

Sem vložte zadání Vaší práce.



ČESKÉ VYSOKÉ UČENÍ TECHNICKÉ V PRAZE  
FAKULTA INFORMAČNÍCH TECHNOLOGIÍ  
KATEDRA SOFTWAREVÉHO INŽENÝRSTVÍ



Diplomová práce

## **Adaptibilní systém pro doporučování obsahu**

***Bc. Jan Bouchner***

Vedoucí práce: Ing. Jaroslav Kuchař

24. dubna 2014



---

## Poděkování

Chci upřímně poděkovat všem, kteří mi věnovali čas, když jsem potřeboval pomoc při psaní této diplomové práce, především vedoucímu práce Ing. Jaroslavu Kuchaři za správné směrování, celkový vhled do technologií a cenné komentáře. Děkuji také své rodině a přátelům za bezvýhradnou podporu během celých mých studií.



---

## Prohlášení

Prohlašuji, že jsem předloženou práci vypracoval(a) samostatně a že jsem uvedl(a) veškeré použité informační zdroje v souladu s Metodickým pokynem o etické 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. § 46 odst. 6 tohoto zákona 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 spracováním 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 24. dubna 2014

.....

České vysoké učení technické v Praze

Fakulta informačních technologií

© 2014 Jan Bouchner. Všechna práva vyhrazena.

*Tato práce vznikla jako školní dílo na Českém vysokém učení technickém v Praze, Fakultě informačních technologií. Práce je chráněna právními předpisy a mezinárodními úmluvami o právu autorském a právech souvisejících s právem autorským. K jejímu užití, s výjimkou bezúplatných zákonných licencí, je nezbytný souhlas autora.*

### **Odkaz na tuto práci**

Bouchner, Jan. *Adaptibilní systém pro doporučování obsahu*. Diplomová práce. Praha: České vysoké učení technické v Praze, Fakulta informačních technologií, 2014.



---

## Abstract

Sem doplňte ekvivalent abstraktu Vaší práce v angličtině.

**Keywords** Nahradte seznamem klíčových slov v angličtině oddělených čárkou.

---

## Abstrakt

V několika větách shrňte obsah a přínos této práce v češtině. Po přečtení abstraktu by se čtenář měl mít čtenář dost informací pro rozhodnutí, zda chce Vaši práci číst.

**Klíčová slova** Nahradte seznamem klíčových slov v češtině oddělených čárkou.



---

# Obsah

<b>Úvod</b>	<b>1</b>
Lidstvo a informace . . . . .	1
Doporučovací systémy . . . . .	3
Motivace . . . . .	4
Cíle práce . . . . .	4
Struktura práce . . . . .	5
<b>1 Aktuální stav na poli doporučování</b>	<b>7</b>
1.1 Příklady systémů . . . . .	7
1.2 Shrnutí poznatků . . . . .	13
<b>2 Analýza a návrh řešení</b>	<b>15</b>
2.1 Požadavky . . . . .	15
2.2 Adaptibilní systém . . . . .	17
2.3 Architektura systému . . . . .	18
2.4 Rozhraní . . . . .	23
2.5 Sada algoritmů . . . . .	25
<b>3 Adaptibilní systém</b>	<b>27</b>
3.1 Strojové učení . . . . .	27
3.2 Multi-armed Bandit algoritmus . . . . .	28
3.3 Bayesian Bandits . . . . .	30
3.4 Nezbytné základy z teorie pravděpodobnosti . . . . .	32
<b>4 Realizace</b>	<b>33</b>
4.1 Principy a technologie . . . . .	33
4.2 Popis implementace . . . . .	34
4.3 Server . . . . .	34
4.4 Klient . . . . .	34
4.5 Sada algoritmů . . . . .	34

4.6	Návrh komunikace . . . . .	34
<b>5</b>	<b>Experimenty a vyhodnocení</b>	<b>35</b>
5.1	Testování různých způsobů chování . . . . .	35
5.2	Experimenty . . . . .	35
5.3	Zhodnocení aplikace . . . . .	35
5.4	Budoucí práce . . . . .	35
	<b>Závěr</b>	<b>37</b>
	<b>Literatura</b>	<b>39</b>
	<b>A Seznam použitých zkratk</b>	<b>41</b>
	<b>B Obsah přiloženého CD</b>	<b>43</b>

---

## Seznam obrázků

0.1	Příklad rozdělení zobrazující popularitu hodnocení . . . . .	3
1.1	Abstraktní model open source systému easyrec . . . . .	13
2.1	Abstraktní pohled na systém společnosti plista . . . . .	18
2.2	Abstraktní návrh architektury a komponent adaptibilního systému	19
2.3	Příklad MQ systému s producentem zpráv a konzumenty . . . . .	21
2.4	Ukázka REST endpointů navrhovaného adaptibilního systému . .	24



---

# Úvod

“We are leaving the age of information and entering the age of recommendation.”

Hned na úvod práce jsem si dovolil použít citát z článku *The Long Tail* [13] (česky Dlouhý chvost) bývalého šéfredaktora časopisu Wired Chrise Andersona, který je též autorem stejnojmenné teorie. Ta je založena na tom, že díky internetu lze nabízet širokou škálu produktů, které by dříve mohly jen sotva slavit prodejní úspěchy, neboť poptávka po nich je příliš malá.

Co je míněno tvrzením „které by dříve mohly jen sotva slavit prodejní úspěchy“?

Myšlenky se lze dopátrat v následující sekci Lidstvo a informace, po kteréžto vysvětlení bude vše zasazeno do kontextu zmíněné teorie (podsektory Informace a The Long Tail Theory) a její souvislosti s potřebou doporučování informací, jež je popsána v sekci Doporučovací systémy.

Sekce Motivace popisuje vizi systému schopného přizpůsobit se podmínkám a predikovat doporučení na základě předchozích a aktuálních zkušeností.

Popisu nároků na podobný systém, jenž jsem prostřednictvím této práce navrhl a následně implementoval, je věnována sekce Cíle práce. V sekci jsou též stanoveny cíle, kterých by měl nabývat implementovaný produkt.

Struktuře práce vzhledem k vytyčeným cílům se věnuje příslušná sekce Struktura práce.

## Lidstvo a informace

Význam informací a dat je pro lidstvo odjakživa nezpochybnitelný. Již před nástupem digitálního věku byl jejich objem značný, takže ke spoustě informací, na základě kterých by si měl jednotlivec utvořit vlastní názor, nebylo snadné se dostat.

Lidé se často spoléhali na obecnou oblíbenost daného produktu (ať už se jednalo o zboží, článek, hudebního interpreta a jiné), která byla ale spíše než

čímkoliv jiným určována aktuálním společenským trendem nebo informace prostě a jednoduše čerpali ze zkušeností svých několika známých na základě jejich doporučení.

### Nástup internetu

S masovým rozvojem internetu začalo množství dostupných informací růst obřím tempem (organizace IDC došla v článku *Extracting Value from Chaos* [17] k závěru, že objem světových dat se každé dva roky zdvojnásobuje). Nové informace jsou produkovány takřka každou sekundu a na jejich setřídění máme mnohem méně času než dříve. Není již v lidských silách udržet si o všem přehled.

Takovým vývojem se přirozeně změnil pohled na data. Cílem není shromáždit co největší objem, neboť při tomto pokusu bychom se zanedlouho dostali do stavu informačního zahlcení. Ceněnou schopností je nyní vytěžit maximální množství užitečných informací a ty využít pro získání nových vědomostí a alespoň přibližně správnou odpověď na otázku potřeb konkrétního jedince.

### Informace a The Long Tail Theory

Vrátím se nyní ještě jednou k teorii Dlouhého chvostu. Pro účely této práce ji nemá význam vysvětlovat dopodrobna, ale je vhodné zmínit alespoň hlavní myšlenku, která vychází z úvah výše.

Nejdůležitějším poznatkem je to, že – z pohledu obchodního – s nárůstem informací dochází k fenoménu snižování prodejů dřívějších hitů (tedy těch několika málo produktů, které se na daném trhu dostanou k masovému publiku, a které určuje především trend) ve prospěch prvků nacházejících se v takzvaném dlouhém chvostu. Uživatel se tak může dostat k informacím, ke kterým by se obvyčejným hledáním přes jeden z vyhledávačů třeba nikdy nedostal (pokud by tedy opravdu přesně nevěděl, co hledá).

Bližší představu si lze utvořit při pohledu na obrázek 0.1. Na vodorovné ose leží jednotlivé produkty, svislá osa pak znázorňuje objem prodeje (stejně tak se může jednat o popularitu). Žlutá část pod grafem vpravo reprezentuje dlouhý chvost (zhruba 80 procent produktů, po kterých je na trhu poptávka v malých objemech). Zelená část nalevo pak znázorňuje těch několik dominujících produktů (označuje se též jako *hlava*). Tento fenomén lze dobře ilustrovat na příkladu hudebního průmyslu, kde díky obrovským databázím interpretů<sup>1</sup> již nedochází k selekci a prodeji či poslechu jen těch potenciálně nejúspěšnějších, ale uživatelům se dostává mnohem většího výběru a je jen na nich, kterého si zvolí.

Všechny výše zmíněné faktory přirozeně zapříčinily rozvoj přístupu zvaného *informační filtrování*, tedy jakési selekce a redukce informací dle volených

---

<sup>1</sup>Jmenujme pár nejznámějších: last.fm, Spotify, Grooveshark, Google Music





Obrázek 0.1: Příklad rozdělení zobrazující popularitu hodnocení. Zdroj: <http://en.wikipedia.org/> [21]

parametrů. Odtud už byl jen krok k doporučování na míru, které se v posledních několika letech rozmohlo zaváděním tzv. *doporučovacích systémů*.

## Doporučovací systémy

Doporučovací systémy jsou systémy, které mohou být uživateli nápomocné v tom smyslu, že z obrovské množiny informací různými způsoby vyfiltrují relevantní podmnožinu. Například prostým pozorováním toho, jak uživatelé službu využívají (v minulosti zakoupené produkty, seznam všech jimi ohodnocených článků či žebříček preferovaných hudebních interpretů), nebo porovnáváním kontextu uživatele s ostatními lidmi využívajícími tu samou službu, lze objevovat pro uživatele nové a jejich vkusu vyhovující položky.

### Typy systémů

Tyto systémy lze dle způsobu práce rozdělit do dvou skupin.

**Kolaborativní filtrování.** První skupinou jsou systémy fungující na principu tzv. kolaborativního filtrování. Tento princip má ještě dvě odnože zahrnující doporučování založené na podobnosti uživatelů (user-based) a doporučení založené na podobnosti položek (item-based). Doporučení z této skupiny nejvíce využívají stránky zabývající se prodejem různých produktů a také stránky nabízející novinové články.

**Doporučení založené na obsahu.** Druhou skupinou jsou pak doporučení založená na obsahu. Pro tato doporučení se nejčastěji využívá textové podobnosti. Výhodou je rychlé absorbování nových položek do systému

a imunita vůči tzv. *cold start problému*<sup>2</sup>. Již na malé množině dat lze dostávat poměrně uspokojivá doporučení.

Další informace o doporučovacích systémech a aktuální stav na poli doporučovacích systémů včetně technologického pozadí dokumentuje následující kapitola 1. Technické detaily jsou v menší míře k nalezení v sekci 4.5 kapitoly 4.

## Motivace

Představme si nyní jakéhosi rádce, který dokáže v každém okamžiku rozhodnout, co je za daných okolností nejvhodnější s tím, že to, o čem rádce rozhoduje, je volba algoritmu, kterým by si měl uživatel v danou chvíli nechat doporučit obsah. Pokud by uživatel dbal těchto rad, s největší pravděpodobností by obdržel vhodné doporučení a svou spokojenost by vyjádřil nějakou zpětnou vazbou (tím, že by klikl na doporučené položky nebo by jiným způsobem vyjádřil svůj zájem).

Dokud rádce explicitně nekomunikuje s uživatelem, neprovádí žádné operace a pouze vyčkává na okamžik, kdy bude vyzván k nějaké akci. V okamžiku, kdy je vyzván k tomu, aby uživateli doporučil obsah, je schopen ihned a bez dlouhé analýzy uskutečnit rozhodnutí.

Dokáže též pružně reagovat na situaci, kdy dojde k náhlé změně preferencí nebo vyvstanou jiné nečekané události, které by měli za následek dlouhodobější doporučování nevhodného obsahu.

## Cíle práce

Cílem práce je tedy vyvinout rádce popsaného výše, a to ve formě aplikačního software.

Nejdůležitějším cílem této práce je tedy návrh a implementace takového adaptibilního systému, jenž by automaticky a vhodně volil a kombinoval metody pro doporučování obsahu. Metodami zde myslí doporučovací algoritmy. Jejich identifikace a výběr vhodné sady je též jednou ze součástí diplomové práce.

Systém bude využívat zpětnou vazbu ohledně kvality doporučení, která principem odměny za dobré doporučení nebo trestu za špatné ovlivní preference při kombinování. Zároveň bude zodpovědný za podporu většího množství uživatelů zasílajících žádosti na systém.

Nezbytnou součástí diplomové práce je též nastudování základních pojmů a pravidel strojového učení a teorie pravděpodobnosti. Zejména kvůli částem zabývajících se zpětnou vazbou a predikcí vhodného algoritmu.

---

<sup>2</sup>TODO

Vzhledem k fragmentované povaze projektu bude nutné navrhnout obecné rozhraní jak pro doporučovací algoritmy, tak pro adaptibilní systém zahrnující zpětnou vazbu.

K nastudování jsou nutné i technologie pro zpracování většího množství dat a pro rychlou odezvu (ta je potřebná již ze samotné podstaty problému).

Funkčnost navrženého systému by měla být ověřena implementací prototypu a spuštěním na modelové úloze, jež bude simulovat doporučování obsahu pro skupinu více uživatelů.

Tato diplomová práce dokumentuje návrh a vývoj takového systému a s ním spolupracujících komponent. Následuje seznam modulů, které byly pro potřeby práce vyvinuty:

- adaptibilní systém pro doporučování obsahu
- sada základních algoritmů určených k doporučování
- rozhraní pro algoritmy a systém pro kombinování
- generátor článků, na kterých bylo prováděno základní měření

## Struktura práce

Tato práce je strukturována do 5 různých kapitol a popisuje celý vývojový cyklus systému. Kapitoly jsou řazeny v tom samém pořadí, v jakém probíhaly fáze vývoje.

**Kapitola** Aktuální stav na poli doporučování popisuje historické pozadí i aktuální trendy používané současnými doporučovacími systémy. Z této rešerše se snaží vyvodit některé závěry, které by mohly být užitečné pro účely některé z dalších kapitol.

**Kapitola** Analýza a návrh řešení shromažďuje veškeré požadavky kladené na systém, zabývá se návrhem architektury, detailními návrhy serverové i klientské části, rozhraním zastřešujícím komunikaci těchto dvou částí a výběrem základní sady algoritmů, které budou využity k doporučování obsahu.

**Kapitola** Adaptibilní systém detailně dokumentuje algoritmické a matematické přístupy použité při návrhu adaptibilního systému.

**Kapitola** Realizace popisuje vývoj aplikace na základě analýzy a návrhu z předchozích kapitol.

**Kapitola** Experimenty a vyhodnocení pak shrnuje poznatky nabyté z experimentů se systémem a hodnotí vyvinutou aplikaci.



# Aktuální stav na poli doporučování

Kamkoliv v dnešní době na internetu zavítáme, máme velkou šanci, že na některý z doporučovacích systémů narazíme. Mohou mít různá jména:

- lidé, které byste mohli znát
- uživatelé kupující produkt X kupují též produkt Y
- položky podobné položce XY

Všechny mají ale stejný význam – zaujmout či upozornit na něco nebo někoho konkrétního. Spousta elektronických obchodů, renomovaných aukčních domů, ale též serverů se zábavou na nich doslova staví svá podnikání, neboť správně navržený a fungující doporučovací systém může firmě přinést výrazné zvýšení zisku.

Sami jejich zákazníci o doporučování stojí. Usnadňují jim navigaci po stránce a existuje vysoká pravděpodobnost, že v závislosti na ní budou přizpůsobovat i své chování. Uživatelská trpělivost ale není neomezená – pokud dostávají špatná doporučení, nenásledují je a ve výsledku odmítají celý systém.

## 1.1 Příklady systémů

Na internetu je k nalezení samozřejmě několik desítek – možná stovek – běžících systémů, záměrně jsem se proto snažil vybrat jen zlomek, který ale pokrývá většinu doporučovaného obsahu (zboží, zábava, text). Níže tedy uvádím jako příklady několik do reálného provozu úspěšně nasazených systémů.

### 1.1.1 Amazon.com

**Amazon.com, Inc.** <sup>3</sup> je jedním z nejstarších a největších internetových prodejců. Společnost začínala svůj provoz jako online knihkupectví, ale postupem let zařadila do své prodejní nabídky též hudební a filmové nosiče, software, elektroniku, nábytek a spoustu dalšího zboží.

Novinkou posledních let je vlastní spotřební elektronika v podobě čtečky elektronických knih a tabletů Kindle či poskytování služeb z oblasti cloud computingu.

Firmu lze řadit mezi průkopníky doporučování na internetu. Jako jeden z prvních internetových prodejců totiž začala svým zákazníkům doporučovat výrobky na základě nákupů jiných uživatelů.

Doporučovací systém je založen na několika zdrojích informací:

- porovnávání uživatelem prohlížených položek a položek umístěných ve virtuálním nákupním košíku s položkami, které se společně s těmito prohlíženými v minulosti často prodávaly <sup>4</sup>.
- udržování informací ohledně hodnocení položek uživateli
- zaznamenávání historie nákupu (pokud uživatel v minulém měsíci zakoupil tři dětské knížky, znamená to, že má dítě?)
- spousta dalších postupů, jako například vyhodnocování demografických informací (dle doručovací adresy), zaznamenávání pohybu po stránce (jaké všechny položky a kolikrát si uživatel prohlédl, než umístil jednu konkrétní do nákupního košíku) nebo sledování prokliků <sup>5</sup> z cílených marketingových e-mailů s odkazy na zboží [5].

Společnost nazývá svou hlavní doporučovací strategii jako *item-to-item kolaborativní filtrování* a používá ji pro přizpůsobení prohlížení webu svým stálým zákazníkům. V tom smyslu, že fanoušek moderních technologií může při své návštěvě stránek nalézt odkazy na technologické novinky všeho druhu, zatímco mladá matka bude mít na těch samých stránkách v nabídce ve větším zastoupení dětské zboží.

Výše jsou popsány pouze základní principy. Doporučovací systém společnosti je samozřejmě velmi komplexní a detailní algoritmus je udržován jako obchodní tajemství. K nahlédnutí jsou ale patenty, např. *Personalized recommendations of items represented within a database* [16] nebo *Collaborative recommendations using item-to-item similarity mappings* [18].

---

<sup>3</sup><http://www.amazon.com>

<sup>4</sup>affinity analysis – nacházení spojení mezi odlišnými položkami. Základním příkladem budiž vztah mezi šamponem a kondicionérem. Kupující je většinou používá v ten samý čas [10]. Při nákupu jednoho by mohl mít tedy zájem i o druhý.

<sup>5</sup>Jako proklik se označuje takové kliknutí na odkaz, které uživatele dovede na cílovou stránku [9].

### 1.1.2 Netflix

**Netflix, Inc.**<sup>6</sup> je společnost, která začínala nabízet své služby jako internetová videopůjčovna. Během posledních pár let (strategicky významný byl rok 2007, kdy byla nabídka rozšířena o filmy streamované prostřednictvím internetu [8]) se rozrostla v obrovskou mediální společnost nabízející obsah v podobě filmů a seriálů pro většinu v dnešní době používaných platforem jako PC, Mac, PlayStation3, Wii, Xbox a také mobilní telefony a tablety.

Vzhledem k tomu, že firma staví své podnikání na tom, že přicházející uživatelé platí za konzumaci zábavy (dle [7] pochází 2/3 zapůjčených filmů z předchozího doporučení), je v jejím vlastním zájmu, aby těmto uživatelům sledujícím filmy a seriály nabízela automaticky další obsahově či žánrově podobné, zkrátka takové, jenž budou co možná nejvíce lahodit jejich vkusu. Úspěšné podnikání společnosti je tak přímo závislé na tom, jak kvalitním doporučovacím systém společnost disponuje.

#### 1.1.2.1 Netflix Prize

Za účelem zkvalitňování doporučování filmů byl vyvinut vlastní systém s názvem *Cinematch*. Potřeba neustálého zlepšování a zpřesňování doporučení vyústila ale v to, že společnost začátkem října 2006 vypsala soutěž známou jako *Netflix Prize*. Jednalo se o pokus ještě více pokročit na poli doporučování filmů a pro tým, který by dokázal zlepšit dosavadní výsledky systému *Cinematch* alespoň o 10 procent, byla vypsána odměna ve výši 1 000 000 dolarů.

K tomuto účelu společnost uvolnila testovací data obsahující ID uživatele, ID filmu, hodnocení na intervalu  $<1,5>$  a datum uskutečnění hodnocení. Testovací data obsahovala 100 480 507 hodnocení pro 17 770 filmů od 480 189 uživatelů. Uvolněna byla ještě další testovací data obsahující stejné informace, jen byla vynechána uživatelská hodnocení. Cílem úkolu pak bylo předpovědět tato chybějící hodnocení opět na intervalu  $<1,5>$ .

Cena byla udělena až v roce 2009 (do té doby docházelo k průběžnému zlepšování, ale nebylo dosaženo stanoveného zlepšení o 10 procent) týmu *Bell-Kor's Pragmatic Chaos* (který vznikl spojením tří do té doby samostatných týmů). Vítězný tým použil k dosažení cíle technik strojového učení, aby při tom zjistil několik zásadních poznatků. Například to, že každé hodnocení filmu je silně subjektivní záležitostí, kterou je dopředu obtížné předpovědět. Ukázalo se také, že velmi záleží na tom, zda uživatel hodnotí právě dosledovaný film nebo film, který zhlédl již před delší dobou. Velkou roli též hraje nálada během dne a další faktory [1].

Výsledný algoritmus je navíc směsicí zhruba stovky menších algoritmů, takže by se dalo s trochou nadsázky prohlásit, že jednou z hlavních taktik je použít tolik doporučovacích algoritmů, kolik je jen možné.

---

<sup>6</sup><https://www.netflix.com>

### 1.1.3 Zite

**Zite.com** <sup>7</sup> je moderní aplikace pro chytré mobilní telefony a tablety.

Mike Klass, CTO společnosti v jednom z článků [11] prozradil, že vizí bylo vyvinout sofistikovaný, na technikách strojového učení postavený systém, jehož účelem nebude pouhé filtrování přichozích novinek vedoucí k úspoře času uživatele. Cílem tohoto systému bylo přizpůsobit se svému uživateli studiem vzorců chování (pozoruje zájmy a návyky při čtení článků) natolik, aby mu dokázal doporučit přesně to, co v danou chvíli hledá, ale kvůli různým okolnostem (například obsah produkovaný neznámým bloggerem) by na to nikdy neměl šanci narazit.

Zite nasazené v systému uživatele je tedy stejně tak unikátní jako jeho uživatel. Sleduje, jaké články si vybírá ke čtení, jak dlouhé tyto články jsou a jak dlouho stráví uživatel jejich čtením. Systém tedy každý den vyhodnocuje miliony nových článků, přičemž se zaměřuje na typ, klíčové atributy i na to, jakým způsobem je článek sdílen skrze web. Následně využije tyto informace ke spárování vybraných článků s osobním vkusem uživatele a tyto články mu doručí do aplikace.

Tak jako Netflix a Amazon doporučují filmy a produkty, které by na základě podobnosti mohli uživatele zajímat, stejně tak i Zite provádí na pozadí porovnání mezi čtenáři – jak v rámci aplikace, tak obecně na webu.

#### 1.1.3.1 Zite & CNN & Flipboard

V srpnu 2011 o technologii vyjádřila zájem společnost CNN [2] a Zite přešlo pod ni. Poslední novinky ze začátku března 2014 ale naznačují [11], že Zite opouští CNN a rozhodlo se spojit síly s aplikací Flipboard <sup>8</sup> vývojářské společnosti Flipboard, Inc., jejíž funkcionalitou je agregování obsahu ze sociálních sítí a jiných stránek a následné nabízení ke čtení v magazínovém formátu umožňujícím jednoduchým způsobem listovat napříč vzájemně souvisejícími tématy.

Společným cílem je vystavět nejlepší systém pro personalizované čtení novinek.

### 1.1.4 Mendeley

**Mendeley** <sup>9</sup> je systém určený k doporučování vědeckých článků využívající jako svou výpočetní vrstvu technologii Apache Mahout <sup>10</sup>. Cílem systému je spojovat dohromady výzkumníky a jejich data. Svým uživatelům tak pomáhá v organizaci výzkumu, umožňuje jim nalézt potenciální spolupráci s dalšími uživateli aplikace a napomáhá též k objevům nových podnětů pro vlastní

---

<sup>7</sup><http://zite.com>

<sup>8</sup><https://flipboard.com>

<sup>9</sup><http://www.mendeley.com>

<sup>10</sup><https://mahout.apache.org>



práci. Uživatelé této aplikace jsou přední světové university jako University of Cambridge, Stanford University, MIT či University of Michigan. Data pro aplikaci pocházejí z vlastních importů svými uživateli i z externích importů skrze různé katalogy prací.

Projekt samotný se v jedné ze svých prezentací [6] přirovnává k největší hudební databázi na internetu – last.fm<sup>11</sup>. Ta funguje na tom principu, že potenciální uživatel si nainstaluje na svůj počítač desktopovou aplikaci, následně s instalovanou aplikací začne poslouchat hudbu a tím je zahájeno automatické odesílání informace o skladbě (interpret, žánr apod.) na server last.fm. Podle těchto odeslaných dat jsou uživateli v budoucnu doporučovány další skladby.

Mendeley tuto analogii vysvětluje tím, že hudební knihovny jsou v jeho případě výzkumné knihovny, roli interpretů zde zastávají jednotliví výzkumníci, hudební skladby jsou pak jimi publikované články a jednotlivé hudební žánry reprezentují vědecké disciplíny.

Doporučení jsou zde generována dvojím způsobem.

**Kolaborativní filtrování** Používá se pro personalizované doporučení. Podporována je jak user-based, tak i item-based varianta doporučení.

**Filtrování založené na obsahu** Používá se k nalezení souvisejícího výzkumu, například nalezení článku ze stejné výzkumné kategorie nebo článku s podobným názvem.

Uživatelé mohou svůj zájem či nezájem o každou z doporučených položek vyjádřit zpětnou vazbou, která je dvojího typu:

**Accept** Vyjadřuje, že uživatel s daným doporučením souhlasí nebo pro něj vylo nějakým způsobem užitečné.

**Remove** Vyjadřuje nevhodné doporučení. Uživatel touto volbou dává najevo, že podobné doporučení by raději již přístě nedostal.

### 1.1.5 Google News

Vlastní platformu pro doporučování obsahu má jako součást svého vývoje též společnost **Google, Inc.**<sup>12</sup>. Její výzkumníci se ve své práci [19] zabývali vývojem prediktivního systému pro personalizované doporučování zpráv na webu Google News<sup>13</sup>. Přihlášeným uživatelům, kteří mají v prohlížeči explicitně povolen záznam historie, jsou generována doporučení založená na zájmu těchto uživatelů vycházejících z profilů sestavených pozorováním chování na stránce.

K porozumění, jak se mění zájem uživatelů o zprávy v čase, napomohla výzkumníkům analýza logů (záznamy chování anonymních uživatelů na stránce).

---

<sup>11</sup><http://www.last.fm>

<sup>12</sup><http://www.google.com/about/company>

<sup>13</sup><http://news.google.com>

Na základě této analýzy byl vyvinut Bayesovský framework pro předvídání zájmu uživatelů o novinky.

Kombinací mechanismu pro filtrování informací, který vzešel z předpovězených uživatelských profilů a již existujícího mechanismu využívajícího principů kolaborativního filtrování vznikl systém pro generování personalizovaných zpráv. Tato kombinace byla nasazena do živého provozu a následné experimenty prokázaly, že kombinováním metod došlo ke zlepšení kvality doporučování, což vedlo k častějším návratům na stránky.

### 1.1.6 Výzkum

Také současný výzkum v oblasti doporučovacích systémů, zdá se, nezahálí. Zde je výčet několika akcí, jejichž náplní či součástí je problematika doporučování a doporučovacích systémů.

- **ACM RecSys conference.** <sup>14</sup> Tato konference je předním mezinárodním fórem pro prezentaci nových výsledků výzkumu a postupů na poli doporučovacích systémů. RecSys sdružuje hlavní mezinárodní výzkumné skupiny a též mnoho předních světových společností na trhu e-commerce. Nabízí také doprovodný program v podobě zvaných přednášek, konzultace týkající se problematiky RecSys a sympózium studentů doktorských programů.

Zajímavé prezentace jsou volně dostupné na internetu, dá se tedy načerpat spousta inspirace do vlastního výzkumu. Mně osobně velmi zaujala prezentace <sup>15</sup> z posledního ročníku konference (Hong Kong, 2013), se kterou vystoupil jeden z klíčových řečníků Torben Brodt. V prezentaci popisuje tzv. *Open Recommendation Platform*, která je založena na výběru vždy toho nejlepšího algoritmu pro doporučení v závislosti na daném kontextu.

- **ICWSM: Weblog and Social Media.** <sup>16</sup> The International AAAI Conference on Weblogs and Social Media je mezinárodní konference, na které se střetávají výzkumní pracovníci z oblasti počítačových a společenských věd. Konference je pořádána za účelem sdílení znalostí, diskutování o nápadech a výměny informací. Probíranými body jsou psychologické a sociální teorie, výpočetní algoritmy pro analýzu sociálních médií a jedním z mnoha témat jsou též doporučovací systémy.
- **ICML: Machine Learning.** <sup>17</sup> The International Conference on Machine Learning je konference s bohatou historií. Její první ročník proběhl

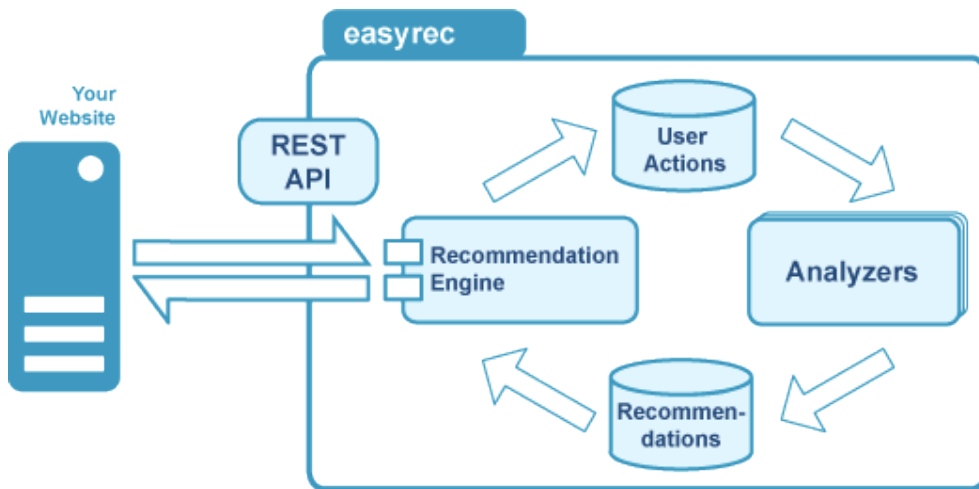
---

<sup>14</sup><http://recsys.acm.org>

<sup>15</sup><http://www.slideshare.net/d0nut/open-recommendation-platform>

<sup>16</sup><http://www.icwsml.org/2014>

<sup>17</sup><http://icml.cc/2014>



Obrázek 1.1: Abstraktní model open source systému easyrec. Zdroj: [3]

již v roce 1980 v Pittsburghu. Jedná se o přední mezinárodní konferenci zabývající se strojovým učením.

O tématu se též píše spousta článků a každým rokem vzniká několik disertačních prací. Dle dostupných informací z ACM RecSys Wiki <sup>18</sup> jich jen za poslední 4 roky bylo přes 50.

Také technická knihovna společnosti IBM obsahuje přehledný seznam <sup>19</sup> několika doporučovacích systémů, z nichž převážná část vznikla jakou součástí univerzitního výzkumu.

Z tohoto seznamu se mi jako zajímavý systém jeví *easyrec* <sup>20</sup>. Jedná se o open source webovou aplikaci napsanou v programovacím jazyce Java nabízející personalizovaná doporučení prostřednictvím RESTful webových služeb. Skrz REST API je možné zasílat uživatelské akce jako prohlížení, nákup či hodnocení položky. Tyto uživatelské akce jsou poté ukládány do databáze *easyrec*, nad kterou je prováděna analýza a z té jsou generována doporučení pro aplikaci uživatele (viz obrázek 1.1).

## 1.2 Shrnutí poznatků

Z výše uvedených příkladů v sekci 1.1 lze vypožorovat určité společné vlastnosti a zformulovat několik závěrů:

- o uživateli je vedena spousta záznamů, například historie jeho chování na stránkách

<sup>18</sup>[http://www.recsyswiki.com/wiki/List\\_of\\_recommender\\_system\\_dissertations](http://www.recsyswiki.com/wiki/List_of_recommender_system_dissertations)

<sup>19</sup><http://www.ibm.com/developerworks/library/os-recommender2/index.html>

<sup>20</sup><http://easyrec.org/recommendation-engine>

## 1. AKTUÁLNÍ STAV NA POLI DOPORUČOVÁNÍ

---

- velký důraz je kladen na zpětnou vazbu (vyjádření zájmu uživatele o obsah stránky)
- stále se využívá osvědčených přístupů kolaborativního filtrování a filtrování na základě obsahu
- síla není v použití jednoho konkrétního algoritmu, ale v kombinaci více metod a přístupů dohromady
- většina výpočtů běží na pozadí v tom samém čase, co uživatel tráví prohlížením stránek, a je schopna se přizpůsobit okolnostem
- problematika má silnou závislost na strojovém učení a teorii pravděpodobnosti

Existujících řešení je tedy celá řada. Každé z těchto řešení je ale využíváno ke svému vlastnímu účelu a není vzájemně přenositelné. Navíc se zdá, že době, kdy obchodníkům stačilo na svůj elektronický obchod nasadit jednoduchý algoritmus kolaborativního filtrování, již odzvonilo a budoucnost patří systémům schopným přizpůsobit vývoj doporučování dle nějaké predikce, a to navíc v reálném čase.

## Analýza a návrh řešení

Zatímco v předchozí kapitole jsem se zabýval popisem existujících řešení a zkoumal, jaké možnosti nabízejí současné doporučovací systémy, tato kapitola je celá věnována analýze požadovaného chování mnou vyvíjeného systému.

Nejprve se budu zabývat požadavky na systém, které vyplynuly z úvodních konzultací s vedoucím práce a vlastním zkoumáním problému. Se znalostí těchto požadavků bude možné provést hrubý návrh architektury systému včetně technického řešení součástí, kterými bude systém disponovat.

Prostřednictvím výstupů získaných v této kapitole budu schopen provést implementaci systému.

### 2.1 Požadavky

Za účelem vyšší přehlednosti jsem se nejprve rozhodl související požadavky strukturovat do zastřešujících skupin trojího typu (globální požadavky, rozhraní a systém). Nakonec se ale ukázalo, že mezi skupinami *rozhraní* a *systém* existuje velmi tenká hranice (z hlediska požadovaného chování, nikoliv implementačního), proto byly skupiny redukovány na dvě.

#### 2.1.1 Globální požadavky na systém

- Systém bude v pravidelných intervalech ukládat do databáze svůj aktuální stav.
- Systém bude schopen v případě pádu aplikace a po jejím opětovném spuštění načíst naposledy uložený stav.
- Systém bude veškeré své operace týkající se doporučování provádět a vyhodnocovat v reálném čase (live read & live write).
- Systém bude na každou klientskou žádost vracet příslušnou odpověď (i chybovou) – neexistuje nic jako odpověď *null*.

- Systém bude klást důraz na zpětnou vazbu, podle které bude přizpůsobovat své chování.
- Systém bude zpětnou vazbu přijímat na principu odměny za dobré doporučení (přičítání v poměru) a trestu za špatné (odečítání v poměru).
- Systém bude schopen automaticky se přizpůsobovat vývoji v čase normalizováním ukládaných hodnoty (z důvodu ochrany proti přetečení datových typů).
- Bude existovat možnost zobrazit dosavadní průběh doporučování systémem v čase.

### 2.1.2 Systém & Rozhraní

- Uživatel bude mít možnost vytvářet skrze systémové rozhraní kolekce se seznamem algoritmů určených pro následnou predikci.
- Uživatel bude mít možnost zaslat systému žádost o radu a obdržet informaci o tom, který algoritmus pro doporučení obsahu by měl v danou chvíli zvolit.
- Uživatel bude mít možnost zaslat systému informaci o tom, jaký algoritmus pro doporučení se rozhodl si zvolit.
- Uživatel bude mít možnost požádat si o doporučení obsahu prostřednictvím algoritmu, jenž mu byl navržen systémem.
- Uživatel bude mít možnost zaslat systému informaci o zpětné vazbě pro daný algoritmus, který si zvolil.
- Aplikace bude umožňovat prostřednictvím svého rozhraní zpracování informace, že jeden uživatel dělá něco s jedním dokumentem (reakce na zpětnou vazbu).
- Aplikace bude umožňovat prostřednictvím svého rozhraní mazání či zneviditelnění pro doporučování již neaktuálních informací.

Dle disciplín softwarového inženýrství je výčet uvedený výše označován jako **funkční požadavky**.

Lze definovat i tzv. **nefunkční požadavky** specifikující vlastnosti a omezující podmínky kladené na systém:

- Systém bude postaven na platformě Java.
- Pro uchování informací o uživateli, položkách a jejich vzájemné interakci bude využita platforma pro vyhledávání v textu Apache Solr.

- Rozhraní systému pro komunikaci s klientem bude navrženo a implementováno v podobě služeb REST (vychází ze zadání).
- Systém bude postaven tak, aby se dal snadno parametrizovat.
- Systém bude připraven na situaci, že jej bude využívat více uživatelů současně.
- Systém poběží jako samostatná aplikace na serveru naslouchající na volném TCP portu. Veškerá komunikace s ní bude probíhat formou zasílaných zpráv.
- Pro snadné nasazení a otestování systému na libovolné pracovní stanici bude nutné vytvořit Chef cookbook a bude jej možno zavést nástrojem Vagrant.

## 2.2 Adaptibilní systém

Pro implementaci systému splňujícího požadavky vzešlé z analýzy výše je třeba důkladně zvážit, jaké všechny komponenty bude tento systém obsahovat a případně jaké algoritmické postupy bude vhodné využít. Na základě zkoumání existujících implementací v kapitole 1 mě zaujal projekt *Open recommendation Platform* (zkr. ORP) společnosti plista<sup>21</sup> prezentovaný na posledním ročníku konference ACM RecSys 2013 v Hong Kongu.

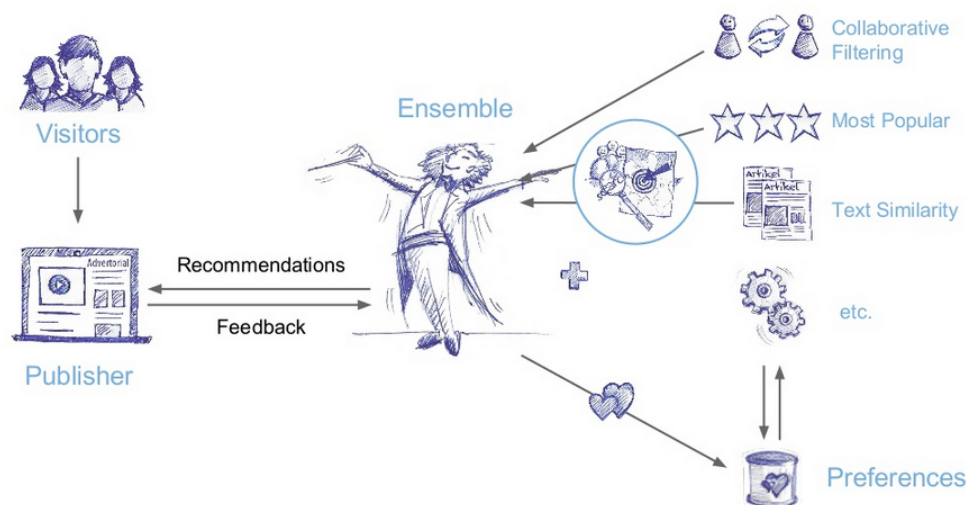
Motivací ORP bylo dosáhnout lepších výsledků kombinací více metod doporučování obsahu společně se zapojením kontextu uživatele (informacemi čerpanými převážně z HTTP hlaviček). Takovými informacemi mohou být:

- IP adresa, která může prozradit geologickou lokaci uživatele
- denní doba, kdy uživatel přistupuje ke stránce
- user agent informující o zařízení, ze kterého bylo k obsahu přistoupeno (mobilní telefon, PC)
- referer pro zjištění způsobu přístupu (přístup z vyhledávání nebo přímý přístup)

Abstraktní pohled na ORP znázorňuje obrázek 2.1, na kterém je systém zachycen jako *Ensemble*. Ensemble je schopen dle uživatelských preferencí vybírat z kolekce doporučovacích algoritmů ten nejvhodnější, jenž použije pro publikování obsahu. Konzumenti doporučeného obsahu (*Visitors*) zasílají poté systému zpětnou vazbu, díky které dojde k rekalkulaci preferencí, což ovlivní výběr budoucího algoritmu.

---

<sup>21</sup><https://www.plista.com>



Obrázek 2.1: Abstraktní pohled na systém společnosti plista. Zdroj: [14]

V prezentaci bylo zmíněno několik užitečných rad ohledně toho, co všechno by měl systém podobného typu umět. Zmíněna je potřeba rychlého webového serveru, na kterém bude aplikace běžet, dále nutnost rychlého síťového protokolu a rychlé fronty zpráv. Padla zde též nutnost rychlého úložiště dat. K dosažení funkcionality pro výběr nejlepšího algoritmu za daných podmínek společnost používá tzv. *multi-armed bayesian bandit*, což byla i jedna ze strategií, které mi na úvodní schůzce doporučil vedoucí této práce.

Se spoustou rad z této prezentace jsem se ztotožnil a vyhodnotil, že s jejich pomocí by mohla být uspokojena velká část požadavků na systém stanovených výše.

Ukázalo se, že strategie *multi-armed bayesian bandit* je pro mou práci natolik významná, že jsem se rozhodl věnovat jí samostatnou kapitolu 3.

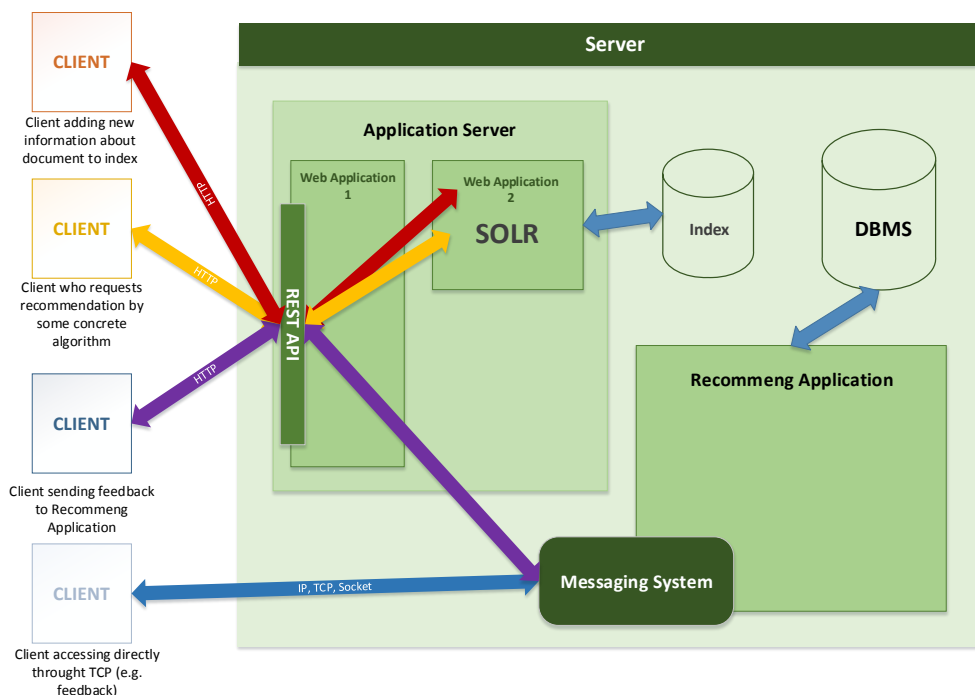
## 2.3 Architektura systému

### 2.3.1 Server

Stěžejní serverovou komponentou je doporučovací platforma. Tu jsem pracovně nazval jako **Recommeng** (zkratka pro recommendation engine), abych o ní již nadále nemusel mluvit jako o *adaptibilním systému* či *systému pro kombinování metod*.

Diagram 2.2 znázorňuje abstraktní návrh architektury takové platformy. Mou snahou bylo zachytit přítomnost jednotlivých komponent systému, formu jejich vzájemné komunikace a též základní interakci uživatele se systémem.





Obrázek 2.2: Abstraktní návrh architektury a komponent adaptibilního systému, ve kterém je též vidět základní interakce uživatele se systémem.

V rámci ucelenosti zde uvádím přehled všech komponent nacházející se na serverové straně.

### 2.3.1.1 Aplikační server

Na aplikačním serveru poběží dvě aplikace.

**Webová aplikace s rozhraním pro snadnou komunikaci s klienty.** (viz diagram 2.2, komponenta *Web Application 1*) Aplikace bude mít na starosti obsluhu klientských požadavků na systém prováděných prostřednictvím protokolu HTTP. Zároveň bude disponovat sadou algoritmů, které lze použít pro doporučení obsahu. Dále bude aplikace obstarávat komunikační režii s druhou aplikací běžící na aplikačním serveru (*Web Application 2*), kterou je platforma pro vyhledávání v textu – Apache Solr.

**Apache Solr** (viz diagram 2.2, komponenta *Web Application 2*) Přítomnost této aplikace vychází z nefunkčních požadavků. Apache Solr funguje jako samostatná komponenta umožňující fulltextové vyhledávání<sup>22</sup> nad

<sup>22</sup> Obecné označení pro vyhledávače fungující na základě porovnávání fráze se všemi ostat-

dokumenty. K indexování a vyhledávání Solr používá knihovnu Apache Lucene (detaily v kapitole 4).

### 2.3.1.2 Databáze

Kvůli potřebě ukládání průběžného stavu aplikace (časové snímky) a snadného obnovení v případě přerušení běhu systému je nutné zapojit do návrhu databázi. Database Management System (DBMS) hraje důležitou roli i při škálování aplikací. V minulosti tolik používané standardní relační DBMS mohou způsobovat zpoždění při provádění čtení/zápisu a v některých případech hrát roli úzkého hrdla aplikace (bottleneck). Ohledně tohoto problému by možná stálo za úvahu prozkoumat možnosti použití NoSQL databází, které jsou již ze svého principu navrženy pro spolupráci s aplikacemi se zaměřením na výkon a škálovatelnost.

### 2.3.1.3 Reccomeng Application

Komunikaci s aplikací bude umožňovat zapojení systému pro zprávy (Messaging System) s využitím fronty zpráv (Message Queue). Obecný příklad takového MQ systému znázorňuje obrázek 2.3

Producenti zpráv (klienti využívající aplikaci) vytvářejí zprávy, které jsou zasílány do fronty. Zprávy jsou z fronty postupně odebírány konzumenty, kteří mají jejich zpracování na starosti. Výhodou je časová nezávislost mezi producenty zpráv a jejich konzumenty – producent tak může zasílat zprávy i tehdy, když žádný z producentů není k dispozici. Zprávy pak existují ve frontě, dokud se o jejich zpracování konzument nepřihlásí.

Aplikace taktéž komunikuje s databází, ale využívá ji pouze pro ukládání informací o stavu, které je prováděno v pravidelných časových intervalech, respektive pro načítání posledního uloženého stavu při startu aplikace. Důvodem pro vynechání komunikačního mezikroku s databází je požadavek na rychlost systému – veškerá data pro výpočet a predikci budou udržována přímo ve vnitřní paměti (uvnitř JVM).

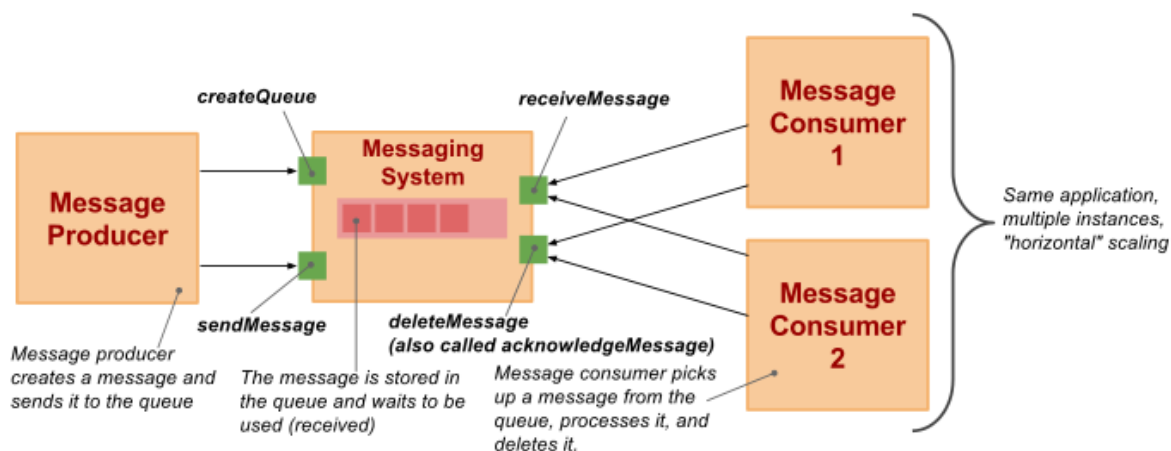
### 2.3.2 Klient

Klientská strana není nikterak složitá. Potenciální uživatel aplikace má v zásadě dvě možnosti, jak s platformou komunikovat:

- jen a pouze zasíláním žádostí na REST API
- kombinací zasílání žádostí na REST API společně s přímou komunikací s Reccomeng systémem prostřednictvím fronty zpráv

---

*ními slovy v daném dokumentu. Fulltext je tedy souhrn algoritmů, které dokáží z daného dokumentu vytvořit podrobnou statistiku výskytu jednotlivých pojmů a tu zanést do databáze (říkáme, že dokument naindexují). Při vyhledávání pak algoritmy fulltextu porovnají hledané klíčové slovo s touto databází. [4]*



Obrázek 2.3: Příklad MQ systému s producentem zpráv a konzumenty. Zdroj: [20]

Některé ze systémových funkcionalit nelze bez komunikace s REST API využívat. Do této kategorie spadá například přidání nového vztahu k položce ve fulltextovém indexu nebo zaslání žádosti o doporučení obsahu některým z podporovaných algoritmů.

Přímé spojení s platformou skrze frontu zpráv bez nutnosti zapojení prostředníka v podobě REST API by ale měly umožňovat všechny možnosti použití Recommeng systému. Jmenovitě vytváření kolekcí se seznamem algoritmů, dále zasílání žádostí o radu pro výběr algoritmu a též metody informující systém o uživatelském chování jako zvolení konkrétního algoritmu nebo zaslání zpětné vazby o doporučení.

V diagramu 2.2 jsem se snažil zachytit různé formy komunikace klienta se systémem. Pro lepší názornost jsou jednotlivé žádosti barevně odlišeny. Ty lze považovat za modelové případy užití inspirované systémovými požadavky.

### 2.3.2.1 Červená varianta

Aneb klient přidávající nový vztah mezi položkou a jejím uživatelem do úložiště.

- klient zašle žádost na rozhraní
- aplikace (Web Application 1) za rozhraním se spojí s indexem
- v případě zdárného průběhu se přidá do indexu informace o tom, že 1 uživatel dělá něco s 1 dokumentem
- klient obdrží odpověď s výsledkem žádosti

### 2.3.2.2 Fialová varianta

Aneb klient zasílající platformě zpětnou vazbu ohledně doporučení, které dostal.

Fialová varianta může též představovat situaci, kdy klient žádá systém o radu, kterým algoritmem si má nechat doporučit obsah ve své budoucí žádosti o doporučení (viz žlutá varianta 2.3.2.3).

- klient zašle žádost na rozhraní
- aplikace (Web Application 1) za rozhraním se pokusí přistoupit k frontě zpráv
- pokud fronta existuje, žádost je předána do fronty pro zpracování konzumentem (aplikace Recommeng)
- konzument zpracuje žádost a zasílá odpověď, kterou systém zpráv předá zpět do aplikace
- klient obdrží odpověď s výsledkem žádosti

### 2.3.2.3 Žlutá varianta

Aneb klient poptávající doporučení obsahu konkrétním algoritmem.

Toto doporučení klient poptává na základě odpovědi na žádost, která mu byla udělena systémem (viz fialová varianta 2.3.2.2)).

- klient zašle žádost na rozhraní
- aplikace (Web Application 1) za rozhraním zvolí vybraný algoritmus pro doporučení
- algoritmus přistoupí k datům v indexu, nad kterými provede doporučení
- klient obdrží odpověď s výsledkem žádosti

### 2.3.2.4 Modrá varianta

Aneb klient, který se rozhodl nevyužít možnosti komunikovat prostřednictvím REST API. Svou žádost zasílá přímo do fronty zpráv.

- klient zašle žádost do fronty
- pokud fronta existuje, žádost je předána do fronty pro zpracování konzumentem (aplikace Recommeng)
- konzument zpracuje žádost a zasílá klientovi odpověď

### 2.3.3 Komunikace

Výše jsme měli možnost poznat dva způsoby komunikace uživatele s doporučovací platformou. Dalším úkolem je navrhnout formát, jakým si budou vyměňovat producent s konzumentem data prostřednictvím fronty zpráv za účelem vzdáleného volání metod Recommeng systému.

Vzhledem k obecné známosti protokolu HTTP jsem se rozhodl napodobit jeho chování a zachovat sémantiku:

- metoda (*method*)
- cesta (*path*)
- tělo zprávy (*body*)

Podobně jako nabízí HTTP, na každou žádost ve tvaru *method*, *path* a *body* přijde odpověď ve tvaru *status* a *body* s využitím různých návratových kódů pro stav (*status*).

## 2.4 Rozhraní

Platforma poskytuje API pro interakci s daty formou RESTful (HTTP implementace REST). Každý uživatel tak může interagovat s rozhraním prostřednictvím veřejných REST endpointů.

Z prováděné analýzy vyplynuly požadavky na aplikační rozhraní takové, že je lze rozdělit do tří skupin dle podstaty úkolu, který mají plnit. K dispozici je diagram 2.4, který by měl celou věc osvětlit.

### 2.4.1 Rozhraní pro Recommeng

Komunikace s rozhraním by měla umožňovat vytvářet v Recommeng aplikaci kolekce s identifikátory algoritmů, jež budou zapojeny do predikce.

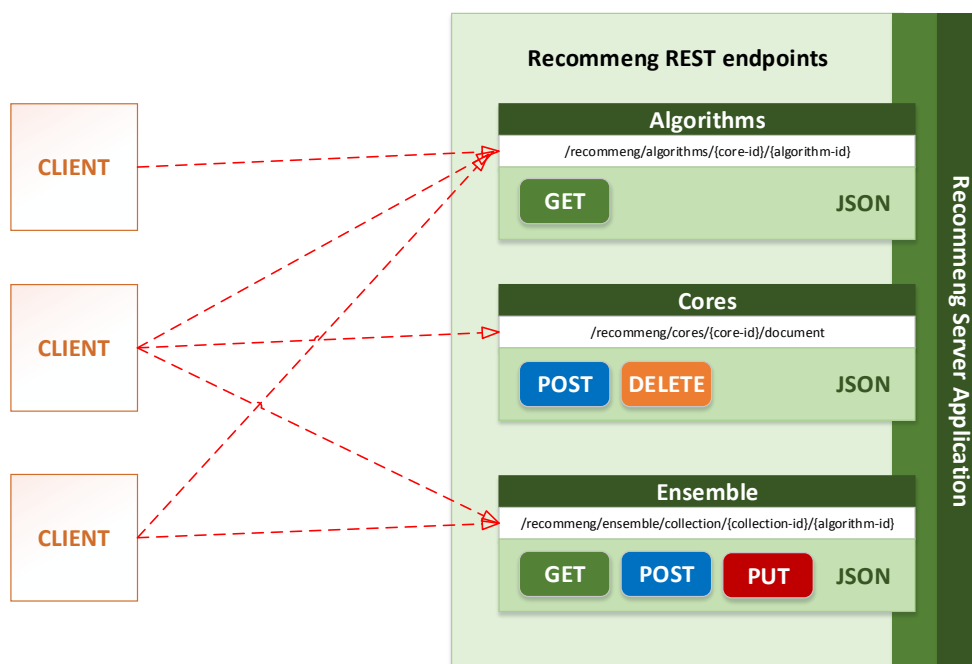
K tomu účelu bude sloužit endpoint:

```
/recommeng/ensemble/collection
```

Pokud uživatel zašle metodou POST žádost na tento endpoint (obsahující identifikátor kolekce a seznam algoritmů), v případě neexistence v systému bude kolekce vytvořena a připravena k predikci.

Dále by měla existovat možnost zaslat na kolekci prosbu o predikci nejlepšího jejího algoritmu pro doporučení.

```
/recommeng/ensemble/collection/{collection-id}
```



Obrázek 2.4: Ukázka REST endpointů navrhovaného adaptibilního systému

Toho dosáhneme žádostí s metodou GET, uvedením identifikátoru kolekce a případným specifikováním požadovaného výstupu (zda chceme vrátit jen nejlepší možnost či všechny možnosti seřazené sestupně od nejlepšího) v parametru dotazu.

Kvůli učení a vývoji znalostí systému je nezbytně nutné mít možnost zaslat informaci o výběru konkrétního algoritmu a též zpětnou vazbu vyjadřující, jak byl žádající uživatel s navrhovanou variantou spokojen.

`/recommeng/ensemble/collection/{collection-id}/{algorithm-id}`

Na endpoint výše v takovém případě zašleme metodou PUT upřesňující informaci o operaci, kterou chceme systému sdělit (výběr vs. zpětná vazba).

### 2.4.2 Rozhraní pro sadu základních algoritmů

Bude se jednat o jednoduchý endpoint, který si při žádosti vystačí s metodou GET.

`/recommeng/algorithms/{core-id}/{algorithm-id}`

Úkolem je zpracovat žádost uživatele o doporučení algoritmem, jehož identifikátor je uveden v cestě. Hledání doporučení bude provedeno nad dokumenty

ze specifikovaného indexu (identifikátor indexu je též nutno uvést). V parametrech dotazu lze specifikovat další vstupy pro doporučení jako limit vrácených výsledků, identifikátor pro doporučení článků z té samé skupiny a další.

### 2.4.3 Rozhraní pro úložiště dat

Důležité rozhraní umožňující zaznamenat chování uživatele vzhledem ke sledovaným položkám.

`/recommeng/cores/{core-id}/document`

Prostřednictvím metody POST budou zasílány veškeré informace o položce (u článku např. identifikátor, text, skupina, datum publikace) a jejím uživateli (identifikátor), která má být uložena do fulltextového indexu. Druhou podporovanou metodou je metoda DELETE pro vyřazení položky z doporučování. Metoda je použita v žádosti na též endpoint jako metoda POST, pouze je nutné v parametru dotazu specifikovat identifikátor článku (z toho důvodu, že identifikátorem článku může být cokoliv, například URL adresa).

## 2.5 Sada algoritmů

Identifikace základní sady algoritmů určených pro kombinování je jeden z požadavků na systém plynoucí přímo ze zadání.

Uživatel žádající o radu pro výběr nejlepšího algoritmu obdrží od systému identifikátor reprezentující tento algoritmus.

Algoritmy pro doporučování vyhodnocují především informace o uživateli, položkách a hodnocení. Ukládány jsou ale i další údaje, například datum zveřejnění položky či její popis (u článku se může jednat o text zprávy). Také s těmito informacemi mohou doporučovací algoritmy pracovat.

### 2.5.1 Algoritmus náhodného výběru

Jak je patrné již z názvu, tento algoritmus vybírá položky pro doporučení naprosto náhodným způsobem. Jeho hlavním úkolem je být zde pro srovnání s ostatními algoritmy.

### 2.5.2 Algoritmus výběru dle nejnovějších položek

Doporučování dle nejnovějších položek je dalším z algoritmů s naivním přístupem k problému. Položky budou v tomto případě doporučovány sestupně dle zveřejněného data.

### 2.5.3 Algoritmus výběru nejlépe hodnocených položek

Jedná se o první algoritmus založený na složitějším výpočtu. Položky vzešlé z doporučení budou dle určitých parametrů nějakým způsobem lepší než ostatní, které se v doporučení neobjevily. Takovým parametrem může být například hodnocení na škále od 1 do 5, počet pozitivních hodnocení, souhrnné číslo udávající počet přečtení článku nebo jiný z mnoha způsobů vyjádření zájmu o položku.

### 2.5.4 Algoritmus výběru dle podobnosti obsahu

Algoritmus se snaží na základě podobnosti obsahu nalézt pro položku několik jí podobných položek z databáze. Podobnost se určuje porovnáním jednotlivých parametrů, například tagů, nadpisů nebo celého textu článku.

### 2.5.5 Algoritmus kolaborativního filtrování

Algoritmus je založen na modelu dřívějšího chování uživatele v systému. Model je většinou konstruován z chování většího množství uživatelů s podobným vkusem. V podstatě lze říci, že doporučení jsou založena na automatické spolupráci více uživatelů a výběru těch, kteří mají co nejpodobnější preference či chování.

Rozlišují se dva hlavní způsoby filtrování.

#### 2.5.5.1 User-based

“You may like it because your friends liked it.” [15]

Aneb filtrování založené na uživateli. Jedná se o starší variantu kolaborativního filtrování. Podstatou je vzít na základě určité podobnosti skupinu (zdroj [15] udává cca 20 až 50) uživatelů s podobným vkusem jako má uživatel, pro něhož je doporučení konstruováno, a poté předpovědět, jak moc zajímavá by pro uživatele byla pro něj neznámá položka, se kterou jsou spojení uživatelé se stejným vkusem.

#### 2.5.5.2 Item-based

“You tend to like that item because you have liked those items.” [15]

Aneb filtrování založené na položkách, které použila v roce 2001 jako první společnost Amazon. Myšlenka je taková, že uživatel, který si v minulosti zakoupil nějakou položku, bude v budoucnu při dalším nákupu vyhledávat položku podobnou. Například předpověď toho, co si uživatel zakoupí v budoucnu, lze uskutečnit analýzou uživatelské historie nákupů [12].



# Adaptibilní systém

## 3.1 Strojové učení

Vzhledem k povaze řešeného problému je nutné zaměřit se na metody tzv. *strojového učení*. Jedná se o vědeckou disciplínu (jednu z větví oboru umělé inteligence), která se zabývá tím, jak se má počítač přizpůsobit určité situaci, aniž by byl pro danou situaci explicitně naprogramován. Metody strojového učení, které hledají generalizaci libovolných dat nebo slouží pro adaptaci existujícího systému na změny okolí, se používají ve všech oblastech informačních věd od analýzy snímků, analýzy DNA a textu, až po simulace chování člověka.

Typickým příkladem je vytvořit z dostupných dat model, který dokáže:

- predikovat cenu akcií za 6 měsíců z aktuální výkonnosti společnosti a ekonomických dat
- rozpoznat spam od regulérního e-mailu
- u pacienta hospitalizovaného s infarktem predikovat riziko dalšího infarktu
- napomoci společnostem zabývajících se internetovou reklamou v rozhodování se, kterou reklamní strategii použít k maximalizaci zisků Používá to ale třeba Google analytics! při správě online experimentů [https://support.google.com/analytics/answer/2844870?hl=cs&ref\\_topic=2844866](https://support.google.com/analytics/answer/2844870?hl=cs&ref_topic=2844866)

Algoritmy strojového učení mohou být děleny do taxonomie (nadtříd a podtříd) založené na požadovaném výsledku algoritmu nebo typu vstupu, který je k dispozici během trénování stroje. Algoritmů je celá řada, bude tedy vhodné zmínit zde alespoň ty nejtypičtější.

### Supervised learning <sup>23</sup> TODO

---

<sup>23</sup>učení s učitelem

Unsupervised learning <sup>24</sup> TODO

Reinforcement learning <sup>25</sup> TODO

TODO zmínit, že pro naše potřeby je nejvhodnější reinforcement learning.

[http://cs.wikipedia.org/wiki/Strojov%C3%A9\\_u%C4%8Den%C3%AD](http://cs.wikipedia.org/wiki/Strojov%C3%A9_u%C4%8Den%C3%AD) [http://cs.wikipedia.org/wiki/Zp%C4%9Bnovazebn%C3%AD\\_u%C4%8Den%C3%AD](http://cs.wikipedia.org/wiki/Zp%C4%9Bnovazebn%C3%AD_u%C4%8Den%C3%AD) [http://en.wikipedia.org/wiki/Machine\\_learning](http://en.wikipedia.org/wiki/Machine_learning) [http://cs.wikipedia.org/wiki/U%C4%8Den%C3%AD\\_s\\_u%C4%8Ditelem](http://cs.wikipedia.org/wiki/U%C4%8Den%C3%AD_s_u%C4%8Ditelem) [http://en.wikipedia.org/wiki/List\\_of\\_machine\\_learning](http://en.wikipedia.org/wiki/List_of_machine_learning) [http://en.wikipedia.org/wiki/Reinforcement\\_learning](http://en.wikipedia.org/wiki/Reinforcement_learning) <http://tdunning.blogspot.cz/2012/10/references-for-on-line-algorithms.html> <http://tdunning.blogspot.cz/search?q=bandit> <http://camdp.com/blogs/multi-armed-bandits> <http://www.recsyswiki.com/wiki/Ensemble>

Algoritmus začne ve stavu nevědomí (ignorant state), kdy neví nic o daných okolnostech a začíná nabývat vědomostí tím, že testuje systém. Tím, jak vstřebává data a vyhodnocuje výsledky, učí se, jaké chování je nejlepší.

Z psychologického hlediska se jedná o následující úvahu: Jakým způsobem dokáže trest a odměna ovlivňovat naše budoucí chování? Na základě čeho se (nejen) lidstvo učí novým věcem?

#### 3.1.1 Online učení

Navrhovaná strategie je též nazývána jako *online učení*. Nutno zmínit, že slovem online zde není míněno něco ve smyslu internetu, ale ve smyslu neustále se vyvíjející aktualizace dat. Učící algoritmus v každém kole vykoná nějakou akci, přijme zpětnou vazbu a připíše si daný zisk či ztrátu.

Z matematického hlediska má online učení propojení na klasické online algoritmy, teorii (opakovaných) her a teorii pravděpodobnosti. Díky těmto znalostem tak můžeme navrhovat pravděpodobností dynamické systémy, kterými lze modelovat složitá průmyslová zařízení nebo třeba výherní automat známý jako mnohoruký bandita (Multi-Armed Bandit).

### 3.2 Multi-armed Bandit algoritmus

The Multi-Armed Bandit Problem je jedním z klasických problémů online učení. Tato herní strategie je podobná tradičnímu hernímu automatu, který představuje one-armed strategii, ale mnohoruká varianta má více herních pák *V každém kole si lze pro hru vybírat mezi  $N$  automaty.*

#### 3.2.1 Teoretický základ nutný k pochopení algoritmu

distribuční funkce a tak podobně

---

<sup>24</sup>učení bez učitele

<sup>25</sup>zpětnovazební učení nebo též učení posilováním

### 3.2.2 Princip algoritmu

Strategii si lze představit tak, že stojíme před  $N$  výherními automaty (též nazývány jako bandité) a v každém kole máme možnost vybrat si jeden z nich, na kterém budeme hrát. Pravděpodobnosti výher u jednotlivých automatů jsou pro nás neznámé.

Formálně lze strategii popsat jako skupinu výnosových (reward) distribučních funkcí  $B = R_1, \dots, R_K$ , kde  $K$  je počet banditů. Každý bandita má tedy přiřazenu jednu distribuční funkci, jež vyjadřuje pravděpodobnost úspěchu.

Zpočátku nemá hráč žádnou informaci o průběhu hry ani o rozložení pravděpodobnosti úspěchu napříč bandity. Tím, že v každém kole vybíráme vždy jen jednoho banditu, se snažíme navrhnout strategii pro maximalizaci celkové výhry.

Samozřejmě, že kdybychom znali banditu s největší pravděpodobností výhry, potom bychom jej vždy vybírali, čímž bychom maximalizovali výhry. Naším úkolem bude tedy nalézt nejlepšího banditu a to co nejrychleji, jak je to jen možné.

K návrhu strategie nám pomáhá to, že jednotlivé bandity nejdříve testujeme, abychom získali nutné znalosti. Poté už se lze zaměřovat na páky, které nám poskytují největší užitekované znalosti.

Úkol trochu komplikuje stochastická povaha banditů. Suboptimální bandita může vracet spoustu výher, což by nás mohlo přimět uvěřit, že právě tento bandita je velice štědrý. Podobně ale naopak nejlepší bandita může vracet spoustu proher. Měli bychom tedy pořád zkoušet i lůzry, nebo se na ně vykašlat?

Co je dalším problémem, tak pokud najdeme banditu, který vrací “docela dobré” výsledky, měli bychom toto zachovat a nadále rozvíjet naše “docela dobré” skóre, nebo bychom měli zkoušet i další bandity v naději, že nalezneme ještě lepšího? Tomuto se říká exploration vs. exploitation dilemma.

NEMOJE Necht  $u_1, \dots, u_K$  jsou průměrné hodnoty výnosových distribučních funkcí. Hráč aktivně zatáhne za páku každé kolo a sleduje příslušný výnos. jeho úkolem je maximalizovat součet těchto výnosů. Oželení (regret) po  $T$  kolech je definováno jako rozdíl mezi součtem výnosů příslušných výnosových funkcí optimální strategie a součtu získaných výnosů (VZOREC VIZ kybernář)

### 3.2.3 Exploration vs. Exploitation

- **Explorace** Nacházení nových oblastí hledání - náhodné procházky, nevyužívá předchozích znalostí
- **Exploatace** Využití stávajících znalostí, uvíznutí v lokálních extrémech, rigidita

### 3.3 Bayesian Bandits

Ukazuje se ale, že nalézt optimální řešení je neuvěřitelně obtížné. A může trvat léta, než je celkové řešení vyvinuto. Existuje totiž také spousta přibližně optimálních řešení, které jsou docela dobré.

Právě to jedno z mnoha takových řešení (které lze velice dobře škálovat) je známo jako Bayesian Bandits a bude se jednat o online algoritmus, o kterém už padla řeč v sekci ?? a má přímou souvislost s zpětnovazebním učením (reinforcement learning).

NEMOJE PRUBEH ALGORITMU Bayesovské řešení začíná stanovením pravděpodobností výhry pro každého banditu z předchozích zkušeností. V našem případě jsou tyto pravděpodobnosti od 0 do 1.

Prior a posterior by Verča "před provedením pokusu"- např. pravděpodobnost, že hodím sudý číslo na kostce je  $1/2$ , a "aposteriori" je po provedení pokusu - nedrív uděláš pokus - to se používá třeba u Bayesova vzorce. ale tý tabulce upřímně moc nerozumím... to prior má být asi jakože to, co předpokládáš, že bys měl mít a to posterior je, jak to jakože "odhadneš" třeba zdat - jakože pokusem

V každém kole: Vzorkování náhodné veličiny  $X_b$  Výběr bandity s největším vzorkem (např. vyber banditu  $B = \operatorname{argmax} X_b$ ) Pozorujme výsledek vybrání bandity  $B$  a updatujme priors (takový ty prostě věci z dřívějšího vyhodnocování - pokusy a výhry) Návrat na 1)

A to je vše. Z výpočetního hlediska algoritmus zahrnuje vzorkování z  $N$  distribucí. Počáteční priors jsou  $\text{Beta}(\alpha = 1, \beta = 1)$  (uniformní rozdělení) a pozorovaná náhodná veličina  $X$  (výhra či prohra, zakódována jako 1, respektive 0) je Binomialní, proto posterior (po provedení pokusu) je  $\text{Beta}(\alpha = 1 + X, \beta = 1 + 1 - X)$

Takže pokud bychom měli odpovědět na otázku z dřívějška (zda vybírat i ty lůzry), tak tento algoritmus navrhuje to, abychom nevyřazovali lůzry, ale měli bychom je vybírat s klesajícím tempem, jakmile nashromáždíme dost jistoty, že existují lepší bandité. Vyplývá to z podstaty, že vždy existuje nenulová šance, že lůzr dosáhne statusu  $B$ , ale pravděpodobnost této události se snižuje s tím, jak hrajeme více kol.

TODO možná nějaký obrázky (grafy) jak se to pulls vyvíjí (všechno je v článku na campdb)

Všimněme si, že se zase až tolik nestaráme o nějaké závěry nad skrytými pravděpodobnostmi, spíše se pro tento problém zabýváme o výběr nejlepšího bandity (nebo přesněji slovy, stále jistějšího během vybírání banditů)

To je třeba vidět v tom grafu níže, že distribuce červeného bandity je velice široká (což představuje neznalost o tom, jaká by skrytá pravděpodobnost vůbec mohla být), ale jsme si docela dobře jistí, že není nejlepší, algoritmus se tedy rozhodne tohoto banditu ignorovat.

ZPETNA VAZBA Multi-armed bandit je příklad Single-stage typu Reinforcement learningu (viz [https://cw.felk.cvut.cz/wiki/\\_media/courses/](https://cw.felk.cvut.cz/wiki/_media/courses/)

a3m33ui/prednasky/files/ui-2010-p11-reinforcement\_learning.pdf). Tedy snažíme se po jedné akci ihned uplatňovat feedback. Každá akce v multi-armed je nazývána jako jedna hra. Po každé hře  $a_t$  obdržíme (stochastický) reward  $r_t$ :  $E[r_t|a_t] = Q^*(a_t)$  a z dlouhodobého hlediska je právě cílem maximalizovat reward. “To solve the multi-armed bandit problem, one must explore a variety of actions and exploit the best of them”

$Q^*(a)$  je “action-value odhad - v čitateli součet rewardu z jednotlivých výběrů / počet výběrů

Pak je ještě “greedy akce” ve hře  $t$  -  $a_t^* = \operatorname{argmax}_a Q_t(a)$

Zajímavé poznámky V případě jakéhokoliv úspěchu doporučení by se měla navýšit někde hodnota pravděpodobnosti, se kterou bude algoritmus znovu vybrán. V případě neúspěchu pak by se tato pravděpodobnost měla exponenciálně snížit. V další iteraci už se bude systém rozhodovat s touto pravděpodobností mezi objevováním a zužitkováním. V případě zužitkování se vybírá z algoritmů, co již předtím něco vynesly. To je ta strategie learnable - u banditů to funguje tak, že poměr mezi objevováním a zužitkováním se mění v čase a ne v závislosti na předchozích výsledcích. Algoritmy jsou rozděleny ve svých dimenzích a každé té dimenzi je přiřazena určitá pravděpodobnost definující míru úspěšnosti nalezení toho, že tenhle algoritmus je fakt nejlepší. V případě, že uživatelé zužitkovávají tyto znalosti, tak si poté vybírají algoritmus, který má maximální pravděpodobnost z dané množiny. epsilon greedy!!!! - varianta epsilon decreasing Jak to má kybernář: Je-li pirát ve stavu objevování, tak v případě útoku na loď nalezne příslušný prostor, který obsahuje místo útoku, a zvýší mu hodnotu pravděpodobnosti. Když je ve stavu zužitkování znalostí, tak pluje do prostoru, kde hledá transportní loď. V případě, že za celý cyklus nenalezne žádnou loď, tak se exponenciálně sníží hodnota pravděpodobnosti daného prostoru. Nalezne-li loď, tak zvýšení pravděpodobnosti proběhne jako ve stavu objevování.

ZAVEREM Algoritmus je velice jednoduchý, proto je také jednoduché jej rozšířit. V mém případě bylo potřeba přidat learning rates - předpokládejme, že to prostředí se prostě může měnit v čase. Technicky by se tedy standardní Bayesian Bandit algoritmus self-updatoval tím, že by se učil tím, že to, co si myslel, že se zdá jako nejlepší, selhává nejčastěji. Také lze docílit toho, aby se algoritmus učil měnicím se prostředím rychleji. Potřebuje pro to jedinou věc - přidat rate napříč updatováním.

Pokud rate menší než 1, algoritmus bude zapomínat předchozí výsledky rychleji a tlak na neznalost bude směrem dolů. naopak rate větší než 1 implikuje to, že náš algoritmus se bude chovat více riskantně a bude vsázet na dřívější výhry častěji a bude tedy více odolný proti změnám měnicího se prostředí.

### 3.4 Nezbytné základy z teorie pravděpodobnosti

distribuční funkce, rozdělení pravděpodobnosti. kvanitily a tak, spojité náhodné veličiny, hustota, střední hodnota

---

# Realizace

Jednou z mnoha výzev pro někoho, kdo se snaží vybudovat doporučovací systém, je to, že je velice těžké dopředu říct, zda budou naše předpovědi dost přesné. Alespoň do té doby, dokud je nezačneme dělat a nebudeme pozorovat, jak často naši uživatelé přijímají naše návrhy. Je zde obrovský prostor možností (možných metod), z čeho vybírat.

Pro implementaci aplikace byly použity technologie uvedené níže.

## 4.1 Principy a technologie

### 4.1.1 Síťová komunikace a fronta zpráv

ZeroMQ

### 4.1.2 REST

### 4.1.3 Aplikační server

### 4.1.4 Databáze

[http://is.muni.cz/th/256499/fi\\_m/thesis\\_sikora.pdf](http://is.muni.cz/th/256499/fi_m/thesis_sikora.pdf)

### 4.1.5 Vyhledávací platforma

### 4.1.6 Doporučování

mahout

### 4.2 Popis implementace

### 4.3 Server

#### 4.3.1 Parametrizace

volba parametrů, ať už těch, co ovlivňují tresty nebo třeba úložiště (redis...)

### 4.4 Klient

### 4.5 Sada algoritmů

### 4.6 Návrh komunikace

router, dealer, worker thready

#### 4.6.1 Komunikační protokol

#### 4.6.2 Formát zasílaných zpráv



---

# Experimenty a vyhodnocení

kecy o testování

## 5.1 Testování různých způsobů chování

viz jak jarda vymyslel těch zhruba 5 příkladů, co mohou nastat

## 5.2 Experimenty

<http://contest.plista.com/wiki/example>

### 5.2.1 Vyhodnocovací technologie

Výpočty. kvůli kombinování budeme počítat s floaty

## 5.3 Zhodnocení aplikace

Slovní zhodnocení

## 5.4 Budoucí práce

bude li nějaká



---

## **Závěr**



---

## Literatura

- [1] BellKor, AT&T Labs, Inc. – Research. [online], stav ze dne 21.4.2012. Dostupné z: <http://www2.research.att.com/~volinsky/netflix/>
- [2] CNN Acquires Zite. [online], stav ze dne 21.4.2012. Dostupné z: <http://cnnpressroom.blogs.cnn.com/2011/08/30/cnnzite/>
- [3] easyrec: Recommendation Engine. [online], stav ze dne 24.4.2012. Dostupné z: <http://easyrec.org/recommendation-engine>
- [4] Fulltext. [online], stav ze dne 24.4.2012. Dostupné z: <http://www.adaptic.cz/znalosti/slovnicek/fulltext/>
- [5] How does the Amazon Recommendation feature work? [online], stav ze dne 21.4.2012. Dostupné z: <http://stackoverflow.com/questions/2323768/how-does-the-amazon-recommendation-feature-work>
- [6] Mendeley: Recommendation Systems for Academic Literature. [online], stav ze dne 21.4.2012. Dostupné z: <http://www.slideshare.net/KrisJack/mendeley-recommendation-systems-for-academic-literature>
- [7] Netflix Contest: 1 Million Dollars for Better Recommendations. [online], stav ze dne 21.4.2012. Dostupné z: <http://www.uie.com/brainsparks/2006/10/02/netflix-contest-1-million-dollars-for-better-recommendations/>
- [8] Netflix offers streaming movies to subscribers. [online], stav ze dne 21.4.2012. Dostupné z: <http://arstechnica.com/uncategorized/2007/01/8627/>
- [9] Proklik. [online], stav ze dne 21.4.2012. Dostupné z: <http://www.adaptic.cz/znalosti/slovnicek/proklik/>

- [10] What Is Affinity Analysis? [online], stav ze dne 21.4.2012. Dostupné z: <http://www.wisegeek.com/what-is-affinity-analysis.htm>
- [11] Zite is Flipping out. [online], stav ze dne 21.4.2012. Dostupné z: <http://blog.zite.com/2014/03/05/zite-is-flipping-out/>
- [12] Almazro, D.; Shahatah, G.; Albdulkarim, L.; aj.: A Survey Paper on Recommender Systems. [online], stav ze dne 24.4.2012. Dostupné z: <http://arxiv.org/pdf/1006.5278v4.pdf>
- [13] Anderson, C.: The Long Tail. [online], říjen 2004. Dostupné z: <http://archive.wired.com/wired/archive/12.10/tail.html>
- [14] Brodt, T.: Open Recommendation Platform. 2013, [Online; stav z 23. dubna 2014]. Dostupné z: <http://www.slideshare.net/d0nut/open-recommendation-platform>
- [15] Drachsler, H.: Recommender Systems and Learning Analytics in TEL. University Lecture, 2014, stav ze dne 24.4.2012. Dostupné z: <http://www.slideshare.net/Drachsler/rec-sys-mupplelecturekmi>
- [16] Jacobi, J.; Benson, E.; Linden, G.: Personalized recommendations of items represented within a database. Zář 26 2006, uS Patent 7,113,917. Dostupné z: <http://www.google.com/patents/US7113917>
- [17] John Gantz, D. R.: Extracting Value from Chaos. [online], červen 2011. Dostupné z: <http://www.emc.com/collateral/analyst-reports/idc-extracting-value-from-chaos-ar.pdf>
- [18] Linden, G.; Jacobi, J.; Benson, E.: Collaborative recommendations using item-to-item similarity mappings. Červenec 24 2001, uS Patent 6,266,649. Dostupné z: <https://www.google.com/patents/US6266649>
- [19] Liu, J.; Pedersen, E.; Dolan, P.: Personalized News Recommendation Based on Click Behavior. In *2010 International Conference on Intelligent User Interfaces*, 2010.
- [20] Vitvar, T.: Lecture 5: Application Server Services. University Lecture, 2014, stav ze dne 24.4.2012. Dostupné z: <http://humla.vitvar.com/slides/mdw/lecture5-1p.pdf>
- [21] Wikipedia: Long tail — Wikipedia, The Free Encyclopedia. 2014, [Online; stav z 19. dubna 2014]. Dostupné z: [http://en.wikipedia.org/w/index.php?title=Long\\_tail&oldid=603431399](http://en.wikipedia.org/w/index.php?title=Long_tail&oldid=603431399)

## Seznam použitých zkratk

**GUI** Graphical user interface

**XML** Extensible markup language





## Obsah přiloženého CD

	readme.txt.....	stručný popis obsahu CD
	exe .....	adresář se spustitelnou formou implementace
	src	
	impl.....	zdrojové kódy implementace
	thesis .....	zdrojová forma práce ve formátu L <sup>A</sup> T <sub>E</sub> X
	text .....	text práce
	thesis.pdf .....	text práce ve formátu PDF
	thesis.ps .....	text práce ve formátu PS