nově příchozího potenciálního uživatele, který se chce přidat k dané skupině [3]:

1. Jsou zdejší členové opravdu aktivní - přispívají často?
2. Interagují členové spolu - odpovídají si navzájem?
3. Mají tu rádi nováčky?
4. Kteří uživatelé tu mají „expertní“ status - jsou tu již dlouho a vytvořili hodně příspěvků?

Pro co nejkomplexnější vizuální zachycení uživatelů vytvořili autoři *datové portréty* (data portrait), tedy vyzualizaci uživatelů založenou na interakčních datech, která může reprezentovat jak jednoho, tak skupinu uživatelů. Uživatel je v tomto případě reprezentován množinou objektů, se kterými interagoval, tedy hlavně příspěvky či zprávy, které vytvořil (to není obecně nutný



Obrázek 1.2: Výstup systému Conversation Clusters [15].

předpoklad, je to platné pouze pro tuto práci). Je nutné provést dva kroky - identifikovat důležité atributy objektů uživatele a rozhodnout se, jak je vizualizovat.

Pro vizualizaci interakcí jednoho uživatele je možné využít motiv květiny či květu. Každý okvětní lístek reprezentuje jeden příspěvek. Lístky v čase přibývají ve směru hodinových ručiček. Celou reprezentaci uživatele pak můžeme vidět na obráku 1.2. Portréty všech uživatelů z jedné skupiny je následně možné poskládat k sobě do „zahrady“, která reprezentuje jednu diskusní skupinu. Z takového portrétu je pak možné rychle zodpovědět všechny 4 výše zmíněné otázky.

Obrázek 1.3: Reprezentace uživatelů ve formě schematických květů. Vlevo vidíme přibývání příspěvků - nejnovější jsou nejsytější. Vpravo je pak kompletní reprezentace několika uživatelů. Žluté objekty na okrajích (odvozené od pestíků) značí počet odpovědí na daný příspěvek. Fialová barva pak určuje, jestli byl samotný příspěvek odpovědí, nebo jestli započal samostatné diskusní vlákno. [3]

Modelování uživatelů pomocí metody LDA

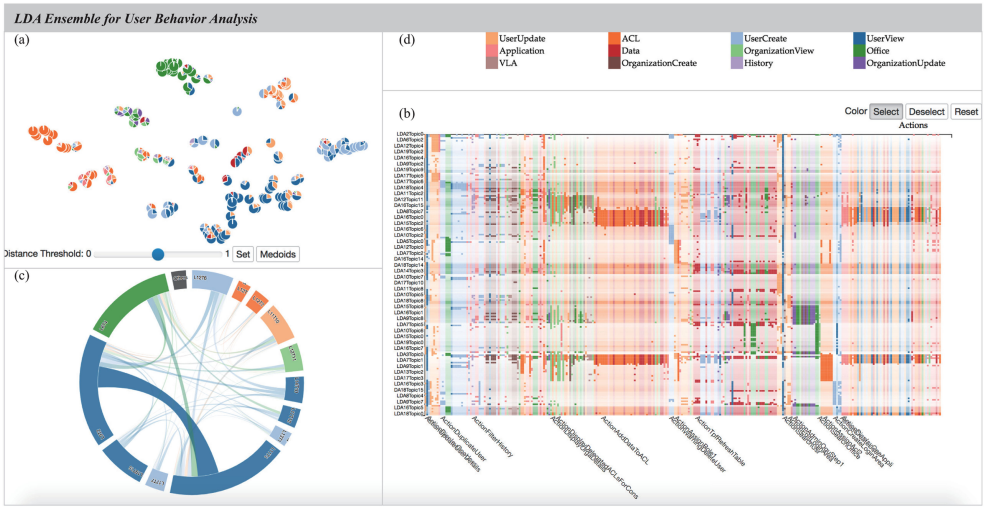
Metoda LDA (Latent Dirichlet Allocation) pochází z oblasti zvané modelování témat (topic modelling). Témata v tomto smyslu nejsou známá, je však předem nutné určit jejich počet. Předpokládáme, že každý dokument je asociován se směsí témat a každé téma je distribuce nad slovy ze slovní zásoby. Každé slovo je tedy asociováno s nějakou pravděpodobností, že se v tématu vyskytne. Distribuce témat, dokumentů a slov je modelována pomocí Dirichletova rozdělní, odtud název metody. [16]

V praxi se LDA nejčastěji využívá právě pro práci s databází textových dokumentů, není to ale pravidlem, jako framework funguje i v jiných doménech. Je tak možné využít ji i pro analýzu chování zákazníků, jen je potřeba najít vhodné namapování. Množinu všech možných akcí, které může uživatel provést, je možné považovat za slovní zásobu a jednotlivé session, tedy sekvence akcí, můžeme označit za dokument. Při takovéto formulaci problému lze pak z dat extrahovat „témata“, která odpovídají typům chování uživatelů. Kvalitu výsledků je potom samozřejmě náročnější posoudit, jelikož význam „tématu“ sestávajícího se z dominantních akcí typu vyhledání uživatele a otevření tabulky nemusí na první pohled zřejmý. [2]

Nevýhodou metody je nutnost předem nastavit množství latentních témat, která chceme objevit. Tento problém se pokouší autoři překonat tím, že LDA spustí několikrát s jiným parametrem n (tedy počtem témat) a následně kombinují výsledky do *ensemblu*, z kterého získají množinu kandidátních témat. Pro průzkum výsledků je pak nutná nějaká forma vizualizace. Různé formy můžeme vidět na obrázku 1.4.

Pro nás může být zajímavá hlavně vizualizace 1.4a, na které vidíme projekci jednotlivých odhalených témat pomocí metody t-SNE (bude rozvedena dále). Každé téma (n-rozměrný vektor) je reprezentováno jedním koláčovým grafem. Jednotlivé barevné výseče pak odpovídají podílu dané třídy akcí v tomto tématu. Je zde možné pozorovat jasné vymezení shluků, což je slibné pro další analýzu - lze očekávat, že shluky budou informativní. [2]

Oba výše zmíněné přístupy mohou bezesporu poskytovat užitečný vhled



Obrázek 1.4: Různé formy vizualizace objevených témat. Tématům je možné přiřadit třídy (d), projektovat je 2D prostoru (a), nebo vytvořit témat a akcí (b). Řádky v matici odpovídají jednotlivým latentním tématům, sloupce pak konkrétním akcím uživatele. Sytost barvy pak odpovídá pravděpodobnosti výskytu akce v daném tématu. [2]

do aktivity uživatelů. U služeb zaměřených na konzumaci obsahu, tedy Netflix, Youtube ale právě i novinové servery je ale jejich přínos pro pochopení uživatelské základny spíše omezený. Akce uživatelů novinových serverů budou podle všeho dost jednotvárné - klikání na články, možná rozkliknutí rubriky. Pro producenty obsahu může být určitě cenné zkoumat uživatele na základě sémantiky obsahu, se kterým interagují.

1.3 Z pohledu doporučovacích systémů

Nabídka obsahu na internetu roste stále rychlejším tempem (internetové obchody, streamovací služby, ...), díky čemuž vzniká potřeba usnadnit uživatelům navigaci v takto rozsáhlé nabídce. V tuto chvíli do hry vstupují doporučovací systémy. Ty pracují se dvěma základními typy entit - *uživatelé* a *položky*. Cílem je pak poskytnout uživateli seznam položek, které by jej mohly jakkoli zajímat. To se děje typicky na základě jednak uživatelova chování v minulosti, jednak na základě libovolného množství různých dalších metadat.

Nejčastěji voleným přístupem při doporučování je *kolaborativní filtrování*, které spoléhá čistě na data o interakcích uživatelů s položkami a je doménově nezávislé. Samotné kolaborativní filtrování je pak celou rodinou metod, které se dají rozdělit na dvě podskupiny. První skupinou jsou metody založené na prohledávání okolí položek, nebo případně uživatelů. V jejich případě do-

poručovací systém predikuje uživatelovo hodnocení nové položky na základě podobných položek (položek z okolí).

Druhým možným přístupem je vytvoření latentní reprezentace uživatelů a položek. Takovou reprezentací se rozumí vektor reálných čísel o dimenzi řádově 50 až 300 (velmi orientační čísla pro představu), které jsou vytvořeny na základě předchozích interakcí uživatele. Jednotlivé dimenze těchto vektorů teoreticky mohou mít nějaký konkrétní význam, například míru příslušnosti k nějakému žánru, obecně ale tyto vektory nelze nijak přímočaře interpretovat. Jediné, na co se spoléháme, je to, že tyto vektory jako celek dobře vystihují danou položku či uživatele. [17]

1.3.1 Sledování interakcí a matice hodnocení

Při sledování interakcí uživatelů s obsahem můžeme rozlišit dva typy zpětné vazby, kterou uživatelé dávají systému. Jedná se o *explicitní* a *implicitní* feedback. V případě explicitní zpětné vazby jde o situaci, kdy uživatel vědomě hodnotí danou položku, tedy uděluje procenta, skóre, hvězdičky, či pouze „líbí se mi“.

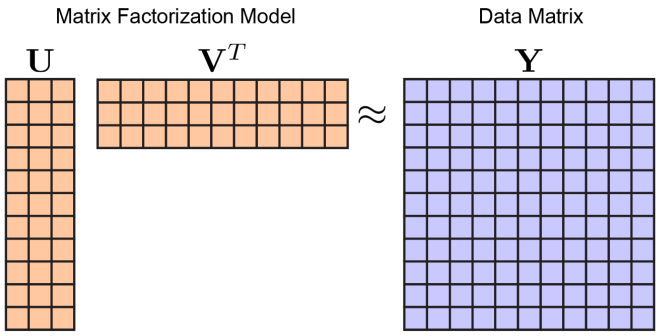
Uživatelé obvykle explicitně ohodnotí jen relativně malé množství položek, se kterými přijdou do styku, je proto přínosné zaměřit se na to, jestli se v jejich chování nedají objevit nějaké implicitní informace o preferenci položek. Typické interakce, ze kterých je možné usuzovat na uživatelovu preferenci jsou nákup položky, vytvoření záložky či uložení položky, historie vyhledávání, či jen pohyb myši. Velmi často lze také započítat pouhé zobrazení detailu položky (přehrání videa či skladby, přečtení novinového článku, ...).

Při sledování interagovaných položek je potřeba nastavit další kritéria, podle kterých posuzovat, jestli se uživateli daná položka opravdu líbila. Typickým přístupem je nastavení nějaké hranice toho, jak dlouho musí uživatel na stránce strávit. Pokud si uživatel poslechne pouze prvních 10 sekund skladby nebo zhlédne pouze prvním pár minut filmu, není asi moudré považovat to za implicitní vyjádření preferencí, ba naopak. [17]

Z takto nasbírané zpětné vazby můžeme sestavit *matici hodnocení* R o rozměrech $m \times n$, kdy čísla m a n značí počet uživatelů, respektive položek. Samozřejmě ani po využití implicitní zpětné vazby neznáme hodnoty pro zdaleka všechny páry uživatel-položka, proto se v matici nacházejí hodnoty z množiny $\mathbb{R} \cup \{?\}$. Cílem je pak predikovat neznámé hodnoty z této matice $r_{u,i} =?$ za pomoci hodnot známých, tedy $r_{u,i} \neq ?$.

1.3.2 Faktorizace matic

Z lineární algebry víme, že součinem dvou matic $A \in \mathbb{R}^{m,l}$ a $B \in \mathbb{R}^{l,n}$ vznikne matice $D \in \mathbb{R}^{m,n}$. Pro faktorizaci matice hodnocení R tedy obecně chceme najít matice $U \in \mathbb{R}^{m,l}$ a $V \in \mathbb{R}^{l,n}$, $l \ll n$, $l \ll m$, takové, že jejich maticový součin dobře aproximuje známé hodnoty z matice R.



Obrázek 1.5: Grafické znázornění faktorizace matic. Matice v maticích U a V jsou známé všechny prvky. Celkový počet prvků je ale signifikantně menší než v matici R (zde Y).

Označíme-li u-tý rádek matice U jako p_u a i-tý sloupec matice V jako q_i, je pak aproximace $r_{u,i}$ dána skalárním součinem $p_u^T q_i$. Chyba takové aproximace se obvykle měří jako kvadrát odchylky, tedy $(r_{u,i} - p_u^T q_i)^2$. Nazveme li množinu všech známých položek $r_{u,i} \neq ?$ $\mathcal{K}$, lze formálně hledání matic U a V zapsat jako:

$$\arg\min_{U,V} \sum_{(u,i)\in\mathcal{K}} (r_{u,i} - p_u^T q_i)^2.$$

V praxi se často chceme vyhnout tomu, aby hodnoty v maticích U a V byly příliš vysoké, což by mohlo mít za následek přeučení. Při faktorizaci matic znamená pojem přeučení to, že by výsledné matice sice velmi dobře aproximovaly známé hodnoty v matici R, ale byly zcela nepoužitelné pro predikci těch neznámých. Při hledání matic tudíž chceme zohlednit to, že preferujeme nižší hodnoty koeficientů - to se nazývá regularizace. Nejčastěji používaná metoda pro regularizaci je takzvaná L2 regularizace. Formálně pak úlohu zapíšeme jako:

$$\arg\min_{U,V} \sum_{(u,i)\in\mathcal{K}} (r_{u,i} - p_u^T q_i)^2 + \lambda \left(\sum_i q_i^T q_i + \sum_u p_u^T p_u \right),$$

kdy $\lambda > 0$ je hyperparametr algoritmu, který řídí, jak silně regularizujeme. Z regularizačního členu je dále vidět, že při L2 regularizaci penalizujeme vektor parametrů jeho euklidovskou normou, která je dána např. vztahem $q_i^T q_i$.

Různé formy faktorizace matic byly a jsou velmi oblíbeným způsobem řešení problému doporučování jak v akademické, tak komerční sféře, existuje proto velké množství různých konkrétních algoritmů. Lze ovšem jmenovat dva základní přístupy k hledání matic U a V.

Singulární rozklad

Pokud bychom znali celou matici R, byla by úloha faktorizace v uvozovkách triviální. Lineární algebra nám totiž poskytuje návod, jak přesně tento rozklad provést. Singulární rozklad (Singular value decomposition, SVD) umožňuje rozložit matici M o rozměrech $m \times n$ a hodnosti r, $r \leq m \leq n$ takto: $M = U\Sigma V^T$, kdy U a V jsou ortogonální matice, zatímco Σ je diagonální a nese *singulární hodnoty*, které jsou kladné a na diagonále seřazené sestupně. [18]

V diagonální matici Σ můžeme dále ponechat pouze l nejvyšších singulárních hodnot (zbytek uvažujeme nastavený na 0). Díky tomu má také smysl ponechat pouze prvních l sloupců matice U a prvních l řádků matice V, jelikož zbytek bychom násobili nulou. Následně můžeme psát $M = \left(U\Sigma^{1/2}\right)\left(\Sigma^{1/2}V\right)$. Díky ponechání pouze prvním l singulárních hodnot mají matice v součinu rozměry právě $m \times l$, respektive $l \times n$, a tudíž jsou přesně naším hledaným rozkladem. Snažíme-li se minimalizovat sumu čtverců reziduí, je to podle Eckhart-Young theorému ta vůbec nejlepší aproximace, kterou můžeme najít. [18]

V praxi bohužel nikdy celou matici R neznáme, proto není možné postupovat takto jednoduše. Existují v zásadě dvě možnosti, jak tento problém překonat. Prvním způsobem je považovat chybějící hodnoty za implicitní nuly, čímž se z matice R stává *řídká* matice, na kterou můžeme aplikovat klasické SVD solvery. Například knohovny `scipy` [19] a `scikit-learn` [20] pro to poskytují efektivní implementaci. Druhou, složitější možností je pokusit se aproximovat SVD na základě dat, která známe, tedy vlastně aproximace aproximace. Tu je možné provádět mnoha způsoby, zde uvádíme gradientní sestup a metodu nejmenších čtverců.

Gradientní sestup

Při řešení této úlohy máme přístup ke gradientu, nabízí se tedy řešení pomocí gradientního sestupu. Uvážíme-li ztrátovou funkci $F(U, V)$ totožnou s výše zmíněný optimalizačním problémem s L2 regularizací, dostaneme tyto parciální derivace:

$$\frac{F}{q_i^{(j)}}(U, V) = -2(r_{u,i} - p_u^T q_i)p_u^{(j)} + \lambda q_i^{(j)}$$

a

$$\frac{F}{p_u^{(j)}}(U, V) = -2(r_{u,i} - p_u^T q_i)q_u^{(j)} + \lambda p_u^{(j)},$$

kde $p_u^{(j)}$ značí j-tou složku u-tého vektoru matice U a $q_i^{(j)}$ značí j-tou složku i-tého vektoru matice V. Vlastní učení pak začíná náhodnou inicializací matic U a V. Následně pak postupujeme iterativně, kdy obě matice updatujeme

podle pravidel:

$$q_i \leftarrow q_i + \gamma \left(\left(r_{u,i} - p_u^T q_i \right) p_u + \lambda q_i \right)$$

a

$$p_u \leftarrow p_u + \gamma \left(\left(r_{u,i} - p_u^T q_i \right) q_i + \lambda p_u, \right)$$

kde γ je uživatelem definovaná konstanta. Jako obvykle při gradientním sestupu se jedná o nekonvexní problém, tudíž konvergence může být pomalá, nemusí se dostavit vůbec a můžeme uváznout v lokálním optimu. [17]

Alternating least squares (ALS)

Tato metoda převádí problém faktorizace na řešení řady konvexních úloh. Pracuje tak, že střídavě fixuje jednu matici a updatuje hodnoty pouze matice druhé. Při tomto přístupu se fixovaná matice stává maticí dat a známé hodnoty z matice hodnocení cílovou proměnnou. Hodnoty v řádcích druhé matice se pak stávají hledanými koeficienty lineární regrese. Pokud označíme vektor známých hodnot z j-tého sloupce matice R jako $R^{\mathcal{K}}_{:,j}$, můžeme při standardní formulaci lineární regrese zapsat heldané minimum takto:

$$\hat{q}_i = \left(\left(U^{\mathcal{K}_j} \right)^T U^{\mathcal{K}_j} \right)^{-1} \left(U^{\mathcal{K}_j} \right)^T R^{\mathcal{K}}_{:,j}.$$

Aplikujeme-li dále výše zmíněnou L2 regularizaci, dostáváme známý problém hřebenové regrese, u které leží minimum v bodě (při zafixování matice U)

$$\hat{q}_i = \left(\left(U^{\mathcal{K}_j} \right)^T U^{\mathcal{K}_j} + \gamma I \right)^{-1} \left(U^{\mathcal{K}_j} \right)^T R^{\mathcal{K}}_{:,j},$$

kdy I je jednotková matice o rozměrech $l \times l$. Při zafixování matice V je pak postup v zásadě totožný, akorát v poslední zmíněné rovnici bude matice hrát roli matice dat matice V místo U a aktualizován bude vektor $\hat{p}_j$. Celý průběh učení se pak řídí tímto jednoduchým postupem:

1. Náhodně inicializuj matice U a V.
2. Dokud není splněna podmínka pro konvergenci:
   a) Aktualizuj matici U s využitím hřebenové regrese.
   b) Aktualizuj matici V s využitím hřebenové regrese.

Vzhledem k tomu, že jsou faktorizační metody široce využívané v praxi, objevuje se velké množství různých vylepšení a změn. Velká část pozorovaného rozptylu v hodnocení položky není vysvětlitelná čistě z interakcí, ale pramení ze zaujetí (bias) daného uživatele (nebo i položky). Skalární součin $q_i^T p_u$ tedy nemusí být dostatečný pro predikci hodnocení.

Pro odhad hodnocení můžeme dále uvážit průměrné hodnocení všech položek, zaujetí položky, tedy to, jak se její průměrné hodnocení liší oproti globálnímu

průměru, a zaujetí uživatele, tedy rozdíl průměru všech hodnocení tohoto uživatele oproti globálu. Predikci pak sestavíme vztahem $\hat{r}_{u,i} = \mu + b_i + b_u + q_i^T p_u$.

1.3.3 Latentní reprezentace

Výstupy z faktorizace matic nemusíme používat pouze přímo pro predikci hodnocení položky uživatelem. Vektory nacházející se v maticích U a V můžeme považovat za reprezentaci uživatelů, potažmo položek, v *latentním prostoru*. Oproti řídkým vektorům z matice hodnocení nemá žádná ze složek latentního vektoru přímo interpretovatelný význam. Takovýmto reprezentacím vektory reálných čísel v latentním prostoru se obvykle říká *embedding*.

Maticová faktorizace je v případě uživatelů a položek přirozeným zdrojem takových embeddingů, není ale v žádném případě jediným možným. [21]. Často se využívají neuronové sítě, dost možná nejznámějším reprezentanty jsou `Word2Vec` [22] a `Fasttext` [23]. Existuje i snaha přenést přímo tyto přístupy do světa doporučování v podobě `Item2Vec` [24]. Obecně je v literatuře týkající se doporučovacích systémů patrný velký zájem o vytvoření co možná nejlepší latentní reprezentace položek a uživatelů a objevují se stále nové přístupy. Možné je dokonce reprezentovat uživatele ne jen jako bod v prostoru, ale jako objekt v tomto prostoru (hypercuboid) [5].

Výhodou embeddingů je, že jsou do určité míry univerzální a dají se použít k různým úlohám. Jsou vlastně jen jinou reprezentací entit, se kterými pracujeme. Neumíme je sice přímo interpretovat, spoléháme ale na to, že své entity reprezentují „dobře“. Je tedy možné očekávat, že vektory podobných entit budou podobné, respektive, že díky podobným vektorům odhalíme podobné entity.

Míra podobnosti či vzdálenosti však zůstává důležitou otázkou sama o sobě. S rostoucí dimenzionalitou přestává euklidovská vzdálenost být užitečná [25] a je potřeba hledat jiné míry podobnosti. Obvykle je využíván úhel, který dva vektory svírají, respektive cosinus vektorů $\vec{u}$ a $\vec{v}$:

$$cos(\vec{u}, \vec{v}) = \frac{\vec{u} \cdot \vec{v}}{\|\vec{u}\| \cdot \|\vec{v}\|}.$$

V této práci je chceme prozkoumat možnost využití latentní reprezentace uživatelů a položek k jinému úkolu, než je doporučování. Z hlediska porozumění uživatelské základně bude zajímavé pokusit se vizualizovat embeddingy jak uživatelů, tak položek, a tím se pokusit porozumět jejich struktuře.

t-SNE

t-Distributed Stochastic Neighborhood Embedding je technika pro redukci dimenzionality a vizualizaci, která je vhodná pro zobrazení vysoce dimenzionálních dat [6]. Oproti možná nejčastěji využívané metodě pro redukci

dimenzionality - PCA, Principal Component Analysis - je t-SNE nelineární projekce. PCA se snaží vysvětlit co nejvyšší množství rozptylu v datech a zachovává velké vzájemné vzdálenosti. Oproti tomu t-SNE zachovává pouze malé vzájemné vzdálenosti. Metoda t-SNE je vhodná hlavně v případě, že vnitřní struktura dat je nelineární [26].

První krok algoritmu je změření podobností bodů v původním vysoce dimenzionálním prostoru. To si můžeme představit jako konstrukci Gaussovy křivky nad každým bodem. Následně změříme hodnotu hustoty pravděpodobnosti nad všemi ostatními body z datasetu a znormalizujeme. Tím dostaneme pravděpodobnosti, které jsou úměrné podobnostem v datech. Rozptyl zkonstruované Gaussovy křivky se řídí parametrem algoritmu zvaným *perplexity*, jehož doporučený rozsah je přibližně 5 až 50. [26]

Druhý krok je podobný tomu prvnímu, akorát použijeme Studentovo t-rozdělení s jedním stupněm volnosti oproti normálnímu rozdělení. Tím dostaneme druhý soubor pravděpodobností, tentokrát v nízkodimenzionálním prostoru. Oproti normálnímu rozdělení má Studentovo t-rozdělení lepší vlastnosti pro modelování větších vzdáleností. [26]

Následně chceme, aby pravděpodobnosti v novém prostoru s nižší dimenzí reflektovaly ty v původním. Pro změření toho, jak si odpovídají dvě distribuce pravděpodobnosti je využívána tzv. Kullback-Leiblerova divergence, kterou vypočteme jako:

$$D_{KL}(P||Q) = -\sum_i P(i) ln \frac{Q(i)}{P(i)},$$

kde P a Q jsou rozdělení pravděpodobnosti. Následně využijeme gradientní sestup k tomu, abychom optimalizovali ztrátovou funkci, kterou je v tomto případě právě KL divergence. [26]. Vzhledem k tomu, že algoritmus spoléhá na náhodnou inicializaci, jsou jeho výsledky nederministické. Různé běhy tedy mohou vyprodukovat různě vypadající výsledky různé kvality (hodnoty KL divergence. Je potom zcela legitimní vybrat si z nich ten nejlepší. [6]

1.3.4 Lepší interpretabilita maticové faktorizace

V předchozí části jsme viděli, že faktorizace matic nabízí sice dobrou a užitečnou reprezentaci uživatelů, hledat její konkrétní význam je ale těžké. Přístup nazvaný *Overlapping Co-Cluster Recommendation Algorithm* (OCULAR) [27] na to reaguje tak, že celou formulací trénování vynucuje lepší interpretabilitu jednotlivých složek vektorů latentní reprezentace.

Základem je detekce *co-clusters*, což jsou shluky uživatelů a produktů s podobnými vlastnostmi. Vzhledem k tomu, že uživatelé mohou mít více vlastností a produkty mohou uspokojovat více potřeb, je přirozené, že se tyto shluky mohou překrývat. Autoři se dále zabývají hlavně Bussiness to Bussiness (B2B) doporučováním, které má některá specifika. O produktech i zákaznících je obvykle k dispozici velké množství metadat, zároveň jsou ale kladeny velké nároky na vysvětlitelnost doporučení - firemním zákazníkům jsou obvykle

nabízeny produkty v rádově vyšších cenových relacích, je ale nutné vědět, proč jsou nabízeny. Přístup k metadatům umožňuje autorům zlepšení v řešení *cold start* problému - situace, kdy přichází nový uživatel bez interakční historie, pro vysvětlení doporučení jsou ale metadata důležitá pouze v tomto případě.

Systém OCULAR pak spočívá v reprezentaci uživatelů i produktů k-dimenzionálními vektory f_u, respektive f_i s tím, že j-tá složka těchto vektorů udává míru příslušnosti k j-tému shluku. Samozřejmě platí, že čím vyšší hodnota, tím silnější příslušnost, a 0 udává, že entita do shluku nepatří vůbec. Pracujeme opět v typické situaci s implicitní zpětnou vazbou, máme tedy k dispozici řídkou matici R, jejíž řádky odpovídají uživatelům, sloupce produktům, hodnota 1 značí pozitivní feedback (uživatel si zakoupil produkt) a 0 neznámou hodnotu. Modelové znázornění situace můžeme vidět na obrázku 1.3.4.

Případy s hodnou 1 v matici R nazveme pozitivní příklady. Pokud uživatel u a produkt i oba náleží do shluku, generuje tento shluk pozitivní příklad s pravděpodobností $1 - e^{-[f_u]_c[f_i]_c}$, kde $[f_i]_c$ značí míru příslušnosti produktu i do shluku c, tedy odpovídající složku vektoru f_i. Předpokládáme-li, že každý shluk generuje pozitivní příklad nezávisle, dostáváme [27]:

$$1 - P\left[r_{u,i} = 1\right] = \prod_c e^{-[f_u]_c[f_i]_c} = e^{-\langle f_u, f_i \rangle},$$

kde $\langle f_u, f_i \rangle$ značí skalární součin vektorů. Parametry modelu následně hledáme metodou nejvyšší věrohodnosti [27]:

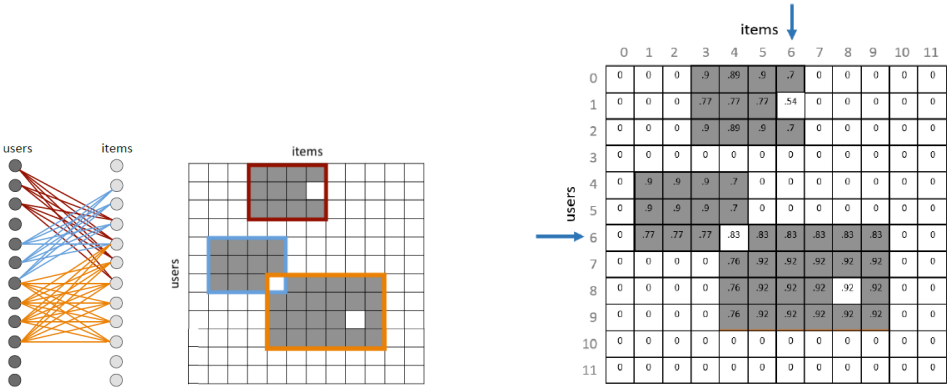
$$\mathcal{L} = \prod_{(u,i):r_{u,i}=1} \left(1 - e^{\langle f_u, f_i \rangle}\right) \prod_{(u,i):r_{u,i}=0} e^{\langle f_u, f_i \rangle}.$$

jako obvykle u těchto problémů je maximalizována log-likelihood a přidána L2 regularizace parametrů. Vlastní řešení pak probíhá stejně jako u metody faktorizace ALS, kdy v jednom kroku vždy fixujeme parametry f_i a odhadujeme f_u a ve druhém kroku naopak. Tím získáváme sérii konvexních problémů, jejichž řešení je relativně jednoduché.

Inference pak v praxi vypadá podobně, jako u faktorizace matic. Na obrázku 1.3.4 můžeme vidět, jak probíbá inference na modelových datech. Naučené (odhadnuté) parametry použijeme k odhadu pravděpodobnosti $P\left[r_{u,i}\right] = e^{-\langle f_u, f_i \rangle}$ u všech neznámých (nulových) hodnot a doporučíme ten produkt s nejvyšší odhadnutou odnotou.

1.3.5 Více než jeden vektor

Zajímavou prací je nedávná publikace autorů z čínského internetového obchodu Alibaba [5]. V ní autoři rozvíjejí inovativní způsob reprezentace uživatele, který není popsán pouze jedním vektorem, ale jedním či několika tělesy v n-rozměrném latentním prostoru. Autoři tato tělesa nazývají *hypercuboids*, česky bychom je mohli nazvat *nadkvádry*, tedy vícerozměrné kvádry. Takový kvádr



Obrázek 1.6: Modelová situace v systému OCULAR. Vlevo vidíme data a identifikované shluky. Někteří uživatelé i produkty mohou patřit do více shluků naráz, míra příslušnosti je dána jejich vektorem. Pro uživatele $u = 6$ dostaneme vektor $f_u = [0, 1.05, 1.25]$ a pro produkt $i = 4$ vektor$f_i = [1.39, 0.73, 0.82]$. Produkt tedy přísluší všem shlukům a uživatel pouze posledním dvěma. Odhad pravděpodobnosti pro tuto dvojici tedy bude relativně vysoký, jelikož se překrývají. Vpravo pak vidíme odhadnuté hodnoty s tím, že odhady pod určitým nízkým prahem byly vynulovány pro přehlednost. [27]

může být popsán dvěma vektory, svým centrem a offsetem, což je vektor, jehož přičtením k centru dostaneme „pravý horní roh“ kvádru. [5]

Reprezentovat uživatele ne jako bod, ale jako těleso nebo několik těles je zajímavý způsob, jak překonat obvyklý problém, na který modelování uživatelů v doporučovacích systémech ale i jinde naráželo. Uživatele typicky nezajímá jen jeden typ položek, ale minimálně několik. Problém se ještě prohlubuje v případě obchodů s tak širokou nabídkou zboží jako je právě Alibaba, potažmo Amazon či česká Alza. V různých kategoriích zboží se obvykle zákazníci rozhodují podle zcela různých kritérií, a mají tak i rozdílný rozsah ceny produktu, kterou jsou ochotni akceptovat. Doporučování založené na výše zmíněné faktorizaci matic sice funguje uspokojivě, pro takto komplexní případy však takto vytvořená reprezentace uživatele již nedostačuje. Jediný bod v prostoru položek nemůže reprezentovat všechny tyto aspekty a rozsahy. Autoři mají důvod domnívat se, že takováto pokročilá metoda modelování by tento nedostatek mohla překonat.

Navržený systém se pak učí reprezentaci uživatelů (tedy dva vektory reprezentující kvádr). Poloha středu nadkvádru v prostoru může dobře zachytit obecný zájem uživatele, naopak velikost tělesa určená druhým vektorem popisuje šíři tohoto zájmu uživatele, potažmo právě rozsah akceptované ceny či rozsah jiných atributů. Jednoho uživatele je dokonce možné reprezentovat ne jedním, ale hned několika nadkvádry, což dále zvyšuje expresivitu a flexibilitu

modelu. Formálně definujeme těleso reprezentující uživatele takto [5]:

$$Hypercuboid \equiv \left\{ p \in \mathbb{R}^d : c - f \preceq p \preceq c + f \right\},$$

kde $c \in \mathbb{R}^d$ reprezentuje střed a $f \in \mathbb{R}^d_{0+}$ reprezentuje offset nebo-li pravý horní roh nadkvádru.

Položky jsou reprezentovány jako obvykle vektory ve stejném prostoru. Vzniká otázka, jak měřit vzdálenost uživatelů, tedy útvarů, a položek. Definujeme ji tak, že pro uživatele u a položku i nejdříve nalezneme bod na povrchu nadkvádru, který je nejblíže vektoru položky, a ten označíme $p_{u,i}$. Rozeznáváme vnější vzdálenost $l_{out}(u, i)$ a vnitřní vzdálenost $l_{in}(u, i)$. Vnější vzdálenost odpovídá vzdálenosti bodu $p_{u,i}$ a vektoru položky, vnitřní pak vzdálenosti $p_{u,i}$ a centra. Celková vzdálenost je pak dána jednoduše vztahem [5]:

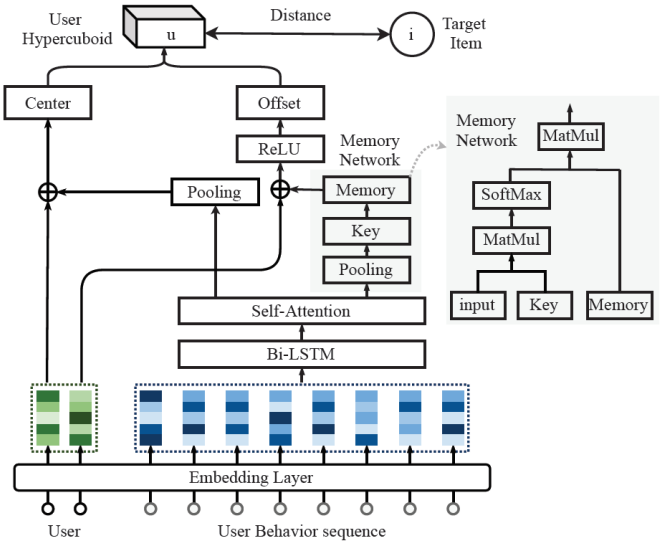
$$l(u, i) = l_{out}(u, i) + \gamma \cdot l_{in}(u, i),$$

kde koeficient γ určuje, jak moc k výsledné hodnotě přispěje vnitřní vzdálenost. Pokud teoreticky nastavíme $\gamma = 0$, znamenalo by to, že očekáváme, že uživatele bude položka zajímat, pokud leží v jeho nadkvádru bez ohledu na její přesnou polohu (vektor). Zde je důležité podotknout, že hovoříme-li o vzdálenosti dvou bodů v prostoru, máme vždy namysli euklidovskou vzdálenost. To může být obecně neobvyklé, většinou bývá v prostorech s vyšší dimenzionalitou využívána vzdálenost kosinová, ale zde je euklidovská vzdálenost nutná pro udržení geometrické interpretace.

Po technické stránce je model navržený autory relativně komplexní a složitý. Jedná se o neuronovou síť sestávající se z mnoha modulů. Vstupem je sekvence akcí uživatele, která je následně zakódována pomocí obousměrné LSTM vrstvy [28] a mechanismem self-attention, který umožňuje modelovat to, jak jednotlivé části vstupní sekvence ovlivňují další. Offset nadkvádru potřebuje dále obsáhnout velké množství informací. Například pro naučení se rozsahu cen, které uživatel akceptuje, je nutné uložit ceny všech položek, které si uživatel koupil. Pro obohacení paměťových schopností využívají autoři paměťovou síť v podobě key-value úložiště.

Paměťová síť svou konstrukcí v něčem připomíná pozornostní mechanismy. Sestává se z matice $M \in \mathbb{R}^{d \times N}$ reprezentující vlastní paměť a matice klíčů $K \in \mathbb{R}^{d \times N}$. Parametr N určuje kapacitu paměti. Označme výstup předchozího self-attention modulu jako $s_u^{(t)}$. Ten je pak použit pro získání klíče z matice klíčů jako $k = softmax(s_u^{(t)}, K)$ a klíčem získáme relevantní části z paměti: $m = k \cdot M^T$. Celou architekturu sítě můžeme vidět na obrázu 1.7

V každém učícím kroku je aktuální reprezentace porovnána s několika náhodně zvolenými pozitivními (uživatel s nimi interagoval) a negativními (uživatel neinteragoval) položkami. Cílem je pak minimalizovat vzdálenost nadkvádru uživatele od pozitivních příkladů a zároveň ji maximalizovat od



Obrázek 1.7: Architektura neuronové sítě využívané pro učení reprezentace nadkvádrů. [5]

negativních příkladů. Formálně zapíšeme ztrátovou funkci takto [5]:

$$\mathcal{L} = \sum_{(u,i)\in\mathcal{T}^+} \sum_{(u,j)\in\mathcal{T}^-} \max(0, l(u,i) + \lambda - l(u,j)),$$

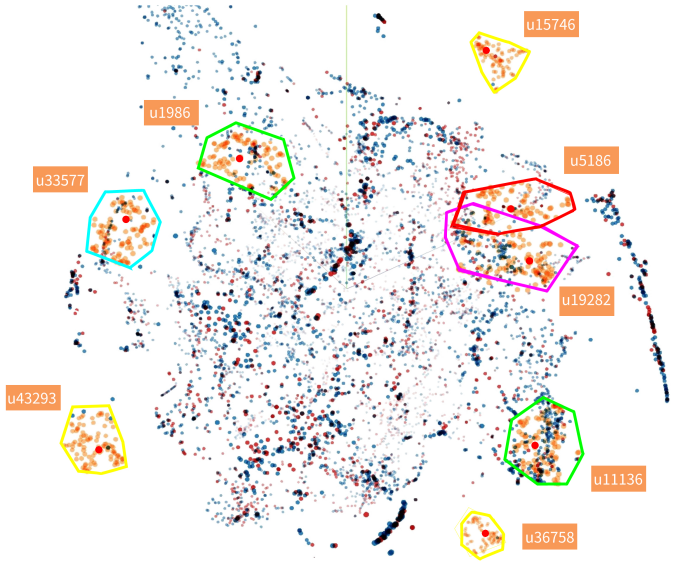
kde $\mathcal{T}^+$ a $\mathcal{T}^-$ značí množinu pozitivních, respektive negativních vzorků.

Možným rozšířením tohoto přístupu je reprezentovat uživatele ne jedním, ale celou sadou nadkvádrů. Ty mohou mít společný střed, nebo být zcela nezávislé. Obě varianty si ovšem vyžádají změnu způsobu, jakým je počítána vzdálenost bodů od těchto reprezentací uživatele. V případě jednoho středu sečteme všechny vnější vzdálenosti a tu nejnižší vnitřní, u nezávislých nadkvádrů s rozdílnými středy pak volíme minimální vzdálenost od všech z nich.

Nás zajímá opět hlavně vizualizace naučených reprezentací. Ta je v tuto chvíli o něco obtížnější, jelikož nepracujeme s n-rozměrnými body, ale s celými tělesy. Na obrázku 1.8 můžeme vidět pokus autorů o vizualizaci toho, kde leží centra jednotlivých nadkvádrů a které položky leží uvnitř. Jedná se o uživatele a položky z datasetu Amazon books. Výsledky jsou relativně úspěšné - je vidět, že systém se naučil velmi odlišné reprezentace pro jednotlivé uživatele.

1.3.6 Velmi mělký autoenkodér

Velmi často využívaným principem či architekturou modelu obecně ve strojovém učení je *autoenkodér*. Jedná se o nesupervizovaný model sestávající se ze dvou částí - *enkodér* a *dekodér*. Vstupem modelu je obvykle vektor vysoké dimenze a úkolem první části modelu je naučit se reprezentovat vstupy vektorem



Obrázek 1.8: Vizualizace naučených nadkvádrů. Červené body jsou centra, modré pak položky. Bylo náhodně vybráno 8 uživatelů a pro ně byly rovnoměrně náhodně vzorkovány body ležící uvnitř jejich reprezentace. Ty jsou vyznačeny oranžově. Barevné ohraničení je vytvořené ručně. [5]

o řádově nižší dimenzi. Dekodér pak naopak rekonstruuje vstup na základě této reprezentace. Typickým modelem pak může být vícevrstvý perceptron schematicky znázorněný na obrázku 1.9. Obě části modelu mohou disponovat libovolným počtem skrytých vrstev, důležitá je ovšem ta nejužší, která tvoří „hrdlo láhve“. Potom, co natrénujeme model pro replikaci vstupu, můžeme dále využít výstup nejužší skryté vrstvy jako kód vstupu, nebo-li jeho reprezentaci v latentním prostoru - embedding [29]. Toto je ta nejpřímočařejší aplikace autoenkodéru, není však jediná možná, uplatní se i při detekci anomálií [30]. V tomto případě je autoenkodér ponechaný vcelku a při inferenci zkoumáme velikost rekonstrukční chyby. Vysoká chyba pak může značit anomální vzorek.

Celou architekturu můžeme nadále zjednodušit a enkodér i dekodér reprezentovat každý pomocí jedné matice. Transformace jsou pak pouhé maticové násobení, tedy lineární operace. Takto sestavený lineární autoenkodér má pak velmi obdobnou funkci jako PCA [31] - pro minimalizaci rekonstrukční chyby musí hledat takovou projekci, která maximalizuje rozptyl [32]. Rozdíl mezi lineární a nelineární redukcí dimenzionality je znázorněný na obrázku 1.10.

V případě doporučovacích systémů často dosahují vysoké přesnosti modely založené na relativně mělkých neuronových sítích, což je signifikantní rozdíl oproti jiným oblastem výzkumu. Jmenovitě můžeme uvést počítačové vidění, kde se naopak setkáváme s velmi hlubokými sítěmi (modely `ResNet` [33], `VGGNet` [34], ...). V práci *Embarrassingly Shallow Autoencoders for Sparse*

Data je princip mělkého lineárního autoenkodéru dotažený do extrému, jelikož model `EASE` využívá pouze jednu jedinou matici.

EASE je doporučovací model, který operuje s běžnou maticí implicitních hodnocení $X \in \mathbb{R}^{|\mathcal{U}|\times|\mathcal{I}|}$, kde $\mathcal{U}$ a $\mathcal{I}$ značí množinu uživatelů, respektive položek. Matice R je pak typicky řídká a binární, i když to není podmínkou. Samotný model je pak dán jedinou maticí $B \in \mathbb{R}^{|\mathcal{I}|\times|\mathcal{I}|}$. K tomu, aby byl model donucen generalizovat, je přidáno jedno důležité omezení - na diagonále matice B, musí být nuly, tedy diag(B)=0, jinak by řešení bylo triviální. Predikce, tedy skóre $S_{u,j}$ pro položku $j \in \mathcal{I}$ a uživatele $u \in \mathcal{U}$ je pak definováno skalárním součinem [9]:

$$S_{u,j} = X_{u,\cdot} \cdot B_{\cdot,j},$$

kde $X_{u,\cdot}$ značí řádek u a $B_{\cdot,j}$ sloupec j. Učení se vah v matici B je pak definováno jako:

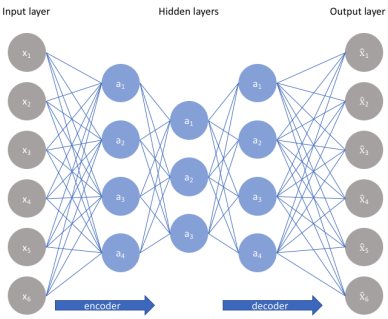
$$\begin{aligned} \min_{B} \quad & ||X - XB||_F^2 + \lambda\,||B||_F^2 \\ \text{s. t.} \quad & \text{diag}(B) = 0 \end{aligned}$$

Vidíme, že odchylka predikce je měřena Frobeniovou normou, minimalizujeme tedy čtverce odchylek, čímž se z hledání matice B stává konvexní problém, jehož řešení umíme jednoduše nalézt pomocí metody Lagrangeových multiplikátorů [9]. Autoři dále rozvíjejí úvahu, že využití jiných ztrátových funkcí by mohlo přinést zlepšení výsledků, ale znamenalo by významné zvýšení výpočetní náročnosti, proto se rozhodli pro tuto formulaci. Dále vidíme, že váhy v matici B jsou regularizovány pomocí L2 regularizace, která penalizuje výši jejich euklidovské normy. To přidává jeden hyperparametr λ, který může být optimalizován na validačním datasetu. Zcela zásadní je pak samozřejmě podmínka nulové diagonály, jinak by existovalo triviální řešení $B = I$, kde I je jednotková matice odpovídajících rozměrů. [9]

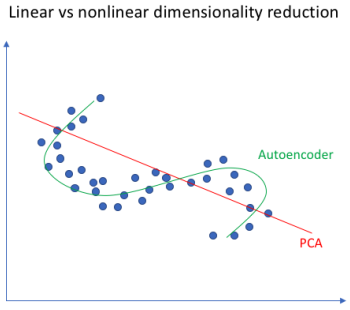
Autoři následně porovnávají EASE s několika dalšími state-of-the-art modely, které jsou řádově komplexnější. Ke srovnání využívají tři různé metriky - recall@n, $n \in \{20, 50\}$ a *Normalized Discounted Cumulative Gain*, NDCG@100. Recall@n značí, kolik z očekávaných položek bylo uživateli doporučeno v top-n modelem doporučených položkách. NDCG@n je typická metrika používaná pro evaluaci doporučování či Information Retrieval, tedy zjednodušeně řečeno vyhledávačů. Každé položce přiřadáme skóre, nebo-li relevanci, a samozřejmě chceme, aby se položky s co nejvyšší relevancí umístily co nejvýše. Nejprve můžeme zavést *Discounted Cumulative Gain*, což je de facto suma relevancí doporučených položek vážená jejich umístěním [36]:

$$DCG_n = \sum_{i=1}^{n} \frac{2^{rel_i} - 1}{\log_2(i+1)},$$

kde rel_i značí relevanci i-té položky. Zde nastává problém při porovnávání výsledků různých doporučovacích systémů, jelikož přiřazené relevance jsou arbitrární, a tudíž vyšší hodnota DCG může znamenat akorát to, že výzkumníci



Obrázek 1.9: Schematické naznačení základní architektury neuronové sítě typu autoenkodér. Prostřední skrytá vrstva slouží jako hrdlo láhve, její výstup je možné použít jako kompaktní kód vstupu. [35]



Obrázek 1.10: Rozdíl mezi lineární a nelineární redukcí dimenzionality. Dvoumaticový autoenkodér s lineární projekcí pak bude odpovídat zde znázorněnému PCA. [35]

zvolili řádově vyšší přiřazení hodnot. Chceme proto tuto hodnotu vhodným způsobem normalizovat. K tomu slouží *Ideální* DCG (IDCG@n) - položky seřadíme podle jejich relevance, čímž spočteme maximální DCG až k n-té pozici, tedy ten ideální, nejvyšší [36]:

$$IDCG_n = \sum_{i=1}^{|REL_n|} \frac{2^{rel_i} - 1}{\log_2(i+1)},$$

kde REL_n reprezentuje seznam relevantních položek seřazený sestupně podle jejich skóre až k n-té pozici. Takto získanou maximální možnou hodnotou následně znormalizujeme spočtený DCG, tedy [36]:

$$NDCG_n = \frac{DCG_n}{IDCG_n}$$

Pro porovnání EASE s dalšími modely si autoři vybrali 3 veřejně dostupné datasety:

- Movielens 20M: 136 677 uživatelů, 20 108 filmů s více než 10 miliony interakcí [37],
- Netflix prize: 463 435 uživatelů, 17 769 filmů a 57 milionů interakcí [38],
- Milion Song Dataset (MSD): 571 355 uživatelů, 41 140 skladeb a 34 milionů interakcí [39].

Mezi modely, se kterými byl EASE srovnáván, můžeme jmenovat *Sparse Linear Method* [40], *Weighted Matrix Factorization* [41] a *Collaborative Denoising Autoencoder* [42]. Dataset Movielens byl pro NEASE nejobtížnější - patřil vždy k nejlepším, ale v žádné ze 3 využitých metrik nedominoval, u zbývajících dvou datasetů dosáhl ale vždy nejvyššího skóre.

Při porovnání modelů je dále možné si všimnout, že EASE je oproti jiným modelům lepší v doporučování obecně méně populárních položek. Jedním že základních baseline přístupů při doporučování může být doporučit všem uživatelům nejpopulárnější položky z datasetu a sledovat, jaké úspěšnosti je možné dosáhnout tímto způsobem. Dobrý personalizovaný doporučovací systém by pak měl samozřejmě dosáhnout lepších hodnot než tento primitivní přístup, jelikož jinak by byla jeho aplikace zcela bezpředmětná. U datasetu MSD je tímto způsobem možné dosáhnout řádově nižších hodnot než u Movielens (NDCG@100 0,058 vs. 0,191) a právě u MSD EASE dominuje nejvíce. Autoři dále zkoumají, jak často modely doporučují položky ze které části spektra (ty často interagované a ty zřídka). Výsledkem je, že EASE položky, které mají méně interakcí, doporučuje opravdu častěji, než jiné modely [9]. Četnost doporučení samozřejmě stále klesá spolu s četností interakcí, pokles ale není tak výrazný.

Neural EASE - NEASE

V předchozí čáti jsme viděli, že i učením mělkého autoenkodéru formulovaným jakožto konvexní problém lze dosáhnout velmi dobrých výsledků, jsme ale velmi limitováni co se možnosti volby ztrátové funkce týče. Systém *Neural EASE* (NEASE) [10] překonává celé omezení tak, že problém zasadí do frameworku hlubokého učení a využije algoritmus zpětného šíření chyby. Díky tomu může chybu měřit jakoukoli diferencovatelnou funkcí.

Ve zmiňované práci využívají autoři model NEASE spolu s variančním autoenkodérem k tomu aby vytvořili ensemble, tedy několik modelů jejichž kombinací vznikne model lepší než všechny jeho části [43]. Nás ale zajímá NEASE sám o sobě kvůli potenciálu pro vizualizaci. Autoři dále experimentují se třemi ztrátovými funkcemi - MSE, kosinová vzdálenost a focal loss [44]. Poslední jmenovaná funkce je relativně novým způsobem, jak se vypořádat s problémem nerovnováhy tříd (velkou většinu datasetu tvoří negativní příklady oproti velmi malému počtu pozitivních). Formálně ji definujeme takto [45]:

$$FL(p_t) = -\alpha_t \, (1 - p_t)^{\gamma} \log(p_t),$$

kde α a γ jsou volitelné parametry. Motivací pro takovou formulaci je snížení hodnoty funkce pro správně klasifikované vzorky v případech, kde si model nebyl „jistý“. V případě jiné často využívané ztrátové funkce pro klasifikaci *cross-entropy* jsou správné, byť nejisté (predikovaná pravděpodobnost je blízko 0,5) stále penalizovány relativně vysokou hodnotou, v případě nerovnováhy tříd je

Ztrátová funkce	NDCG@100	Recall@20	Recall@50
MSE	0,425	0,393	0,523
Kosinová vzd.	0,431	0,403	0,532
Focal loss	0,377	0,343	0,426

Tabulka 1.1: Výsledky modelu NEASE při použití různých ztrátových funkcí na datasetu Movielens.

ale celý problém velmi obtížný a obvykle stačí se spokojit se správnou predikcí a není nutné ještě vynucovat její jistotu z hlediska modelu. [45]

Vliv parametru γ je ten, že s rostoucí hodnotou se snižuje výsledná hodnota funkce pro správně klasifikované vzorky, což umožňuje obrátit pozornost ke vzorkům, jejichž klasifikace je stále problematická. Při nastavení $\gamma = 0$ se z funkce stává de facto cross-entropy. Parametr α pak značí váhu vzorku, což je obecně používaná technika v případě nerovnováhy tříd i mimo focal loss. Autoři však upozorňují, že to samo o sobě nestačí pro rozlišení případů obtížných a jednoduchých pro klasifikaci. [44]

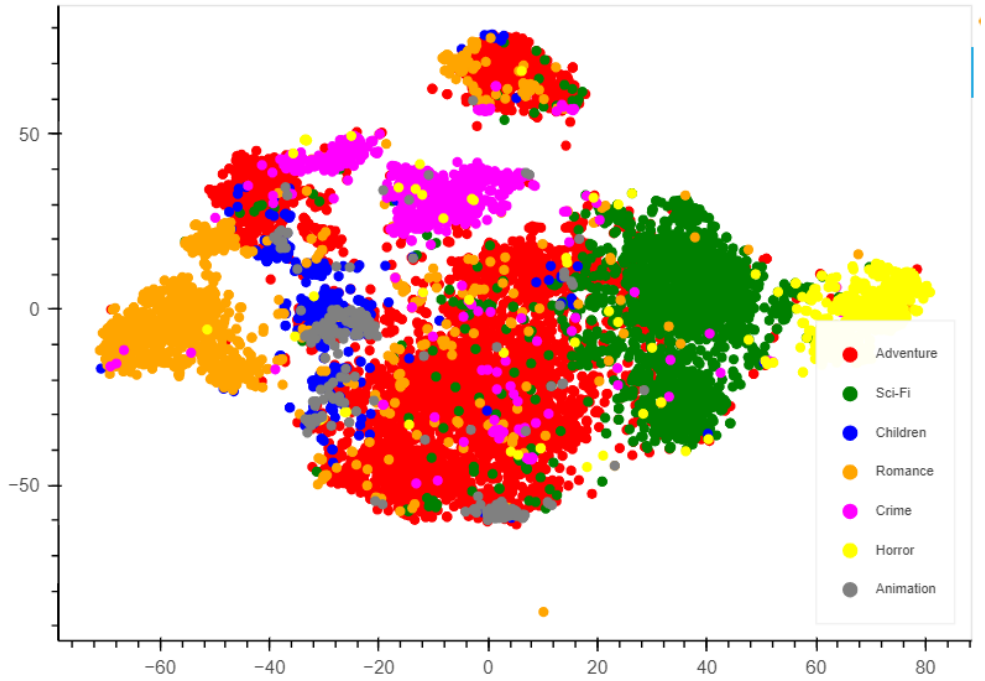
Podíváme-li se na výsledky v tabulce 1.1, můžeme vidět, že NEASE opravdu těží z možnosti využití jiných ztrátových funkcí. Nejlepších výsledků na datasetu Movielens dosahuje s kosinovou vzdáleností, s jejímž využitím překonává EASE ve všech zvolených metrikách.

Výsledkem trénování je matice o rozměru $|\mathcal{I}| \times |\mathcal{I}|$. Její řádky můžeme chápat jako svého druhu embedding či prostě reprezentaci jednotlivých položek. Zvolíme-li si metriku (vzhledem k dimenzionalitě kosinovou vzdálenost), můžeme následně spočítat distanční matici o stejných rozměrech. Vhledem k tomu, že t-SNE pracuje nad distribucemi vzdáleností, může taková matice být taktéž vstupem pro výpočet t-SNE a pro následnou vizualizaci.

V případě vizualizace uživatelů můžeme postupovat podobně. Matici interakcí uživatelů vynásobíme naučenou maticí B, čímž získáme predikci - vektory uživatelů o stejné dimenzi. Mezi nimi opět spočítáme matici kosinových vzdáleností a můžeme aplikovat t-SNE. Jak vypadají výsledky u vybraných uživatelů z datasetu Movielens vidíme na obrázku 1.11. Pro přiřazení kategorie uživateli byl zvolen nejčastější žánr filmů, se kterými uživatel interagoval. Velká většina uživatelů Movielens spadá do kategorií drama a komedie, to je však těžko odlišitelný mainstream, proto byli pro vizualizaci vybráni uživatelé z méně častých, lépe separovatelných kategorií.

1.4 Reprezentace uživatelů na základě jimi generovaného obsahu

Při pokusu o zachycení uživatelů ve světě sociálních sítí může být velmi hodnotné reprezentovat uživatele na základě obsahu, se kterým interagují. Obecně lze říci, že na sociálních sítích produkují uživatelé příspěvky v podobě textu



Obrázek 1.11: Zobrazení vybraných uživatelů z datasetu Movielens pomocí modelu NEASE. Obecně můžeme vidět dobrou míru separace tříd, byť ta je u různých tříd různá. Třídy Sci-fi a Horror jsou velmi dobře odlišitelná a zároveň leží blízko u sebe, což odpovídá i vlastnostem žánrů. Uživatelé sledující romantické filmy (oranžová) jsou taktéž velmi dobře separováni v jednom shluku, ale malé množství jich tvoří další shluky v blízkosti jiných uživatelů. U dětských a animovanách filmů (modrá a šedá) se situace zdá na první pohled hroší, jelikož tito uživatelé jsou silně promíšení. Zamyslíme-li se ale nad tím, kdo jsou tito uživatelé - v obou případech pravděpodobně dětští diváci sledující dětské animované filmy, může být jejich blízkost naopak žádoucí. U krimifilmů vidíme dva dobře separované shluky, kdy jeden z nich zasahuje k romantickým filmům. Nejrozsáhlejší kategorií jsou zde pak užívtelé s oblibou dobrodružných filmů (červená), kteří jsou rozdělení do tří oblastí. V té majoritní ve středu vizualizace pozorujeme velkou prolnutí se Sci-fi, které u této kateorie opět dává smysl. U dvou dalších meších shluků se ale jedná o skupiny uživatelů více kategorií a přisuzovat jim hlubší význam by už byla tvrzení stavěná na velmi mělkých základech.

nebo audiovizuálního obsahu (fotografie, obrázky vlastní tvorby, videa, ...). Autoři metody Author2Vec [4] se zaměřili na textové příspěvky uživatelů na platformě Reddit [4]. [45]

Author2Vec vytváří vektorovou reprezentaci uživatele čistě na základě textu příspěvků, které uživatel vytvořil. Experimentují s databází 10 000 uživatelů, kdy každý uživatel vytvořil alespoň 20 příspěvků. V případě velmi aktivních uživatelů je bráno v potaz pouze 500 nejnovějších příspěvků. Každý příspěvek je potom reprezentován pomocí předtrénované sítě typu BERT [46]. V této chvíli tedy autora reprezentuje množina vektorů jeho příspěvků, kdy každý vektor má dimenzi 3072.

Vlastní trénování vektorů autorů pak probíhá při úkolu určit autora z náhodně vybrané podmnožiny jeho příspěvků. K řešení této klasifikační úlohy sestavili autoři neuronovou síť, jejímž vstupem je vybraná podmnožina vektorů příspěvků. První vrstvou sítě je pak obousměrná rekurentní vrstva typu GRU [47] s 512 neurony. Za ní následuje K-Sparse vrstva [48] se 768 neurony, u které se předpokládalo, že se naučí řídké zakódování autora. Při učení sítě se za touto vrstvou nacházela ještě MLP vrstva s ReLU aktivacemi, která klasifikovala autora. [4]

Výše zmíněná K-Sparse vrstva je vrstva neuronů, která dovolí projít pouze k nejvyšším hodnotám a zbytek nastaví na nulu. Intuice za tímto mechanismem je taková, že výsledné příznaky by měly mít vyšší sémantickou hodnotu a také slouží jako regularizační metoda, která zabraňuje přeučení. V tomto konkrétním případě byl parametr nastaven na $k = 32$ při učení a $k = 64$ v době inference. Architektura celého systému je zachycena na obrázku 1.12. [4]

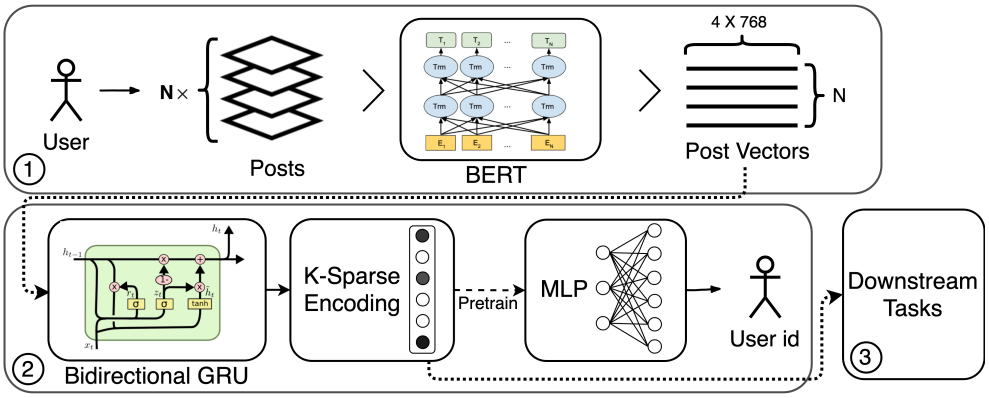
Pro ověření modelu využili autoři 10-fold křížovou validaci na úloze identifikace uživatele. U té dosáhl model průměrného F1 skóre 0,933, což je relativně dobrý výsledek, z čehož autoři odvozují, že latentní reprezentace uživatelů bude mít velkou diskriminativní sílu. Tu je možné vizuálně posoudit, pokud vektory uživatelů projektujeme do 2D pomocí metody t-SNE, což můžeme vidět na obrázku 1.4.

1.5 Samoorganizační mapy

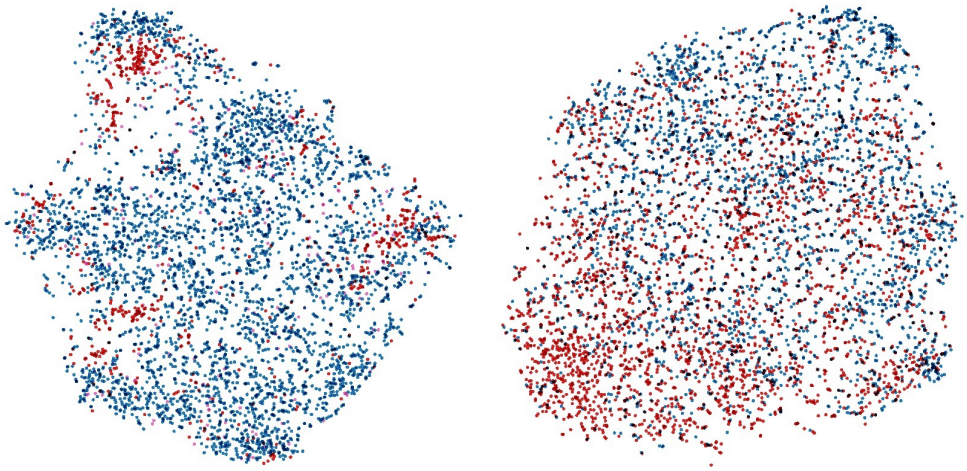
Mezi ne úplně časté techniky pro vizualizaci dat patří *Self-organizing Map* (SOM) taktéž zvaná Kohonenova síť podle svého tvůrce Teuvo Kohonena. Jedná se o svého druhu neuronovou síť, jejíž učení ovšem neprobíhá pomocí zpětného šíření chyby, ale pomocí soutěžení jednotlivých neuronů [7]. Neurony jsou obvykle inicializovány do podoby dvourozměrné mřížky, která může být buď čtvercová, nebo hexagonální (to je obvyklý postup, nikoliv nutnost). Učení pak probíhá v několika jednoduchých krocích [49]:

1. Síti je představen jeden vzorek z trénovacího datasetu.

[4] https://www.reddit.com/



Obrázek 1.12: Celý proces Author2Vec se sestává ze tří částí: 1) Převedení příspěvků uživatele do vektorové reprezentace (embedding). 2) Předtrénování na úloze klasifikace uživatele. 3) Využití reprezentace uživatele v dalších úlohách. [4]



Obrázek 1.13: Uživatelé reprezentovaní pomocí systému Author2Vec zobrazení pomocí metody t-SNE. Vlevo vidíme uživatele z trénovacího setu obarvené podle pohlaví a můžeme si všimnout tří červených (ženských) shluků. Vizualizace vpravo pak zobrazuje uživatele obarvené podle toho, trpí-li (trpěli-li) uživatel depresí. Množina uživatelů z pravé vizualizace je disjunktní s tou trénovací. [4]

2. Je nalezen neuron ze sítě, který je trénovacímu vzorku nejblíže (*best matching unit*, BMU)

3. Váhy BMU a jejích sousedů jsou aktualizovány tak, aby se neurony přesunuly blíže k trénovacímu vzorku. Množství aktualizovaných sousedů a velikost jejich aktualizace se obvykle snižuje s rostoucím počtem iterací.

4. Pokud nastala ukončovací podmínka (konvergence, počet epoch), učení končí, jinak pokračuje krokem 1.

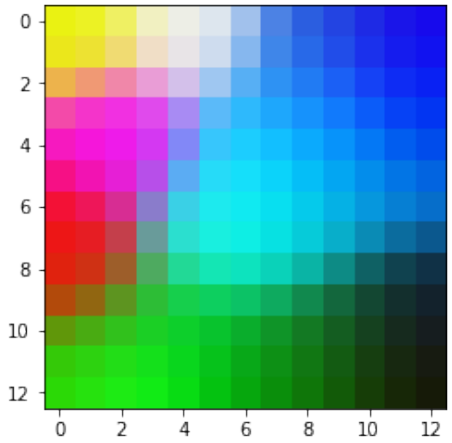
Pro pro výpočet vzdáleností neuronů a trénovacích vzorků se obvykle používá Euklidovská vzdálenost, je ale možné využít i např. Manhattanskou nebo i kosinovou.

Mezi hlavní aplikace SOM patří vizualizace dat pomocí zobrazení vlastní sítě. Ta obvykle probíhá pomocí tepelné mapy. Zobrazíme čtvercovou či hexagonální mřížku podle zvolené topologie a jednotlivé segmenty obarvíme podle nějakého klíče. Prvním způsobem je obvykle obarvení neuronů na základě průměrné vzdálenosti k jejich sousedům. To nám prozradí, které jednotky se vyskytují v oblastech s vysokou hustotou trénovacích dat (malé vzdálenosti) a které leží v prostoru osamocené daleko od svých sousedů.

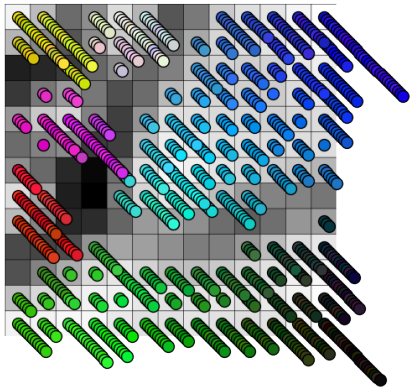
Další možností je obarvení jednotek podle hodnoty atributu vstupních dat. Každá jednotka získá svou barvu podle hodnot svých nejblíže ležících trénovacích vzorků. Pro analýzu pak obvykle slouží celá množina takových grafů, kdy u každého je využit jiný atribut a až prozkoumáním více z nich dohromady je možné odhalit nové závislosti a korelace.

Pro ilustraci různých způsobů vizualizace uvedeme, jak vizualizace vypadají na uměle vytvořeném příkladu [50]. Data můžeme vygenerovat náhodně ve třírozměrném prostoru a interpretovat je jako barvy zapsané v RGB. Na těchto datech následně natrénujeme SOM v podobě čtvercové mřížky 13×13. Na obrázku 1.14 vidíme zmiňovanou tepelnou mapu. Jelikož pozice ve třírozměrném prostoru lze interpretovat přímo jako barva, je obarvení jednotek určeno jejich pozicí, ne sousedy.

Nejlepší a nejvíce informativní metodou zobrazení SOM, se kterou jsme se při rešerši setkali, je zobrazení všech, nebo části trénovacích vzorků nad tepelnou mapu v podobě bodů či koleček obarvených podle hodnoty zvoleného atributu. Takové zobrazení nám totiž umožní si všimnout, kolik vzorků z jakých kategorií se v té či oné části prostoru nachází. Pokud bychom jednotku obarvili podle majoritní kategorie jejích vzorků. Ztratila by se veškerá informace o dalších vzorcích z jiných kategorií a jejich počtu, zatímco takto dokážeme pro každou jednotku prezentovat informaci odpovídající histogramu, akorát ve velmi stravitelnější podobě.



Obrázek 1.14: SOM natrénovaná na RGB datech. Barva každé jednotky je dána její polohou v prostoru. [50]



Obrázek 1.15: Jaká data v prostoru odpovídají kterým jednotkám můžeme zobrazit takto - každý bod zobrazíme jako útvar (kolečko), nad jeho BMU. Tak vidíme rozložení dat v prostoru i jejich kategorii. Barva políčka na šachovnici je pak dána průměrnou vzdáleností neuronu k jeho sousedům - světlejší barva značí menší vzdálenost.[50]

1.6 Minimum-Distortion Embedding

Existuje velké množství různých způsobů, jak vizualizovat vícerozměrné vektory ve dvou či třech rozměrech, můžeme jmenovat zmíněné t-SNE, SOM, nebo Sammonovu projekci [51]. Všechny tyto metody jsou bezesporu užitečné a často využívané, ale trpí několika problémy. Obecným problémem je špatná škálovatelnost, kdy doba výpočtu značně stoupá s množstvím dat, možná citelnějším problémem je ale častá nemožnost přidat do projekce nová data (SOM v tomto budiž výjimkou). Nově publikovaný framework nazvaný *Minimum-Distortion Embedding* (MDE) [8] překonává oba tyto problémy, zatímco dosahuje výsledků zcela srovnatelných s výše zmíněnými metodami.

Jednoduchým způsobem, jak formulovat měřítko kvality embeddingu, je pomocí vzdálenosti $d_{ij} = ||x_i - x_j||$ obrazů vektorů i a j. Vlastní kvalita pak závisí na hodnotě *distorzní funkce* $f_{ij}(d_{ij})$, u které se předpokládá diferencovatelnost. U této funkce chceme, aby nabývala nízkých hodnot u dvojic vektorů reprezentujících položky, které si jsou podobné, a vysokých hodnot u rozdílných dvojic. Jednoduchým příkladem může být funkce $f_{ij}(d_{ij}) = w_{ij}d_{ij}^2$, kde w_{ij} indikuje váhu páru a je pozitivní pro podobné páry a negativní pro

rozdílné páry.

Formálně pracujeme s množinou položek $\mathcal{V}$ a množinou jejich párů $\mathcal{E} \in \mathcal{V} \times \mathcal{V}$. Distorzní funkce odvozená od vah párů položek má obecně tvar:

$$f_{ij}(d_{ij}) = \begin{cases} w_{ij} p_S(d_{ij}) & (i,j) \in \mathcal{E}_{sim} \\ w_{ij} p_D(d_{ij}) & (i,j) \in \mathcal{E}_{dis}, \end{cases}$$

kde p_S a p_D jsou penalizační funkce pro podobné a nepodobné páry a $\mathcal{E}_{sim}$ spolu s $\mathcal{E}_{dis}$ jejich množiny. Zde je dobré si všimnout, že p_S a p_D mohou být dvě zcela rozdílné funkce. Dalším způsobem, jak odvodit distorzní funkci, je od vzdáleností v původním vektorovém prostoru. V takovém případě chceme, aby vzdálenosti mezi páry položek byly stejné jak v původním prostoru, tak v tom tom novém:

$$f_{ij}(d_{ij}) = (\delta_{ij} - d_{ij})^2,$$

kde δ_{ij} označuje vzdálenost páru položek v původním prostoru. [8]

Řešení MDE úlohy se pak sestává z nalezení takových obrazů vektorů, které minimalizují průměrnou hodnotu distorzní funkce, zatímco splňují některá možná omezení, která na ně můžeme klást. Můžeme vyžadovat centrování, ukotvení, nebo standardizaci. Standardizace vyžaduje, aby výsledná kolekce vektorů měla nulový průměr a jednotkovou kovarianci, což jsou žádané vlastnosti, pokud chceme embeddingy využít pro další úlohy a nejen pro vizualizaci.

Zajímavým omezením je ukotvení, kdy fixujeme nebo zadáváme obrazy některých vektorů a dopočítáváme pouze ty nefixované. Přesně tímto způsobem můžeme získat projekce dalších položek, které nebyly součástí původní trénovací množiny. Všechny původní, třeba standardizované obrazy zafixujeme a následně řešíme druhý problém s tímto omezením. Výpočet je také možné akcelerovat pomocí GPU, což přináší velké zrychlení oproti zmiňovanému t-SNE.

Distorzní funkce jsou důležitým hyperparametrem v tomto framworku. Obecně chceme volit jednu distorzní funkci pro podobné páry $(i,j) \in \mathcal{E}_{sim}$ (atraktivní, přitahující funkci) a jinou pro odlišné páry $(i,j) \in \mathcal{E}_{dis}$ (odpuzující). Z atraktivních funkcí můžeme dále uvést logistickou distorzní funkci:

$$p_S(d) = \log(1 + e^{\alpha(d-\tau)}),$$

kde $\alpha > 0$ a $\tau > 0$ jsou parametry. Tato funkce tlačí podobné položky, aby měly vzdálenost menší než τ, zatímco ty s vyšší vzdáleností penalizuje přibližně lineárně. Druhým příkladem může být *log-plus-one* distorzní funkce:

$$p_S(d) = \log(1 + d^{\alpha}),$$

kde $\alpha > 0$ je parametr. Můžeme si všimnout, že metody t-SNE [52] a UMAP [53] využívají nějakou variantu této funkce. [8]

Při volbě odpuzující funkce $p_D(d)$ jsou užitečné takové funkce, které mají pro $d \to 0$ limitu $+\inf$ a pro $d \to \infty$ konvergují k 0. Konkrétními příklady

pak mohou být obrácená hodnota mocniny vzdálenosti $p_D(d) = -\frac{1}{d^\alpha}$, nebo logaritmicky:

$$p_D(d) = \log(1 - exp(-d^\alpha)),$$

kde $\alpha > 0$ v obou případech. Další možností je volit funkci ve formě:

$$p_D(d) = \log\left(\frac{d^\alpha}{1 + d^\alpha}\right),$$

kde opět $\alpha > 0$. Tato poslední funkce je zajímavá zejména proto, že MDE problém bez omezujících podmínek založený na *log-plus-one* atraktivní penalizaci a této odpuzující je vlastně ekvivalentní k UMAP. [8]

Kapitola 2

Praktická část

V praktické části práce jsme se rozhodli ověřit, jak užitečné budou jednotlivé přístupy zmíněné v rešeršní části pro vizualizaci uživatelů, pochopení jejich chování, potažmo budou-li užitečné i pro jiná data. Nejdříve na menším datasetu ověříme interpretabilitu shluků uživatelů a položek s využitím základního přístupu - maticové faktorizace a důsledně prozkoumáme vzniklé shluky. Následně navrhneme vlastní přístup pro trénování reprezentace uživatelů i položek v jednom latentním prostoru, který je inspirovaný přístupem výzkumníků ze společnosti Alibaba [5], který se pokoušíme zjednodušit a trénovat s důrazem ne na doporučování, ale právě na dobré vlastnosti latentních reprezentací z hlediska interpretability.

Bodové grafy embeddingů vzniklé pomocí t-sne samozřejmě nejsou jedinou možností vizualizace, mají určitá omezení, jelikož do bodového grafu není možné dostat všechny informace o uživatelích. Jednou z možností je přidat množství histogramů, které zobrazí distribuci hodnot různých atributů uživatelů tak, jak to uvidíme v následující sekci, takové množství různých grafů může být ale nepřehledné. V sekci 2.2 proto prozkoumáme i jinou možnost zobrazení interakcí, která dokáže obsáhnout více informací.

2.1 Úvodní experimenty na malém data setu

Pro pilotní experimenty, pomocí kterých jsme chtěli odhadnout, jeslti má zvolený přístup smysl, jsme zvolili data set menšího rozsahu, který dodala firma Recombee. Tento data set obsahuje dva soubory - `cooklist-prod-items.csv` s informacemi o jednotlivých položkách (což jsou v tomto případě recepty) a `cooklist-prod-events.csv` s interakcemi uživatelů. V souboru s informacemi o položkách najdeme tyto sloupce:

- **itemid**: string, id dané položky,
- **course**: string, určuje, jeslti jde o snídani, přílohu, oběd atp.,

- **cuisine**: string, typ kuchyně, tedy např. italská,
- **brand**: string, značka, které dodává suroviny,
- **direction_count**: integer, počet instrukcí,
- **ingredient_count**: integer, počet ingrediencí,
- **image_url**: url, obrázek zobrazovaný u receptu,
- **ingredient_list**, množina stringů, suroviny pro přípravu receptu,
- **title**: string, název receptu,
- **date_published**: timestamp, datum publikace,
- **time_seconds**: integer, odhadovaný čas přípravy,
- **calories**: integer, množství kalorií v jídle,
- **average_price_per_serving**: float, průměrná cena za porci,
- **diet**: množina stringů: dieta, pro kterou je recept vhodný, např. vegetariánská.

Ve druhém souboru `cooklist-prod-events.csv` jsou pak tyto sloupce:

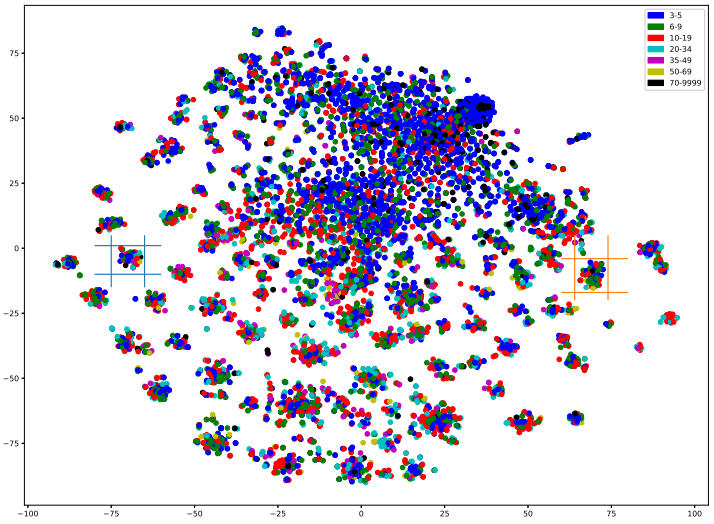
- **userid** string, id uživatele,
- **data** string, id položky, se kterou uživatel interagoval,
- **timestamp** čas interakce,
- **type** string, typ interakce. Možné hodnoty jsou: `DetailView`, `Bookmark`, `CartAddition` a `Purchase`.

Tento data set obsahuje pouze implicitní hodnocení položek - uživatelé neudělují žádný explicitní rating. Z povahy některých uživatelských akcí je však možné usuzovat na kladný vztah uživatele vůči receptu, a to zejména z akcí typu `CartAddition` a `Purchase`. Pro pročištění data setu jsme si ponechali pouze uživatele, kteří měli počet interakcí mezi 3 a 100. Těch tak zůstalo 10 382 spolu s 35 321 recepty.

Z takto vyfiltrovaných uživatelů a receptů jsme následně sestavili matici hodnocení s hodnotou 1, pokud si uživatel pouze přidal recept do košíku a s hodnotou 2, pokud došlo k nákupu. Na takto sestavenou matici bylo aplikováno SVD, kterým byla data redukována do pouze 2 dimenzí (z důvodu vizualizace, obvyklé jsou spíš desítky či nízko stovky). Na dvourozměrná data byla dále aplikována metoda t-SNE.

Výsledky se v tomto případě dají označit za povzbudivé. Na obrázku 2.1 vidíme, že uživatelé jsou takto separování do množství menších shluků. V grafu

je samozřejmě patrná i masa nevyprofilovaných uživatelů, ale důležité je, že zde vidíme i shluky. Pro analýzu toho, jestli jsou shluky nějakým způsobem užitečné či vypovídající a ne jen artefakty zanesené metodou t-SNE, byly vybrány dva shluky uživatelů, které jsou označeny modrými a oranžovými čarami.

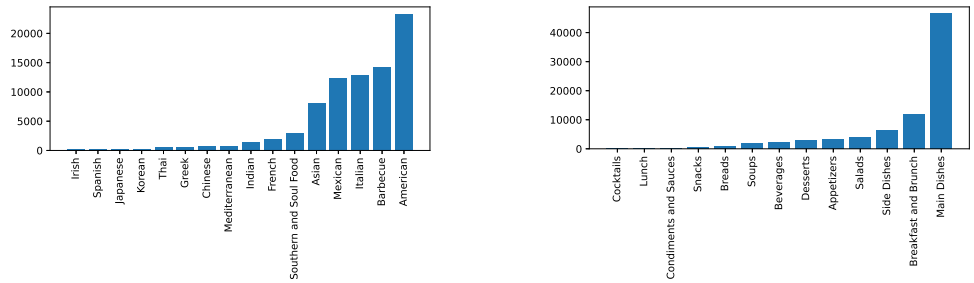


Obrázek 2.1: Vizualizace uživatelů po aplikaci SVD a t-SNE. Oranžovými a modrými čarami jsou označeny vybrané shluky pro další analýzu. Barvy bodů označují, s kolika recepty uživatel interagoval.

Průzkum shluků uživatelů

K analýze shluků jsme přistoupili tak, že jsme se pokusili najít společného jmenovatele receptů, se kterými tito uživatelé interagovali. K receptům máme velké množství meta dat a my jsme se zaměřili na dvě přirozené možnosti, jak od sebe odlišit recepty - kuchyň, ze které pocházejí, a typ chodu - tedy jedná-li se o např. snídani, přílohu, nebo hlavní jídlo.

Prvním krokem k analýze je zjištění, jaká je distribuce jednotlivých hodnot těchto dvou kategorií v datasetu. Pro správné vyhodnocení není možné pouze spočítat kolik receptů pochází z jaké kuchyně, je potřeba tyto počty vážit tím, kolik uživatelů s receptem interagovalo, jelikož přesně tak potom budeme vyhodnocovat jednotlivé shluky (tedy pokud s jedním konkrétním receptem na pizzu interagovalo 5 uživatelů, přičítáme k italské kuchyni +5, i když se jedná o jeden recept). Z histogramů na obrázku 2.2 vidíme, že uživatelé nejčastěji



Obrázek 2.2: Globální distribuce kategorií „kuchyň“ a „chod“ v datasetu (pole `cuisine` a `course`).

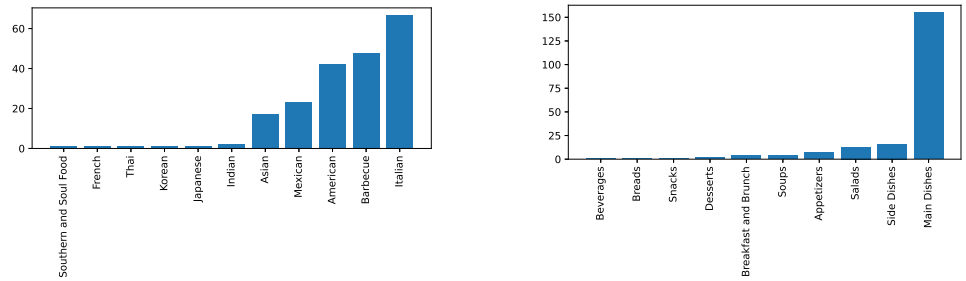
interagovali s recepty americké kuchyně a na 2. až 4. jsou s minimálními rozdíly recepty na grilování s italskou a mexickou kuchyní. U typu chodu jsou pak výrazně ve vedení hlavní jídla následovaná snídaněmi a přílohami.

V případě identifikovaných shluků se tedy dá předpokládat, že budou vykazovat nějaké podobné rozdělení. Rovnoměrná distribuce, kterou bychom mohli na první pohled pokládat za neúspěch, by ve skutečnosti byla důkazem relativně silného zaujetí uživatelů v daném shluku. Na obrázku 2.1 vidíme rozdělení hodnot pro obě kategorie u receptů uživatelů ze shluku, který je na obrázku 2.1 ohraničen modrými čarami. Vidíme, že tito uživatelé oproti globální distribuci výrazně preferují italskou kuchyň na úkor té americké. Co se typů chodů týče, nejsou preference tak vyhraněné. Stejně jako v globálu výrazně vedou hlavní jídla, akorát snídaně se posunuly ze druhého místa až na šesté a na druhé místo se tím dostaly přílohy.

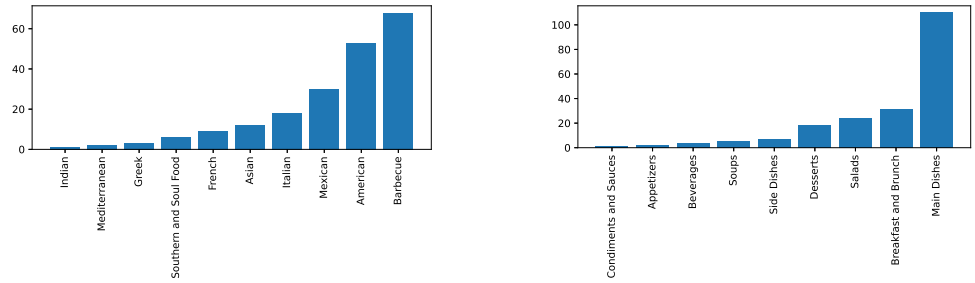
U shluku ohraničeného oranžovými čarami je situace relativně podobná. Uživatelé výrazněji preferují recepty na grilování (barbeque) a americká kuchyně opět ztrácí oproti globálu. Typ chodu se ani zde neukázal být dobrý pro odlišení uživatelů ze shluku od ostatních. Distribuce v oranžovém shluku je prakticky shodná s tou globální.

Shluky receptů

Při analýze shluků uživatelů se shluky vytvořené pomocí metody t-SNE ukázaly být jako relativně informativní, nabízí se tedy vyzkoušet to samé pro položky, tedy recepty. Z praktického hlediska se jedná v zásadě o totožný algoritmus, akorát provedený na transponované matici hodnocení. Na obrázku 2.5 vidíme výsledek opět po aplikaci t-SNE. Je možné vidět, že v případě receptů se netvoří tak dobře separovatelné shluky jako v případě uživatelů. Na okrajích ústředního „oblaku“ bodů můžeme rozeznat několik malých shluků, u těch je ale těžko najít společného jmenovatele. Zde na obrázku vidíme body obarvené podle kuchyně, v průběhu experimentů jsme zkoušeli opět i typ chodu, který



Obrázek 2.3: Distribuce kategorií „kuchyň“ a „chod“ v ve shluku označeném modře na obrázku 2.1.



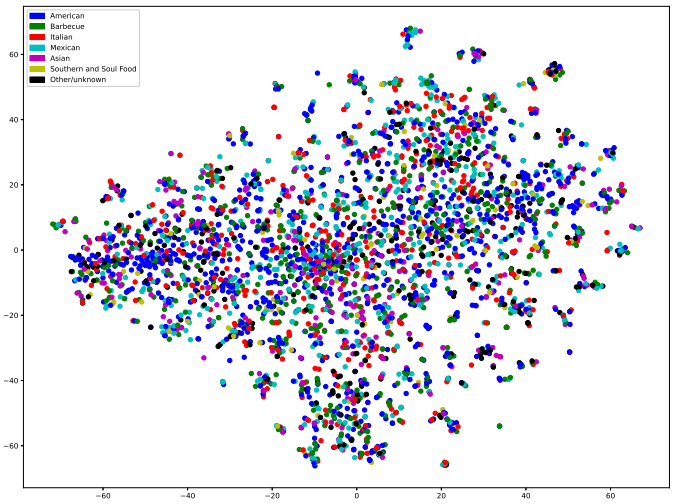
Obrázek 2.4: Distribuce kategorií „kuchyň“ a „chod“ v ve shluku označeném oranžově na obrázku 2.1.

se ovšem neukázal být nikterak informativnější.

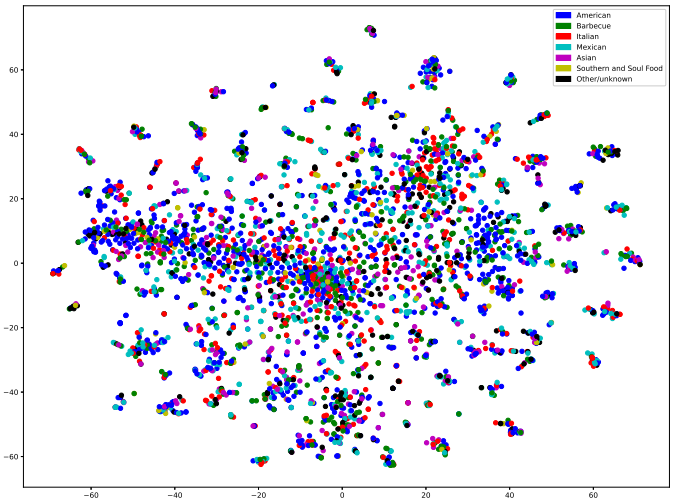
Jednou z možností, jak dále zpracovat matici hodnocení ještě před její faktorizací, je aplikace tf-idf. Stejně jako při práci s textovými dokumenty a jazykem, může i v našem případě tf-idf eliminovat vliv populárních položek, se kterými interagují skoro všichni uživatelé (analogicky ke slovům, která se vyskytují ve skoro každém dokumentu, těm tf-idf přiřadí nízké skóre). Na obrázku 2.6 vidíme, jak vypadají recepty, pokud místo matice hodnocení použijeme matici s hodnotami tf-idf skóre. Je možné pozorovat, větší množství vizuálně lépe separovatelných shluků, byť tím zlepšení končí. Ani zde se shluky nezdají výrazně informativní a je těžké analyzovat, co mají recepty společného.

2.2 Aplikace systému Author2Vec

Autoři systému Author2Vec při své práci využívali dataset s prakticky totožným počtem počtem uživatelů jako máme k dispozici zde v případě receptů, je proto nasnadě vyzkoušet tento přístup i na tomto datasetu. Samozřejmě existují i rozdíly, které pravděpodobně budou mít vliv na výslednou kvalitu. Au-



Obrázek 2.5: Vizualizace položek (receptů) po aplikaci t-SNE (bez tf-idf). Body jsou obarvené dle kuchyně, ze které recept pochází.



Obrázek 2.6: Vizualizace položek (receptů) s využitím tf-idf a po aplikaci t-SNE. Body jsou obarvené dle kuchyně, ze které recept pochází. Můžeme pozorovat mírné vizuální zlepšení v separaci shluků oproti verzi bez tf-idf (obrázek 2.5).

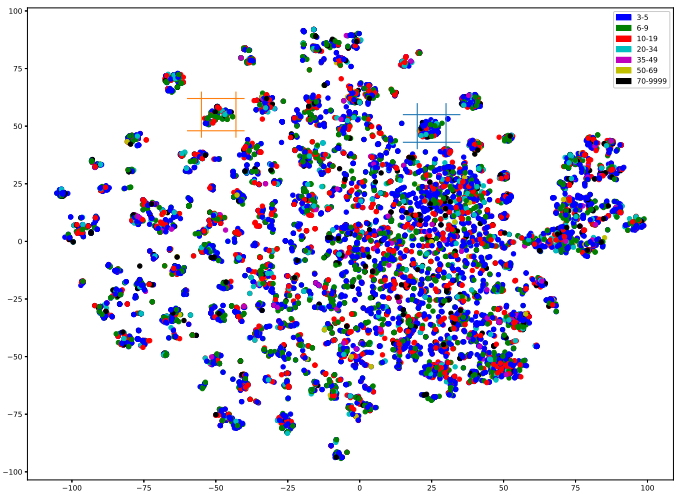
thor2Vec je vytvořený pro uživatele sítě Reddit, kde každý uživatel vytváří své vlastní příspěvky. My zde příspěvky nemáme, pouze položky, se kterými interaguje více uživatelů najednou, vypovídající hodnota množiny položek je tudíž pro identifikaci uživatele znatelně nižší, než množina jím vyprodukovaných unikátních příspěvků.

V případě receptů reprezentujeme jednotlivé položky jejich názvem, který bude sloužit stejně jako „text příspěvku“ v případě práce s uživateli Redditu. Celý text příspěvku následně reprezentujeme jediným embeddingem dimenze 768, pro který využijeme model BERT[5] předtrénovaný pro anglický jazyk. Vlastní neuronovou síť jsme pak sestavili podle popisu autorů za pomoci knihovny Tensorflow [6] a jejího rozhraní Keras. Za vstupní vrstvou následuje GRU [47] s obousměrným průchodem a 100 neurony. Za ní se pak nachází K-Sparse vrstva, která propustí pouze k nejvyšších aktivací (zde $k = 32$). Z tého vrstvy budeme následně čerpat latentní reprezentaci uživatelů. Další část sítě byla využita pouze pro předtrénování - MLP s jednou vrstvou s 256 neurony a ReLU aktivacemi a na závěr softmax vrstva pro klasifikaci 9556 uživatelů. Knihovna Tensorflow nenabízí připravenou implementaci pro K-Sparse vrstvu, tudíž bylo nutné ji ručně implementovat, pro což ovšem knihovna nabízí vhodné nástroje.

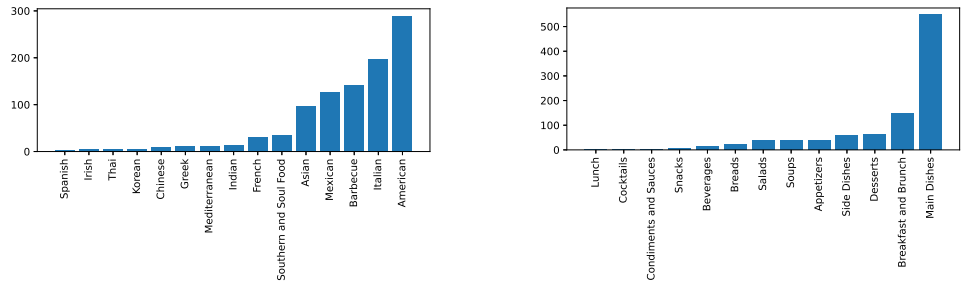
Při trénování byl každý uživatel reprezentován náhodně vybranou podmnožinou 80 % receptů, se kterými interagoval, s maximálním omezením na 25 položek. Toto omezení se dotklo pouze asi 2 % uživatelů z datasetu, jelikož většina uživatelů interagovala s méně než 10 položkami. Obava, že v tomto případě nejsou množiny položek pro uživatele dostatečně diskriminativní, se při trénování potvrdila. Při klasifikační úloze se *accuracy* nedostala přes 6 % (to je sice velmi nízké číslo, ale při počtu tříd skoro 10 tisíc je to stále násobně lepší výsledek, než náhodná klasifikace). Rozdíl oproti násobně lepším výsledkům z publikace Author2Vec je možné asi převážně tím, že každý uživatel má zde méně interakcí - přibližně 3 až 15 oproti 20 až 500 v případě uživatelů Redditu. Malé množiny položek, které se navíc mohou překrývat, tudíž pro identifikaci uživatele neslouží příliš dobře. Při rozboru výsledků se potvrdilo, že síť byla úspěšná převážně na uživatelích, kteří měli vyšší počet interakcí, s ostatními si v zásadě neporadila.

I přes to, že výsledky trénování nebyly vyloženě slibné, se můžeme podívat na vizualizaci výsledné latentní reprezentace uživatelů. Na obrázku 1.4 můžeme vidět vizualizaci latentní reprezentace uživatelů opět s pomocí metody t-SNE. Výsledky nejsou na první pohled vyloženě špatné, i zde vidíme jasně definované shluky uživatelů, byť vizuální kvalita nedosahuje té u předchozích. Nás zajímá opět hlavně vysvětlitelnost shluků - na obrázcích 2.2 a 2.2 vidíme histogramy k modře, respektive oranžově vyznačenému shluku. Můžeme si všimnout, že zde zobrazené distribuce se nijak významně neliší od těch glo-

[5] https://github.com/hanxiao/bert-as-service
[6] https://www.tensorflow.org/



Obrázek 2.7: Shluky uživatelů vytvořené pomocí systému Author2Vec a metody t-SNE. Obarvení uživatelů odpovídá počtu jejich interakcí.

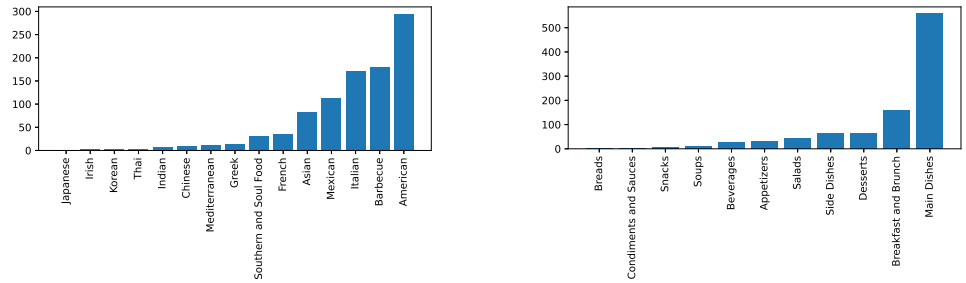


Obrázek 2.8: Distribuce kategorií „kuchyň“ a „chod“ v ve shluku označeném modře na obrázku 1.4

bálních z obrázku 2.2, tudíž se shluky minimálně z tohoto úhlu pohledu nejeví jako vyloženě informativní.

2.3 Inspirace hypercuboidy

Nevýhodou obou výše zmíněných přístupů, tedy faktorizace a systému Author2Vec, je, že nepracují s reprezentací uživatelů a položek v jednom sdíleném vektorovém prostoru. Tuto nesnáz elegantně překonává práce výzkumníků z čínské společnosti Alibaba [5], kteří reprezentují uživatele pomocí jednoho



Obrázek 2.9: Distribuce kategorií „kuchyň“ a „chod“ v ve shluku označeném oranžově na obrázku 1.4

nebo více nadkvádrů v latentním prostoru, ve kterém existují položky jakožto body.

Pro pilotní experimenty s malým datasetem jsme se rozhodli tuto metodu zjednodušit a ověřit její potenciál. Při tomto jednodušším přístupu reprezentujeme uživatele implicitně jakožto centroid položek, se kterými interagoval. Při trénování pak vycházíme z jednoduché úvahy. Položky jednoho uživatele by mu měly být co nejblíž, zároveň by však disjunktní[7] uživatelé - centroidy by od sebe měli být co nejdál.

S takto definovaným cílem můžeme pro učení vektorové reprezentace položek využít gradientní sestup. Označíme-li množinu položek uživatele u jako $\mathcal{I}_u$, definujeme uživatele - centroid c_u takto:

$$c_u = \frac{1}{|\mathcal{I}_u|} \sum_{i \in \mathcal{I}_u} i.$$

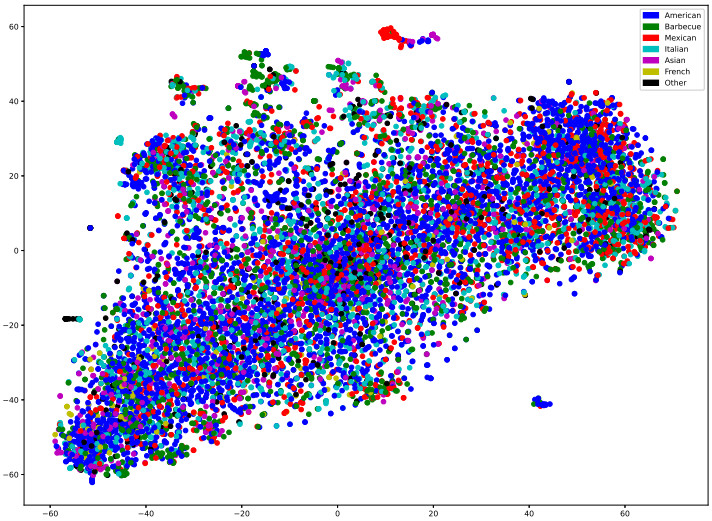
Následně můžeme zvolit libovolnou metriku vzdálenosti d ve spojitém vektorovém prostoru. My budeme experimentovat se dvěma obvykle používanými, euklidovskou a cosinovou vzdáleností. V našem případě dále platí $d(u_1, u_2) = d(c_{u_1}, c_{u_2})$ a $d(i, u) = d(i, c_u)$. Pro vzorek dvou uživatelů pak definujeme ztrátovou funkci takto:

$$L(u_1, u_2) = \sum_{i \in \mathcal{I}_{u_1}} d(i, u_1) + \sum_{i \in \mathcal{I}_{u_2}} d(i, u_2) + \frac{1}{d(u_1, u_2)}.$$

Učení probíhá tak, že modelu (tedy neuronové síti) představíme vždy pár disjunktních uživatelů a po každém páru aktualizujeme váhy tak, abychom minimalizovali ztrátovou funkci. Při experimentech se dále ukázalo, že účinnější než představovat modelu vždy celou množinu položek uživatele je náhodně z ní vybírat podmnožinu.

V prvním experimentu jsme se rozhodli učit přímo vektory položek, tedy náhodně inicializovat vektory pro všechny položky a následně je iterativně

[7] Průnik množin interagovaných položek je prázdný



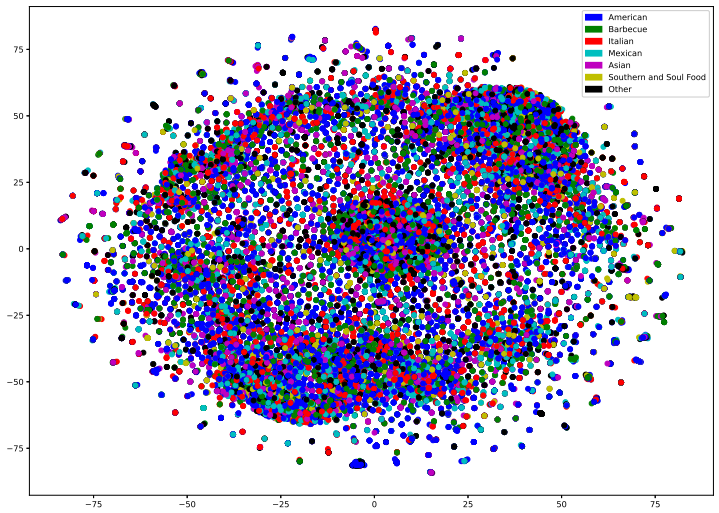
Obrázek 2.10: Reprezentace uživatelů pomocí přímo trénovaných vektorů s kosinovou vzdáleností. Barva uživatelů je dána podle majoritní kategorie receptů, se kterými interagovali.

upravovat. V knihovně Tensorflow toto odpovídá modelu s jedinou vrstvou typu `Embedding` dimenze 30. Tento model samozřejmě není nejsilnější, mohl by nám ale umožnit vhled do situace. Při posuzování výsledků nás budou zajímat opět hlavně vizualizace pomocí t-SNE. Na obrázku 2.10 vidíme reprezentaci uživatelů natrénovanou s pomocí cosinové vzdálenosti.

V případě ani jedné ze dvou uvažovaných vzdáleností nebylo možné pozorovat velkou míru separace shluků, a to jak prostorovým uspořádáním, tak obarvením uživatelů podle různých kategorií, což pro nás znamená nízkou interpretabilitu. V případě cosinové vzdálenosti (obrázek 2.10) je situace o něco málo lepší, ale ani zde nejsou výsledky slibné. Podíváme-li se přímo na naučenou reprezentaci položek a ne na centroidy (obrázek 2.11, vidíme, že ani zde není znatelná silná struktura, byť výsledky nevypadají náhodně.

Pro druhý experiment jsme sestavili neuronovou síť typu MLP (Multi Layer Perceptron) které by měla sloužit jakožto *encoder*. Jejím vstupem tudíž budou příznaky jednotlivých položek a výstupem latentní reprezentace. Tento přístup má samozřejmě tu výhodu, že dokáže následně pracovat i s položkami, které nejsou součástí trénovací množiny, což v případě přímo trénovaných vektorů nebylo možné. Vektorová reprezentace položek byla sestavena následovně:

1. Název receptu zakódovaný pomocí sítě BERT, tak, jako u systému Au-



Obrázek 2.11: Natrénovaná reprezentace položek s pomocí kosinové vzdálenosti.

thor2Vec. Zde ovšem navíc proběhla redukce dimenzionality pomocí metody PCA na výsledný rozměr 30.

2. Kuchyně, ze které recept pochází zakódovaná 1 z n. Použito bylo 8 nejčastějších kategorií plus devátá jakožto „jiné“.

3. Chod, opět 1 z n. Zde 6 kategorií plus jedna.

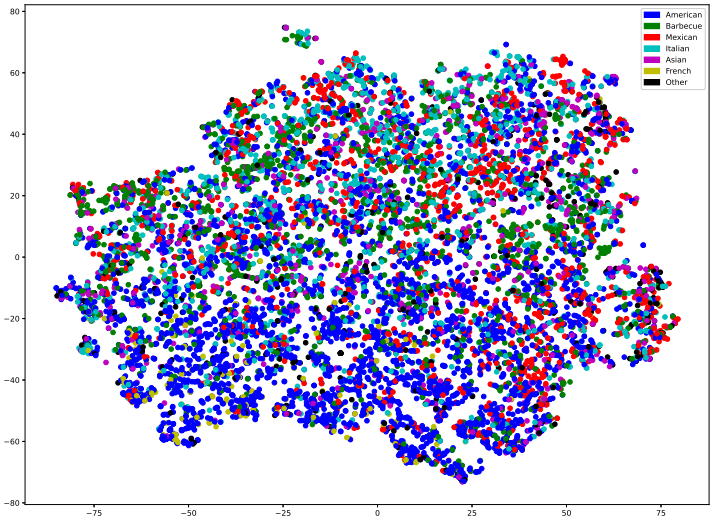
4. Dieta, ke které je pokrm vhodný (Veganská, Keto, ...). Zde je použito kódování pomocí k-hot vektoru, jelikož každá položka má přiřazenu množinu hodnot.

5. Numerické příznaky receptu - počet instrukcí, surovin, čas přípravy, kalorie a průměrná cena za porci. Tyto příznaky měly různé rozsahy, a proto byly pomocí z-score normalizace transformovány do standartního normálního rozdělení.

Celková dimenze vstupního vektoru pak byla 57. Experimentovali jsme s několika verzemi vícevrstvého perceptronu, které se lišily počtem skrytých vrstev a počtem neuronů. Všechny architektury jsou zaznamenané v tabulce 2.1. I přes výrazný rozdíl v počtu parametrů jednotlivých vrstev nebyly rozdíly ve výsledcích značné. Při trénování se hodnota ztrátové funkce ustálila vždy na přibližně stejných hodnotách a při vizuálním posouzení nejevily reprezentace uživatelů výraznější rozdíly. Vizualizaci uživatelů vidíme na obrázku 2.12.

Verze	Skryté vrstvy	Počty neuronů
MLP_1	1	50, 30
MLP_2	2	100, 50, 30
MLP_3	3	200, 120, 50, 30

Tabulka 2.1: Všechny vyzkoušené architektury vícevrstvého perceptronu. Udávané počty neuronů vždy obsahují i dimenzi poslední výstupní vrstvy, která byla vždy 30. Aktivační funkce skrytých vrstev byla vždy ReLU, u výstupní pak sigmoida.



Obrázek 2.12: Reprezentace uživatelů pomocí enkodéru (typ MLP_3 v tabulce 2.1). Obarvení uživatelů podle majoritní kategorie položek, se kterými interagovali.

2.3.1 Dataset Movielens

Naše metoda se do této chvíle neprojevila jako účinná, jelikož nevytvořila signifikantně oddělené shluky či jinak interpretovatelnou strukturu. Rozhodli jsme se proto pro dvě změny v přístupu. Přešli jsme k datasetu MovieLens [8], v jehož případě existuje naprostá jistota, že data v něm obsažená jsou smysluplná, a špatné výsledky jsou tedy zcela jistě chybou metody, nikoli dat. Dataset obsahuje 25 milionů uživatelských hodnocení provedených 162 tisíci uživateli na 62 tisících filmech.

Druhou změnou byla mírná úprava ztrátové funkce, kterou minimalizujeme při trénování. V průběhu předchozích experimentů bylo možné pozorovat, že je při trénování minimalizován hlavně poslední člen ztrátové funkce, tedy ten, který penalizuje blízkost disjunktních uživatelů. Jelikož se pohybujeme v relativně vysocedimenzionálním prostoru, rozhodli jsme se nadále využívat pouze kosinovou podobnost. Její výpočet je v knihovně Tensorflow implementován tak, že výsledek -1 značí totožný úhel vektorů, 0 pak jako obvykle kolmost a 1 značí opačný směr, tedy nejnižší similaritu. S přihlédnutím k tomuto formulujeme ztrátovou funkci jako:

$$L(u_1, u_2) = \sum_{i \in \mathcal{I}_{u_1}} d(i, u_1) + \sum_{i \in \mathcal{I}_{u_2}} d(i, u_2) - d(u_1, u_2).$$

V průběhu trénování klesala první složka ztrátové funkce (vzdálenost položek od centroidu) dle očekávání, ale hodnota druhé složky zůstávala konstantní (od začátku oscilovala kolem stabilní hodnoty). Dosud vektory aktualizovány po každém páru disjunktních uživatelů. Nyní jsme experimentovali s aktualizacemi po 1, 8 a 16 párech uživatelů. Žádná taková hodnota ovšem nepřinesla změnu v průběhu trénování a průměrnou vzdálenost disjunktních uživatelů se nedařilo navýšit.

I přes stagnaci hodnoty ztrátové funkce v průběhu trénování se ale měnila struktura natrénovaných vektorů. Při zobrazení pomocí t-SNE bylo možné vidět, že položky, které byly při trénování viděny pouze několikrát, zůstávaly u sebe, zatímco položky, které aktualizovány více než 500krát prokazovaly daleko větší rozptyl. Toto pozorování samo o sobě není překvapivé, ale mohlo by vnést vhled do toho, proč reprezentace uživatelů neprokazuje silnou strukturu.

Zobrazíme-li uživatele po zatím nejdelší době trénování (500 000 iterací, tedy párů uživatelů). Je možné pozorovat určitou strukturu ne v prostorovém rozmístění bodů, ale v obarvení uživatelů podle majoritní kategorie položek, se kterými interagovali. Při obarvování uživatelů v dataset Movielens narážíme na úskalí, že naprostá většina uživatelů nejčastěji sleduje dramata, a takové obarvení pak není moc informativní. Vzhledem k tomu, že většina filmů je často asociována s více než jedním žánrem, můžeme uživatele obarvit ne podle jednoho žánru, ale podle převažující podmnožiny žánrů. Teoreticky samozřejmě existuje velké množství všech možných podmnožin, ale v praxi se

[8] https://grouplens.org/datasets/movielens/

ukazuje, že při zvolení pouze 6 nejčastějších podmnožin žánrů obarvíme velkou většinu uživatelů.

Reprezentaci uživatelů s takovým obarvením můžeme vidět na obrázku 2.13. Strukturálně připomíná nejspíše mozek, což u t-SNE neznamená nic dobrého, ale co se obarvení týče, můžeme si povšimnout určitých zákonitostí.

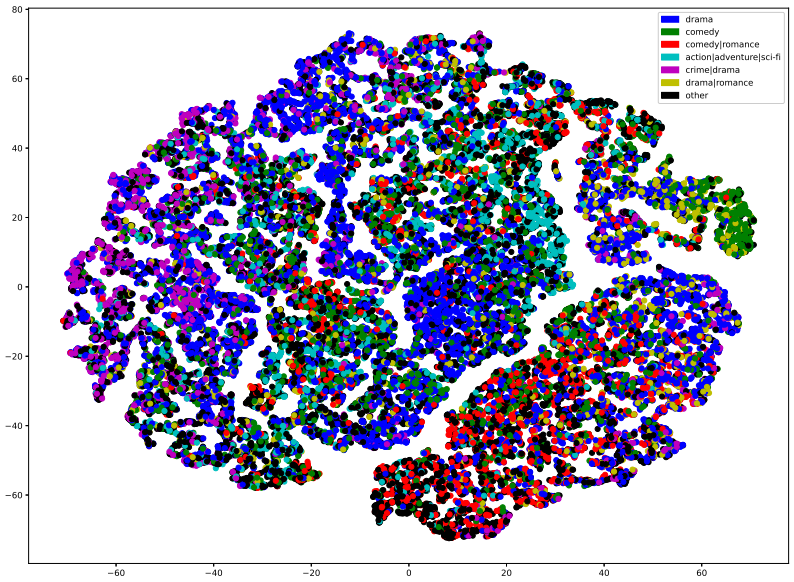
Obecným problémem v tomto ale do určitém míry asi všech datasetech je nevyhraněnost většiny uživatelů. U Movielens se většina uživatelů dívá na dramata a komedia a zůstává otázkou, kolik informace zde vlastně je k nalezení a zobrazení. Rozhodli jsme se proto zabývat se pouze ideálními, vyhraněnými uživateli. Kategorie uživatelů určujeme opět podle nejčastěji sledovaného žánru, vyloučíme ale 4 nejčastější a pokračujeme s 9 dalšími kategoriemi. S omezením počtu interakcí na interval $\langle 10; 70\rangle$ tak získáme asi 8500 uživatelů. Lze říci, že většina uživatelů je obvykle pasivních a menší část aktivních uživatelů objevuje obsah pro tuto pasivní většinu, což dále zvyšuje důležitost této menšiny vyhraněných uživatelů.

Zobrazení výsledků na takto omezeném datasetu můžeme vidět na obrázku 2.14. Pozitivní je, že zde vidíme jasnou separaci uživatelů dle kategorií, byť ne tak vyhraněnou, jako např. při využití NEASE na obrázku 1.11. Při bližším prozkoumání naše vizualizace potvrzuje některé domněnky, které jsme vyslovili v rešeršní části u analýzy uživatelů pomocí NEASE, týkající se blízkosti a prolnutí konkrétních žánrů.

Obvykle se prolínají ty třídy, u kterých to dává smysl. Ve středu vidíme dva velké shluky uživatelů preferujících dobrodružné filmy (červená) a sci-fi (modrá). Můžeme pozorovat, že tyto shluky se prolínají, což dává smysl vzhledem k podobnosti žánrů. Zároveň jsme toto pozorovali i na obrázku 1.11. Mezi další podobnosti patří rozdělení kategorie dobrodružných filmů do tří shluků, jednoho velkého a dvou srovnatelně velkých menších. U jednoho z menších shluků vidíme silné zastoupení dalších kategorií - krimifilmů (žlutá), romantických filmů (oranžová) a filmů pro děti (zelená). Přesně takový shluk můžeme opět najít i na obrázku 1.11.

I u posledního shluku kategorie dobrodružných filmů můžeme nalézt jistou podobnost. Zde i na srovnávané vizualizaci se jedná o červený shluk položený nejvýše. V obou případech jsou dále přítomni uživatelé z kategorií romantických filmů a sci-fi, byť je potřeba poznamenat, že na naší vizualizaci se tento shluk jeví „čistší“, a proto je podobnost slabší. Vlevo dole pak vidíme jednak uživatele s preferencí romantických filmů, kdy tato kategorie je opět vizuálně dobře separovaná, jednak oblast s překrývajícími se kategoriemi dětských (zelená), animovaných (šedá) a dobrodružných filmů. Opět lze říci, že takové spojení dává smysl - dětské filmy často bývají animované, jedná se tedy pravděpodobně o velmi podobné uživatele.

Samozřejmě lze nalézt i rozdíly. Podle naší vizualizace k sobě mají uživatelé z kategorie hororů a krimi filmů velmi blízko, a dokonce se prolínají, což je velký rozdíl oproti vizualizace pomocí NEASE 1.11, kde jsou tyto dvě kategorie dobře separované a daleko od sebe. Obecně lze ale říci, že jsme identifikovali

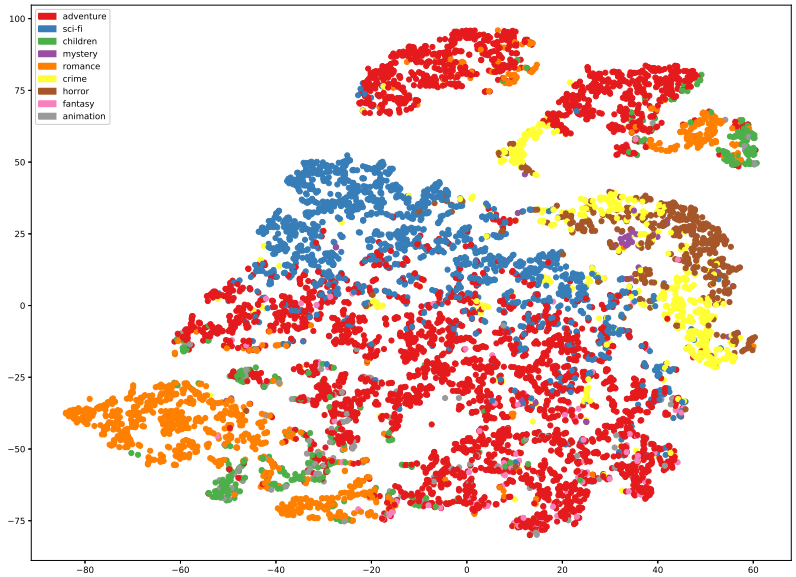


Obrázek 2.13: Reprezentace uživatelů z datasetu Movielens po 500 000 trénovacích párech. Zcela vpravo vidíme velmi dobře znatelný shluk zelených uživatelů preferujících komedie. Nalevo od něj je pak viditelná oblast bledě modrých uživatelů s preferencí dobrodružných filmů a scifi. Úplně vlevo se pak nachází oblast fialových uživatelů s oblibou krimi dramat. V pravém dolním rohu pak můžeme vidět dobře separovaný shluk uživatelů preferujích komedie, romantické komedie a dramata. Barevně je relativně dobře odlišený od zbytku, jedná se ale o tři nejčastější kategorie mezi uživateli, tudíž informační přínos je diskutabilní.

velké množství podobností, které snad potvrzují vyslovené doměnky o podobnosti kategorií.

2.4 Propojení uživatelů a položek

Doteď jsme shluky uživatelů vysvětlovali buď pomocí histogramů, nebo pomocí obarvení majoritní kategorií. Oba dva tyto přístupy ovšem mají velké nedostatky. Histogramy jsou sice informativní, jelikož je můžeme vytvořit pro každou kategorii a doplnit o další grafy a statistiky, ale provedeme-li to pro každý shluk, vznikne velké množství položek, které bude nutné zobrazit uživateli - analytikovi, a zůstává otázkou, jestli mu takto prezentované

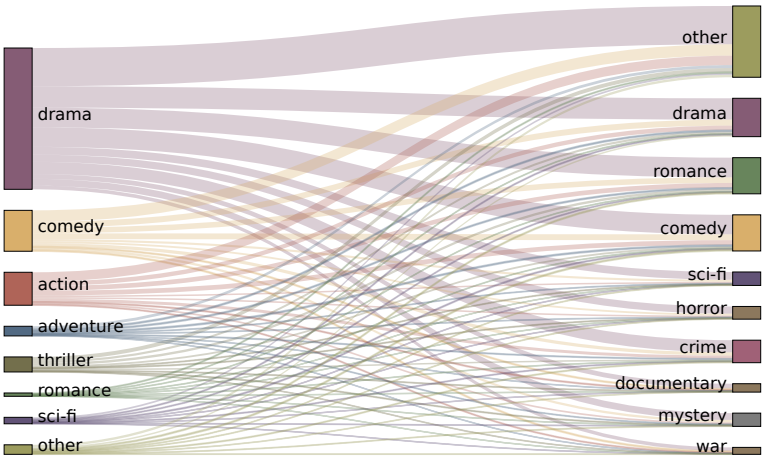


Obrázek 2.14: Zobrazení vybrané podmnožiny vyhraněných uživatelů. Trénování probíhalo pouze na vyhrané podmnožině, zobrazen je tudíž celý trénovací set.

informace budou přínosné.

Při obarvením uživatelů nastává přesně opačný problém, jelikož se ztrácí velká většina informací. V případě datasetu Movielens to ještě není tolik patrné, jelikož u filmů známe jen jeden atribut, a to žánr filmu, i v tomto jednoduchém případě můžeme ale obarvením podle jednoho majoritního žánru zahodit většinu informací, kterou o uživateli máme (vkus jednoho uživatele obvykle nebude definován pouze majoritním žánrem). Má-li každá položka více atributů, jako tomu je například u výše zmíněného datasetu s recepty, kde rozeznáváme kategorické atributy jako kuchyň a chod a spojité atributy jako kalorickou hodnotu porce, je pak informační ztráta ještě citelnější. O relevanci některých dalších atributů je asi možné polemizovat, ale minimálně tyto tři zmíněné určitě mají vypovídací hodnotu o vkusu uživatele.

Kompromisním řešením pro vizualizaci by mohly být Sankeyovy neboli bilanční diagramy, které se nejčastěji používají pro zobrazení účinnosti zařízení, pro naše účely se ovšem jeví taktéž vhodné. Obecně se jedná o zobrazení toku v systému, ve kterém je síla šipek úměrná velikosti toku. Adaptace pro náš případ užití je relativně jednoduchá - rozeznáváme dva typy uzlů, skupiny uživatelů a skupiny položek. Tok mezi nimi (od uživatelů k položkám) jsou

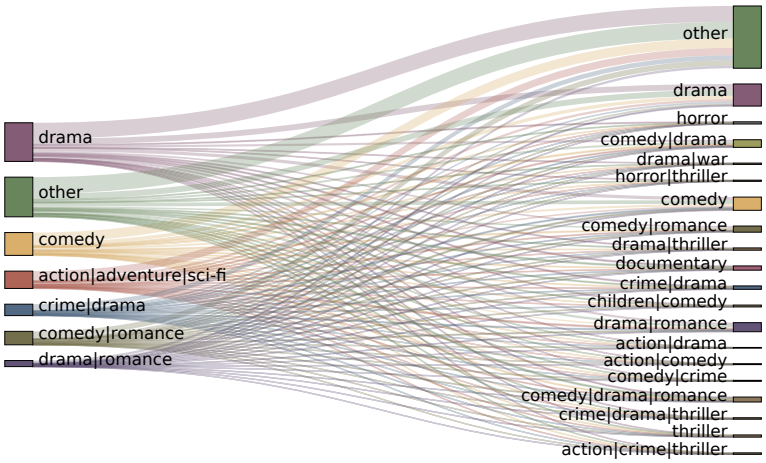


Obrázek 2.15: Sankeyův diagram interakcí skupin uživatelů se skupinami filmů. Label pro každého uživatele a film byl vybrán jako ten s nejvyšším tf-idf skóre.

interakce. Takové zobrazení by nám mělo umožnit nahlédnout přesně to, co chceme, tedy jaké skupiny uživatelů interagují s jakými skupinami položek a má tu výhodu, že se neomezí pouze na jednu nejčastější či jinak zvolenou kategorii, ale obsáhne je všechny, nebo alespoň značně více. [54]

Jak vypadá takový diagram pro dataset Movielens můžeme vidět na obrázku 2.15. Při jeho vytváření bylo potřeba překonat několik problémů způsobených specifiky datasetu, zároveň ale vyšly najevo důležité skutečnosti, které nebyly zřejmé z předchozích vizualizací. Jak již bylo zmíněno dříve, obarvíme-li (či shlukneme) uživatele čistě podle majoritního žánru filmů, se kterými interagovali, dostane se naprostá většina uživatelů do kategorií „drama“, potažmo „komedie“, což je pro naše účely nevhodné.

V předchozí části se relativně dobře osvědčilo obarvení uživatelů podle nejčastější skupiny žánrů (obrázek 2.13), ale Sankeyův diagram upozornil na slabiny tohoto přístupu. K obarvení většiny uživatelů sice opravdu stačí několik málo kombinací žánrů, v případě filmů je tomu ale jinak. Shlukujeme-li filmy podle přesné kombinace žánrů, potřebujeme 20 kategorií pro to, abychom obsáhli rozumnou většinu filmů. S tímto počtem uzlů na straně filmů už je diagram velmi nepřehledný. Bez takto vysokého počtu by naprostá většina interakcí uživatelů z uživatelských shluků, které se zdají být podle názvu dobře vy-



Obrázek 2.16: Sankeyův diagram interakcí skupin uživatelů se skupinami filmů, pokud bychom filmy shlukovaly podle množin jejich žánrů. Vidíme, že v datasetu se vyskytuje velké množství různých podmnožin, tudíž tento přístup vyústí ve velmi nepřehledný diagram.

hraněné, tedy např. `action|adventure|sci-fi`, vedla ne do stejnojmenného shluku či uzlu na straně filmů, ale do zbytkového uzlu „jiné“. Jak taková vizualizace vypadá můžeme vidět na diagramu 2.16. Je tedy zřejmé, že filmy s touto kombinací žánrů jsou sice u těchto uživatelů nejčastější, v žádném případě ale netvoří většinu. Většina interakcí je rovnoměrně rozvrstvená mezi filmy s dalšími kombinacemi žánrů. Toto zjištění zavdává pochyby o informativnosti takového obarvení, i když se na předchozí vizualizaci jevilo jako nadějné.

Rozhodli jsme se proto aplikovat tf-idf, tedy *term frequency - inverse document frequency*, což by nám mělo umožnit vybrat signifikantní žánr uživatle, který nutně nemusí být ten majoritní. Tf-idf je technika dobře známá z oblasti *information retrieval*, kde se používá k převodu textových dokumentů na vektory. Skóre daného termu (slova) v dokumentu je určeno jako součin dvou výrazů: $tf \times idf$, kde tf značí počet výskytů slova v dokumentu a idf je konkrétně v našem případě a implementaci v knihovně `Scikit-learn` [20] vypočteno jako:

$$idf(t) = log\frac{1+n}{1+df(t)} + 1,$$

kde n značí počet dokumentů (uživatelů) v kolekci a df počet dokumentů, ve kterých se vyskytuje term t. Pro zvolení žánru uživatele tedy stačí sestavit dokumenty reprezentující uživatele, což provedeme tak, že za každý film přidáme do dokumentu všechny žánry, se kterými je film asociován, spočteme skóre a následně zvolíme žánr s nejvyšším skóre.

V případě filmů narážíme na podobné obtíže. Nevýhoda shlukování filmů podle celých množin jejich žánrů byla popsána výše. Nabízí se možnost vybrat s pouze filmy s právě jedním přiřazeným žánrem, kterých je v datasetu přibližně třetina. Vede na ně ale pouze asi 10 % interakcí, tudíž by takové zobrazení opět zahazovalo velkou většinu informací, čemuž se chceme vyhnout. Nakonec jsme se rozhodli využít i pro samotné filmy tf-idf, byť filmy jakožto textové dokumenty mají několik velmi specifických vlastností, které ovlivní chování tohoto skóre. Jeden film má přiřazený 1 až 4 žánry, které se logicky neopakují, první složka součinu, *term frequency*, bude tudíž vždy rovna jedné. Je zřejmé, že výsledná hodnota bude záviset pouze na druhé složce, *inverse document frequency*, která bude nejvyšší pro nejméně častý žánr v datasetu. Náš přístup tedy ohodnotí každý film tím z jeho žánrů, který se vyskytuje u nejnižšího počtu dalších filmů.

Využití tf-idf na obou stranách diagramu vyústí ve skupiny uživatelů i filmů s rozumnou granularitou, zdá se proto být nejslibnější cestou. Takový relativně přehledný diagram by mohl pomoci odpovědět na otázku, jaké další filmy sledují uživatelé ze shluku např. komedií. Z obrázku 2.15 ale vidíme, že neprokazují žánrovou vyhraněnost a pozornost uživatelů je rozdělena mezi shluky filmů více méně rovnoměrně podle velikosti shluku. Dalším logickým krokem je pak analýza jednotlivých shluků uživatelů.

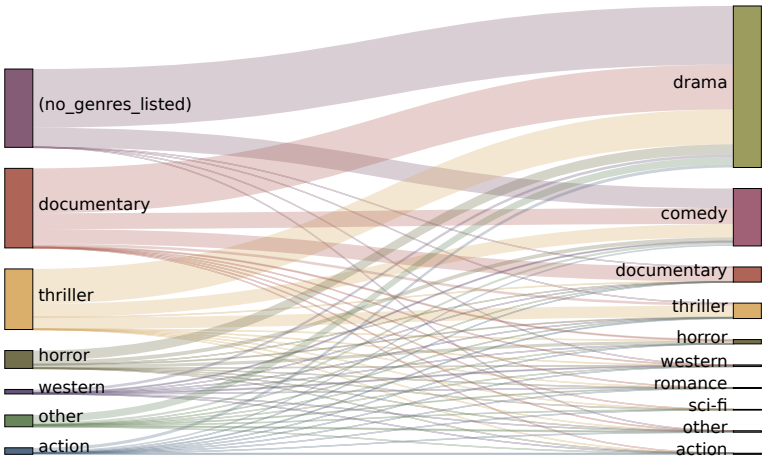
Pro další analýzu jsme si vybrali tři největší shluky uživatelů odpovídající kategoriím `drama`, `comedy` a `action`. Při podrobnější analýze se ukázalo jako nutné částečně revidovat dosavadní přístup. Hlavní změny je možné shrnout takto:

1. Omezili jsme se pouze na filmy s jednou přiřazenou kategorií, aby příslušnost k filmu ke shluku byla jasně dána. Tím sice vyřadíme více než půlku filmů, zbudou nám ale dobře interpretovatelná data.

2. Z uživatelských shluků jsme vyřadili kategorie `drama` a `comedy`, jelikož ty byly natolik dominantní, že znemožňovaly další analýzu. Víme tedy, že všichni sledují dramata a komedie, zajímá nás ale, co dalšího je zajímá.

3. Díky vynechání dominantních kategorií můžeme opustit tf-idf skóre, čímž získají shluky uživatelů ještě jasnější interpretovatelnost.

Z takto vytvořených diagramů už jsou patrné některé odlišnosti mezi shluky. Na diagramu 2.17 vidíme detailně rozebraný shluk uživatelů z kategorie `drama` a na diagramu 2.18 to samé pro kategorii `action`. Můžeme si všimnout, že zde už vstupují do hry filmy bez přiřazeného žánru, které tvoří

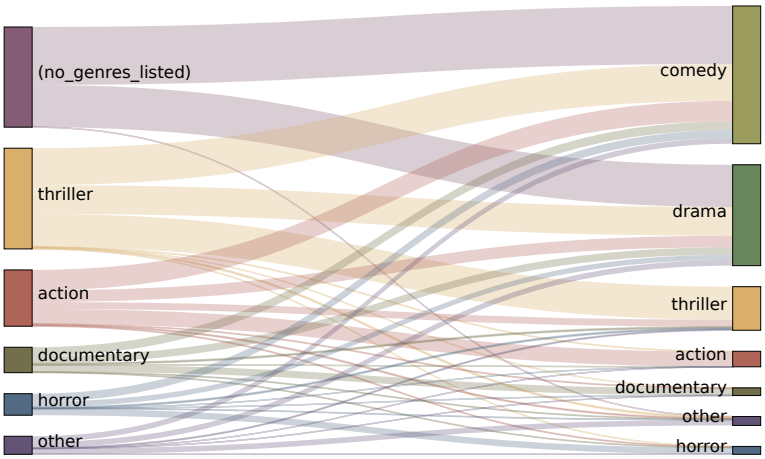


Obrázek 2.17: Sankeyův diagram uživatelů z kategorie `drama` z diagramu 2.15. Ani na jedné straně není využito tf-idf, filmy byly omezeny pouze na ty s jednou, potažmo žádnou přiřazenou kategorií.

signifikantní část interakcí, ale i přes to je možné pozorovat rozdíly. U dramat vidíme, že třetí nejčastější kategorií jsou dokumenty následované thrillery. V případě obou kategorií je dále možné si všimnout, že tito uživatelé jsou zodpovědní za v zásadě všechny interakce s příslušnými kategoriemi filmů. Lze říci, že nikdo jiný než uživatelé ze shluku dokumentů už dokumenty nesleduje, a to samé platí pro shluk thrillerů.

V případě uživatelů z kategorie `action`, které vidíme na diagramu 2.18, jsou asi dle očekávání dominantní kategorie thrillery a akční filmy. Opět vidíme, že s filmy z dané kategorie interagují prakticky výlučně uživatelé z odpovídajícího shluku s výjimkou toho, že uživatelé ze shluku akčních filmů ještě mají viditelnou část interakcí s thrillery. Oproti předchozímu zde pozorujeme výrazně méně interakcí s dokumentárními filmy. Je na místě důkladně se vyvarovat stereotypů, ale dvojice drama+dokumenty a thrillery+akční filmy nás asi nepřekvapí a jsou v souladu s očekáváním. Ze společných rysů obou shluků je možné si všimnout podobně velkého shluku uživatelů s preferencí hororů.

Experimentovali jsme i s daty z druhého datasetu s recepty. Výsledky obsahovaly velmi podobné problémy, jako jsme pozorovali u Sankeyových di-



Obrázek 2.18: Sankeyův diagram uživatelů z kategorie `action` z diagramu 2.15. Ani na jedné straně není využito tf-idf, filmy byly omezeny pouze na ty s jednou, potažmo žádnou přiřazenou kategorií.

agramů na datasetu Movielens - při shlukování podle kategorií nebyla patrná vyhraněnost uživatelů vůči shlukům položek. Pro stručnost uvedeme výsledné diagramy pouze v příloze.

Pro vytvoření všech diagramů v této sekci byla využita knihovna `Plotly` [9], která nabízí velmi dobré uživatelské rozhraní. Grafy je možné exportovat do vektorových formátů (svg, v této práci pdf), ale knihovna nabízí i export do `html`, což umožňuje jednoduché zobrazení ve webových aplikacích. Takto exportované diagramy jsou navíc interaktivní, přehlednější než statické obrázky a je možné zobrazovat různé další informace při přejetí myší nad částmi diagramu. Mají tak dobrý potenciál být součástí aplikace pro analýzu či podporu rozhodování. V této práci jsou proto přiloženy v příloze.

2.5 Aplikace SOM

Při experimentování se samoorganizačními mapami se využití celého datasetu se všemi uživateli ukázalo být neúnosně náročné. Abychom tento problém vyřešili, rozhodli jsme se pracovat pouze s uživateli, kteří mají počet interakcí

[9] https://plotly.com/

mezi 25 a 60. Z nich dále vybereme náhodný vzorek, kdy každý uživatel je vybrán s pravděpodobností 0,5. Tímto způsobem nakonec dostáváme 15 849 uživatelů, kteří interagovali s 9 280 filmy. Na těchto uživatelích natrénujeme SOM se čtercovou topologií o rozměrech 30×20 neuronů. Při trénování jsme využili kosinovou vzdálenost, jelikož v případě interakčních dat, tedy řídkých vektorů s vysokým počtem dimenzí, se jedná o přirozenou volbu.

Pro další zefektivnění trénování byla vždy v průběhu jedné epochy náhodně vybrána polovina trénovacích dat, která byla představena síti. Tento postup samozřejmě zkrátí trénovací čas přibližně na polovinu, může ale přinést potíže s konvergencí, pokud vybíráme moc malou část dat. Naštěstí to nebyl tento případ a průměrná vzdálenost dat od jejich BMU se v průběhu trénování stále snižovala. Trénování probíhalo po 30 epoch, kdy v průběhu posledních 5 průměrná vzdálenost již spíše stagnovala. Všechny experimenty probíhaly s veřejně dostupnou implementací SOM, kterou je možné nalézt na stránce Gitlab [10]. Jedinou odchylkou od výše popsaného trénovacího algoritmu bylo, že aktualizace vah neuronů neprobíhala po představení každého jednotlivého trénvacího vzorku, ale až po celé dávce (batch), která obsahovala 32 vzorků.

Pro vizualizaci výsledků zobrazíme čtvercovou síť jakožto teplotní mapu. Každá jednotka odpovídá jednomu čtverečku se sloupcem značek nad ní. Každá značka odpovídá jednomu uživateli, jehož vektor leží nejblíže právě této jednotce. Jako i v předchozích případech zobrazujeme i zde pouze uživatele kategorií, které považujeme za reprezentativní. Dle našeho názoru je z výsledného grafu vidět dobrá separace jednotlivých tříd a hlavně rozumné vztahy (blízkost a překryv) mezi jednotlivými žánry.

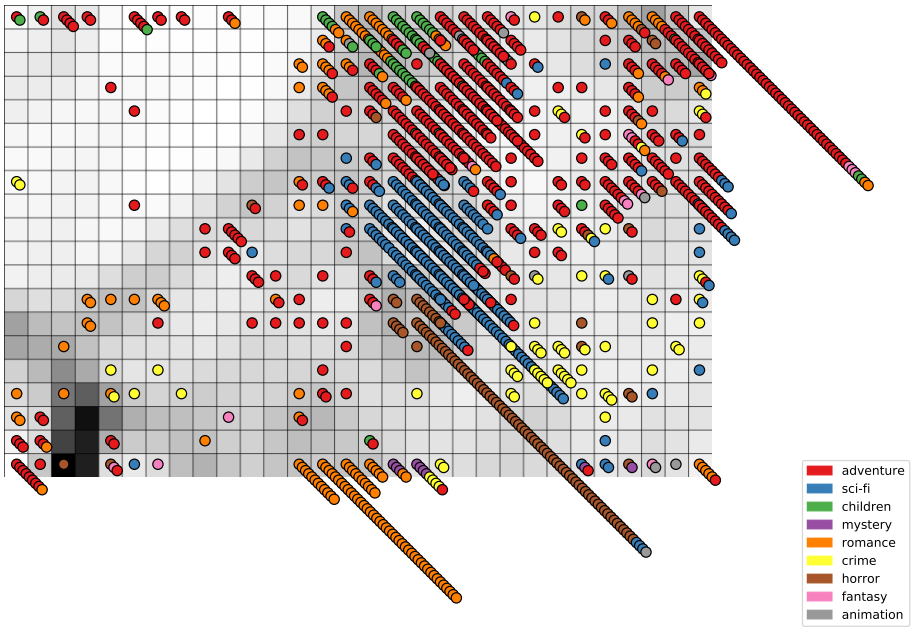
Experimentovali jsme i s daty z druhého datasetu s recepty. Data jsme omezili na uživatele a recepty s alespoň 5 interakcemi, což omezilo celkovou velikost datasetu na 3743 uživatelů a 2196 receptů, díky čemuž nebylo nutné další předzpracování (výběr) dat včetně náhodného výběru dat pro každou epochu. U tohoto data setu se opakovaly obtíže z předchozích experimentů s centroidy, kdy se nedařilo vizuálně separovat třídy. Vizualizaci výsledků ponecháváme vzhledem k jejímu rozsahu do přílohy práce.

V samoorganizačních mapách vidíme užitečný způsob znázornění vysokodimenzionálního prostoru řídkých interakčních vektorů. Zvolený způsob vizualizace ve spojení s tepelnou mapou je netradiční, ale v zásadě přehledný a díky své podobnosti k histogramu umožňuje na první pohled posoudit hustotu různých částí prostoru. Další výhodou SOM je zcela triviální a výpočetně nenáročná projekce nových uživatelů.

2.6 Využití frameworku MDE

V případě Minimum-Distortion Embedding jsme se rozhodli ověřit jeho schopnost dobře vizualizovat uživatele a hlavně kvalitu projekce dalších uživatelů do

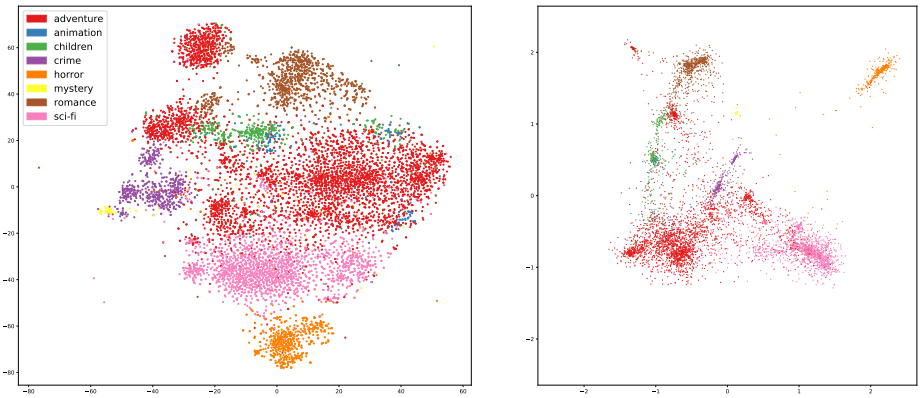
[10] https://github.com/Kursula/Kohonen_SOM



Obrázek 2.19: Vizualizace natrénované samoorganizační mapy na datasetu Movielens. Podkladová mřížka je teplotní mapa zobrazující průměrnou vzdálenost jednotky k jejím sousedům. Světlejší barva značí vyšší vzdálenost. Uživatelé jsou pak zobrazeni jako značky nad jejich odpovídající BMU. U kategorií uživatelů jako sci-fi a horror můžeme vidět velmi dobrou separaci tříd, stejně tak u dětských filmů a záhad. Uživatelé preferující romantické filmy se nacházejí v několika oblastech, jejich separace je ale také dobrá. Třída s nejhorší separací jsou pravděpodobně uživatelé mající v oblibě krimi filmy.

již existující vizualizace. Autoři MDE nabízejí efektivní implementaci v jazyce Python v podobě balíčku `pymde`, který stále prochází rychlým vývojem.

Prvním krokem je získání podobných a rozdílných párů uživatelů. Teoreticky je možné je získat arbitrárně a systému pouze předložit, je ale také možné získat páry automaticky pomocí náhodného vzorkování grafu nejbližších sousedů. Pro k-NN graf jsme experimentovali s různými hodnotami k, konstantně nejlepších výsledků bylo dosahováno s hodnotou $k = 15$, která se kterou jsme dosáhli všech výsledků prezentovaných v této sekci. Nejbližší sousedi jsou hledáni pomocí Euklidovské vzdálenosti. Její využití pro vektory s vysokou dimenzí jako jsou řídké interakční vektory je netradiční a Euklidovská vzdálenost je v tomto případě zcela jistě špatnou globální metrikou. V případě binárních interakčních vektorů je ale jejich vzdálenost redukována na počet indexů, ve kterých spolu vektory nesouhlasí, a proto lze říci, že uživatelé s malou Euklidovskou vzdáleností jsou si téměř jistě podobní. Stejně tak vysvětlují autoři MDE využití Euklidovské vzdálenosti při jejich experimentech s data-



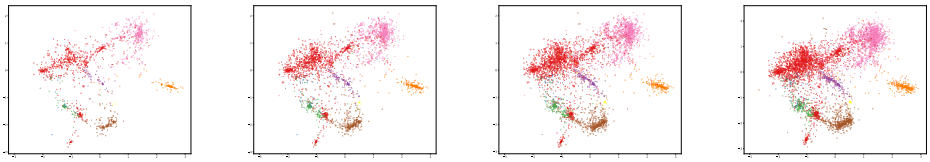
(a) t-SNE (b) MDE

Obrázek 2.20: Srovnání t-SNE a MDE vizualizací na vybraných uživatelích. MDE obecně produkuje hustší shluky oproti t-SNE ovšem ve srovnatelné kvalitě. U MDE může být hustota shluků regulována pomocí poměru podobných a rozdílných párů tak, jak navrhují autoři [8]. Na obou grafech vidíme, že nejlépe je separovaná třída horror, třída adventure je rozdělená do tří shluků, sci-fi a crime leží blízko sebe a dětské filmy leží mezi shluky třídy adventure. Rozdíl vidíme v pozici shluku třídy mystery, která leží blízko krimi shluku (daleko blíže v případě t-SNE), ale u MDE je také blízko romantickým filmům, což neplatí pro t-SNE.

setem MNIST [8].

Pro vyhodnocení vizualizačního potenciálu MDE nejdříve porovnáme, jak jsme pomocí MDE schopni vizualizovat vybrané uživatele. Použili jsme přibližně 6.000 uživatelů z méně častých kategorií a ty jsme vizualizovali jak pomocí t-SNE, tak pomocí MDE. Výsledky vidíme na obrázku 2.20. U obou technik je patrná dobrá separace kategorií, což zde považujeme za úspěch MDE - vyrovná se jedné z nejpoužívanějších vizualizačních technik. Relativní poloha (blízkost) jednotlivých tříd navíc z větší části odpovídá vztahům pozorovaným u předchozí vizualizace pomocí SOM na obrázku 2.19.

Silnou deklarovanou výhodou MDE oproti t-SNE je možnost přidávat do projekce další uživatele. Kvalitu takové projekce v praxi jsme se rozhodli systematicky ověřit v dalším experimentu. Používáme stále stejnou množinu vybraných uživatelů, tu ale náhodně rozdělíme do čtyř skupin po 1.500 uživatelích a zobrazení budujeme inkrementálně. Nejdříve zobrazíme první skupinu, následně ji fixujeme pomocí *anchor constraint*, přidáme do projekce druhou skupinu atd. Tento postup můžeme vidět na obrázku 2.21. I při inkrementální stavbě embeddingu vznikne vizualizace srovnatelné kvality jako na obrázku 2.20 (b), díky čemuž považujeme kvalitu projekce uživatelů pro naše účely za prokázanou.



(a) 1.500 (b) 3.000 (c) 4.500 (d) 6.000

Obrázek 2.21: Inkrementální stavba embeddingu pomocí MDE. Začneme s 1.500 uživateli, jejich projekci fixujeme, přidíme dalších 1.500 atd. Obarvení uživatelů odpovídá tomu z obrázku 2.20.

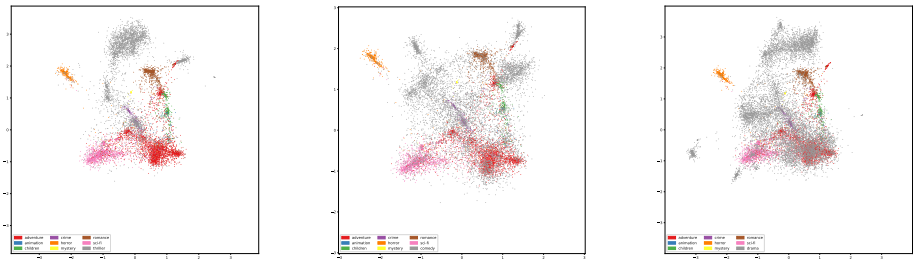
Jelikož se projekce dalších uživatelů ukázala být dle našeho názoru možná, experimentujeme dále se všech vybraných uživatelů a následným přidáním 3 velkých tříd - drama, komedie a thrillery. Tyto třídy přidáváme vždy jen jednu v daný okamžik, což nám umožňuje zachovat přehlednost grafu a pozorovat, které části prostoru jsou sdíleny uživateli více kategorií a které jsou opravdu z tohoto pohledu čisté. Výsledky jsou na obrázku 2.22. Všechny zde uvedené grafy se nacházejí v příloze práce ve vektorovém formátu pro možnost lepšího prozkoumání.

Je možné pozorovat, že stále nejlépe separovanou kategorií uživatelů jsou ti s preferencí hororových filmů. V jejich blízkosti se nachází velmi málo uživatelů z majoritních kategorií. Z toho vyvstává otázka mají-li tito uživatelé vůbec zájem o jiný obsah, než který sami vyhledávají. Dále tyto vizualizace prozrazují, že oblast uživatelů z kategorie dobrodružných filmů (červená), která se zdála relativně dobře separovaná od ostatních kategorií, vidíme ale, že v této oblasti se nachází také velké množství uživatelů z kategorií dramat a komedií, zatímco žádní s preferencí thrillerů.

Dvě stále ještě velmi dobře separované kategorie jsou sci-fi (růžová) a romantické filmy (hnědá). Naopak kategorie krimi zcela splývá s thrillery (přesněji je zcela překryta malou částí uživatelů z kategorie thrillerů), což odpovídá žánrové blízkosti. U dětských filmů a mystery je opět patrná lepší separace až na částečný překryv s komediemi, které jsou ale obecně hodně rozptýlené.

Práce s knihovnou `pymde` se při experimentech obecně ukázala být jako velmi jednoduchá a rychlá. Také doba výpočtu je řádově kratší, `pymde` dokáže výpočet akcelerovat pomocí GPU s využitím PyTorch [11]. Doba výpočtu byla řádově nižší než u t-SNE a o několik řádů rychlejší než v případě SOM.

[11] https://pytorch.org/



(a) Thriller (b) Komedie (c) Drama

Obrázek 2.22: Zde jsme spočítali embedding všech zvolených uživatelů, se kterými jsme pracovali v předchozích krocích. Následně do projekce přidali 3 velké třídy uživatelů - drama, komedie a thrillery. Vzhledem ke stochastické povaze algoritmu nemůžeme porovnávat vzájemnou polohu projektovaných tříd. Abychom grafy udrželi čitelné, bylo zobrazeno pouze 50 % komediálních uživatelů a 20 % těch z kategorie drama.

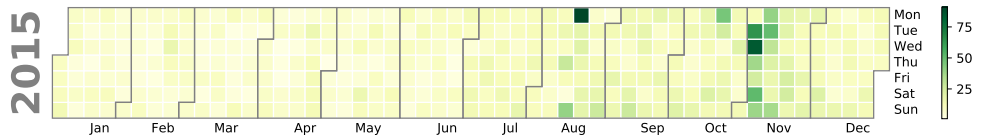
2.7 Zobrazení interakcí v čase

Ještě neprozkoumaným aspektem zůstává distribuce interakcí s položkou v čase. Doba od přidání položky do databáze může u některých typů komodit mít signifikantní vliv na jejich relevanci pro uživatele. Dobře to můžeme ilustrovat v případě novinových článků, které jsou relevantní pouze několik dnů. Doporučení několik let starého článku o politickém dění pravděpodobně nebude relevantní, leda, že by měl silné konotace k aktuálním událostem. Takových článků bude ale v každém okamžiku jen malá menšina oproti všem ostatním.

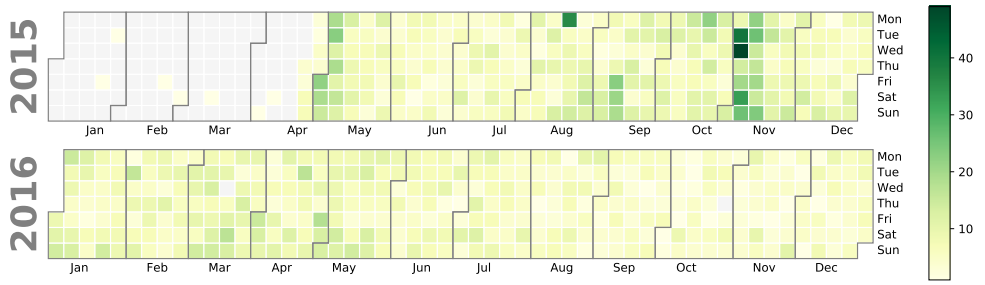
Zde chceme prozkoumat, jestli i položky či uživatelé z našich datasetů vykazují nějaké chování v čase, např. peaky, periodicita a jiné. Zde uvádíme zobrazení pro několik málo jednotlivých položek z obou datasetů, další zobrazení ponecháváme vzhledem k jejich rozsahu do přílohy. Interakce zobrazujeme jako tepelnou mapu v podobě kalendáře, kdy každé políčko odpovídá jednomu dnu a jeho barva množství interakcí v tomto dnu. Pro vytvoření všech grafů byla využita volně dostupná knihovna `Calplot`[12].

V případě filmů můžeme vypozorovat smysluplnou korelaci v případě některých titulů. Film Avengers měl premiéru v roce 2012. V dalších letech množství jeho interakcí stagnovalo bez viditelnějších výchylek, tedy tak, jak vidíme v první polovině obrázku 2.23. V druhé polovině roku 2015 ovšem přichází peak, který je možné spojit s vydáním dalšího dílu série, filmem Avengers: Age of Ultron. Na obrázku 2.24 vidíme, že druhému filmu se největší divácké pozornosti dostalo v listopadu roku 2015, což přesně koresponduje s prvním filmem. Většina prozkoumaných filmů se chová přesně takto - peak po

[12] https://github.com/tomkwok/calplot



Obrázek 2.23: Film Avengers a peak jeho hodnocení v roce 2015 - tři roky po premiéře.

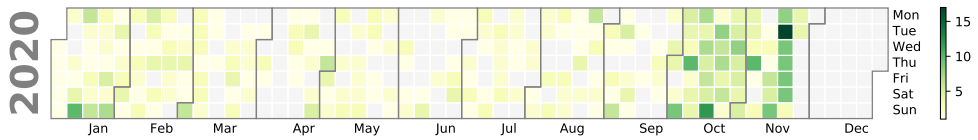


Obrázek 2.24: Interakce s Filmem Avengers: Age of Ultron ve dvou letech po preméře.

jejich vydání a následně vymizení zájmu. Výjimkou zůstávají nejoblíbenější tituly jako Pulp Fiction nebo Vykoupení z věznice Shawshank, které se těší konstantnímu zájmu publika.

Druhý dataset s recepty má interakce rozvrstvené rovnoměrně v čase. U prakticky všech receptů můžeme pozorovat nárůst ve druhé polovině roku 2020 tak jako na obrázku 2.25, to je ale do velké míry zapříčiněné probíhající marketingovou kampaní. Obecně bylo patrné, že u receptů nevykazují uživatelé žádnou periodicitu či jinou proměnlivost zájmu. Vzhledem k povaze receptů dávají takové interakce více méně smysl, u oblíbených receptů bude přízeň publika v čase konstantní a rozhodně neopadne chvíli po jejich zveřejnění.

Obecně můžeme říci, že v této analýze vidíme přínos a potenciál. Můžeme identifikovat různé na první pohled ne zcela zřejmé vztahy, jako jsme viděli v případě filmů Avengers, kdy interakce jednoho filmu korelují s druhým (což neznamená apriorní kauzalitu, ale zde se vyloženě nabízí). Dovozujeme, že výsledky se budou signifikantně lišit v závislosti na typu dat, a tedy by byly zajímavější v případě zmiňovaných novinových článků.



Obrázek 2.25: Interakce s nejoblíbenějším receptem - krémové toskánské kuře s česnekem.

Závěr

V této práci jsme zmapovali množství technik pro exploraci, zobrazení a vysvětlení vztahu publika k obsahu a experimentálně ověřili jejich přínos na dvou datasetech. Navrhli jsme vlastní metodu pro trénování vektorové reprezentace uživatelů a položek v jednom společném latentním prostoru. Menší dataset s recepty se obecně ukázal být obtížný pro všechny vizualizační metody, ovšem u dataasetu Movielens dosáhla námi navržená metoda úspěšných vizualizací.

Filmový dataset obsahuje velké množství uživatelů, které považujeme za nevyhraněné - ti s preferencí nejčastějších žánrů jako komedie a dramata. Metoda centroidů dokázala mezi nimi objevit shluky vyhraněnějších uživatelů z méně častých kategorií (romantické filmy, horrory, ...) tak, jak vidíme na obrázku 2.13. Při omezení trénovací množiny na pouze tyto kategorie pak bylo dosaženo vysoké míry separace tříd na obrázku 2.14. Překryvy některých tříd zde dle našeho názoru odpovídají jejich sémantické blízkosti. Zjištěná struktura (rozdělení do shluků v prostoru) navíc odpovídá konvenčním vizualizačním technikám, proto považujeme tuto vizualizaci za přesnou.

Vztah uživatelů i různých dalších typům položek (tedy ne jen k jednomu majoritnímu pomocí přiřazení třídy) jsme zkoumali pomocí Sankeyových diagramů. Ty dle našeho názoru nabízí jiný a velmi užitečný vhled do interakcí poublika - se kterými dalšími kategoriemi položek interagují uživatelé ze shluků. Ukázalo se ovšem, že shlukování podle atributů (tříd) není dostatečné a v budoucnu bude třeba nalézt lepší způsob pro vytvoření shluků, který bude však stále zachovávat vysvětlitelnost.

Pro vizualizaci interakčních vektorů uživatelů jsme dále využili dvě další techniky, samoorganizační mapy a MDE. Dle našeho nejlepšího vědomí je využití SOM pro řídká interakční data prakticky neprobádané, samoorganizační mapy ale dosáhly kvalitních vizualizací, které jsou dle našeho názoru více informativní, než klasický bodový graf.

Minimum-Distortion Embedding zcela nový framework, který může sloužit jak pro vizualizaci, tak pro extrakci příznaků. Má několik deklarovaných výhod

oproti rozšířenému t-SNE. Ověřili jsme jeho schopnost přidávání dalších uživatelů do projekce, která se ukázala být velmi dobrá. MDE se tedy vyrovná t-SNE kvalitou vizualizace a předčí jej všestranností a rychlostí výpočtu.

V poslední části jsme se zaměřili na porozumění vývoje interakcí v čase. Zvolená forma zobrazení pomocí tepelné mapy po dnech nabízí detailní vhled do časové řady interakcí. Oba zvolené datasety podle ní neobsahují výrazné změny v chování publika, ty by byly pravděpodobně znatelnější v případě jiného typu dat, jako mohou být novinové články.

Věříme, že navržené metody mohou být užitečné při komplexní analýze publika a přinesou nový vhled do chování uživatelů.

Literatura

[1] Editorial Analytics: How News Media Are Developing and Using Audience Data and Metrics. Online; navštíveno: 10.02.2021. Dostupné z: https://www.digitalnewsreport.org/publications/2016/editorial-analytics-2016/

[2] Chen, S.; Andrienko, N.; Andrienko, G.; aj.: LDA Ensembles for Interactive Exploration and Categorization of Behaviors. *IEEE Transactions on Visualization and Computer Graphics*, ročník 26, č. 9, 2020: s. 2775–2792, doi:10.1109/TVCG.2019.2904069.

[3] Xiong, R.; Donath, J.: PeopleGarden: Creating Data Portraits for Users. In *Proceedings of the 12th Annual ACM Symposium on User Interface Software and Technology*, UIST '99, New York, NY, USA: Association for Computing Machinery, 1999, ISBN 1581130759, str. 37–44, doi:10.1145/320719.322581. Dostupné z: https://doi.org/10.1145/320719.322581

[4] J, G.; Ganguly, S.; Gupta, M.; aj.: Author2Vec: Learning Author Representations by Combining Content and Link Information. In *Proceedings of the 25th International Conference Companion on World Wide Web*, WWW '16 Companion, Republic and Canton of Geneva, CHE: International World Wide Web Conferences Steering Committee, 2016, ISBN 9781450341448, str. 49–50, doi:10.1145/2872518.2889382. Dostupné z: https://doi.org/10.1145/2872518.2889382

[5] Zhang, S.; Liu, H.; Zhang, A.; aj.: Learning User Representations with Hypercuboids for Recommender Systems. 11 2020.

[6] van der Maaten, L.: t-SNE - Laurens van der Maaten. Online; navštíveno: 08.02.2021. Dostupné z: https://lvdmaaten.github.io/tsne/

[7] Kohonen, T.; Honkela, T.: Kohonen network. Online; navštíveno: 31.03.2021. Dostupné z: http://www.scholarpedia.org/article/Kohonen_network

[8] Agrawal, A.; Ali, A.; Boyd, S.: Minimum-Distortion Embedding. *arXiv*, 2021.

[9] Steck, H.: Embarrassingly Shallow Autoencoders for Sparse Data. *CoRR*, ročník abs/1905.03375, 2019, 1905.03375. Dostupné z: http://arxiv.org/abs/1905.03375

[10] Vančura, V.; Kordík, P.: Deep Variational Autoencoder with Shallow Parallel Path for Top-N Recommendation (VASP). 2021, 2102.05774.

[11] How media dashboards help online editorial teams boost readership and engagement. Online; navštíveno: 10.02.2021. Dostupné z: https://www.journalism.co.uk/news/how-media-dashboards-help-online-editorial-teams-boost-readership-and-engagement/s2/a699512/

[12] Podmazina, V.: Audience Segmentation - KDnuggets. Online; navštíveno: 10.02.2021. Dostupné z: https://www.kdnuggets.com/2018/06/audience-segmentation.html

[13] Bremner, J.: Segmentation.ai: Automated Audience-Clustering-as-a-Service in Adobe Experience Platform. Online; navštíveno: 10.02.2021. Dostupné z: https://medium.com/adobetech/segmentation-ai-automated-audience-clustering-as-a-service-in-adobe-experience-platform-261f4099462c

[14] Kumari, N.; R., S.; Rupela, A.; aj.: ShapeVis: High-dimensional Data Visualization at Scale. 2020, 2001.05166.

[15] Opitz, L.: 4 ways to level up your trend research with Conversation Clusters. Online; navštíveno: 10.02.2021. Dostupné z: https://www.talkwalker.com/blog/trend-research-with-conversation-clusters

[16] Blei, D.; Ng, A.; Jordan, M.; aj.: Journal of Machine Learning Research 3 (2003) 993-1022 Submitted 2/02; Published 1/03 Latent Dirichlet Allocation. 02 2003.

[17] Koren, Y.; Bell, R.; Volinsky, C.: Matrix Factorization Techniques for Recommender Systems. *Computer*, ročník 42, č. 8, 2009: s. 30–37, doi: 10.1109/MC.2009.263.

[18] The Eckart-Young Theorem. Online; navštíveno: 07.02.2021. Dostupné z: https://legacy.voteview.com/ideal_point_eckart_young_theorem.htm

[19] Virtanen, P.; Gommers, R.; Oliphant, T. E.; aj.: SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, ročník 17, 2020: s. 261–272, doi:10.1038/s41592-019-0686-2.

[20] Pedregosa, F.; Varoquaux, G.; Gramfort, A.; aj.: Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, ročník 12, 2011: s. 2825–2830.

[21] Embeddings — Crash Course — Google developers. Online; navštíveno: 05.02.2021. Dostupné z: https://developers.google.com/machine-learning/crash-course/embeddings/video-lecture

[22] Mikolov, T.; Chen, K.; Corrado, G.; aj.: Efficient Estimation of Word Representations in Vector Space. 2013, 1301.3781.

[23] Bojanowski, P.; Grave, E.; Joulin, A.; aj.: Enriching Word Vectors with Subword Information. 2017, 1607.04606.

[24] Barkan, O.; Koenigstein, N.: Item2Vec: Neural Item Embedding for Collaborative Filtering. 2017, 1603.04259.

[25] Domingos, P.: A Few Useful Things to Know about Machine Learning. *Commun. ACM*, ročník 55, č. 10, Říjen 2012: str. 78–87, ISSN 0001-0782, doi:10.1145/2347736.2347755. Dostupné z: https://doi.org/10.1145/2347736.2347755

[26] Violante, A.: An Introduction to t-SNE with Python Example. Online; navštíveno: 08.02.2021. Dostupné z: https://towardsdatascience.com/an-introduction-to-t-sne-with-python-example-5a3a293108d1

[27] Vlachos, M.; Dünner, C.; Heckel, R.; aj.: Addressing Interpretability and Cold-Start in Matrix Factorization for Recommender Systems. *IEEE Transactions on Knowledge and Data Engineering*, ročník 31, č. 7, 2019: s. 1253–1266, doi:10.1109/TKDE.2018.2829521.

[28] Hochreiter, S.; Schmidhuber, J.: Long Short-Term Memory. *Neural Comput.*, ročník 9, č. 8, Listopad 1997: str. 1735–1780, ISSN 0899-7667, doi:10.1162/neco.1997.9.8.1735. Dostupné z: https://doi.org/10.1162/neco.1997.9.8.1735

[29] Hinton, G. E.; Salakhutdinov, R. R.: Reducing the Dimensionality of Data with Neural Networks. *Science*, ročník 313, č. 5786, 2006: s. 504–507, ISSN 0036-8075, doi:10.1126/science.1127647, https://science.sciencemag.org/content/313/5786/504.full.pdf. Dostupné z: https://science.sciencemag.org/content/313/5786/504

[30] Sakurada, M.; Yairi, T.: Anomaly Detection Using Autoencoders with Nonlinear Dimensionality Reduction. In *Proceedings of the MLSDA 2014 2nd Workshop on Machine Learning for Sensory Data Analysis*, MLSDA'14, New York, NY, USA: Association for Computing Machinery, 2014, ISBN 9781450331593, str. 4–11, doi:10.1145/2689746.2689747. Dostupné z: https://doi.org/10.1145/2689746.2689747

[31] Jolliffe, I.: *Principal Component Analysis*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2011, ISBN 978-3-642-04898-2, s. 1094–1096, doi: 10.1007/978-3-642-04898-2_455. Dostupné z: https://doi.org/10.1007/978-3-642-04898-2_455

[32] Kordík, P.: Convnets and Autoencoders. 6. přednáška předmětu Metody výpočetní inteligence v zimním semestru akademického roku 2020/21. Dostupné z: https://courses.fit.cvut.cz/MI-MVI/lectures/05/index.html

[33] He, K.; Zhang, X.; Ren, S.; aj.: Deep Residual Learning for Image Recognition. *CoRR*, ročník abs/1512.03385, 2015, 1512.03385. Dostupné z: http://arxiv.org/abs/1512.03385

[34] Simonyan, K.; Zisserman, A.: Very Deep Convolutional Networks for Large-Scale Image Recognition. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, editace Y. Bengio; Y. LeCun, 2015. Dostupné z: http://arxiv.org/abs/1409.1556

[35] Jordan, J.: Introduction to Autoencoders. Online; navštíveno: 10.03.2021. Dostupné z: https://www.jeremyjordan.me/autoencoders/

[36] Karim, R.: Illustrated: Self-Attention. Online; navštíveno: 25.03.2021. Dostupné z: https://towardsdatascience.com/illustrated-self-attention-2d627e33b20a

[37] Harper, F. M.; Konstan, J.: The MovieLens Datasets: History and Context. *ACM Trans. Interact. Intell. Syst.*, ročník 5, 2015: s. 19:1–19:19.

[38] Bennett, J.; Lanning, S.; Netflix, N.: The Netflix Prize. In *Workshop at SIGKDD-07, ACM Conference on Knowledge Discovery and Data Mining.*, 01 2009, str. 8.

[39] Bertin-Mahieux, T.; Ellis, D. P.; Whitman, B.; aj.: The Million Song Dataset. In *Proceedings of the 12th International Conference on Music Information Retrieval (ISMIR 2011)*, 2011.

[40] Ning, X.; Karypis, G.: Sparse linear methods with side information for Top-N recommendations. 04 2012, s. 581–582, doi:10.1145/2187980.2188137.

[41] Hu, Y.; Koren, Y.; Volinsky, C.: Collaborative Filtering for Implicit Feedback Datasets. In *2008 Eighth IEEE International Conference on Data Mining*, 2008, s. 263–272, doi:10.1109/ICDM.2008.22.

[42] Wu, Y.; DuBois, C.; Zheng, A. X.; aj.: Collaborative Denoising Auto-Encoders for Top-N Recommender Systems. In *Proceedings of the Ninth ACM International Conference on Web Search and Data Mining*, WSDM '16, New York, NY, USA: Association for Computing Machinery, 2016, ISBN 9781450337168, str. 153–162, doi:10.1145/2835776.2835837. Dostupné z: https://doi.org/10.1145/2835776.2835837

[43] Lutins, E.: Ensemble Methods in Machine Learning: What are They and Why Use Them? Online; navštíveno: 31.03.2021. Dostupné z: https://towardsdatascience.com/ensemble-methods-in-machine-learning-what-are-they-and-why-use-them-68ec3f9fef5f

[44] Lin, T.; Goyal, P.; Girshick, R. B.; aj.: Focal Loss for Dense Object Detection. *CoRR*, ročník abs/1708.02002, 2017, 1708.02002. Dostupné z: http://arxiv.org/abs/1708.02002

[45] Arora, A.: What is Focal Loss and when should you use it? Online; navštíveno: 31.03.2021. Dostupné z: https://amaarora.github.io/2020/06/29/FocalLoss.html

[46] Devlin, J.; Chang, M.; Lee, K.; aj.: BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *CoRR*, ročník abs/1810.04805, 2018, 1810.04805. Dostupné z: http://arxiv.org/abs/1810.04805

[47] Cho, K.; van Merrienboer, B.; Bahdanau, D.; aj.: On the Properties of Neural Machine Translation: Encoder-Decoder Approaches. *CoRR*, ročník abs/1409.1259, 2014, 1409.1259. Dostupné z: http://arxiv.org/abs/1409.1259

[48] Chung, J.; Gülçehre, Ç.; Cho, K.; aj.: Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling. *CoRR*, ročník abs/1412.3555, 2014, 1412.3555. Dostupné z: http://arxiv.org/abs/1412.3555

[49] Pang, K.: Self-organizing Maps. Online; navštíveno: 31.03.2021. Dostupné z: https://www.cs.hmc.edu/~kpang/nn/som.html

[50] Kursula: Kohonen Self-Organizing Map in Python and NumPy. Online; navštíveno: 31.03.2021. Dostupné z: https://github.com/Kursula/Kohonen_SOM

[51] Sammon, J. W.: A Nonlinear Mapping for Data Structure Analysis. *IEEE Transactions on Computers*, ročník C-18, č. 5, 1969: s. 401–409, doi: 10.1109/T-C.1969.222678.

[52] van der Maaten, L.; Hinton, G.: Viualizing data using t-SNE. *Journal of Machine Learning Research*, ročník 9, 11 2008: s. 2579–2605.

[53] McInnes, L.; Healy, J.; Melville, J.: UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction. 2020, 1802.03426.

[54] Sankey Diagrams - A Sankey diagram says more than 1000 pie charts. Online; navštíveno: 10.03.2021. Dostupné z: http://www.sankey-diagrams.com/

Příloha A

Rozsáhlejší vizualizace

V této příloze dodáváme pro úplnost vizualizace, které pro svůj rozsah nebyly vhodné do vlastní práce.

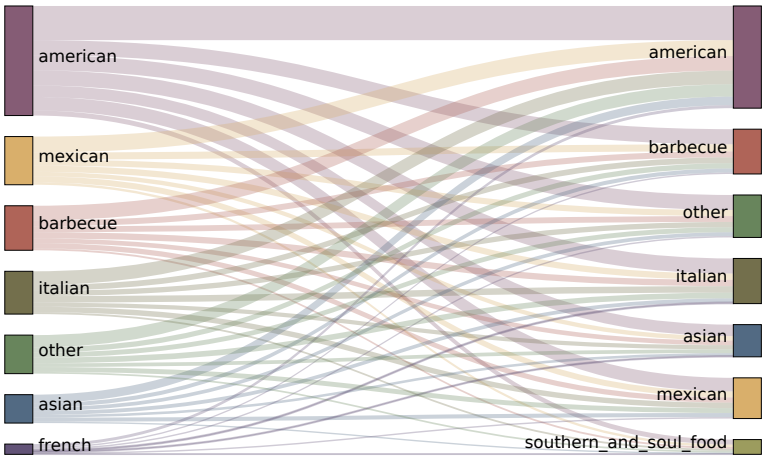
A.1 Sankeyovy diagramy

U datasetu s recepty trpěly Sankeyovy diagramy podobnými nedostatky jako ty u Movielens - nevyhraněnost uživatelů vůči shlukům položek. Na obrázku A.1 vidíme propojení uživatelů ohodnocených kuchyní, která u nich dosáhla nejvyššího tf-idf skóre, se shluky receptů taktéž podle kuchyně. Na obrázku A.2 pak vidíme jiné shluky uživatelů, zde podle chodu a taktéž s využitím tf-idf.

A.2 SOM

V této sekci uvádíme dvě další zobrazení SOM na datasetu s recepty. Na obrázku A.4 jsou zobrazeni uživatelé obarvení podle kategorie kuchyně. Ta byla zvolena podle kuchyně, ze které pocházely recepty, se kterými interagovali nejvíce. Pro přehlednost byli vynecháni uživatelé z kategorie americké kuchyně, která je majoritní třídou.

Z vizualizací můžeme usoudit, že v prostoru interakcí těchto uživatelů neexistují výrazné shluky - mřížka je vyplněna daleko rovnoměrněji než v případě Movielens. Taktéž zvolené obarvení uživatelů podle příslušné kuchyně nepřispívá k separaci, což přináší otázku, vyskytují-li se v daném prostoru vůbec shluky uživatelů - pokud ano, nedaří se nám najít atribut, který by je separoval.

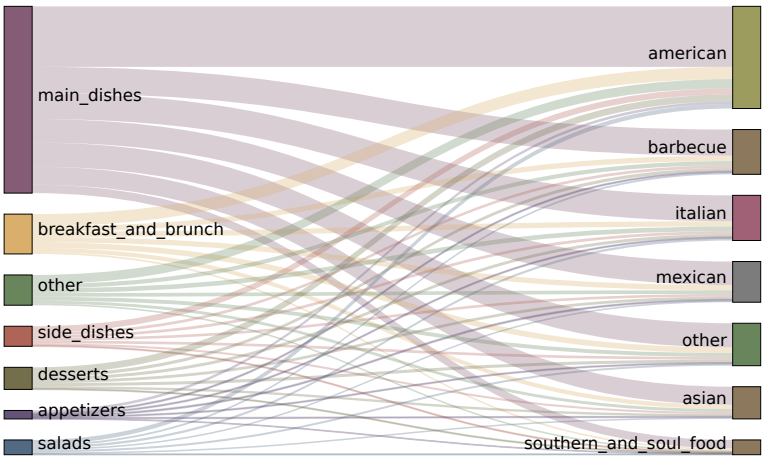


Obrázek A.1: Sankeyův diagram interakcí. Vlevo uživatelé ohodnocení podle nejvyššího tf-idf skóre, vpravo položky - recepty.

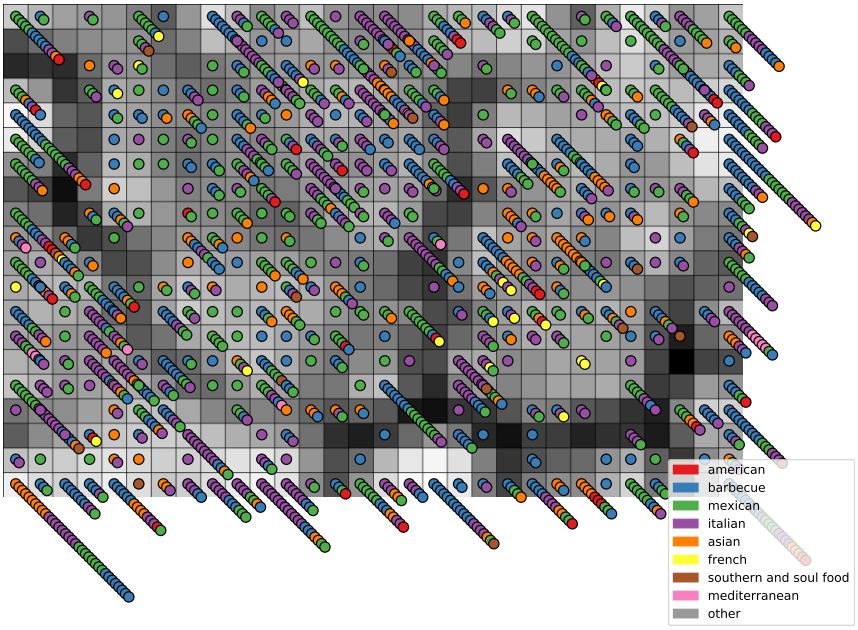
A.3 Interakce v čase

Zde doplňujeme zobrazení interakcí různých skupin uživatelů. Můžeme se podívat na interakce uživatelů z kategorie komedie na obrázku A.6, tedy těch, kteří nejčastěji dobře hodnotili filmy s tímto žánrem. Můžeme vidět, že různé roky znamenají různou aktivitu uživatelů - signifikantní nárůst interakcí po roce 2014, ovšem ani v tomto dlouhodobém časovém horizontu není patrná nějaká periodická aktivita. V případě jednotlivých filmů je proměnlivá aktivita publika dobře vidět u snímku Dunkirk z roku 2017 na obrázku A.5.

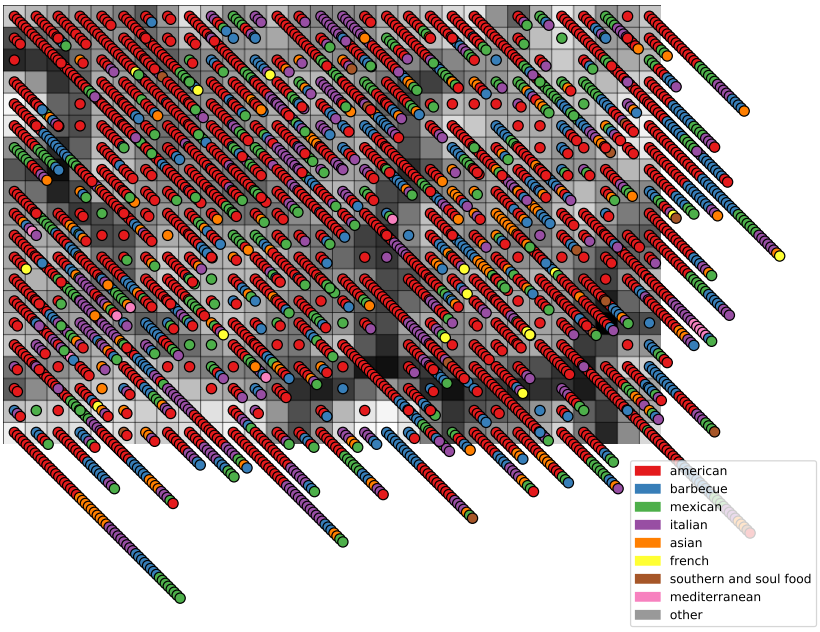
V rámci receptů se můžeme podívat na delší časové období interakcí uživatelů, kteří preferují italskou kuchyni. Jejich interakce vidíme na obrázku A.7. V tomto případě je distribuce dána hlavně životním cyklem celého projektu - vznik v roce 2017 a marketingová kampaň ve druhé polovině roku 2020.



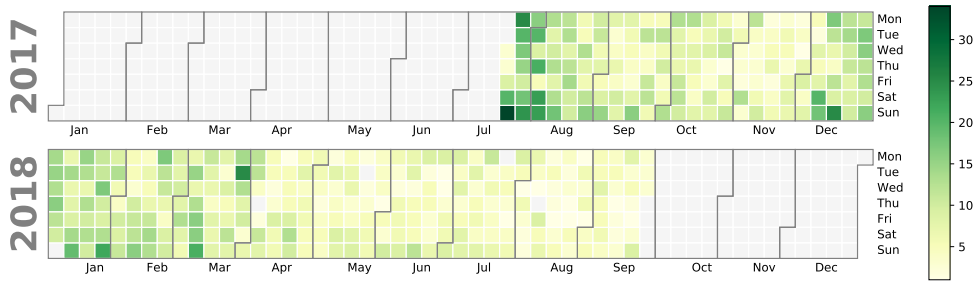
Obrázek A.2: Sankeyův diagram interakcí. Vlevo uživatelé ohodnocení podle nejvyššího tf-idf skóre v kategori chod, vpravo položky - recepty podle kuchyně.



Obrázek A.3: SOM trénovaná na datasetu s recepty. Vynecháni jsou uživatelé z kategorie americké kuchyně.

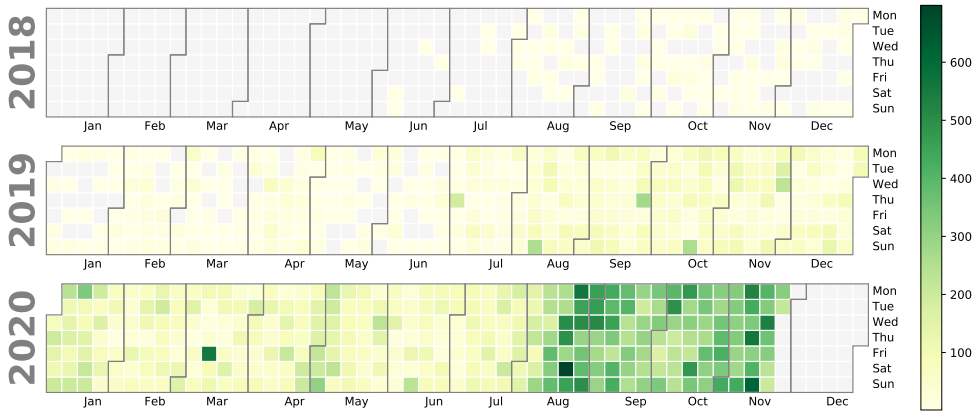


Obrázek A.4: SOM trénovaná na datasetu s recepty. Zde jsou přítomni všichni uživatelé.



Obrázek A.5: Interakce se snímkem Dunkirk

Obrázek A.6: Interakce v čase uživatelů z kategorie komedie.



Obrázek A.7: Interakce uživatelů s preferencí italské kuchyně

Příloha B

Seznam použitých zkratek

MDE Minimum-distortion Embedding

BBC British Broadcasting Company

LDA Latent Dirichlet Allocation

SOM Self Organizing Map

(N)EASE (Neural) Embarrassingly Shallow Autoencoder

SVD Singular Value Decomposition

PCA Principal Component Analysis

DCG Discounted Cumulative Gain

t-SNE t-Distributed Stochastic Neighbor Embedding

GRU Gated recurrent unit

MLP Multi layer perceptron

BMU Best matching unit

UMAP Uniform Manifold Approximation and Projection

Tf-idf Term frequency - inverse document frequency

BERT Bidirectional Encoder Representations from Transformers

ReLU Rectified Linear Units

Příloha C

Obsah přiloženého CD

- `readme.txt` ..... stručný popis obsahu CD
- `hlubik_thesis.pdf` ..... výsledný výtisk práce v podobě pdf
- `latex` ..... zrojové soubory práce ve formátu LaTeX
  - `img` ..... obrázky použité v práci
- `src` ..... zdrojový kód práce v jazyce python
- `sankey` ..... Sankeyho diagramy ve formátu html