CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS
PROGRAMA DE PÓS-GRADUAÇÃO EM MODELAGEM MATEMÁTICA E COMPUTACIONAL

APRENDIZADO POR REFORÇO BASEADO EM AGRUPAMENTOS PARA RECOMENDAÇÃO NA AUSÊNCIA DE INFORMAÇÃO PRÉVIA

OTÁVIO AUGUSTO MALHEIROS RODRIGUES

BELO HORIZONTE
JULHO DE 2017

OTÁVIO AUGUSTO MALHEIROS RODRIGUES

APRENDIZADO POR REFORÇO BASEADO EM AGRUPAMENTOS PARA RECOMENDAÇÃO NA AUSÊNCIA DE INFORMAÇÃO PRÉVIA

Dissertação apresentada ao Programa de Pós-graduação em Modelagem Matemática e Computacional do Centro Federal de Educação Tecnológica de Minas Gerais, como requisito parcial para a obtenção do título de Mestre em Modelagem Matemática e Computacional.

Área de concentração: Modelagem Matemática e Computacional

Linha de pesquisa: Métodos Matemáticos Aplicados

Orientador: Anisio Mendes Lacerda
Coorientador: Flávio Luis Cardeal Pádua

Rodrigues, Otávio Augusto Malheiros
R696a Aprendizado por reforço baseado em agrupamentos para recomendação na ausência de informação prévia. / Otávio Augusto Malheiros Rodrigues. – – Belo Horizonte, 2017.
xiv, 49 f. : il.

Dissertação (mestrado) – Centro Federal de Educação Tecnológica de Minas Gerais, Programa de Pós-Graduação em Modelagem Matemática e Computacional, 2017.
Orientador: Prof. Dr. Anísio Mendes Lacerda
Coorientador: Prof. Dr. Flávio Luis Cardeal Pádua

Bibliografia

1. Aprendizado do Computador. 2. Análise por Agrupamento. 3.Recuperação da Informação I. Lacerda, Anísio Mendes. II. Centro Federal de Educação Tecnológica de Minas Gerais. III. Título

CDD 001.535

Elaboração da ficha catalográfica pela Biblioteca-Campus II / CEFET-MG

SERVIÇO PÚBLICO FEDERAL
MINISTÉRIO DA EDUCAÇÃO
CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS
COORDENAÇÃO DO PROGRAMA DE PÓS-GRADUAÇÃO EM MODELAGEM MATEMÁTICA E COMPUTACIONAL

"APRENDIZADO POR REFORÇO BASEADO EM AGRUPAMENTOS PARA RECOMENDAÇÃO NA AUSÊNCIA DE INFORMAÇÃO PRÉVIA"

Dissertação de Mestrado apresentada por **Otavio Augusto Malheiros Rodrigues**, em 2 de agosto de 2017, ao Programa de Pós-Graduação em Modelagem Matemática e Computacional do CEFET-MG, e aprovada pela banca examinadora constituída pelos professores:

Prof. Dr. Anísio Mendes Lacerda (Orientador)
Centro Federal de Educação Tecnológica de Minas Gerais

Prof. Dr. Flávio Luis Cardeal Pádua (Coorientador)
Centro Federal de Educação Tecnológica de Minas Gerais

Prof. Dr. Marcelo Peixoto Bax
Universidade Federal de Minas Gerais

Prof. Dr. Adriano César Machado Pereira
Universidade Federal de Minas Gerais

Visto e permitida a impressão,

Prof. Dr. José Geraldo Peixoto de Faria
Coordenador do Programa de Pós-Graduação em
Modelagem Matemática e Computacional

Dedico este trabalho aos meus pais Luiz Pedro e Miracy, com muito amor e gratidão, por tudo que fizeram por mim ao longo da minha vida. Espero ter sido merecedor do esforço dedicado por vocês em todos os aspectos desde meu nascimento e em especial quanto à minha formação.

Agradecimentos

Agradeço ao meu orientador Prof. Dr. Anísio Mendes Lacerda por ter me recebido e sempre me direcionado sobre o melhor caminho a ser seguido, de forma clara e única, além também da busca pelo perfeccionismo durante todo o desenvolvimento. Espero poder contribuir à ciência com a mesma ética e dedicação que me transmitiu.

Da mesma forma, agradeço ao Prof. Dr. Flávio Luiz Cardeal Pádua pelos direcionamentos dados em diversos momentos desde o início do meu programa de mestrado, além do auxílio na escolha de disciplinas extremamente relevantes para minha formação durante o programa.

Não posso deixar de agradecer também ao Prof. Me. Moisés Henrique Ramos Pereira e ao Prof. Me. Felipe Leandro Andrade da Conceição pelas informações passadas já no início da minha busca pelo programa de mestrado, além do auxílio em momentos importantes do estudo.

Agradeço à CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior) pela concessão da bolsa durante todo o período de realização deste mestrado.

Por fim, agradeço aos meus amigos e familiares, por compreenderem meus momentos de ausência, além também do estímulo constante, essencial para a motivação durante todo o período desta fase importante da minha vida.

“Conhecimento não é aquilo que você sabe, mas o que você faz com aquilo que você sabe.”
(Aldous Huxley)

Resumo

Atualmente, com a popularização e aumento da quantidade de páginas Web, o volume de informação compartilhada tem crescido, ampliando cada vez mais a quantidade de conteúdo disponível para os usuários dessa rede. Assim, surge a necessidade de ferramentas capazes de identificar conteúdo relevante a partir desse grande volume de informação disponível. Os sistemas de recomendação são ferramentas computacionais que possuem este objetivo, ou seja, esses sistemas focam em auxiliar os usuários a encontrar informação relacionada às suas preferências de forma personalizada. As técnicas estado-da-arte na literatura de sistemas de recomendação são baseadas no histórico de interação dos usuários com os itens disponíveis no sistema. Assim, estas técnicas são limitadas no cenário de ausência de informação prévia sobre as preferências dos usuários. Este é um problema bem conhecido na literatura de sistemas de recomendação, chamado cold-start de usuários e é o foco desta dissertação. Mais especificamente,aborda-se o problema de recomendar itens para usuários de forma contínua (i.e.,online) por meio de algoritmos baseados em aprendizado por reforço. A classe de algoritmos na qual baseia-se este trabalho é Multi-Armed Bandits (MAB) e refere-se a algoritmos de tomada de decisão sequencial com retro-alimentação. Neste trabalho, modelamos a recomendação no cenário de cold-start como um algoritmo MAB de dois passos. Assumimos que os itens podem ser agrupados considerando sua descrição textual, o que leva a agrupamentos de itens semanticamente semelhantes. Primeiramente, seleciona-se o agrupamento de itens mais relevante para o usuário-alvo da recomendação. A seguir, dado o agrupamento previamente selecionado, escolhe-se o item mais relevante dentro desse agrupamento, o qual é sugerido para o usuário-alvo. Para avaliar a qualidade do algoritmo proposto, o seu desempenho foi mensurado utilizando um conjunto de dados real. A avaliação experimental mostra que o algoritmo proposto produz melhorias significativas em termos de qualidade de recomendação em relação aos algoritmos MAB estado-da-arte.

Palavras-chave: Sistemas de Recomendação. Agrupamentos. Aprendizado por Reforço. Recomendação Sequencial.

Abstract

Nowadays, with the popularization and increase of the number of Web pages, the volume of shared information has grown, increasing more and more the amount of content available to users on this web. Hence, there is an increasing need for tools capable of filtering relevant content from the large volume of information. Recommender systems are computational tools that have this goal, i.e., these systems focus on helping users to find information related to their preferences in a personalized way. State-of-the-art techniques in the literature of recommender systems are based on the history of user interaction with the items available in the system. Hence, these techniques are limited in the scenario of lack of prior information about the preferences for users. This is a well-known problem in the literature of recommender systems, called *users cold-start*, which is the focus of this dissertation. More specifically, the problem of recommending items for users on an sequencial basis (i.e., online) is addressed by means of algorithm based on reinforcement learning paradigm. The class of algorithms on which this work is based is Multi-Armed Bandits (MAB) and refers to the class of sequential decision-making algorithms with feedback. In this work, we model the recommendation in the *cold-start* scenario as a MAB algorithm with two steps. We assume that items can be grouped considering their textual description, which leads to clusters of semantically similar items. First, the most relevant cluster of items is selected for the target user of the recommendation. Then, given the cluster previously selected, the most relevant item is chosen within that cluster, and it is suggested for the target user. To evaluate the quality of the proposed algorithm, we measured its performance using a real-world data set. The experimental evaluation shows that the proposed algorithm produces significant improvements about the recommendation quality in terms of accuracy when compared to state-of-the-art MAB algorithms.

Keywords: Recommender Systems. Clustering. Reinforcement Learning. Online Recommendation.

Lista de Figuras

Lista de Tabelas

Lista de Algoritmos

Lista de Abreviaturas e Siglas

CTR	*Click Through-Rate* (Taxa Através de Cliques)

Lista de Símbolos

ϵ	Letra Grega Epsilon

Sumário

1 Introdução

O objetivo dos sistemas de recomendação é reconhecer itens relevantes de acordo com as preferências do usuário. Estes sistemas se baseiam nas preferências explícitas e implícitas dos usuários, nas preferências de outros usuários e nos atributos do usuário e do item. Um sistema de recomendação de filmes pode incorporar tanto dados de avaliação explícita (por exemplo, a usuária Alice avaliou o filme *Toy Story* com uma nota 4 em 5), bem como informação de conteúdo dos filmes (por exemplo, *Forrest Gump* pertence ao segmento comédia) para sugerir filmes que combinem com os gostos dos usuários.

Existe uma incerteza inerente às informações de preferência do usuário, que podem ser coletadas através da interação online dos usuários com o sistema de recomendação. Além disso, um número considerável de usuários e itens podem ser completamente novos ao sistema de recomendação, ou seja, não há informações suficientes sobre suas preferências históricas. Este cenário é conhecido como problema de *cold-start* (SCHEIN et al., 2002) e representa um grande desafio para os sistemas de recomendação.Em uma configuração de *cold-start*, as abordagens de recomendação tradicionais sofrem para aprender uma função de correspondência para gostos dos usuários e popularidade dos itens. Nesta pesquisa tratamos especificamente do problema de *cold-start* de usuários.

O cenário de *cold-start* de usuários representa um desafio para os sistemas de recomendação e são reconhecidos como um problema de prospecção/exploração (*exploration/exploitation*). Esse problema refere-se a encontrar uma compensação entre dois objetivos concorrentes: (I) prospectar incertezas dos gostos dos usuários (AGARWAL et al., 2009), enquanto (II) maximiza a satisfação dos usuários a longo prazo. Por exemplo, um sistema de recomendação de filmes deve encontrar o filme mais relevante para os usuários enquanto continua tentando melhorar a compreensão de suas preferências.

Tradicionalmente, o dilema da prospecção/exploração é formulado como um problema *multi-armed bandits* (LI et al., 2010; TANG et al., 2013; VERMOREL; MOHRI, 2005). No contexto desta pesquisa, ou seja, sistemas de recomendação, formula-se o problema como: Para cada rodada de recomendação, o algoritmo seleciona um item (por exemplo, um filme) para recomendar, e após isso ele recebe uma recompensa. Portanto, a recomendação personalizada pode ser vista como uma instanciação do problema de *multi-armed bandits*. Observe que a recompensa é extraída de uma distribuição com probabilidade desconhecida determinada pelo item selecionado e refere-se à resposta do usuário (por exemplo, um clique). A recompensa é retornada ao algoritmo *bandit* e usada para otimizar sua estratégia de recomendação. A curto prazo a estratégia ideal será puxar o braço(*arm*) com a maior recompensa esperada em relação ao usuário em cada rodada e assim maximizar a

recompensa cumulativa total para o conjunto de rodadas.

Nesta pesquisa, para capturar a dependência entre os itens, é proposto um arcabouço através de um algoritmo *bandit* de dois níveis baseado em agrupamentos. Mais especificamente, primeiro, consideram-se os braços como grupos de itens e usa-se um algoritmo *bandit* para selecionar um grupo de itens a serem recomendados. Assim, dado um agrupamento selecionado, usa-se um algoritmo *bandit* distinto para selecionar o item mais relevante neste agrupamento. A estrutura proposta é flexível para considerar, como entrada, (i) um método de agrupamento e (ii) dois algoritmos *bandits* (ou seja, um para cada nível), o que leva a vários algoritmos como instâncias do arcabouço.

1.1 Justificativa

A justificativa para este trabalho apoia-se no grande número de informações em vídeos disponibilizadas diariamente pela rede, além de usuários novos que consomem esse conteúdo. Não há histórico prévio de avaliações ou gostos desses usuários para realizar as recomendações. A pesquisa visa maximizar a satisfação dos usuários, a longo prazo, além de explorar as incertezas dos gostos dos utilizadores (AGARWAL et al., 2009). Por exemplo, um sistema de recomendação de filmes deve encontrar o filme mais relevante para os usuários enquanto continuam tentando melhorar a compreensão de suas preferências.

1.2 Motivação

Sistemas de Recomendação são usados atualmente por várias aplicações que visam atender melhor as preferências dos usuários. Nota-se uma forte migração da Era da Informação, para a Era da Recomendação e melhorar a recomendação visa encontrar informação relevante em meio ao conteúdo disponibilizado na Internet, de forma a mostrar ao usuário aquilo que ele realmente tem afinidade e interesse. Assim, a motivação deste trabalho é melhorar a relevância dos itens para os usuários, enquanto aprende-se mais sobre os seus gostos.

1.3 Objetivos: Geral e Específicos

O objetivo geral deste trabalho consiste em desenvolver um Sistema de Recomendação, baseado em técnicas de Aprendizado por Reforço, para apoiar em tempo real as escolhas dos melhores vídeos para usuários novos, que não possuem nenhuma informação histórica disponível, os quais possam fornecer avaliações de relevância para se aprender mais sobre os gostos destes usuários e melhorar continuamente as recomendações feitas.

A intuição na qual se baseia o método proposto nesta dissertação é que vídeos parecidos podem ser agrupados para melhorar a qualidade da recomendação. Vídeos de um mesmo agrupamento podem agradar o usuário-alvo da recomendação e ao invés de escolher dentre todo o universo de vídeos, busca-se um processo de tomada de decisão em dois níveis. Por fim, espera-se que vídeos com conteúdo parecido possuem taxas de acesso parecidas e podem ser agrupados de acordo com as características que os descrevem. Para confirmar essa intuição, uma série de objetivos específicos devem ser considerados. Abaixo se encontra uma lista desses objetivos:

- Organizar cada vídeo baseando em contextos presentes para identificação do mesmo, com posterior agrupamento dos itens semelhantes;
- Implementar as soluções do *Multi-Armed Bandits* mais conhecidas com o objetivo de sugerir itens relevantes para os novos usuários e aprender mais sobre eles.
- Propor um novo algoritmo para lidar com o *cold-start* de usuários, que é baseado em um processo de decisão em dois níveis de recomendação dos itens agrupados;
- Realizar experimentos extensivos em um conjunto de dados de filmes do mundo real para validar a precisão da família de algoritmos proposta.

1.4 Organização do trabalho

Esta pesquisa está organizada da seguinte forma: Na Seção 3, são apresentados os trabalhos anteriores relacionados e que contribuíram nos rumos tomados nessa pesquisa. Na Seção 2, são detalhados os fundamentos teóricos e os algoritmos propostos para a recomendação em cenários de *cold-start* de usuários. Na Seção 4, são exibidos os métodos e a configuração experimental utilizada pela pesquisa para a criação do arcabouço de dois níveis baseado em *multi-armed bandits*. Na Seção 5, são listados e discutidos os resultados obtidos, mostrando ganhos significativos através do arcabouço proposto quando comparado ao método *baseline* de recomendação, considerando dois níveis de recomendação. Finalmente, na Seção 6, são feitas as conclusões, deixando claro que com os métodos propostos foi possível recomendar vídeos à usuários aos quais não havia nenhuma informação prévia do seu histórico de preferências e são apresentadas possíveis linhas futuras de desenvolvimento.

2 Fundamentação Teórica

Neste capítulo são apresentados os conceitos necessários para o entendimento do trabalho. Na Seção 2.1, apresenta-se os tipos de sistemas de recomendação. A modelagem de sistemas de recomendação *online* por meio de estratégias de aprendizado por reforço é detalhada na Seção 2.2. Por fim, na Seção 2.3, detalha-se o uso de algoritmos do tipo *Multi-armed Bandits* para recomendação de itens para novos usuários.

2.1 Sistemas de Recomendação

Sistemas de Recomendação são uma sub-área de Aprendizado de Máquina (*Machine Learning*) e sua finalidade é sugerir itens a usuários, com base em suas de preferências. A principal motivação que levou ao desenvolvimento dos sistemas de recomendação foi o grande volume de conteúdo disponível (RESNICK; VARIAN, 1997). Atualmente, os sistemas de recomendação têm sido utilizados numa variedade de aplicações, dentre as quais podemos citar, sugestões para filmes (CHOI; KO; HAN, 2012), programas de TV (BARRAGÁNS-MARTÍNEZ et al., 2010), músicas (AIZENBERG; KOREN; SOMEKH, 2012), notícias de interesse (MORALES; GIONIS; LUCCHESE, 2012) e produtos (SCHAFER; KONSTAN; RIEDL, 1999).

Mais formalmente, o problema de recomendação de itens pode ser definido da seguinte forma: seja $U = \{u_1, u_2, \ldots, u_m\}$ e $I = \{i_1, i_2, \ldots, i_n\}$ o conjunto de todos os usuários e itens, respectivamente. Cada usuário u é associado a um subconjunto $I_u \subseteq I$ de itens avaliados por u. De forma similar, cada item i é associado a um subconjunto $U_i \subseteq U$ de usuários que avaliaram o item i. As avaliações dos usuários são armazenadas em uma matriz V, na qual cada elemento v_{ui} representa o valor da avaliação dado pelo usuário u ao item i. Assim, o sistema de recomendação pode ser interpretado como uma função $\delta : U \times I \rightarrow \Re$ que mapeia pares de usuários/itens para números reais que indicam a relevância do item para o usuário. Quanto maior o valor predito de relevância, maior a probabilidade que o usuário irá gostar do item.

As preferências dos usuários podem ser extraídas de forma implícita ou explícita. Na forma implícita, as informações são obtidas por meio de ações do usuário ao interagir com o sistema de recomendação. Por exemplo, ao acessar uma página Web, se pode inferir que o usuário demonstrou interesse por essa página. Outro exemplo de informação de preferência de forma implícita são as compras feitas pelo usuário em um sítio Web. Por outro lado, quando o usuário indica de forma clara suas preferências temos a forma explícita de representação de preferências. Por exemplo, quando o usuário indica o número numa faixa de 0 a 5 (onde 1 é ruim e 5 é ótimo) tem-se a representação explícita do gosto do

usuário. Nesse trabalho estamos interessados em representação implícita de preferências.

Os algoritmos de recomendação podem ser classificados em personalizados e não-personalizados. O mais comum e simples algoritmo não personalizado é o *Most Popular*, que recomenda os itens mais populares para qualquer usuário, sem considerar seus gostos pessoais. Tipicamente, esse algoritmo serve como método de referência para a maior parte dos algoritmos personalizados mais sofisticados.

O passo mais importante para recomendar um item à um usuário é definir seus interesses. O interesse do usuário pode ser representado de diversas maneiras, dependendo da natureza da informação utilizada para gerar as recomendações (ADOMAVICIUS; KWON, 2009). Assim, sistemas de recomendação podem ser classificados em três tipos principais: (i) sistemas baseados em filtragem colaborativa, (ii) sistemas baseados em conteúdo, (iii) sistemas híbridos. A seguir, detalhamos cada um dos desses tipos de sistemas.

2.1.1 Abordagem Baseada em Conteúdo

Em sistemas de recomendação baseados em conteúdo(CB), o usuário recebe recomendações de itens semelhantes aos que ele preferiu no passado. Em particular, os métodos baseados em conteúdo estimam a utilidade $u(a, i)$ de um item i para o usuário-alvo a da recomendação utilizando as utilidades $u(a, i_k)$ dadas por a para itens $i_k \in I$ que são similares ao item i. No contexto de filmes, um sistema de recomendação baseado em conteúdo pode utilizar a sinopse dos filmes para descrevê-los e assim encontrar filmes similares. Uma das vantagens desta abordagem é que o conteúdo dos itens pode ser utilizado para inferir preferências de usuários novos no sistema. O foco deste trabalho são sistemas de recomendação baseados em conteúdo.

2.1.2 Abordagem Baseada em Filtragem Colaborativa

A intuição dos sistemas de recomendação baseados em informação colaborativa é que usuários com preferências similares no passado tendem a ter preferências similares no futuro (ADOMAVICIUS; TUZHILIN, 2005a). Normalmente, as técnicas de filtragem colaborativa atuam combinando as preferências dos usuários alvo com as preferências de usuários semelhantes, no caso, os vizinhos mais próximos, para então produzir recomendações de itens que ainda não foram vistos pelo usuário-alvo (ADOMAVICIUS; TUZHILIN, 2005b). Nota-se, neste caso, que não há necessidade de informações dos perfis dos usuários, como por exemplo, localização, idade, dentre outras, assim como não necessita-se do conteúdo dos itens, como descrição, preço, etc, tornando a filtragem colaborativa uma estratégia independente. Essa abordagem representa o estado-da-arte em sistemas de recomendação (LINDEN; SMITH; YORK, 2003). Porém, um problema particular com o qual as abordagens colaborativas é que elas são incapazes de lidar com a chegada de novos usuários. Pelo fato

dessa abordagem utilizar algoritmos que contam com as preferências do usuário para fazer sugestões, o sistema não terá capacidade de recomendar um item para um novo usuário até que ele avaliado um número substancial de itens.

2.1.3 Abordagem Híbrida

Uma variedade de sistemas de recomendação utilizam uma abordagem, i.e., são sistemas que utilizam tanto informação de conteúdo dos itens quanto informação colaborativa. A intuição é que a combinação de diferentes formas de descrição dos itens e dos usuários permitem lidar com problemas específicos das técnicas mencionadas acima. Assim, por exemplo, técnicas baseadas em conteúdo tendem a recomendar itens muito similares aos itens para os quais os usuários já demonstraram preferência. Por outro lado, técnicas baseadas em informação colaborativa são incapazes de recomendar itens para usuários que nunca demonstraram preferência por nenhum item no passado(BURKE, 2002). Enfim, as técnicas híbridas tendem a possuir os pontos fortes e a lidar com deficiências das abordagens anteriores.

2.2 Aprendizado por Reforço

Conforme discutido acima, os sistemas de recomendação possuem como objetivo auxiliar usuários a encontrar informação relevante de forma personalizada. Porém, as técnicas do estado-da-arte (i.e., técnicas baseadas em informação colaborativa) assumem que existe informação prévia sobre as preferências dos usuários. Esse não é o cenário dos sistemas em que novos usuários surgem e para os quais há necessidade de sugerir itens.

O processo de recomendação geralmente é modelado como uma tarefa sequencial (i.e., *online*) de tomada de decisão. Assim, a cada instante de tempo t o algoritmo de recomendação precisa decidir qual item deve ser recomendado para o usuário-alvo da recomendação. Para cada item recomendado, o algoritmo recebe uma recompensa, que pode ser positiva ou negativa. Esta modelagem é a base do Aprendizado por Reforço (*Reinforcement Learning*) (SUTTON; BARTO, 1998). O contra-ponto do aprendizado por reforço é o aprendizado em *batch*, para o qual todos os dados estão disponíveis. Obviamente, dado que temos ausência de informação prévia dos usuários, a abordagem em *batch* não pode ser utilizada quando novos usuários surgem no sistema.

O paradigma de aprendizado por reforço foi introduzido em (ROBBINS, 1952) e trata do aprendizado sequencial para tomada de decisão em cenários de *feedback* limitado. Um aspecto chave que diferencia aprendizado por reforço de outras técnicas de aprendizado é que no primeiro caso tem-se um processo de tentativa e erro (SUTTON; BARTO, 1998).

Na Figura 1 é apresentado a visão geral do processo de aprendizado por reforço. Nessa classe de algoritmos, tem-se um *agente* que interage com um ambiente desconhecido

ao longo de uma sequência de instantes de tempo, esse agente observa o *estado* atual do ambiente para escolher a melhor *ação* a ser realizada e recebe uma *recompensa* para cada ação escolhida. Cabe ressaltar que o objetivo do agente é aprender sobre o ambiente fazendo escolhas que maximizem as recompensas recebidas ao longo das interações.

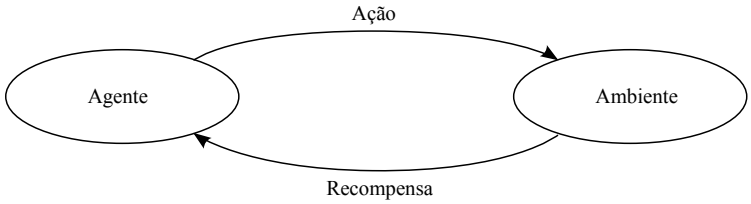

Figura 1 – Paradigma de Aprendizado por Reforço. Nesse paradigma, o agente interage com o ambiente por meio de uma ação, para a qual recebe uma recompensa. O objetivo é aprender sobre o ambiente maximizando as recompensas recebidos ao longo do tempo.

O aprendizado por reforço é o paradigma adotado em diversos domínios, e muitas vezes, esse é o único paradigma aplicável a determinados problemas. Por exemplo, a construção de um algoritmo que aprenda a jogar xadrez é inviável em um aprendizado baseado em *batch*. Lembrando que no aprendizado em batch, ao contrário do aprendizado por reforço, toda a informação precisa estar disponível a priori.

Assim, o aprendizado em batch para construção de um programa que aprenda a jogar xadrez é inviável por dois motivos: (i) primeiro, o custo de ter uma base de dados com todas as possibilidades de movimentos para aprendizado do melhor movimento no jogo é proibitivo e (ii) segundo, em muitas situações do jogo de xadrez, não existe a noção de melhor movimento, em outras palavras, o quão bom é um movimento depende dos movimentos subsequentes.

Nesta dissertação, o foco são os algoritmos do tipo *Multi-armed Bandits* (ROBBINS, 1952), que são um caso específico de algoritmos do paradigma de aprendizado por reforço. Na próxima Seção detalha-se esses algoritmos.

2.3 Algoritmos *Multi-armed Bandits*

Os algoritmos *Multi-armed Bandits* possuem origem no paradigma de aprendizagem por reforço. O problema MAB refere-se à um processo sequencial de decisão de um agente que tenta otimizar suas ações enquanto melhora seu conhecimento sobre as opções (*arms*). O desafio central é a necessidade de equilibrar prospecção (*exploration*) e exploração (*exploitation*). Interativamente, o algoritmo $\mathcal{A}$ *explora* o conhecimento prévio do ambiente para escolher a opção que parece ser a melhor no momento. Porém, a opção escolhida aparentemente como ótima pode ser uma escolha sub-ótima, devido a imprecisões no conhecimento do ambiente pelo algoritmo $\mathcal{A}$. Para evitar essa situação indesejada, o algoritmo $\mathcal{A}$ tem que *prospectar* selecionando opções aparentemente sub-ótimas para

coletar mais informações sobre elas. A etapa de prospecção permitirá ter um entendimento melhor do ambiente e, assim, definir a opção realmente ótima.

O problema acima pode ser descrito através de um exemplo abstrato: um jogador em um cassino está diante de um conjunto de N máquinas caça-níqueis, chamadas de *bandits*, sendo que cada *bandit* possui um braço *(arm)* que pode ser puxado. Quando puxado, qualquer braço vai dar uma recompensa. Porém essas recompensas podem variar entre as máquinas. Por exemplo, um determinado braço pode retornar moedas de recompensa em 1% do tempo, enquanto um outro braço pode retornar moedas de recompensa em 5% do tempo. Qualquer puxão específico de qualquer braço específico é dado como um risco. Além disso, o jogador não conhece a recompensa de cada máquina *a priori*. Logo, o jogador deve descobrir experimentalmente a taxa de recompensa das máquinas obrigatoriamente jogando em cada uma delas. Até então, o que há é um problema estatístico em que é necessário lidar com o risco para descobrir qual máquina tem a maior recompensa média. O que diferencia um problema do tipo MAB é que se recebe uma pequena quantidade de informação sobre as recompensas de cada máquina. Especificamente, o algoritmo aprende algo somente sobre a escolha feita e não sobre outras escolhas que poderiam ter sido feitas (WHITE, 2012).

No cenário de recomendação, o conjunto de máquinas caça-níquel são os itens, ou seja, são as opções possíveis e para as quais deseja-se determinar o percentual de sucesso. O ato de “puxar” o braço de uma máquina equivale a escolher um item a ser recomendado para o usuário. Por fim, o equivalente ao retorno de moedas da máquina é o usuário considerar a recomendação relevante.

Formalmente, um cenário de recomendação utilizando MABs é descrito como: seja $|I|$ o número de itens que o algoritmo deve escolher. A cada instante de tempo t, o recomendador seleciona um item $i(t)$ e uma recompensa $r(t)$ é retornada. As recompensas são selecionadas a partir de um distribuição de probabilidade desconhecida para cada item. O processo é sequencial e o objetivo é maximizar a soma de recompensas. No contexto dessa dissertação tem-se um recomendador de filmes, i.e., os itens sugeridos para os usuários são filmes.

Um algoritmo MAB é uma estratégia que determina a sequência de itens (i.e., *arms*) escolhida a cada instante de tempo que retorna o valor máximo de recompensa nesse instante de tempo. Assim, uma vez que a cada instante de tempo procura-se maximizar a recompensa, espera-se que o algoritmo maximize a soma das recompensas ao longo do tempo.

Os algoritmos do tipo MAB buscam pelo melhor compromisso entre aprender sobre os itens (*exploration*) e usar o conhecimento atual para selecionar o melhor item (*exploitation*). Com o objetivo de maximizar a recompensa acumulativa no longo prazo, o desafio fundamental é a necessidade de balancear entre prospecção e exploração (i.e., *the*

Exploration/Exploitation Dilemma).

Tradicionalmente, o foco de algoritmos MAB são as classes de algoritmos livres de contexto (*context-free*), que modelam situações nas quais não existe informação de contexto e a recompensa depende somente do item escolhido. Em contraste, os algoritmos *Contextual Multi-armed Bandits* levam em consideração o contexto da recomendação (i.e., informação do usuário) para selecionar o item a ser recomendado. O foco deste trabalho são algoritmos livres do contexto, dado que não assume-se nenhuma informação prévia das preferências dos usuários. Foram comparados seis algoritmos do tipo MAB: ε-greedy, UCB1 e UCB2, Bayes UCB, Thompson Sampling e Random Choice. A seguir, detalha-se cada um deles.

2.3.1 ε-greedy

No algoritmo ε-greedy (SUTTON, 1996) o melhor item é considerado como sendo a opção com a maior probabilidade de recompensa. Abaixo apresenta-se um diagrama que expressa esse comportamento.

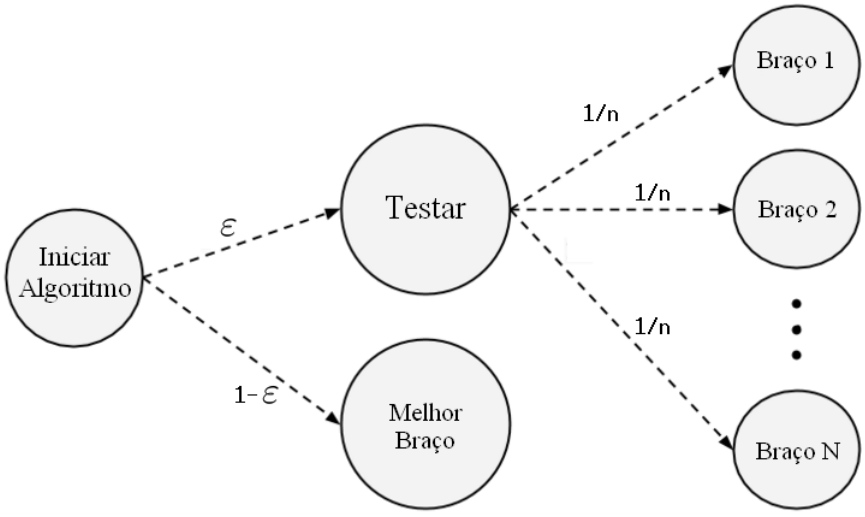

Figura 2 – Funcionamento básico do algoritmo Epsilon Greedy

A execução do algoritmo consiste em balancear entre prospecção e exploração por meio do parâmetro ε, ou seja, esse balanceamento é estocástico. Assim, em uma fração proporcional a ε, o algoritmo escolhe um dentre todos os itens disponíveis de forma aleatória. Essa fase refere-se à prospecção. Por outro lado, a fase de exploração ocorre proporcionalmente à uma fração dada por $1 - \epsilon$. Por exemplo, quando tem-se $\varepsilon = 0.7$, significa que em 70% das escolhas o algoritmo irá fazer prospecção, ou seja, irá tentar aprender sobre suas escolhas. Enquanto em 30% das escolhas, o algoritmo irá explorar, ou seja, escolherá a melhor opção conhecida. Conforme mostra a Figura 2 o balanceamento entre prospecção/exploração pode ser resumido da seguinte maneira:

- Com probabilidade ε, o algoritmo ε-greedy prospecta uma opção aleatória entre todas as disponíveis.
- Com probabilidade *1* - ε, o algoritmo ε-greedy explora a melhor opção conhecida.

O algoritmo ε-greedy trabalha entre a visão da experimentação puramente aleatória e a maximização das recompensas baseada na melhor opção conhecida. Ele é um dos algoritmos MAB mais conhecidos e largamente utilizados pois busca balancear prospecção e exploração de forma explícita.

De forma geral, o algoritmo ε-greedy possui boa acurácia em uma variedade de aplicações. Uma desvantagem desse algoritmo é que ao prospectar ele escolhe igualmente entre os itens disponíveis. Em outras palavras, não existe preferência explícita por nenhum item e as opções *pior possível* e *próximo do ótimo* são igualmente prospectadas. Assim, outros algoritmos MAB foram propostos, os quais focam em garantias de convergência ótimas. A seguir, discutimos algoritmos com esse foco.

2.3.2 Classe de algoritmos UCB (*Upper Confidence Bound*)

A política que examinamos é chamada UCB1, e pode ser resumida pelo princípio do otimismo diante da incerteza. Ou seja, apesar da nossa falta de conhecimento sobre quais ações são melhores, vamos construir um palpite otimista sobre a boa recompensa esperada de cada ação e escolher a ação com o melhor palpite. Se o nosso palpite for errado, nosso palpite otimista diminuirá rapidamente e seremos obrigados a mudar para uma ação diferente. Mas se escolhermos bem, seremos capazes de explorar essa ação e nos arrependeremos muito. Desta forma, equilibramos exploração e exploração. Existem vários algoritmos pertencentes à classe UCB, dos quais iremos discutir os seguintes: (i) UCB1, (ii) UCB2 e (iii) BayesUCB.

2.3.2.1 UCB1

O algoritmo epsilon-greedy apresentado anteriormente possui uma fraqueza: ele não mantém o controle do quanto ele sabe sobre os itens disponíveis. Assim, ele foca no quanto de recompensa foi obtida para cada um dos itens. Assim, o algoritmo ε-greedy tende a explorar de forma intensiva itens para os quais não existe confiança na estimação de sua recompensa.

O algoritmo UCB aqui apresentado não usa aleatoriedade em tudo. Em vez disso, o UCB evita ser crédulo, exigindo manter o controle da confiança nas avaliações dos valores estimados de todas os braços(*arms*). Para fazer isso, é necessário ter algumas métricas de quanto sabe-se sobre cada braço.

Além de manter explicitamente o controle de confiança nos valores estimados de cada braço, o algoritmo UCB1 é especial por dois outros motivos:

- O UCB não usa aleatoriedade em tudo. Ao contrário do epsilon-Greedy, é possível saber exatamente como o UCB vai se comportar em qualquer situação. Isso pode torná-lo mais fácil para raciocinar às vezes.
- O UCB não possui parâmetros livres que você precise configurar antes de poder executá-lo. Esta é uma grande melhoria, se houver interesse em executá-lo em um ambiente de difícil aprendizado, porque isso significa que pode-se começar a usar UCB sem ter uma noção clara do que se espera do mundo e como se comportar.

Tomadas em conjunto, o uso de uma medida de confiança explícita, a ausência de aleatoriedade desnecessária e a ausência de parâmetros configuráveis faz do UCB um algoritmo muito eficaz. Além disso, o algoritmo é de fácil implementação.

Devemos a princípio entender o porquê é importante manter o controle de nossa confiança nos valores dos braços. A razão tem a ver com a natureza das recompensas que recebemos de braços: elas são ruidosas. Se forem usadas as experiências passadas com um braço, em seguida, o valor estimado de qualquer braço é sempre uma estimativa ruidosa do verdadeiro retorno sobre o investimento que pode-se esperar dele. Por causa desse ruído, pode ser apenas uma coincidência que Braço A pareça melhor do Braço B; se tivesse mais experiência com os dois braços, eventualmente seria percebido que Braço B é realmente melhor. O UCB não usa aleatoriedade em tudo. Em vez disso, o algoritmo evita ser crédulo, exigindo manter o controle da confiança nas avaliações dos valores estimados de todas as armas. Para fazer isso, é necessário ter algumas métricas do quanto se sabe sobre cada braço (WHITE, 2012). O algoritmo busca uma forma de assegurar que não há uma totalidade de ausência das informações prévias do ambiente antes de começar a aplicar a sua regra de decisão baseada em confiança. É importante manter esta etapa de inicialização em mente quando se considera a implementação do UCB1: se você só vai deixar o algoritmo em execução para um pequeno número de itens (digamos, um número M de itens) e há muitos braços para explorar (digamos, um número N de braços), é possível que o UCB1 vá apenas tentar cada braço em sucessão e nem mesmo o fará até o fim. Se M < N, isto é definitivamente o que vai ocorrer. Se M está perto de N, ainda será gasto muito tempo apenas fazendo este passo a passo inicial. Se isso é uma coisa boa ou ruim é algo que precisa ser considerado antes de usar UCB.

2.3.2.2 UCB2

O algoritmo UCB2 pode ser considerado uma versão um pouco mais complicada do UCB1, com melhores constantes associadas aos limites de confiança. Trabalha com um parâmetro alpha entre 0 e 1, onde quanto menor o valor, maior será o nível de preferência que é dado para explorar ou prospectar as opções disponíveis e diversificando a recomendação definida com o intuito de encontrar itens adequados para recomendar e, em seguida, se estabelecendo em explorar a melhor opção dentre os itens encontrados até

agora de acordo com a avaliação dos usuários (AUER; CESA-BIANCHI; FISCHER, 2002). Em outras palavras, esse algoritmo tende a fazer mais prospecções ao longo das iterações para só depois de um longo aprendizado, passar a explorar as melhores opções com o que aprendeu através do tempo.

Nesta pesquisa, foram usados valores entre 0.0 (que é quando o algoritmo tende a prospectar mais sobre opções desconhecidas) e 1.0 (que é quando o algoritmo tende a explorar mais a melhor opção conhecida) para o parâmetro $alpha$. A mesma ideia é usada no algoritmo Epsilon Greedy 2.3.1 para valores de $beta$.

2.3.2.3 Bayes-UCB

O algoritmo Bayes-UCB (KAUFMANN; CAPPÉ; GARIVIER, 2012) assume um método de modelagem *Bayesiana* para o *multi-armed bandits*. A tomada de decisão entre prospectar e explorar baseia-se na função quantil, que são pontos estabelecidos em intervalos regulares a partir da função de distribuição acumulada da distribuição Beta. Deve-se modelar as ações com distribuições Beta e considerar que uma distribuição inicial para as ações é dada da mesma forma que a variável de parâmetro que visa afinar a sua resposta.

A função quantil é utilizada para criação de intervalos de confiança sobre a probabilidade de sucesso de cada uma das ações, para com isso escolher a ação que possui a maior taxa de confiança de ser a melhor ação entre todas as outras em cada instante de tempo.

2.3.3 Thompson Sampling

Da mesma forma que o Bayes-UCB, o Thompson Sampling (THOMPSON, 1933) também utiliza uma formulação baeysiana para o problema do MAB. Assim como o algoritmo anterior, ele escolhe as ações de acordo com a probabilidade de a ação escolhida ser a melhor. O algoritmo possui um conhecimento a priori das opções disponíveis e conforme passam as interações com o ambiente, ele atualiza esse conhecimento visando maximizar os ganhos (AGRAWAL; GOYAL, 2012). A decisão da ação a ser tomada é feita por amostragem de cada uma das distribuições que modelam as opções disponíveis, escolhendo assim a ação que possuir o maior valor de amostragem.

A distribuição Beta possui dois parâmetros, que são α e β, podendo ser entendidos como simples contagens dos valores de sucesso ou falha, ou em outras palavras, a contagem do número de vezes em que uma ação retornou 1 ou 0.

2.3.4 Random Choice

Este algoritmo não possui parâmetros de execução, assim como também não possui critérios para fazer as seleções dos itens. Ele não é capaz de verificar o ambiente para

saber qual é a melhor opção conhecida ou quais são as opções até então sub-ótimas. Para o Random Choice, não há distinção entre os itens presentes no sistema e com isso, o algoritmo basicamente possui a mesma probabilidade de recomendação para todos os itens disponíveis, procurando não ser criterioso na hora de carregar um novo braço a cada iteração.

Por conta de ser um algoritmo que apenas seleciona os itens sem nenhum tipo de critério, bem diferente dos algoritmos citados anteriormente que equilibram prospecção e exploração, o Random Choice serve como *baseline* dentre os algoritmos MAB utilizados.

2.4 Latent Dirichlet Allocation - LDA

A alocação latente de Dirichlet (LDA, Latent Dirichlet Allocation, em inglês), é um modelo probabilístico gerador de tópicos. Ela baseia-se na distribuição multinominal para realizar as contagens de palavras e uma distribuição Dirichlet, que é uma generalização da distribuição beta, para a realização da estrutura subjacente. A idéia básica é que os descritores textuais dos vídeos(i.e. título, categoria e sinopse) são representados como misturas aleatórias sobre temas latentes, em que cada tópico é caracterizado por uma distribuição sobre palavras (BLEI; NG; JORDAN, 2003).

Ela utiliza uma abordagem bayesiana para aprender a estrutura latente de temas que compõem cada um dos vídeos. É baseado em um modelo generativo levando em consideração que cada um dos documentos em particular na coleção, é uma mistura de temas. Por exemplo, uma determinada descrição de um vídeo de notícias sobre esportes poderia ser composto por uma mistura de 50% sobre o tema Marta, 30% sobre o tema Phelps e 20% sobre o Bolt, enquanto outro vídeo poderá ser composto por 40% sobre as Olimpíadas, 20% sobre Complexo Olímpico, 30% sobre Rio de Janeiro e 10% sobre o COI.

Basicamente, o algoritmo seleciona uma certa distribuição sobre os temas distintos que a descrição do vídeo contenha. Assim, algumas descrições darão mais peso ao tópico 1, enquanto outras darão mais peso ao tópico 2. Após isso, para cada palavra existente em cada um dos tópicos no vídeo, seu tema é escolhido. Depois de selecionar o tema, a própria palavra é escolhida de acordo com o dicionário de distribuição de probabilidade de seu tema atribuído.

A característica mais marcante deste modelo é o mínimo de intervenção humana requerida para sua aplicação. O LDA tem capacidade de descobrir temas subjacentes nos vídeos selecionados e assim estabelecer links entre vídeos com os quais não se possui nenhuma informação anterior sobre estes temas. Além disso, os vídeos não são obrigados a serem rotulados com tópicos ou palavras-chave que antecedam a montagem LDA. O método é não supervisionado e com o mínimo de dados de entrada prévia por parte do usuário (BLEI, 2012).

3 Trabalhos Relacionados

Neste capítulo apresentam-se os trabalhos relacionados a esta dissertação. Especificamente, o foco deste trabalho são sistemas de recomendação *online*, nos quais existe a inserção de novos usuários ao longo do tempo. Este problema é conhecido como *cold-start*. Além disso, o cenário de interesse desta dissertação são sistemas de recomendação de vídeos. Assim, iremos dividir a revisão da literatura em duas partes: (i) recomendação para novos usuários e (ii) recomendação de vídeos. Nas próximas seções, detalham-se cada um dos dois temas relacionados a este trabalho.

3.1 Recomendação para novos usuários

O problema de recomendação para novos usuários, consiste em recomendar itens para usuários que não interagiram com o sistema, ou seja, o sistema deve recomendar itens para usuários sem conhecer suas preferências. Diversas abordagens foram propostas para tratar o problema de novos usuários. Tradicionalmente, a modelagem das preferências dos novos usuários é obtida por meio de entrevistas durante as quais são apresentados vários itens aos usuários e estes, por sua vez, os classificam de acordo com sua relevância (GOLBANDI; KOREN; LEMPEL, 2010). Esta estratégia é conhecida como conjuntos semente (i.e., *seed sets*).

Porém, a estratégia de conjuntos semente apresenta desafios. O primeiro é que os conjuntos são computados *a priori* e não mudam de acordo com o usuário. Outro desafio é que muitas vezes os usuários não estão dispostos a classificar um número suficiente de itens no momento inicial de uso do sistema de recomendação. Assim, foram propostas abordagens que modelam o sistema de recomendação como um sistema de aprendizado *online* (KOHRS; MERIALDO, 1999).

Com isso, a estratégia de elicitação das preferência de usuário precisa garantir que o mesmo (i) não abandone um longo processo de inscrição em um sistema e (ii) não perca o interesse em retornar ao site, devido à baixa qualidade das recomendações iniciais. Tem-se desenvolvido, então, gradualmente um conjunto de estratégias teóricas de informação para o problema dos novos usuários. As propostas envolvem estruturas de simulação offline e avaliações das estratégias através de amplas simulações de uma experiência online com usuários reais em um sistema de recomendação (RASHID; KARYPIS; RIEDL, 2008). Essa é a ideia que se aplica em nossa pesquisa, pois o processo de recomendação sequencial simula um ambiente online através de experimentos offline.

Alguns trabalhos utilizam do aprendizado por reforço buscando soluções para o

problema da recomendação para novos usuários, como no caso de (BOBADILLA et al., 2012) em que os autores usam a otimização baseada no aprendizado neural. Os testes foram realizados nas bases de dados do Netflix e do Movielens, obtendo melhorias importantes nas medidas de acurácia, precisão e revocação quando aplicadas ao problema de *cold-start* para novos usuários. Já em (TANG; WU; CHEN, 2017) as recomendações propostas são baseadas em Redes Neurais Recorrentes (RNN) para fazer recomendações tomando em consideração apenas o comportamento dos usuários em um período de tempo, levando em consideração que são usuários novos no sistema. Esse trabalho mostra como a ideia de como o aprendizado por reforço pode ser aplicado, da mesma forma que em nossa pesquisa, porém através de abordagens diferentes.

Em outro desses trabalhos, (HERNANDO et al., 2017) mostra como usuários não registrados podem ser considerados como um caso particular do problema de *cold-start* de novos usuários. Como os usuários não registrados de um determinado serviço não criaram uma conta de perfil nem avaliaram nenhum item, os sistemas de recomendação não podem conhecer os gostos dos usuários não registrados e normalmente fornecem a eles recomendações baseadas na média de cada item. No entanto, os usuários não registrados são uma proporção importante do nicho de muitos sistemas de recomendação. Portanto, são desejadas formas mais avançadas para recomendar a esses usuários não registrados. Os autores propõem oferecer um modelo de inferência natural baseado em regras de incerteza que lhes permita inferir suas próprias recomendações.

Nesta dissertação, considera-se o cenário de recomendação *online*, no qual não se conhece as preferências dos usuários e a recomendação deve ser computada à medida que novos usuários surgem no sistema. Diversos trabalhos na literatura modelaram o problema de recomendação *online* como um problema do tipo *Multi-armed Bandit* (RASHID et al., 2002), porém nenhum ainda na literatura utilizou da estratégia de dois níveis com experimentos extensivos através de uma base de dados do mundo real, como proposto nessa pesquisa.

Tradicionalmente, na literatura, o problema para quando não há informação prévia de um novo usuário é tratado através do *multi-armed bandits*. Várias abordagens para a solução do problema são apresentadas em (WHITE, 2012), visando comparar os vários algoritmos que compõem a literatura do problema e dando uma visão aprofundada sobre o balanceamento de prospecção e exploração, que são tratados pelos nomes *exploration* e *exploitation* respectivamente. O uso dessa abordagem para tratamento do problema de novos usuários pode também ser visto em (FELICIO et al., 2017; PAIXÃO et al., 2017), onde os autores buscam novos modelos para recomendações nesse contexto.

No contexto da recomendação para dispositivos móveis, os autores em (BOUNEFFOUF; BOUZEGHOUB; GANÇARSKI, 2012) propõem um algoritmo que leva em conta a localização geográfica dos usuários, desta vez levando em conta o problema da evolução

do conteúdo dos usuários ao longo do tempo em que ele interage com o sistema. Em (PAZZANI, 1999), os autores propuseram no modelo de filtragem colaborativa o uso de *filterbots*, que são agentes cujas notas são diretamente baseadas no conteúdo e automaticamente injetam avaliações no sistema recomendador para reduzir a dispersão de dados. Assim, essa abordagem simples insere avaliações ”falsas” no sistema para melhorar a qualidade de recomendação em um ambiente *cold-start*.

Em (ZHOU, 2015), os autores detalham como as recompensas dadas podem ser diferentes de acordo com o contexto. Disserta-se sobre a situação em que é observado o contexto da informação em cada momento *t* nos algoritmos de *multi-armed bandits*. O *arm* (braço em tradução livre) que tem a maior recompensa esperada pode estar vinculado a diferentes contextos. Esta variante do *multi-armed bandits* é chamada de *bandits* contextuais. Geralmente em um problema de *bandits* contextuais há um conjunto de políticas (i.e, regras) e cada regra dentro do contexto de um *arm*. Por exemplo, em um sistema de recomendação para notícias personalizadas, pode-se tratar cada página de notícia como um *arm*, e as características de ambos, artigos e usuários, como contextos. O sistema, em seguida, seleciona notícias para cada usuário de acordo com o contexto e assim maximiza a taxa de cliques nas páginas ou o tempo de visualização das mesmas. Essa é a estratégia utilizada nessa pesquisa para avaliar o método de recomendação.

Sobre trabalhos que fazem recomendações utilizando a mineração de dados, os autores em (CHEN et al., 2016) notam que descrever os itens através de *tags* proporciona uma grande oportunidade para melhorar o desempenho da recomendação. Os autores propuseram um método de *tags* e avaliação com base em filtragem colaborativa para o modelo de recomendação de itens, que primeiro utiliza modelagem de tópicos para explorar a informação semântica das *tags* para cada usuário e para cada item, respectivamente.

Além disso, em (YOU et al., 2013), é proposto um método de agrupamento conforme os interesses dos usuários, interesses esses extraídos através das redes sociais e da avaliação dos filmes para resolver o problema de *cold-start*. O método consiste no agrupamento de usuários que são o destino da recomendação juntamente com os outros usuários, combinando o interesse do gênero de usuários e as informações de avaliação. É importante perceber a enorme quantidade de interessante e informações úteis dos usuários, onde são extraídas informações através de avaliações baseadas em curtidas. Além disso, eles usam o *Internet Movie Database* (IMDb)[1] como o principal conjunto de dados. A base de dados online do IMDb consiste em uma grande quantidade de informações relacionadas à filmes e programas de TV, incluindo atores. Este conjunto de dados, não só é usado para fornecer informações dos filmes, mas também como recurso para fornecer informações do gênero que é extraído. Em nossa pesquisa, essa base de dados foi utilizada conjuntamente com a base de dados do Movielens1M para agregar mais conteúdo textual para extração no

[1] Internet Movie Database - IMDb: http://www.imdb.com/

momento de criar os agrupamentos.

Em outro dos trabalhos relacionados (LI; GENTILE; KARATZOGLOU, 2016), é investigada uma técnica de agrupamento dependente do contexto para sistemas de recomendação com base em estratégias de prospecção (*exploration*) e exploração (*exploitation*) através do *Multi-armed bandits* através de vários usuários. O algoritmo verifica dinamicamente os usuários nos grupos com base na sua semelhança comportamental observado durante uma sequência de atividades. Ao fazê-lo, o algoritmo reage ao usuário para moldar agrupamentos em torno dele, mas, ao mesmo tempo, explorando a geração de agrupamentos sobre os utilizadores que não estão até então envolvidos. Outros trabalhos também, desenvolvem técnicas que visam melhorar a implementação dos agrupamentos especialmente para sistemas de recomendação visando melhorar qualidade e precisão (KHALID et al., 2017), enquanto há trabalhos que verificam a política de recomendação de acordo com as características dos agrupamentos criados (PANDEY; CHAKRABARTI; AGARWAL, 2007).

Além dos trabalhos acima, cita-se também a pesquisa apresentada em (LI et al., 2010), na qual os autores propõem um método para melhorar as recomendações personalizadas no contexto do *bandit problem*, baseado em princípios onde um algoritmo de aprendizagem sequencial seleciona notícias para recomendar aos usuários com base em informações de avaliação implícita. A proposta desta pesquisa se difere, pois visa fazer recomendações personalizadas de notícias, usando tanto de avaliação explícita, que são o número de avaliações através dos cliques que as notícias contêm, bem como usa também as informações sobre o item, pré computadas no momento em que são carregadas no sistema, para aumentar as características que descrevem aquele item e com isso geram uma forma de aprender sobre os gostos do novo usuário, além, de também gerar uma recomendação mais eficaz.

Por fim, em relação à avaliação dos métodos propostos, em (MARY; PREUX; NICOL, 2014) buscam-se métodos para avaliação offline de sistemas de recomendação online. Depois de destacar as limitações dos métodos existentes, eles apresentam um novo método, baseado em técnicas de *bootstrapping*, que é muito mais preciso e proporciona uma medida de qualidade da sua estimativa. O último é uma propriedade altamente desejável, a fim de minimizar os riscos inerentes ao colocar sistemas de recomendação no ar pela primeira vez. Eles fornecem metodologia teórica e experimental que provam a sua superioridade em relação aos métodos estado da arte, bem como uma análise da convergência das medidas de qualidade. Nesse trabalho utilizou-se da mesma abordagem para avaliar os métodos de recomendação em dois níveis.

3.2 Recomendação de vídeos

Recomendação de vídeos vem sendo pesquisada a muito tempo e podemos destacar dentre os estudos a interface criada para o serviço de recomendação de filmes do MovieLens[2] para auxiliar seus usuários (MILLER et al., 2003b). Da mesma forma, em (DAVIDSON et al., 2010) se discute o sistema de recomendação de vídeos do YouTube, que é a mais popular comunidade de vídeos online do mundo [3], mostrando os desafios para recomendar conjuntos personalizados de vídeos para usuários tendo como base a sua atividade no site.

Os conjuntos de dados do MovieLens são amplamente utilizados na educação, pesquisa e indústria. Eles são baixados por centenas de milhares de vezes por ano, refletindo a sua utilização em livros populares de programação, cursos tradicionais e online, além do uso em softwares. Esses conjuntos de dados são um produto das atividades dos membros no sistema *MovieLens Movie Recommendation*, uma plataforma de pesquisa ativa que hospedou muitos experimentos desde o seu lançamento em 1997. Sua importância é tão grande que alguns pesquisadores documentam as melhores práticas e limitações do uso do MovieLens em novas pesquisas (HARPER; KONSTAN, 2016). Por conta disso, essa pesquisa opta por utilizar essa base de dados, conjuntamente com a base do IMDb.

Em geral este problema pode ser resolvido usando Sistemas de Recomendação baseados em conteúdo porque eles permitem que o usuário encontre novos conteúdos que não tenham sido avaliados antes, uma vez que o sistema deve ser capaz de combinar as características de um item, além dos relevantes recursos no perfil do usuário. Da mesma forma, eles permitem que o novo usuário possa encontrar conteúdos interessantes (ASABERE, 2012). Além disso, o contexto é baseado em sistemas de Recomendação Online, como visto em (ZHOU et al., 2015), tornando a avaliação dos métodos um trabalho que exige extremo cuidado de interpretação. Inclusive há pesquisas, como em (DAVIES; LEWIS, 2016) que estudam algoritmos apenas para recomendação de vídeos, nesse caso em especial, o algoritmo monitora vídeos que um usuário assistiu e identifica um grupo de outros usuários que também assistiram ao mesmo gênero de vídeo ou similar para então gerar recomendações mais interessantes baseada em interesses comuns.

Frisando que a recomendação de vídeo tornou-se uma parte essencial dos serviços de vídeo online, em (YANG et al., 2016) os autores propõem um algoritmo baseado em grupos sociais com interesses em comum para produzir recomendações de vídeos personalizados ao avaliar os vídeos candidatos, a partir dos grupos aos quais um usuário está afiliado. Para verificar a eficácia do método, os autores utilizam a taxa de cliques como métrica relacionada ao ganho cumulativo das recomendações. Em outra pesquisa, (ZHOU et al., 2017) propõe um método para recomendar vídeos baseado em dividir os usuários

[2] MovieLens, site de recomendações personalizadas de filmes: https://movielens.org/
[3] Estatísticas do Youtube: https://www.youtube.com/yt/press/pt-BR/statistics.html

em vários grupos, enquanto verificam os cliques em vídeos feitos pelos novos usuários para relacioná-los aos grupos ao mesmo tempo em que fazem recomendações para cada um deles. Isso permite uma exploração melhor das comunidades de compartilhamento de mídia.

Além disso, dentre os trabalhos relacionados à essa linha de pesquisa, cita-se (YANG et al., 2007), o qual propõe um sistema de recomendação de vídeos online que tenha capacidade de recomendar uma lista de vídeos relevantes de acordo com o vídeo atual visualizado pelo usuário sem informações do seu perfil. A pesquisa também mostra como a avaliação de relevância é aproveitada para automaticamente ajustar os intra-pesos dentro de cada gênero e inter-pesos entre os gêneros com base nos dados de cliques dos usuários. Em seguida, busca-se explorar a variância de relevância entre diferentes gêneros. Solucionar estes problemas representa um grande impacto nas visualizações de conteúdo pelos usuários (ZHOU; KHEMMARAT; GAO, 2010), além da inovação e do alinhamento com as propostas de negócio, como ocorre no Netflix[4] (GOMEZ-URIBE; HUNT, 2016).

Na pesquisa (ZHOU et al., 2016), os autores propõem um sistema de recomendação para vídeos baseado na aprendizagem online distribuída em nuvem. A ideia é recomendar vídeos de acordo com o contexto do usuário, ao mesmo tempo que adaptam a estratégia de seleção de vídeo com base na avaliação do usuário, baseada em clique, para maximizar o total de cliques do usuário, ou seja, a recompensa dada pela avaliação de relevância.

Em outros casos, como em (SYED-ABDUL et al., 2013) a recomendação de vídeos tem sido usada como uma forma de barrar conteúdos enganosos que podem prejudicar a saúde dos usuários, com isso mostrando que sistemas de recomendação podem além de recomendar vídeos conforme os gostos de seus usuários, identificar possíveis conteúdos que poderiam prejudicar ou induzir à práticas enganosas por parte dos usuários que consomem esse conteúdo. Esses estudos ajudam a desenvolver algoritmos que irão detectar e filtrar automaticamente esses vídeos antes que eles se tornem populares.

Existem trabalhos, como (CONCEIÇÃO et al., 2016), que propõem uma abordagem que explora a informação visual e textual dos vídeos para descrevê-los adequadamente no sistema de recomendação, buscando tratar o problema de *cold-start*. A idéia do trabalho é que dado um documento de vídeo, que geralmente consiste de uma sequência de imagens, áudio e informações relacionadas – como o título, *tags* (rótulos) e descrição –, a recomendação de vídeos visa encontrar uma lista dos vídeos mais relevantes para direcionar os usuários. Ainda no problema de *cold-start* para recomendação de vídeos, (YI et al., 2016) propõe uma abordagem visando otimizar a medida de similaridade do filme, calculando as semelhanças entre diretores e atores. Todo esse processo para testar a utilidade do método é realizado no conjunto de dados do MovieLens-1M.

[4]Netflix, provedora global de filmes e séries de televisão via streaming: https://www.netflix.com/

Dentre outros métodos para recomendação, pode-se citar (ZHANG et al., 2017), onde os autores procuram descobrir indicadores de relevância implícitos que podem indicar o interesse dos usuários de vídeos com base no gênero. Esses indicadores de interesse implícitos são amplamente compostos pelos movimentos de cursor, pelas atividades de rolamento da tela bem como a rapidez na movimentação do mouse, visando contornar o problema da falta de avaliação explícita e informações prévias para realizar recomendações.

Alguns trabalhos buscam também adaptar esses sistemas recomendadores de vídeos para contextos mais avançados, atualizando as técnicas já conhecidas. É o caso de (RAIGOZA; KARANDE, 2017) que estudam o sistema de recomendação em um ambiente baseado na nuvem proposto para produzir uma lista de filmes recomendados com base nas informações do perfil de um usuário, através do conjunto de dados do Movielens, para assim melhorar o desempenho do sistema levando em conta também um cenário de *cold-start*.

Os vídeos utilizados nesta pesquisa, são organizados de duas formas: Na primeira são categorizados através de tópicos e posteriormente, agrupados para utilização em um dos métodos desta pesquisa, enquanto na segunda forma é aplicado um algoritmo popular de agrupamento. Ambos os métodos são vistos em (LI; YANG; JIANG, 2017), onde os autores buscam melhor forma de organizar os documentos através dos agrupamentos. Realizadas estas etapas previamente, pode-se então utilizar a estratégia de recomendação para escolher os agrupamentos e então selecionar um vídeo, como é feito em (KIM; AHN, 2008) que busca encontrar os agrupamentos mais relevantes e mostrar como usar esse método pode melhorar o desempenho dos sistemas de recomendação.

4 Recomendação Sequencial Baseada em Agrupamentos

No capítulo anterior foram descritos os aspectos mais relevantes para o entendimento desse trabalho e dos trabalhos relacionados. Nesse capítulo é descrita a estratégia de recomendação sequencial proposta nesta dissertação. O restante desse capítulo está organizado da seguinte forma: na Seção 4.1 é definido o problema de pesquisa foco desse trabalho; na Seção 4.2 é detalhada a modelagem do problema de recomendação sequencial por meio de técnicas de aprendizado por reforço (especificamente, algoritmos do tipo *Multi-armed Bandits*); na Seção 4.3 apresenta-se a estratégia de recomendação que desconsidera agrupamentos; por fim, na Seção 4.4 apresenta-se o algoritmo de recomendação proposto.

4.1 Formulação do problema

O foco deste trabalho é o cenário no qual existe total escassez de dados sobre os usuários do sistema de recomendação, ou seja, todos os usuários do sistema são novos. Essa escassez de dados leva a um alto grau de incerteza na recomendação e representa um dos principais desafios de pesquisa na área de recomendação, o chamado problema de *cold-start* de usuários. Assim, quando aplicados à cenários reais, sistemas de recomendação são dinâmicos, i.e., novos usuários surgem no sistema ao longo do tempo e esses devem ser considerados no momento da recomendação.

O problema de apresentar recomendações para novos usuários é comumente conhecido como problema de *cold-start*. Por contraste, o problema de *warm-start* refere-se à situação em que tem-se dados históricos de usuários para realizar a recomendação. Assim, os sistemas de recomendação precisam ser capazes de entender as preferências dos usuários tão rápido quanto possível. O dilema consiste em encontrar o melhor compromisso entre: (i) maximizar a satisfação dos usuários no longo prazo (*exploitation*) e, ao mesmo tempo, (ii) explorar a incerteza das preferências dos usuários (*exploration*).

Os métodos mais populares para tratamento desse dilema são baseados em técnicas de Aprendizado por Reforço (*Reinforcement Learning*). Mais especificamente, adota-se a modelagem da recomendação como um problema do tipo *Multi-armed Bandit* (MAB). Cabe ressaltar que a recompensa será positiva se o usuário gostar da recomendação e, negativa, caso contrário. Além disso, o algoritmo de recomendação terá acesso somente à recompensa relativa ao item recomendado, i.e., o algoritmo não terá acesso à recompensa dos itens não recomendados. O objetivo do algoritmo é maximizar as recompensas, i.e., as

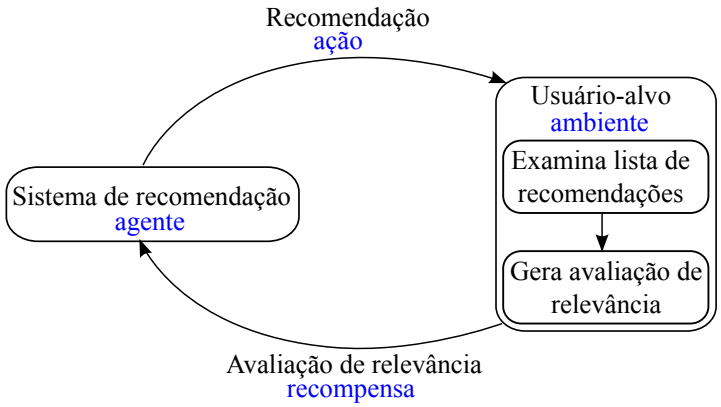

Figura 3 – A tarefa de recomendação online modelada como um problema do tipo *Multi-armed bandit*.

recomendações relevantes para os usuários.

Para trabalhar em cima desse problema há uma intuição por trás dos vídeos que é: vídeos com conteúdos parecidos, possuem padrões de acesso similares. Dessa forma, pode-se dizer que de acordo com uma categoria que descreve o vídeo, este possui um determinado padrão de acesso e essas categorias, definem diferentes padrões. Com isso podemos organizar os vídeos através de semelhança semântica, diminuindo o escopo da escolha dos algoritmos MAB.

4.2 Recomendação Sequencial baseada em *MAB*s

Tradicionalmente, sistemas de recomendação, tais como os baseados em conteúdo, baseados em filtragem colaborativa e os híbridos, sugerem itens relevantes para os usuários baseando-se em suas preferências, conforme já descrito anteriormente. Entretanto, aplicações reais que utilizam sistemas de recomendação precisam tratar o problema de usuários para os quais não existe informação de preferência.

Na Figura 3 é apresentado o ciclo de interação entre um usuário-alvo de recomendações e um sistema de recomendação online. O usuário surge e o sistema de recomendação seleciona um item relevante, que é então apresentado ao usuário. O usuário interage com a recomendação clicando no item sugerido se esse item é relevante para ele. A interação é interpretada como uma avaliação de relevância sobre a qualidade do item recomendado. Por isso, esta avaliação é usada para atualizar o modelo de recomendação com o objetivo de oferecer melhores recomendações no futuro. O ciclo termina e o próximo usuário é considerado. Esta formulação se traduz no problema de Aprendizado por Reforço (veja a terminologia da área de aprendizado por reforço em cor azul na Figura 3).

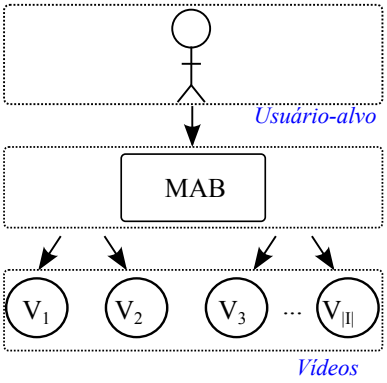

Figura 4 – Modelo sem agrupamento

4.3 Recomendação Sequencial sem Agrupamento de Itens

Antes de apresentar o algoritmo proposto nesta dissertação, detalha-se a abordagem tradicional de recomendação sequencial que desconsidera agrupamentos de itens. A abordagem de recomendação que utiliza MABs e não considera agrupamentos é o estado-da-arte em recomendação sequencial em cenários de escassez de informação sobre os usuários.

Na Figura 4, detalha-se a recomendação considerando que o algoritmo MAB precisa escolher um dos vídeos disponíveis. Assim, o problema de tomada de decisão refere-se à escolha do vídeo mais relevante, ou seja, os *arms*, neste caso, são os vídeos. Para cada vídeo recomendado, o algoritmo irá receber uma recompensa que dependerá do usuário ter gostado do vídeo ou não.

No Algoritmo 1 é apresentado o processo de recomendação sequencial baseado em um algoritmo do tipo MAB conforme as etapas descritas acima. O algoritmo MAB itera em instantes de tempo $t = 1, 2, \ldots$ (Linha 1). Em cada instante t, o algoritmo seleciona um item $i(t) \in I$ com base no conhecimento que coletou das rodadas anteriores (Linha 2), apresenta o item selecionado para o usuário alvo (Linha 3). Portanto, a recompensa $r(t)$ [1] é revelada (Linha 4-8). Finalmente, o algoritmo melhora sua política de seleção e itens com a nova observação $i(t), r(t)$ (Linha 9). Observe que o algoritmo escolhe um único item para cada instante de tempo.

4.4 Algoritmo *Multi-armed Bandit* de Dois Níveis - *(TL-Bandits)*

A suposição básica da modelagem tradicional de recomendação que utiliza MABs é que os *arms* (ou itens) são independentes entre si (PANDEY; CHAKRABARTI; AGARWAL, 2007). Nesta dissertação, considera-se que os itens a serem escolhidos não são independentes. Em outras palavras, o trabalho proposto consiste em explorar similaridade semântica entre os itens a serem recomendados.

[1] Usa-se $r(t)$ como a recompensa retornada para o item $i(t)$ no tempo t

Algoritmo 1: Recomendação sequencial baseada em MABs

Input: *Algoritmo MAB* $\mathcal{A}$
for $t = 1, \ldots, T$ **do**
  $i(t) \leftarrow$ seleciona-item$(i(1, \ldots, t-1), \mathcal{A})$
  mostra $i(t)$ ao usuário; registra acesso
  if *usuário acessou* $i(t)$ **then**
    $r(t) = 1$
  end
  else
    $r(t) = 0$
  end
end
atualiza-mab$(i(t), r(t), \mathcal{A})$

Em um sistema de recomendação de filmes, por exemplo, cada filme pertence a uma ou mais categorias (e.g., drama, comédia, entre outras). Assim, fica explícito que filmes de uma mesma categoria possuem similaridade de gênero. Outro exemplo de vídeos com conteúdo similar são aqueles que possuem sinopse parecida. Dessa forma nota-se que pode-se usar vários tipos de descrição dos vídeos para agrupá-los por semelhança. Podemos destacar, além de gênero e sinopse, o título e as *tags*.

É importante frisar sempre, que a intuição na qual se baseia o método proposto nesta dissertação é que vídeos parecidos podem ser agrupados para melhorar a qualidade da recomendação. Vídeos de um mesmo agrupamento podem agradar o usuário-alvo da recomendação e ao invés de escolher dentre todo o universo de vídeos, busca-se um processo de tomada de decisão em dois níveis. Por fim, espera-se que vídeos com conteúdo parecido possuem taxas de acesso parecidas e podem ser agrupados de acordo com as características que os descrevem.

No primeiro nível busca-se escolher o agrupamento de vídeos que seja mais relevante para o usuário. Note que nesta etapa, existe um algoritmo do tipo MAB que irá escolher o agrupamento mais relevante. Assim, diferentemente da modelagem tradicional, o número de escolhas é muito inferior, pois o número de agrupamentos é muito inferior ao número de vídeos da base de dados.

No segundo nível, outro algoritmo do tipo MAB precisa escolher o vídeo a ser recomendado, dado que o algoritmo MAB do nível anterior já selecionou o agrupamento. Em outras palavras, dada a escolha do agrupamento, tem-se de escolher um vídeo que faça parte desse agrupamento.

De forma abstrata as próximas Figuras visam mostrar como é o funcionamento dessa abordagem de recomendação em um site de vídeos. Suponha que um usuário esteja assistindo um vídeo muito popular sobre esportes, conforme a Figura 5. O sistema de

Recomendação deve agora fazer a recomendação de um novo vídeo, e na abordagem TL-Bandits, ele deve primeiro escolher um agrupamento antes de escolher o vídeo.

Figura 5 – Vídeo que está sendo assistido no momento pelo usuário. Fonte:(youtube.com)

Suponha que esses agrupamentos sejam definidos pela categoria a qual o vídeo pertence, conforme a Figura 6. Dentro da sua função de recomendação, o algoritmo MAB pode escolher qualquer um dos agrupamentos existentes, mas ele resolve escolher a categoria do mesmo vídeo que o usuário está assistindo no momento, ou seja, esportes.

Dessa forma, agora o segundo nível de MAB precisa selecionar um vídeo dentro da categoria esportes para recomendar ao usuário. Essa recomendação vai aparecer ao lado do vídeo que o usuário assiste no momento, que é o vídeo da Figura 5. Através da seleção, o algoritmo exibe o vídeo que pode ser relevante ao usuário, como na Figura 7 e aguarda para verificar se o usuário vai clicar nesse vídeo ou não, para assim gerar a recompensa e repassá-la aos dois níveis do TL-Bandits para gerar a próxima recomendação.

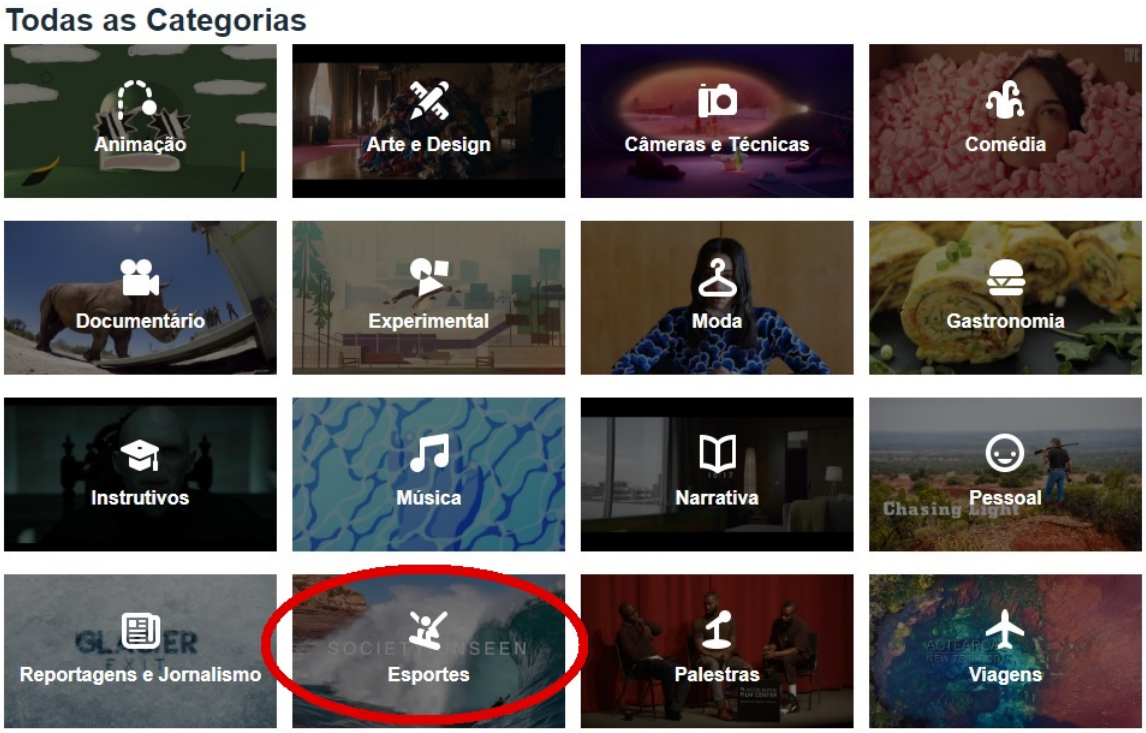

Figura 6 – Categorias de vídeos. Fonte:(vimeo.com/categories)

Figura 7 – Vídeo recomendado pelo TL-Bandits. Fonte:(youtube.com)

Cabe ressaltar que o agrupamento dos vídeos é feito de forma *offline* ou *a priori*, ou seja, as informações dos vídeos são processadas antes de iniciar-se o processo de recomendação. Essa abordagem faz com que o custo de agrupamento dos vídeos não seja determinante no custo da recomendação.

Na Figura 8 é apresentado o esquema geral do algoritmo *bandit* de dois níveis proposto nesta dissertação. Como se pode ver, a seleção do item sugerido para o usuário-alvo é realizada em duas etapas. Primeiro, é necessário selecionar o agrupamento que seja o mais provável a conter um vídeo relevante para o usuário. A seguir, dado o agrupamento selecionado, o próximo passo é a seleção do vídeo dentro do agrupamento escolhido. Em ambos os níveis, existe um algoritmo MAB responsável por selecionar um agrupamento (nível 1) e um vídeo (nível 2), sendo que estes algoritmos são independentes. A recompensa depende da escolha feita pelos MABs.

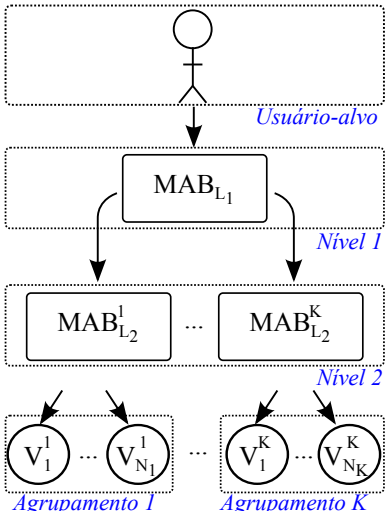

Figura 8 – Visão geral do *TL-Bandits*. Assume-se K agrupamentos de itens.

No Algoritmo 2, detalha-se o problema de recomendação sequencial como um algoritmo MAB de dois níveis. Observa-se que o algoritmo proposto assume como entrada duas instâncias de algoritmos do tipo *bandits*, uma para cada nível. Também considera-se que os itens (ou seja, vídeos) foram previamente agrupados, conforme mencionado acima.

Assim, como no Algoritmo 1, o processo de recomendação acontece em instantes discretos de tempo $t = 1, 2, \ldots$ (Linha 1). Em cada instante de tempo t, o algoritmo primeiro seleciona o agrupamento mais relevante a partir do qual um item será selecionado (Linha 2). Na Linha 3, o *MAB* de nível 2 seleciona um vídeo no agrupamento previamente selecionado. O vídeo selecionado é apresentado ao usuário alvo e suas avaliações de relevância são capturadas (Linhas 4 a 9). O passo final refere-se a atualizar a recompensa extraída da ação do usuário para os MABs em cada nível de forma independente (Linhas 10 a 11).

Neste trabalho, assume-se que os MAB em ambos os níveis são da mesma família, ou seja, se tivermos o algoritmo Bayes UCB no nível 1, então temos outra instanciação do

mesmo algoritmo no nível 2. Há que enfatizar que o algoritmo *TL-Bandits* proposto é flexível para considerar qualquer algoritmo MAB para escolher o agrupamento e os vídeos dentro dos agrupamentos.

Algoritmo 2: Algoritmo *Bandit* de Dois Níveis

Input: Algoritmo *Bandit* de nível-1 $\mathcal{A}_1$ e nível-2 $\mathcal{A}_2$

for $t = 1, \ldots, T$ **do**
  $c(t) \leftarrow$ seleciona-*agrupamento*$(u(t), \mathcal{A}_1)$
  $i(t) \leftarrow$ seleciona-item$(u(t), c(t), \mathcal{A}_2)$
  exibe $i(t)$ ao usuário; registra o clique
  if *usuário clicou em* $i(t)$ **then**
    $r(t) = 1$
  end
  else
    $r(t) = 0$
  end
  atualiza-mab$(u(t), i(t), r(t), \mathcal{A}_1)$
  atualiza-mab$(u(t), i(t), r(t), \mathcal{A}_2)$
end

5 Avaliação Experimental

Neste capítulo apresenta-se os experimentos realizados para validação do algoritmo proposto. Assim, o capítulo esta organizado da seguinte forma. Na Seção 5.1, detalha-se o conjunto de dados utilizado, bem como apresenta-se uma caracterização do mesmo para demonstrar a necessidade de tratar-se o caso de escassez de informação de preferências dos usuários. Na Seção 5.2, apresenta-se as métricas de avaliação da qualidade da recomendação utilizadas nesta dissertação. Na Seção 5.3 é detalhado o protocolo de avaliação utilizado neste trabalho. Na Seção 5.4 são apresentados os métodos estado-da-arte em recomendação livre de contexto. Na Seção 5.5 são detalhados os algoritmos de agrupamento de vídeos utilizados nesta dissertação. Nas Seções 5.6, 5.7, 5.8 e 5.9 apresenta-se e discute-se os resultados dos experimentos. Na Seção 5.11 ressalta-se a reprodutibilidade de todas as etapas de experimentação realizadas nesta dissertação.

5.1 Conjunto de Dados

Para avaliar a qualidade da recomendação do método proposto nesta dissertação, foi utilizado o conjunto de dados *MovieLens1M*, que contém 1.000.209 avaliações para 3.883 filmes, realizadas por 6.040 usuários (MILLER et al., 2003a). As avaliações coletadas estão em uma escala de 1 a 5 estrelas, o que para esse trabalho teve uma implicação em seu uso. Por conta disso foi necessário adaptá-la para o contexto dessa pesquisa, que baseia a avaliação em cliques. A explicação para contornar essa limitação encontra-se na Seção 5.2.

É importante enfatizar que este é um conjunto de dados sobre classificação de filmes disponível publicamente. A escolha desse conjunto de dados se deu pelo fato dele ser amplamente utilizado na literatura de sistemas de recomendação. A plataforma onde o mesmo é disponibilizado hospeda experimentos desde o seu lançamento em 1997, ou seja, 20 anos de atualizações e pesquisas em sua base de dados (HARPER; KONSTAN, 2016). O Movielens1M em particular concentra dados de usuários que entraram na plataforma desde o ano 2000, além de sua formatação ser de fácil compreensão e manuseio aliado ao fato de ser um conjunto de dados de referência estável.

Como podemos ver nos dados extraídos para a geração da Figura 9, há uma baixa taxa de usuários (isto é, avaliações) para um grande número de filmes avaliados; enquanto que, uma grande quantidade de usuários avalia um pequeno número de filmes. Esta distribuição é um exemplo do problema de *cold-start* de usuários, para o qual existe uma escassez acentuada nas preferências dos usuários para com os filmes. Assim, pode-se concluir que a hipótese fundamental deste trabalho, i.e. a escassez de informação de

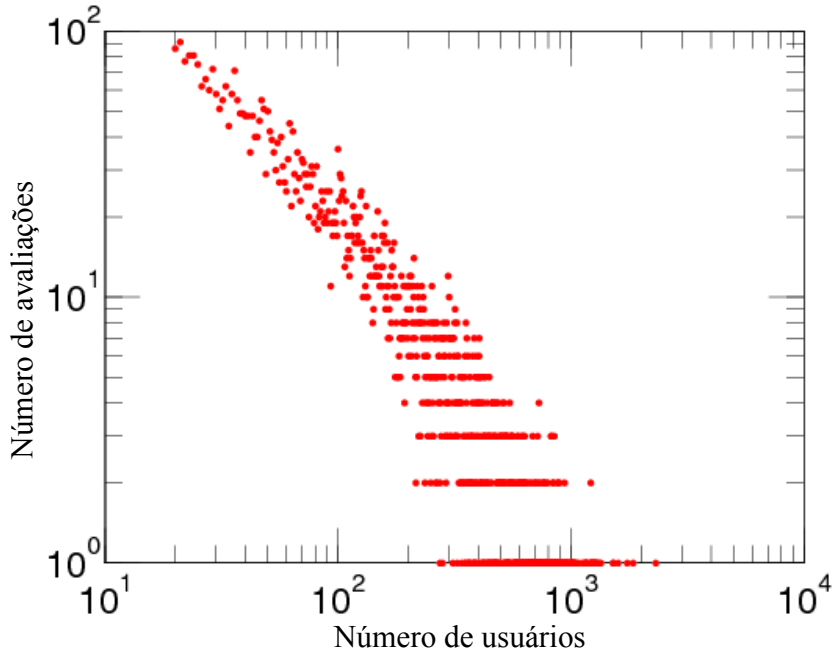

Figura 9 – Distribuição exponencial inversa de usuários no conjunto de dados do Movielens1M.

preferências dos usuários, está presente no conjunto de dados utilizado.

5.2 Métricas

No contexto da recomendação de itens, quando um item é clicado, é retornada uma recompensa igual a 1; caso contrário, a recompensa é igual a 0. Assim, a recompensa esperada de um item é sua Taxa de Cliques *(Click-Through-Rate ou CTR)* e escolher um item com previsão máxima de CTR é o mesmo que maximizar o número esperado de cliques dos usuários, o que, por sua vez, equivale a maximizar a recompensa total esperada da formulação dos algoritmos *bandits* propostos. A recompensa média é equivalente à métrica CTR, que é a recompensa total dividido pelo número total de tentativas:

$$CTR \;\; Relativa = \frac{1}{T}\sum_{t=1}^{T} r_t, \tag{1}$$

onde r_t é a recompensa em rodada t e T é o número total de rodadas. É relatado o CTR relativo, que é a CTR do algoritmo dividida pelo recomendador aleatório.

Também são relatados os resultados com base na Taxa de Cliques Acumulada *Cumulative Click-Through Rate*, que se refere a soma das recompensas recebidas por um algoritmo MAB até a rodada i. Formalmente:

$$CTR(i) \;\; Acumulada = \sum_{t=1}^{i} r_t, \tag{2}$$

Em outras palavras, a CTR acumulada mede a quantidade total de recompensa que um algoritmo recebe até um ponto fixo no tempo.

O conjunto de dados do Movielens apresenta uma escala de avaliação de relevância baseada em valores entre 1 e 5. Para adaptá-la à métrica CTR foi necessário considerar um filme avaliado pelo usuário como um filme que tenha sido clicado e assistido pelo mesmo, ou seja, considera-se como avaliação de relevância o interesse do usuário no filme. Se ele clicou para assistir ao filme, há uma indicação implícita de interesse em ver esse filme, independentemente dele dar uma nota baixa ou alta ao filme.

Sendo assim use-se este dado(i.e. clique) como avaliação de relevância implícita, o que faz mais sentido para o problema de novos usuários, já que aqui trata-se de um cenário em que não há informações prévias das preferências do usuário. Sendo assim a entrada do algoritmo MAB é uma tupla composta de <usuário r, item i>. Esse modelo de avaliação pode ser utilizado não apenas no contexto dessa pesquisa, mas também em outras aplicações, como marketing digital, notícias, indicação de links (Adwords), dentre outras.

5.3 Protocolo de Avaliação

A avaliação de sistemas de recomendação sequencial é uma tarefa difícil e objeto de estudo em trabalhos da área de recomendação (LI et al., 2011; CHAPELLE; LI, 2011; STREHL et al., 2010; HOFMANN et al., 2016). Nesta dissertação, seguiu-se a abordagem proposta pelos autores em (LI et al., 2011), que permite o uso de dados estáticos para avaliação de sistemas de recomendação sequencial.

Por estar simulando um ambiente *online* em um experimento *offline*, não há como prever o que poderá acontecer com os resultados após um determinado instante de tempo, pois o experimento é finito e não contínuo. Para isso é necessário repeti-lo várias vezes, porém é necessário manter-se a coerência de cada experimento realizado. O método utilizado nesta dissertação baseia-se no conceito de *bootstrapping* introduzido por (EFRON, 1979). Trata-se de uma abordagem do cálculo de intervalos de confiança dos parâmetros em circunstâncias particulares, nas quais o número de amostras é reduzido. Essa técnica foi extrapolada para a solução de muitos outros problemas de difícil resolução através de técnicas para análise estatística tradicionais(EFRON; TIBSHIRANI, 1994).

Partindo desta ideia, nesta dissertação é utilizado o método estado-da-arte para avaliação *offline* de algoritmos *bandits*: BRED (*Bootstrapped Replay on Expanded Data* ou Bootstrap Replay sob Dados Expandidos em tradução livre)(LI et al., 2011). O método BRED permite a avaliação imparcial utilizando dados históricos, ou seja, dados previamente coletados. O método pressupõe que o conjunto de dados é estático para executar as re-amostragens baseadas em *bootstrap*. A pesquisa segue a mesma abordagem apresentada

em (MARY; PREUX; NICOL, 2014) e assume o mundo estático em pequenas porções do conjunto de dados MovieLens1M. Toma-se o menor número de porções de tal forma que uma determinada parcela tem um número fixo de filmes. Consideremos cada porção como um único conjunto de dados D de tamanho T com K possíveis escolhas (i.e., vídeos). A seguir, a partir de D, em cada etapa, gera-se P novos conjuntos de dados amostrados $D_1, D_2, ..., D_P$ de tamanho $K \times T$ por amostragem com substituição. Para cada conjunto de dados i de tamanho D_i com K_i vídeos, computa-se o CTR estimado dado pela média da CTR de cada algoritmo *bandit* em 100 permutações aleatórias dos dados. São relatadas a CTR média e a CTR acumulada em todas as porções.

5.4 Modelos de Referência

Conforme descrito na Seção 2.3, foram utilizados os seguintes algoritmos estado-da-arte em tomada de decisão sequencial com recompensa limitada:

$\rightarrow$ Random Choice: recomenda um dos vídeos disponíveis com igual probabilidade.

$\rightarrow$ ϵ-greedy (ϵ) (EG): recomenda um vídeo com probabilidade ϵ, e recomenda o vídeo com maior CTR estimada com probabilidade $(1 - \epsilon)$ (TOKIC, 2010).

$\rightarrow$ UCB1: recomenda o vídeo com maior valor UCB (*Upper Confidence Value*), de acordo com (AUER; CESA-BIANCHI; FISCHER, 2002).

$\rightarrow$ UCB2 (η): similar ao anterior, porém utilizando outra forma de computar o valor UCB (AUER; CESA-BIANCHI; FISCHER, 2002). O parâmetro η balanceia a quantidade de prospecção e exploração.

$\rightarrow$ Bayes UCB: é uma adaptação bayesiana do algoritmo UCB, no qual o valor UCB é computado levando-se em consideração o quantil superior da média de CTRs estimados de cada vídeo (KAUFMANN; CAPPÉ; GARIVIER, 2012).

$\rightarrow$ Thompson sampling (TS): é uma abordagem bayesiana de um algoritmo MAB no qual os valores de recompensa são inferidos a partir de dados históricos e resumidos utilizando um distribuição *a posteriori*. Resumidamente, o algoritmo recomenda itens com probabilidade de sucesso proporcional ao sucesso de recomendações anteriores (THOMPSON, 1933).

5.5 Métodos de Agrupamento

Nesta dissertação utilizou-se três métodos distintos para agrupamentos dos vídeos de acordo com suas similaridades: agrupamento por categoria do vídeo (Seção 5.5.1), agrupamento de acordo com o algoritmo k-means (Seção 5.5.2) e agrupamento de acordo

com o algoritmo de extração de tópicos *Latent Dirichlet Allocation* (Seção 5.5.3). Nas próximas seções detalha-se cada método de agrupamento utilizado.

5.5.1 Categoria

Nesta abordagem de agrupamento é usada a categoria pré-definida na qual todos os filmes no conjunto de dados são classificados. Por exemplo, temos as categorias *Ação*, *Aventura*, *Romance*, etc. As principais desvantagens desta abordagem de agrupamento são que (i) o número de agrupamentos é fixo e (ii) é dispendioso, uma vez que é necessário fazer uma classificação manual para cada item.

A modelagem de extração de dados é a mais simples das três utilizadas nessa pesquisa, já que o algoritmo que busca as informações foi criado para fazer um verificação simples dentre as categorias de um vídeo. Dado que um vídeo v possui uma categoria principal C e um número n de sub-categorias S, é escolhida sempre a categoria principal C para ser o agrupamento em que v será inserido. Naturalmente, para cada um dos vídeo existe apenas uma categoria principal, apesar de existir mais de uma sub-categoria relacionada ao mesmo vídeo.

5.5.2 k-Means

Seja $M = \{m_1, \ldots, m_n\}$ o conjunto de títulos e sinopses concatenados para cada vídeo m_i, para o qual deseja-se calcular um conjunto de k grupos, $C = c_1, \ldots, c_k$ (JAIN, 2010). O algoritmo k-Means encontra uma partição tal que o erro quadrático entre a média empírica de um agrupamento e os pontos dentro do agrupamento são minimizadas. Dado μ_k como o centróide do agrupamento c_k o erro quadrático $E(c_k)$ entre μ_k e os pontos no agrupamento c_k são definidos como:

$$E(c_k) = \sum_{m_i \in c_k} ||m_i - \mu_k||^2. \tag{3}$$

O objetivo final do k-Means é minimizar a soma do erro quadrático em todos os K agrupamentos,

$$E(C) = \sum_{k=1}^{K} \sum_{m_i \in c_k} ||m_i - \mu_k||^2. \tag{4}$$

Para extrair os dados do Movielens foi utilizada a biblioteca *sckit learn* [1]. A abordagem utilizada é uma variante do algoritmo de Lloyd, descrita em (ARTHUR; VASSILVITSKII, 2007) e para utilizá-la foi necessário primeiramente extrair todos os descritores de cada vídeo contido no conjunto de dados criando uma *bag-of-words*, que é uma técnica para criar um dicionário de palavras, onde cada vídeo possui uma descrição da frequência em que

[1] Biblioteca sckit-learn k-Means: http://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html

cada uma das palavras aparece na descrição do vídeo. Também foi realizada a remoção das *stopwords*, que são os artigos e preposições em inglês, como *"the"*, *"of"*, *"a"*, etc. Somente após esse processo, o método k-Means foi aplicado, passando-se o parâmetro de quantos agrupamentos devem ser gerados e o número de iterações a serem feitas. Para cada observação é atribuído um agrupamento de modo a minimizar a soma de quadrados dentro do agrupamento. Logo após, a média das observações em cada agrupamento é calculada e usada como o centróide do novo agrupamento. Em seguida, as observações são reatribuídas a grupos e centróides recalculados em um processo iterativo até o algoritmo alcançar a convergência, que é encontrada através do erro quadrático. Dado tempo suficiente, K-means sempre convergirá, no entanto, isso pode ser para um mínimo local. Isso é altamente dependente da inicialização dos centróides. Como resultado, a computação é muitas vezes feita repetidamente, com diferentes inicializações dos centróides. Pelo fato da biblioteca ser inicializada aleatoriamente, isso pode ser um ponto fraco, já que não há como inicializar os centróides para ficarem mais distantes um do outro.

5.5.3 Latent Dirichlet Allocation (LDA)

Como visto na sessão 2.4, o modelo de tópico *Latent Dirichlet Allocation* é um modelo probabilístico generativo para coleções de dados discretos (por exemplo, conjunto de documentos) (BLEI; NG; JORDAN, 2003). Um modelo de tópico é um algoritmo que se concentra na descoberta de estruturas latentes, ou seja, padrões ocultos que ajudam a descrever os dados. Uma vez que é um modelo generativo, ele define como as palavras em um documento são geradas através do controle de tópicos latentes. No contexto da recomendação de vídeos, as palavras vêm dos títulos e das sinopses, e os tópicos referem-se a grupos de vídeos com características semelhantes.

No LDA, assume-se que cada descrição de vídeo observada (título e sinopse) foi gerada por misturas ponderadas de tópicos latentes não-observados, que podem ser aprendidos com as descrições e se referem a temas semânticos presentes no conjunto de dados dos vídeos. Formalmente, o LDA é definido usando os seguintes termos (BLEI; NG; JORDAN, 2003):

- A *palavra* é a unidade básica de dados discretos em um vocabulário V.
- A *descrição do filme* é uma sequência de N palavras extraídas do título e sinopse do filme.
- O *conjunto de dados* é uma coleção $|I|$ de descrições dos filmes.

O processo gerador do LDA especifica um procedimento probabilístico simples para gerar novas descrições dos vídeos de acordo com um conjunto de tópicos ϕ. A descrição do filme é gerada primeiro selecionando uma distribuição sobre os tópicos θ de uma distribuição *Dirichlet*, que determina a probabilidade $P(z)$ (probabilidade do tópico z) para as palavras nessa descrição. As palavras na descrição do filme são geradas selecionando um tópico i

desta distribuição e, em seguida, escolhendo uma palavra desse tópico seguindo $P(w|z = i)$. A estimativa de parâmetros do modelo utilizado aqui segue a abordagem de Amostragem de Gibbs colapsada (GRIFFITHS; STEYVERS, 2004).

Para modelar a solução para gerar os tópicos, foram utilizados dois pacotes. O *Natural Language Toolkit* (NLTK) [2] descrita como uma plataforma líder para a construção de programas escritos na linguagem Python para trabalhar com dados de linguagem humana, baseada no livro de 2009 (BIRD; KLEIN; LOPER, 2009) e todos seus algoritmos em licença *open source*. Também foi utilizado o pacote Gensim para aplicar o LDA, baseado nas descrições realizadas em (HOFFMAN; BACH; BLEI, 2010). Este módulo permite estimativa do modelo LDA a partir de um corpo de treinamento através do conjunto de dados e inferência da distribuição de tópicos nos vídeos novos e não vistos. O modelo também pode ser atualizado com novos vídeos para treinamento online.

Como para gerar o LDA é necessário extrair muita informação para uma geração fiel de tópicos, foi utilizada a base de dados do IMDb [3], para adicionar ao conjunto de dados do Movielens as sinopses de cada um dos filmes. Os dados foram extraídos através de um código que gera um arquivo xml ao final do processo. Por fim, os dados são adicionados ao Movielens 1M.

Na etapa seguinte, é carregada a lista NLTK de artigos e preposições em inglês, conhecidas como *stopwords*. *Stopwords* são palavras como "a", "the"ou "in"que não transmitem significados relevantes para geração do modelo LDA. A partir desse momento é então utlizada a biblioteca Gensim [4], pré-processando as sinopses, inicialmente usando uma função para remover qualquer substantivo próprio a partir do modelo gerado pelo NLTK. O modelo para geração do LDA é então executado. De posse dos tópicos que descrevem cada vídeo é utilizado aquele que possui o maior valor dentre todos os que compõem a descrição do vídeo e ele passa a ser o agrupamento ao qual o vídeo pertence.

[2] *Natural Language Toolkit*: http://www.nltk.org/index.html
[3] *Internet Movie Database*: http://www.imdb.com/
[4] *Models.ldamodel*: https://radimrehurek.com/gensim/models/ldamodel.html

5.6 Desempenho dos métodos *Baseline* de recomendação

Tabela 1 – Média Relativa CTR. Em negrito, estão destacados os ganhos sobre o *baseline*. Entre parênteses, são mostrados os números de agrupamentos.

	Baseline	Categoria	Agrupamentos k-Means						Agrupamentos LDA					
	-	(17)	(5)	(10)	(17)	(25)	(50)	(100)	(5)	(10)	(17)	(25)	(50)	(100)
Random	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000	1,000
EG(0,0)	1,146	**3,752**	**1,326**	0,233	**2,974**	0,323	0,364	0,010	**2,126**	0,045	0,045	0,138	0,000	0,087
EG(0,1)	1,034	**6,296**	**1,670**	**1,757**	**3,466**	**4,836**	**4,248**	**3,182**	**6,551**	**5,568**	**6,348**	**5,601**	**5,026**	**6,317**
EG(0,2)	0,921	**5,872**	**1,545**	**1,929**	**3,345**	**4,079**	**3,467**	**4,924**	**5,669**	**5,288**	**6,536**	**4,978**	**5,517**	**6,260**
EG(0,3)	0,966	**4,576**	**1,602**	**1,981**	**2,569**	**3,360**	**3,486**	**4,667**	**4,795**	**3,909**	**4,446**	**4,290**	**4,810**	**5,308**
EG(0,4)	0,989	**3,608**	**1,511**	**1,871**	**2,586**	**3,503**	**3,168**	**2,444**	**3,819**	**3,432**	**4,259**	**3,043**	**3,698**	**3,817**
EG(0,5)	0,933	**3,248**	**1,371**	**1,743**	**1,629**	**3,053**	**2,364**	**3,298**	**3,472**	**2,750**	**2,205**	**2,036**	**2,879**	**3,490**
EG(1,0)	0,944	**1,008**	**0,947**	0,929	**1,009**	**1,116**	0,841	0,803	**0,992**	**0,977**	**1,036**	**1,014**	0,931	0,904
UCB1	0,876	**1,824**	0,852	0,833	**1,560**	**0,910**	0,864	**0,955**	**2,835**	**1,818**	**1,411**	**1,275**	**1,328**	**1,452**
UCB2 (0,0)	2,135	**3,752**	1,326	0,233	**2,974**	0,323	0,364	0,010	2,126	0,045	0,045	0,138	1,681	0,087
UCB2 (0,1)	1,011	**2,104**	**1,061**	**1,033**	**1,690**	**1,048**	**1,028**	**1,020**	**3,630**	**2,220**	**1,679**	**1,522**	**1,422**	**1,798**
UCB2 (0,2)	0,831	**2,040**	**1,000**	**1,057**	**1,707**	**1,063**	**1,061**	**1,081**	**3,748**	**2,083**	**1,634**	**1,449**	**1,543**	**1,538**
UCB2 (0,3)	0,978	**2,024**	**1,027**	**1,090**	**1,595**	**1,111**	**1,028**	**0,980**	**3,567**	**2,167**	**1,563**	**1,551**	**1,491**	**1,462**
UCB2 (0,4)	0,933	**2,216**	**1,030**	**0,976**	**1,750**	**1,201**	**0,972**	**0,975**	**3,567**	**2,000**	**1,598**	**1,725**	**1,543**	**1,615**
UCB2 (0,5)	1,135	**2,112**	1,057	0,981	**1,931**	1,106	0,986	1,015	**3,709**	**1,977**	**1,813**	**1,406**	**1,474**	**1,846**
UCB2 (0,7)	1,079	**2,216**	0,932	**1,124**	**1,948**	**1,196**	1,056	1,020	**3,614**	**1,932**	**1,598**	**1,565**	**1,672**	**1,769**
UCB2 (1,0)	1,045	**2,192**	**1,102**	**1,167**	**1,983**	**1,317**	**1,276**	**1,172**	**3,386**	**2,106**	**1,688**	**1,609**	**1,741**	**1,904**
Bayes UCB	0,697	**6,344**	**1,652**	**1,862**	**2,897**	**3,804**	**2,430**	**2,157**	**7,551**	**6,811**	**7,107**	**5,181**	**4,267**	**2,846**
TS	1,079	**1,272**	**1,856**	**2,205**	**1,207**	**4,529**	**2,907**	**3,616**	1,016	0,970	**1,098**	**1,246**	**1,181**	**1,231**

Na Tabela 1 é apresentada a métrica de desempenho CTR de todos os algoritmos *bandit* testados. Note que todos os resultados são normalizados sobre o método Random, que é explicado na subseção 2.3.4. Portanto, as estratégias com números abaixo de 1.0 são piores em relação à CTR do que o método Random. A coluna *Baseline* refere-se às estratégias *bandits* sem considerar agrupamentos de itens, ou seja, eles se referem às estratégias para as quais o braço é o item. Além disso, quando se fala na estratégia *Baseline*, significa que os teste são realizados com um algoritmo MAB padrão de apenas um nível. Em negrito, estão destacados os resultados para os quais foram obtidos um resultado melhor do que os métodos de referência (*baselines*). O algoritmo ε-greedy é tratado na Tabela 1 como EG. É necessário lembrar que todos os resultados foram obtidos usando a técnica de re-amostragem através da técnica de Bootstrapping, ou seja, o valor é em cima das médias de 100 permutações. A seguir são descritos e exibidos os resultados mais detalhados desta Tabela 1 para melhor visualização e entendimento.

Tabela 2 – Melhores resultados *Baseline* e Categoria para a Média Relativa CTR

	Baseline	Category
	-	17
UCB2 (0,0)	2,135	3,752
Bayes UCB	0,697	6,344

A Tabela 2 mostra um comparativo entre os melhores resultados do *Baseline* e Categoria respectivamente. Nos experimentos da coluna Categoria, considera-se primeiramente os agrupamentos como sendo as categorias dos filmes (e.g., Drama, Comédia,

etc). Ao usar a categoria para agrupar os filmes, obtém-se um número predefinido de agrupamentos. Nesse conjunto de dados, os filmes são agrupados em 17 categorias. Ao usar essa instanciação para a estrutura é possível conseguir resultados promissores em relação à CTR. Por exemplo, o algoritmo Bayes UCB atinge uma CTR relativa média de $6,344$ e mesmo o melhor valor atingido pelo método *Baseline* através do algoritmo UCB(0.0) ainda é superado quando comparado com a Categoria.

Também foram testadas outras duas abordagens de agrupamento para as quais pode-se definir o número de agrupamentos, ou seja, k-Means e LDA. Para ambos os casos, optou-se por usar valores de 5, 10, 17, 25, 50 e 100 agrupamentos. Ao analisar os melhores resultados, pode-se ver que o LDA com 5 agrupamentos supera todas as configurações testadas da estrutura proposta e todas as instâncias dos métodos de referência. Esses resultados são discutidos a seguir na Seção 5.7.

5.7 Efeito dos métodos de agrupamento

Tabela 3 – Resultados LDA com algoritmo Bayes UCB para a Média Relativa CTR

Agrupamentos LDA					
5	10	17	25	50	100
7,551	6,811	7,107	5,181	4,267	2,846

Conforme apresentado na Tabela 3, os melhores resultados em relação à métrica CTR se referem a itens agrupados com a representação de tópicos LDA. Lembrando que o LDA extrai tópicos das sinopses dos filmes e os representa num espaço latente. Foi usada essa representação para descrever os filmes e agrupá-los de acordo com o tópico mais provável. Uma vantagem do LDA e do k-Means é que o número de agrupamentos é flexível e pode ser ajustado de acordo com cada conjunto de dados. O fato dos melhores resultados serem atribuídos ao método LDA pode ser explicado devido à sua robustez ao lidar com a descrição muito densa das sinopses dos filmes. Alguns detalhes sobre como os agrupamentos influenciaram os resultados obtidos através do LDA, podem ser vistos na Seção 5.8.

Tabela 4 – Resultados K-Means com algoritmo ε-Greedy (0,2) para a Média Relativa CTR

Agrupamentos K-Means					
5	10	17	25	50	100
1,545	1,929	3,345	4,079	3,467	4,924

Também podemos ver na Tabela 4, que o desempenho do k-Means é melhor do que o método de referência que não utiliza agrupamentos (i.e., *Baseline*). No entanto, quando comparado com as estratégias de agrupamento de Categoria e LDA, o k-Means mostrou-se

incapaz de apresentar melhores recomendações, como é explicado na Seção 5.8. Acredita-se que a escassez da representação textual de itens (no caso, foram usados tags, título e categorias) leva à agrupamentos pobres dos itens que são gerados pelo algoritmo de k-Means.

5.8 Efeito do número de agrupamentos

Na Figura 10, são apresentados os valores relativos de CTR em relação ao número de agrupamentos. Observe que tanto os métodos de referência quanto a abordagem de agrupamento baseada em categorias são independentes dos números de agrupamentos. No primeiro, ou seja, nos métodos de referência, cada item é um braço, portanto, não há agrupamento de vídeos. No último, isto é, na abordagem baseada em categorias, o número de agrupamentos é pré-definido e é igual a 17 nesse conjunto de dados. Por isso, para essas abordagens insensíveis ao agrupamento, na Figura 10, temos duas linhas paralelas ao eixo x. No entanto pode-se considerá-las um ponto fixo, no caso do *Baseline* como um ponto em zero no eixo x e para o Categoria, como um ponto em 17 no eixo x.

É importante frisar também, que a abordagem de Bootstrapping torna os resultados mais fiéis, já que a re-amostragem com substituição permite que se obtenha 100 vezes mais dados do que o conjunto de dados original permite. Em outras palavras, simula-se da melhor forma o ambiente online através de experimentos offline.

Ao considerar as abordagens de agrupamento k-Means e LDA, tem-se padrões diferentes. Enquanto no k-Means à medida em que aumentam o número de agrupamentos ocorrem melhorias no desempenho; na LDA existe uma tendência oposta, ou seja, o melhor desempenho do LDA acontece quando possui apenas 5 agrupamentos. Isso significa que quanto menor o número de agrupamentos, melhores são as escolhas dos tópicos gerados pelo LDA, além do que se atinge uma melhor taxa de cliques relativa, consequentemente deixando claro que essa abordagem gera melhores recomendações. No caso do k-Means quanto mais agrupamentos se tem, melhores são as taxas de cliques relativa, o que significa que as melhores recomendações serão geradas a partir de um maior número de agrupamentos para o k-Means. Nota-se também que mesmo com sua ascendência no gráfico, terão de ser gerado um grande número de agrupamentos para que ele tenha a possibilidade de em determinado momento bater o resultado obtido pelo LDA com 5 agrupamentos.

É interessante notar também, que por volta de pouco mais de 60 agrupamentos, o LDA e o k-Means se cruzam no gráfico e a partir desse momento, o LDA tende a perder e fazer recomendações piores em relação ao k-Means. De todo modo, olhando de uma forma geral todos os resultados exibidos para todos os agrupamentos criados, o LDA bate por muito a abordagem k-Means.

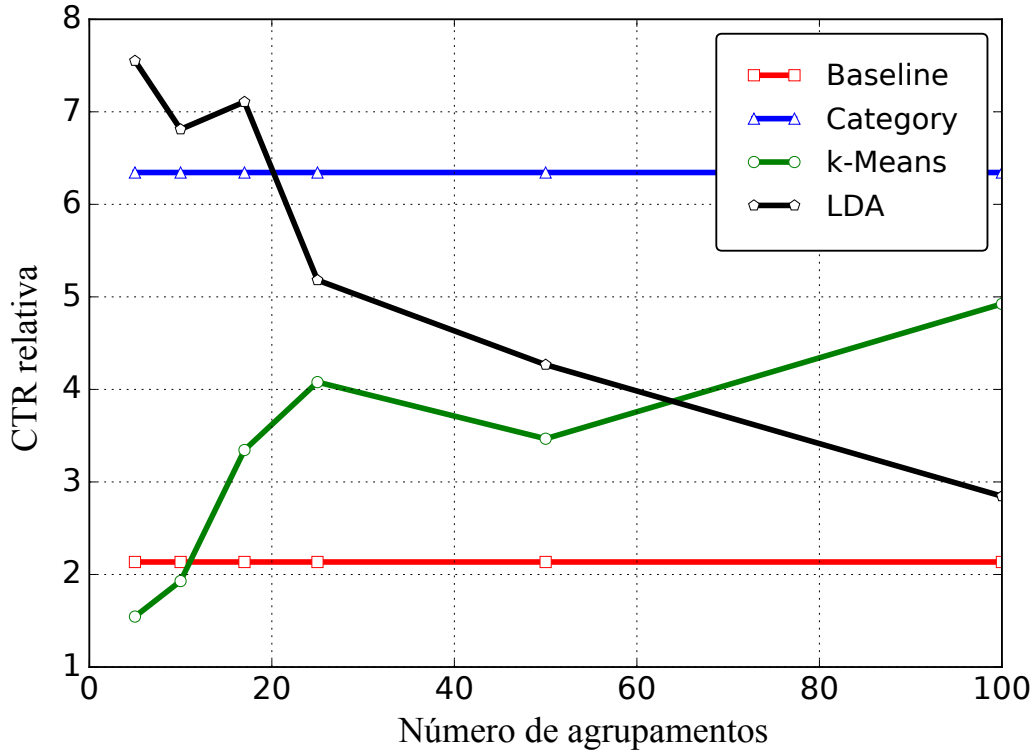

Figura 10 – CTR vs Número de *Agrupamentos*. Os algoritmos MAB usados com *Baseline*, Categoria, k-Means, e LDA, são UCB2(0.0), Bayes UCB, EG(0.2), e Bayes UCB, respectivamente.

Outro dado interessante é que a abordagem de agrupamento por meio da categoria do vídeo, que no caso é uma de 17 categorias, superar a abordagem k-Means utilizando 17 agrupamentos. Pelo fato das categorias serem fixas, não há como saber como seria a curva de tendência para o caso de haver mais categorias, mas o que se supõe, é que em determinado momento o k-Means superaria a abordagem de categorias, dado que quanto maior o número de agrupamentos para essa abordagem, mais eficaz se tornam as recomendações realizadas.

5.9 Análise da qualidade da recomendação através das rodadas

Nesta seção, é investigado o comportamento de cada estratégia de recomendação em função do tempo. A partir da Figura 11, nota-se que cada rodada se refere a uma interação específica entre o usuário-alvo e o sistema de recomendação. A abordagem de agrupamento LDA combinada com o algoritmo MAB Bayes UCB atinge o melhor desempenho durante as rodadas de recomendação.

Note que na Figura 11, ao contrário da Figura 10, o k-Means ganha da estratégia de agrupamento por categoria. Isso pode ser explicado pelo fato de que há mais ganho de recompensa em curtos espaços de recomendações, ou seja, a cada recomendação o

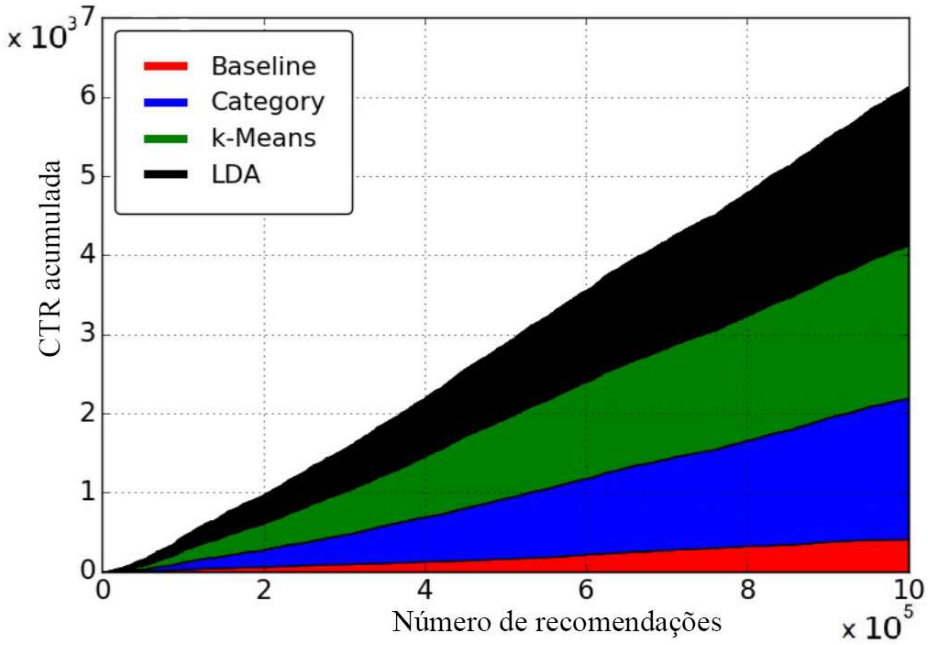

Figura 11 – Recompensa Cumulativa. Os algoritmos MAB com o *Baseline*, Categoria, k-Means e LDA, são UCB2(0.0), Bayes UCB, EG(0.2), e Bayes UCB, respectivamente.

usuário tende a dar mais avaliações positivas do que negativas, impactando na soma final da recompensa. Em outras palavras, esse gráfico mostra uma simulação do clique dado pelo usuário no vídeo a cada vez que uma recomendação é feita. Se o usuário clica com mais frequência nas recomendações exibidas então a soma de cada clique, onde cada clique possui o valor 1, terá um crescimento em relação ao eixo y mais rápido do que em casos que o usuário tende a clicar poucas vezes nas recomendações apresentadas. No final a soma será maior se mais recomendações que geraram interesse do usuário foram apresentadas.

5.10 Contribuição da Dissertação

Esta pesquisa gerou uma contribuição à ciência através do artigo *Improved User Cold-Start Recommendation via Two-Level Bandit Algorithms*, publicado no ENIAC (Encontro Nacional de Inteligência Artificial e Computacional) de 2017, organizado pelas Comissões Especiais de Inteligência Artificial e de Inteligência Computacional da Sociedade Brasileira de Computação. O evento oferece um fórum para pesquisadores, profissionais, educadores e estudantes apresentarem e discutirem as inovações, tendências, experiências e evolução nos campos de Inteligência Artificial e Computacional [5].

[5] ENIAC 2017: http://www.bracis2017.ufu.br/eniac-encontro-nacional-de-inteligencia-artificial-e-computacional

5.11 Reprodutibilidade

Visando permitir à comunidade acadêmica obter acesso fácil aos experimentos e resultados gerados nesta dissertação o conjunto de dados tratado que foi utilizado nos experimentos está disponível publicamente e pode ser obtido em (MILLER et al., 2003a). Todos os métodos de referência utilizados (cujos *links* são mostrados em suas respectivas sessões), bem como as implementações (acessíveis no GitHub), também estão disponíveis para testes e pesquisa. Assim, enfatiza-se que os resultados obtidos são facilmente reprodutíveis, o que permite desenvolvimentos futuros e possíveis melhorias, tendo como base o algoritmo proposto.

Todos os algoritmos utilizados foram implementados na linguagem de programação Python 2.7 e executados no sistema operacional Ubuntu Linux 15.04. A máquina de execução dos experimentos possui processador Intel® Core™ i5 de 6ª geração, com 8GB de memória e disco rígido de 1TB.

6 Conclusão e Trabalhos futuros

O desenvolvimento de um sistema de recomendação é uma tarefa bastante desafiadora do ponto de vista científico. Uma vasta quantidade de técnicas diferentes devem ser utilizadas para que se possa fornecer um sistema que seja capaz de fazer recomendações relevantes em tempo real. Durante todo o processo, desde a aquisição dos metadados dos vídeos, até a recomendação em si, necessita-se de um bom nível de robustez e desempenho visando atingir os objetivos. A utilização deste tipo de sistema atrai grande interesse tanto da comunidade acadêmica e da indústria devido aos desafios computacionais envolvidos e ao sucesso de sua utilização, respectivamente.

Neste trabalho, para solucionar o problema de recomendação em um ambiente com escassez de informação sobre as preferências dos usuários foi proposto um novo algoritmo (*TL-BANDIT*) do tipo *Multi-Armed Bandit* que funciona em duas etapas. A intuição na qual se baseou o método proposto foi de que vídeos parecidos podem ser agrupados para melhorar a qualidade da recomendação. Vídeos de um mesmo agrupamento podem agradar o usuário-alvo da recomendação e ao invés de escolher dentre todo o universo de vídeos, busca-se um processo de tomada de decisão em dois níveis. Com isso, esperava-se que vídeos com conteúdo parecido possuíssem taxas de acesso parecidas e poderiam ser agrupados de acordo com as características que os descrevessem.

A estrutura do algoritmo proposto é flexível e pode ser instanciada de maneiras diferentes dependendo do cenário de aplicação. Foram realizados vários experimentos utilizando um conjunto de dados do mundo real para recomendação de vídeos. Os resultados experimentais mostraram que esta estrutura pode melhorar a qualidade da recomendação e fornecer ganhos significativos sobre métodos estado-da-arte.

As técnicas utilizadas para realizar o agrupamento buscavam extrair o máximo de informação textual possível que descrevesse os vídeos, com isso, poderia-se ter uma relação de similaridade semântica, para então organizar esses vídeos e posteriormente agrupá-los. Foram usadas três técnicas para níveis de comparação, uma pela Categoria dos vídeos, outra utilizando o algoritmo de agrupamentos k-Means e por fim uma que utiliza o método LDA. Durante os experimentos o LDA mostrou-se mais robusto em obter melhores agrupamentos, já que era possível obter as melhores recomendações através dessa técnica.

Foram implementadas as soluções do *Multi-Armed Bandits* mais conhecidas com o objetivo de sugerir itens relevantes para os novos usuários e aprender mais sobre eles. A pesquisa utilizou 6 algoritmos, sendo que somando também os casos em que um único algoritmo pode ter o parâmetro de prospecção e exploração alterado para o experimento,

foram utilizadas no total 19 formas diferentes para o arcabouço de dois níveis. Levando em conta que cada um desses algoritmos foi executado para cada modelo de agrupamento e também para o modelo *Baseline*, que não considera agrupamentos, foram obtidos no total 266 resultados comparativos sobre a eficácia da proposta da pesquisa. Além disso, todas elas foram validadas através do método de *bootstrapping*, que é estado-da-arte para avaliação de sistema de recomendação offline, que simulam um ambiente online.

Com isso, por meio das técnicas descritas nesta pesquisa, escolhendo agrupamentos de vídeos e dentro destes, selecionando um vídeo relevante, foi possível melhorar a qualidade da recomendação para novos usuários, sobre os quais não havia nenhuma informação prévia disponível sobre suas preferências. Ao mesmo tempo, permitiu que se aprendesse mais sobre esses usuários, para então gerar melhores recomendações ao longo do tempo. Além disso, o arcabouço proposto possui poucos estudos em relação à literatura, o que há de parecido não chega a utilizar bases de dados com grande número de volumes para simular um ambiente real como essa pesquisa propôs em seus objetivos e com êxito utilizou para gerar resultados relevantes para uma ampla discussão. Além disso, o arcabouço proposto pode ser instanciado para outros contextos, sem necessidade de adaptações demoradas para começar a testar.

Como trabalho futuro, pretende-se investigar outras combinações de algoritmos *bandits* para cada nível do algoritmo proposto, ou seja, poderia-se utilizar um algoritmo Bayes-UCB no primeiro nível e o UCB1 no segundo nível e vice-versa. Isso permitiria uma vasta gama de combinações e uma ainda mais resultados para serem avaliados. Por exemplo, pode-se investigar o uso de algoritmos MAB diferentes em cada nível do algoritmo TL-BANDIT. Acredita-se também que uma linha atraente de pesquisa é utilizar as características visuais dos vídeos juntamente das características textuais já existentes para agrupá-los.

Outra proposta futura, seria testar o arcabouço em outras bases de dados de vídeos que pudessem estar disponíveis da mesma forma que o Movielens1M. Além disso, poderia-se também, investigar a performance do arcabouço e dos algoritmos utilizados, visando também melhoras nessa área.

Por fim, uma linha de pesquisa interessante pode ser acumular informação sobre os usuários ao longo do tempo. Assim, usuários que interagirem mais de uma vez com o sistema de recomendação podem ter um perfil mais elaborado e futuras recomendações podem considerar este perfil. Em resumo, apesar do foco deste trabalho ter sido a recomendação em cenários de escassez de informação de preferência dos usuários, o algoritmo proposto poderia também levar em consideração o contexto da recomendação (i.e., o usuário) para recomendar vídeos mais relevantes.

Referências

ADOMAVICIUS, G.; KWON, Y. Toward more diverse recommendations: Item re-ranking methods for recommender systems. In: **Workshop on Information Technologies and Systems**. [S.l.: s.n.], 2009. Citado na página 5.

ADOMAVICIUS, G.; TUZHILIN, A. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. **IEEE Transactions on Knowledge and Data Engineering**, v. 17, n. 6, p. 734–749, 2005. Citado na página 5.

ADOMAVICIUS, G.; TUZHILIN, A. Toward the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. **IEEE transactions on knowledge and data engineering**, IEEE, v. 17, n. 6, p. 734–749, 2005. Citado na página 5.

AGARWAL, D. et al. Online models for content optimization. In: **Advances in Neural Information Processing Systems**. [S.l.: s.n.], 2009. p. 17–24. Citado 2 vezes nas páginas 1 e 2.

AGRAWAL, S.; GOYAL, N. Analysis of thompson sampling for the multi-armed bandit problem. In: **COLT**. [S.l.: s.n.], 2012. p. 39–1. Citado na página 12.

AIZENBERG, N.; KOREN, Y.; SOMEKH, O. Build your own music recommender by modeling internet radio streams. In: ACM. **Proceedings of the 21st international conference on World Wide Web**. [S.l.], 2012. p. 1–10. Citado na página 4.

ARTHUR, D.; VASSILVITSKII, S. k-means++: The advantages of careful seeding. In: SOCIETY FOR INDUSTRIAL AND APPLIED MATHEMATICS. **Proceedings of the eighteenth annual ACM-SIAM symposium on Discrete algorithms**. [S.l.], 2007. p. 1027–1035. Citado na página 33.

ASABERE, N. Y. A survey of personalized television and video recommender systems and techniques. **Int. J. Inf. Commun. Technol. Res**, v. 2, n. 7, p. 602–608, 2012. Citado na página 18.

AUER, P.; CESA-BIANCHI, N.; FISCHER, P. Finite-time analysis of the multiarmed bandit problem. **Machine learning**, Springer, v. 47, n. 2-3, p. 235–256, 2002. Citado 2 vezes nas páginas 12 e 32.

BARRAGÁNS-MARTÍNEZ, A. B. et al. A hybrid content-based and item-based collaborative filtering approach to recommend tv programs enhanced with singular value decomposition. **Information Sciences**, Elsevier, v. 180, n. 22, p. 4290–4311, 2010. Citado na página 4.

BIRD, S.; KLEIN, E.; LOPER, E. **Natural language processing with Python: analyzing text with the natural language toolkit**. [S.l.]: "O'Reilly Media, Inc.", 2009. Citado na página 35.

BLEI, D. M. Probabilistic topic models. **Communications of the ACM**, ACM, v. 55, n. 4, p. 77–84, 2012. Citado na página 13.

BLEI, D. M.; NG, A. Y.; JORDAN, M. I. Latent dirichlet allocation. **Journal of machine Learning research**, v. 3, n. Jan, p. 993–1022, 2003. Citado 2 vezes nas páginas 13 e 34.

BOBADILLA, J. et al. A collaborative filtering approach to mitigate the new user cold start problem. **Knowledge-Based Systems**, Elsevier, v. 26, p. 225–238, 2012. Citado na página 15.

BOUNEFFOUF, D.; BOUZEGHOUB, A.; GANÇARSKI, A. A contextual-bandit algorithm for mobile context-aware recommender system. In: SPRINGER. **Neural Information Processing**. [S.l.], 2012. p. 324–331. Citado na página 15.

BURKE, R. Hybrid recommender systems: Survey and experiments. **User modeling and user-adapted interaction**, v. 12, n. 4, p. 331–370, 2002. Citado na página 6.

CHAPELLE, O.; LI, L. An empirical evaluation of thompson sampling. In: **Advances in neural information processing systems**. [S.l.: s.n.], 2011. p. 2249–2257. Citado na página 31.

CHEN, C. et al. Capturing semantic correlation for item recommendation in tagging systems. In: **AAAI**. [S.l.: s.n.], 2016. p. 108–114. Citado na página 16.

CHOI, S.-M.; KO, S.-K.; HAN, Y.-S. A movie recommendation algorithm based on genre correlations. **Expert Systems with Applications**, Elsevier, v. 39, n. 9, p. 8079–8085, 2012. Citado na página 4.

CONCEIÇÃO, F. L. A. da et al. Metodologia para recomendação de vídeos baseada em descritores de conteúdo visuais e textuais. **Tendências da Pesquisa Brasileira em Ciência da Informação**, v. 9, n. 1, 2016. Citado na página 19.

DAVIDSON, J. et al. The youtube video recommendation system. In: ACM. **Proceedings of the fourth ACM conference on Recommender systems**. [S.l.], 2010. p. 293–296. Citado na página 18.

DAVIES, S.; LEWIS, J. Video recommendation algorithm. 2016. Citado na página 18.

EFRON, B. Computers and the theory of statistics: thinking the unthinkable. **SIAM review**, SIAM, v. 21, n. 4, p. 460–480, 1979. Citado na página 31.

EFRON, B.; TIBSHIRANI, R. J. **An introduction to the bootstrap**. [S.l.]: CRC press, 1994. Citado na página 31.

FELICIO, C. et al. A multi-armed bandit model selection for cold-start user recommendation. 2017. Citado na página 15.

GOLBANDI, N.; KOREN, Y.; LEMPEL, R. On bootstrapping recommender systems. In: ACM. **Proceedings of the 19th ACM international conference on Information and knowledge management**. [S.l.], 2010. p. 1805–1808. Citado na página 14.

GOMEZ-URIBE, C. A.; HUNT, N. The netflix recommender system: Algorithms, business value, and innovation. **ACM Transactions on Management Information Systems (TMIS)**, ACM, v. 6, n. 4, p. 13, 2016. Citado na página 19.

GRIFFITHS, T. L.; STEYVERS, M. Finding scientific topics. **Proceedings of the National academy of Sciences**, National Acad Sciences, v. 101, n. suppl 1, p. 5228–5235, 2004. Citado na página 35.

HARPER, F. M.; KONSTAN, J. A. The movielens datasets: History and context. **ACM Transactions on Interactive Intelligent Systems (TiiS)**, ACM, v. 5, n. 4, p. 19, 2016. Citado 2 vezes nas páginas 18 e 29.

HERNANDO, A. et al. A probabilistic model for recommending to new cold-start non-registered users. **Information Sciences**, Elsevier, v. 376, p. 216–232, 2017. Citado na página 15.

HOFFMAN, M.; BACH, F. R.; BLEI, D. M. Online learning for latent dirichlet allocation. In: **advances in neural information processing systems**. [S.l.: s.n.], 2010. p. 856–864. Citado na página 35.

HOFMANN, K. et al. Online evaluation for information retrieval. **Foundations and Trends® in Information Retrieval**, Now Publishers, Inc., v. 10, n. 1, p. 1–117, 2016. Citado na página 31.

JAIN, A. K. Data clustering: 50 years beyond k-means. **Pattern recognition letters**, Elsevier, v. 31, n. 8, p. 651–666, 2010. Citado na página 33.

KAUFMANN, E.; CAPPÉ, O.; GARIVIER, A. On bayesian upper confidence bounds for bandit problems. In: **AISTATS**. [S.l.: s.n.], 2012. p. 592–600. Citado 2 vezes nas páginas 12 e 32.

KHALID, A. et al. Scalable and practical one-pass clustering algorithm for recommender system. **Intelligent Data Analysis**, IOS Press, v. 21, n. 2, p. 279–310, 2017. Citado na página 17.

KIM, K.-j.; AHN, H. A recommender system using ga k-means clustering in an online shopping market. **Expert systems with applications**, Elsevier, v. 34, n. 2, p. 1200–1209, 2008. Citado na página 20.

KOHRS, A.; MERIALDO, B. Improving collaborative filtering with multimedia indexing techniques to create user-adapting web sites. In: ACM. **Proceedings of the seventh ACM international conference on Multimedia (Part 1)**. [S.l.], 1999. p. 27–36. Citado na página 14.

LI, C.; YANG, C.; JIANG, Q. The research on text clustering based on lda joint model. **Journal of Intelligent & Fuzzy Systems**, IOS Press, v. 32, n. 5, p. 3655–3667, 2017. Citado na página 20.

LI, L. et al. A contextual-bandit approach to personalized news article recommendation. In: ACM. **Proceedings of the 19th international conference on World wide web**. [S.l.], 2010. p. 661–670. Citado 2 vezes nas páginas 1 e 17.

LI, L. et al. Unbiased offline evaluation of contextual-bandit-based news article recommendation algorithms. In: ACM. **Proceedings of the fourth ACM international conference on Web search and data mining**. [S.l.], 2011. p. 297–306. Citado na página 31.

LI, S.; GENTILE, C.; KARATZOGLOU, A. Graph clustering bandits for recommendation. **arXiv preprint arXiv:1605.00596**, 2016. Citado na página 17.

LINDEN, G.; SMITH, B.; YORK, J. Amazon. com recommendations: Item-to-item collaborative filtering. **IEEE Internet computing**, IEEE, v. 7, n. 1, p. 76–80, 2003. Citado na página 5.

MARY, J.; PREUX, P.; NICOL, O. Improving offline evaluation of contextual bandit algorithms via bootstrapping techniques. In: **International Conference on Machine Learning**. [S.l.: s.n.], 2014. p. 172–180. Citado 2 vezes nas páginas 17 e 32.

MILLER, B. N. et al. Movielens unplugged: experiences with an occasionally connected recommender system. In: **IUI**. [s.n.], 2003. p. 263–266. ISBN 1-58113-586-6. Disponível em: <http://doi.acm.org/10.1145/604045.604094>. Citado 2 vezes nas páginas 29 e 41.

MILLER, B. N. et al. Movielens unplugged: experiences with an occasionally connected recommender system. In: ACM. **Proceedings of the 8th international conference on Intelligent user interfaces**. [S.l.], 2003. p. 263–266. Citado na página 18.

MORALES, G. D. F.; GIONIS, A.; LUCCHESE, C. From chatter to headlines: harnessing the real-time web for personalized news recommendation. In: ACM. **Proceedings of the fifth ACM international conference on Web search and data mining**. [S.l.], 2012. p. 153–162. Citado na página 4.

PAIXÃO, C. Z. F. et al. Assessing and improving recommender systems to deal with user cold-start problem. Universidade Federal de Uberlândia, 2017. Citado na página 15.

PANDEY, S.; CHAKRABARTI, D.; AGARWAL, D. Multi-armed bandit problems with dependent arms. In: ACM. **Proceedings of the 24th international conference on Machine learning**. [S.l.], 2007. p. 721–728. Citado 2 vezes nas páginas 17 e 23.

PAZZANI, M. J. A framework for collaborative, content-based and demographic filtering. **Artificial Intelligence Review**, v. 13, n. 5-6, p. 393–408, 1999. ISSN 0269-2821. Disponível em: <http://dx.doi.org/10.1023/A:1006544522159>. Citado na página 16.

RAIGOZA, J.; KARANDE, V. A study and implementation of a movie recommendation system in a cloud-based environment. **International Journal of Grid and High Performance Computing (IJGHPC)**, IGI Global, v. 9, n. 1, p. 25–36, 2017. Citado na página 20.

RASHID, A. M. et al. Getting to know you: learning new user preferences in recommender systems. In: ACM. **Proceedings of the 7th international conference on Intelligent user interfaces**. [S.l.], 2002. p. 127–134. Citado na página 15.

RASHID, A. M.; KARYPIS, G.; RIEDL, J. Learning preferences of new users in recommender systems: an information theoretic approach. **ACM SIGKDD Explorations Newsletter**, ACM, v. 10, n. 2, p. 90–100, 2008. Citado na página 14.

RESNICK, P.; VARIAN, H. R. Recommender systems. **Communications of the ACM**, ACM, v. 40, n. 3, p. 56–58, 1997. Citado na página 4.

ROBBINS, H. Some aspects of the sequential design of experiments. **Bulletin of the American Mathematical Society**, v. 58, n. 5, p. 527–535, 1952. Citado 2 vezes nas páginas 6 e 7.

ROBBINS, H. Some aspects of the sequential design of experiments. In: **Herbert Robbins Selected Papers**. [S.l.]: Springer, 1985. p. 169–177. Nenhuma citação no texto.

SCHAFER, J. B.; KONSTAN, J.; RIEDL, J. Recommender systems in e-commerce. In: ACM. **Proceedings of the 1st ACM conference on Electronic commerce**. [S.l.], 1999. p. 158–166. Citado na página 4.

SCHEIN, A. I. et al. Methods and metrics for cold-start recommendations. In: **ACM SIGIR**. [S.l.: s.n.], 2002. p. 253–260. Citado na página 1.

STREHL, A. et al. Learning from logged implicit exploration data. In: **Advances in Neural Information Processing Systems**. [S.l.: s.n.], 2010. p. 2217–2225. Citado na página 31.

SUTTON, R. S. Generalization in learning: Successful examples using sparse coarse coding. **Advances in neural information processing systems**, p. 1038–1044, 1996. Citado na página 9.

SUTTON, R. S.; BARTO, A. G. **Reinforcement learning: An introduction**. [S.l.: s.n.], 1998. v. 1. Citado na página 6.

SYED-ABDUL, S. et al. Misleading health-related information promoted through video-based social media: anorexia on youtube. **Journal of medical Internet research**, JMIR Publications Inc., Toronto, Canada, v. 15, n. 2, p. e30, 2013. Citado na página 19.

TANG, L. et al. Automatic ad format selection via contextual bandits. In: ACM. **Proceedings of the 22nd ACM international conference on Conference on information & knowledge management**. [S.l.], 2013. p. 1587–1594. Citado na página 1.

TANG, S.; WU, Z.; CHEN, K. Movie recommendation via blstm. In: SPRINGER. **International Conference on Multimedia Modeling**. [S.l.], 2017. p. 269–279. Citado na página 15.

THOMPSON, W. R. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. **Biometrika**, JSTOR, v. 25, n. 3/4, p. 285–294, 1933. Citado 2 vezes nas páginas 12 e 32.

TOKIC, M. Adaptive ε-greedy exploration in reinforcement learning based on value differences. In: SPRINGER. **Annual Conference on Artificial Intelligence**. [S.l.], 2010. p. 203–210. Citado na página 32.

VERMOREL, J.; MOHRI, M. Multi-armed bandit algorithms and empirical evaluation. In: SPRINGER. **European conference on machine learning**. [S.l.], 2005. p. 437–448. Citado na página 1.

WHITE, J. **Bandit algorithms for website optimization**. [S.l.]: "O'Reilly Media, Inc.", 2012. Citado 3 vezes nas páginas 8, 11 e 15.

YANG, B. et al. Online video recommendation based on multimodal fusion and relevance feedback. In: ACM. **Proceedings of the 6th ACM international conference on Image and video retrieval**. [S.l.], 2007. p. 73–80. Citado na página 19.

YANG, C. et al. Social group based video recommendation addressing the cold-start problem. In: SPRINGER-VERLAG NEW YORK, INC. **Proceedings, Part II, of the 20th Pacific-Asia Conference on Advances in Knowledge Discovery and Data Mining-Volume 9652**. [S.l.], 2016. p. 515–527. Citado na página 18.

YI, P. et al. A movie cold-start recommendation method optimized similarity measure. In: IEEE. **Communications and Information Technologies (ISCIT), 2016 16th International Symposium on**. [S.l.], 2016. p. 231–234. Citado na página 19.

YOU, T. et al. Clustering method based on genre interest for cold-start problem in movie recommendation. **Journal of Intelligence and Information Systems**, Korea Inteligent Information System Society, v. 19, n. 1, p. 57–77, 2013. Citado na página 16.

ZHANG, J. et al. Improving video recommendation systems from implicit feedback in the e-marketing environment. In: **Proceedings of the International MultiConference of Engineers and Computer Scientists**. [S.l.: s.n.], 2017. v. 2. Citado na página 20.

ZHOU, L. A survey on contextual multi-armed bandits. **CoRR, abs/1508.03326**, 2015. Citado na página 16.

ZHOU, P. et al. Differentially private online learning for cloud-based video recommendation with multimedia big data in social networks. **IEEE transactions on multimedia**, IEEE, v. 18, n. 6, p. 1217–1229, 2016. Citado na página 19.

ZHOU, R.; KHEMMARAT, S.; GAO, L. The impact of youtube recommendation system on video views. In: ACM. **Proceedings of the 10th ACM SIGCOMM conference on Internet measurement**. [S.l.], 2010. p. 404–410. Citado na página 19.

ZHOU, X. et al. Online video recommendation in sharing community. In: ACM. **Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data**. [S.l.], 2015. p. 1645–1656. Citado na página 18.

ZHOU, X. et al. Enhancing online video recommendation using social user interactions. **The VLDB Journal**, Springer, p. 1–20, 2017. Citado na página 18.