



# UNIVERSIDADE TÉCNICA DE LISBOA INSTITUTO SUPERIOR TÉCNICO

## Recomendação de Programas de Televisão

*André de Trigueiros Pinção Lucas*  
(Licenciado em Engenharia Informática e de Computadores)

Dissertação para obtenção do Grau de Mestre em  
Engenharia Informática e de Computadores

Orientador: Prof. Doutor Bruno Emanuel da Graça Martins

Co-orientador: Prof. Doutor Andreas Miroslaus Wichert

Outubro de 2010







# Sumário

O crescente volume de programas de televisão, disponibilizado pelas operadoras, tem dificultado aos espectadores a tarefa de seleccionar conteúdos relevantes. Neste cenário, os sistemas de recomendação destacam-se como uma possível solução, muito embora a aplicação concreta ao domínio dos programas de televisão permaneça relativamente inexplorada na literatura relacionada. No meu projecto de mestrado estudei o problema da recomendação de programas de televisão, nomeadamente através de abordagens híbridas que combinam filtragem colaborativa e técnicas baseadas na similaridade ao nível dos meta-dados descritivos dos programas. Em particular, abordei o problema através de informação extraída da Web (e.g., redes sociais onde os utilizadores discutem programas de televisão) e usando técnicas de aprendizagem automática de forma a combinar os resultados de diferentes algoritmos de recomendação. Os resultados experimentais confirmam a adequação dos métodos propostos, mostrando que a combinação de diferentes abordagens pode resultar numa melhoria de cerca de 7% em comparação com as melhores técnicas colaborativas.

**Keywords:** Sistemas de Recomendação , Sistemas Híbridos , Sistemas Baseados no Conteúdo , Filtragem Colaborativa , Aprendizagem Automática , Learning-to-Rank , Conjuntos de Dados , Recomendação de Programas de Televisão









# Agradecimentos

Passados largos meses de trabalho com alguns períodos de dedicação quase exclusiva à dissertação de mestrado, gostaria de agradecer ao meu orientador, sempre disponível para me ajudar e dar o seu feedback, mesmo em dias não úteis (fins-de-semana, feriados e férias), e sem o qual teria sido ainda mais duro realizar este longo trabalho.

Além disso, agradeço aos meus familiares, namorada e amigos que acabaram por se envolver neste trabalho devido à minha indisponibilidade para estar com eles em diversos momentos. Pela sua compreensão e apoio, o meu muito obrigado. Por esta razão, aproveito para lhes dedicar este trabalho.



# Conteúdo

<b>Sumário</b>	<b>v</b>
<b>Agradecimentos</b>	<b>ix</b>
<b>1 Introdução</b>	<b>3</b>
1.1 Hipótese e metodologia . . . . .	4
1.2 Resumo das principais contribuições . . . . .	4
1.3 Organização do documento . . . . .	5
<b>2 Conceitos e Trabalho Relacionado</b>	<b>7</b>
2.1 Conceitos . . . . .	7
2.1.1 Avaliação de sistemas de recomendação . . . . .	10
2.2 Trabalho relacionado . . . . .	15
2.2.1 Sistemas de recomendação baseados no conteúdo . . . . .	15
2.2.2 Sistemas de filtragem colaborativa . . . . .	16
2.2.3 Sistemas de recomendação híbridos . . . . .	18
2.2.4 Aplicações no domínio da recomendação de programas de televisão	20
2.3 Software existente e conjuntos de dados . . . . .	21
2.3.1 <i>Frameworks</i> para desenvolvimento de sistemas de recomendação .	21
2.3.2 Conjuntos de dados públicos . . . . .	23
2.4 Sumário e discussão crítica . . . . .	25
<b>3 Técnicas de Recomendação</b>	<b>29</b>
3.1 Técnicas de predição básicas . . . . .	29
3.2 Técnicas de recomendação baseadas no conteúdo . . . . .	30
3.2.1 Métricas de similaridade . . . . .	30
3.2.2 Técnicas implementadas . . . . .	32
3.3 Técnicas de recomendação colaborativas . . . . .	34

3.3.1	Técnicas de recomendação colaborativas <i>user-to-user</i> . . . . .	34
3.3.2	Técnicas de recomendação colaborativas <i>item-to-item</i> . . . . .	36
3.3.3	Outras técnicas de recomendação colaborativas . . . . .	37
3.4	Sumário . . . . .	38
<b>4</b>	<b>Estratégias Híbridas</b>	<b>41</b>
4.1	Ponderação por heurísticas . . . . .	41
4.1.1	Combinação por troca por ordem fixa . . . . .	41
4.1.2	Combinação por troca por ordem variável . . . . .	41
4.1.3	Combinação por ponderação fixa com três conjuntos de técnicas combinadas por troca . . . . .	42
4.1.4	Combinação por ponderação variável . . . . .	42
4.1.5	Combinação por ponderação fixa . . . . .	43
4.2	Sumário . . . . .	45
<b>5</b>	<b>Avaliação Experimental</b>	<b>47</b>
5.1	Metodologia de avaliação . . . . .	47
5.2	Conjunto de dados de validação . . . . .	48
5.3	Métricas de avaliação consideradas . . . . .	54
5.4	Avaliação da distância entre previsões e valores reais . . . . .	56
5.4.1	Resultados . . . . .	56
5.4.2	Discussão . . . . .	56
5.5	Avaliação das listas de itens recomendadas . . . . .	59
5.5.1	Resultados . . . . .	60
5.5.2	Discussão . . . . .	61
5.6	Conclusões . . . . .	62
<b>6</b>	<b>Conclusões</b>	<b>65</b>
6.1	Sumário das contribuições . . . . .	66
6.2	Trabalho futuro . . . . .	66
	<b>Bibliografia</b>	<b>68</b>





# Lista de Tabelas

2.1	Matriz de contingência com recomendações de itens a utilizadores. . . . .	11
2.2	<i>Frameworks</i> para desenvolvimento de sistemas de recomendação. . . . .	23
2.3	Conjuntos de dados para sistemas de recomendação. . . . .	26
5.4	Caracterização das fontes de dados. . . . .	49
5.5	Caracterização do conjunto de dados TV Programs Dataset . . . . .	53
5.6	Exemplo de uma matriz utilizadores/itens. . . . .	54
5.7	Valores das métricas para listas ordenadas para o caso de exemplo. . . . .	54
5.8	Valores das métricas para listas ordenadas para o caso de exemplo. . . . .	55







# Lista de Figuras

2.1	Topologia da rede <i>Bayesiana</i> de Zhang & Koren (2007) . . . . .	17
2.2	Topologia da máquina de Boltzmann de Truyen <i>et al.</i> (2009). . . . .	18
2.3	Topologia da máquina de Boltzmann de Salakhutdinov <i>et al.</i> (2007). . . . .	18
2.4	Topologia da Rede <i>Bayesiana</i> de de Campos <i>et al.</i> (2006) . . . . .	20
2.5	Distribuição das classificações nos conjuntos de dados da Netflix e BookCrossing. . . . .	24
2.6	Distribuição das classificações nos conjuntos de dados do MovieLens e Jester. . . . .	25
3.7	Diagrama da metodologia baseada no conteúdo. . . . .	34
4.8	Diagrama da metodologia de combinação por troca por ordem fixa. . . . .	41
4.9	Diagrama da metodologia de combinação por troca por ordem variável. . . . .	42
4.10	Diagrama da metodologia de combinação por ponderação fixa com três conjuntos de técnicas combinadas por troca. . . . .	43
4.11	Diagrama da metodologia de combinação por ponderação com pesos variáveis. . . . .	43
4.12	Framework do processo de <i>learning-to-rank</i> . . . . .	44
5.13	Construção do conjunto de dados para o domínio dos programas de televisão. . . . .	49
5.14	Distribuição das classificações dos utilizadores e itens antes da filtragem. . . . .	51
5.15	Distribuição das classificações dos utilizadores e itens após realização da filtragem. . . . .	51
5.16	Proporção de cada classificação no conjunto de dados. . . . .	52
5.17	Valores mínimos e máximos das classificações dos utilizadores. . . . .	52
5.18	Comparação entre a percentagem de utilizadores que tem cada desvio padrão com o número médio de classificações desses utilizadores. . . . .	53
5.19	Resumo dos resultados obtidos de acordo com a avaliação da distância entre predições e valores reais, bem como da cobertura de cada técnica de predição. . . . .	57

5.20	Resumo dos resultados obtidos de acordo com a avaliação das listas de 5 itens recomendadas. . . . .	60
5.21	Variação do valor de nDRCG consoante o número de itens considerados nas listas de recomendação. . . . .	61



# Capítulo 1

## Introdução

O actual volume de programas de televisão, disponibilizado pelas operadoras, dificulta aos espectadores a tarefa de seleccionar conteúdos televisivos relevantes. As ferramentas tradicionais de procura de conteúdos televisivos, i.e. os guias de programação impressos ou electrónicos, não atendem de forma eficiente às actuais necessidades de informação dos espectadores. Estes guias apresentam listas extensas de programas de televisão, tendo-se que os espectadores despendem muito tempo na busca de informações sobre programas potencialmente interessantes (Hsu *et al.*, 2007). O problema tem-se vindo a agravar, dado o aumento da quantidade de conteúdos associado à popularização da Televisão Digital (terrestre, satélite e cabo).

Neste cenário, os sistemas de recomendação destacam-se como uma possível solução. Estes sistemas auxiliam os utilizadores na busca de conteúdos relevantes de acordo com os seus interesses, tal como definidos explicitamente pelos próprios e/ou aferidos implicitamente pelo sistema através do comportamento dos utilizadores no passado. Desde meados da década de 90, os sistemas de recomendação têm vindo a ganhar popularidade, tendo sido aplicados em diversos domínios (Adomavicius & Tuzhilin, 2005). Estudos na área comprovam que a satisfação dos utilizadores com os sistemas de recomendação é muito elevada e que, em alguns sistemas, o número de recomendações efectivamente úteis (i.e., itens relevantes recomendados que eram desconhecidos pelos utilizadores) ultrapassa os 40% (Swearingen & Sinha, 2001).

Apesar da crescente popularidade destes sistemas, a aplicação concreta ao domínio dos programas de televisão permanece relativamente inexplorada na literatura relacionada. No meu trabalho de mestrado, estudei o problema da recomendação de programas de televisão, através de abordagens híbridas que combinam a filtragem colaborativa com medidas de similaridade ao nível dos conteúdos dos programas. Em particular, abordei o problema através de informação extraída da Web (e.g., informação extraída de redes sociais onde os utilizadores discutem os seus programas de televisão preferidos) e usando técnicas de aprendizagem automática por forma a combinar os resultados de diferentes algoritmos de recomendação, aprendendo efectivamente a ordenar os programas de televisão a serem recomendados de acordo com a sua relevância para com os interesses dos utilizadores.

## 1.1 Hipótese e metodologia

No meu trabalho de mestrado abordei o problema da construção de um sistema de recomendação para o domínio específico dos programas de televisão, procurando responder às seguintes duas questões:

1. Quais as técnicas da área dos sistemas de recomendação que melhor se adequam ao domínio concreto das recomendações de programas de televisão?
2. A combinação de várias técnicas de recomendação, obtida através de métodos recentemente propostos na área de *learning-to-rank*, resulta numa melhoria efectiva na qualidade das recomendações efectuadas?

Para responder a estas questões, desenvolvi um protótipo de um sistema de recomendação implementando técnicas recentes para a geração das recomendações. Especificamente, explorei o uso de técnicas baseadas no conteúdo, técnicas de filtragem colaborativa, e ainda a combinação de ambas as técnicas para obter os benefícios dos sistemas híbridos.

Para comparar as diferentes técnicas estudadas, comecei por recolher e caracterizar estatisticamente uma colecção de dados de teste, dada a inexistência de uma colecção de dados específica para o domínio dos programas de televisão, pelo que esta colecção constitui por si só um dos contributos importantes deste trabalho. A colecção de dados foi construída com base em informação publicamente disponibilizada na Web, através das seguintes redes sociais onde os utilizadores classificam programas de televisão: TV.com<sup>1</sup>, IMDb<sup>2</sup> e Living Social<sup>3</sup>. Através da tecnologia XQuery (que é uma linguagem de consulta, com alguns recursos de programação, desenhada para realizar consultas em colecções de dados XML e é semanticamente similar ao SQL (Chamberlin, 2003)), foi possível extrair os dados destes sites, estruturá-los e armazená-los para serem usados para o desenvolvimento e teste de diferentes abordagens.

## 1.2 Resumo das principais contribuições

As principais contribuições apresentadas neste trabalho são:

- Criação de uma colecção de testes específica para o domínio da recomendação de programas de televisão.
- Avaliação de diversas técnicas de recomendação no contexto da recomendação de programas de televisão, bem como através de estratégias híbridas que combinam diferentes tipos de técnicas.
- Proposta e avaliação de várias estratégias híbridas que combinam diferentes tipos de técnicas, incluindo algoritmos de aprendizagem automática que visam melhorar a precisão dos resultados.

---

<sup>1</sup><http://www.tv.com/>

<sup>2</sup><http://www.imdb.com/>

<sup>3</sup><http://tv.livingsocial.com/>

- Apresentação de novas métricas de avaliação que consideram o desvio padrão das classificações de cada utilizador.

### **1.3 Organização do documento**

Os restantes conteúdos desta dissertação estão organizados da seguinte forma. O Capítulo 2 apresenta os conceitos fundamentais na área dos sistemas de recomendação e analisa os principais trabalhos relacionados. O Capítulo 3 descreve as técnicas de predição básicas, baseadas na análise dos conteúdos e na filtragem colaborativa, implementadas neste trabalho. No Capítulo 4 são descritas as diversas abordagens híbridas que foram testadas. No Capítulo 5 é apresentada a metodologia de avaliação, assim como os resultados obtidos e a sua respectiva discussão. Finalmente, o Capítulo 6 sumariza as principais conclusões do trabalho e discute possíveis caminhos para trabalho futuro.





## Capítulo 2

# Conceitos e Trabalho Relacionado

Este capítulo apresenta os principais conceitos teóricos associados aos sistemas de recomendação. São ainda descritas, de forma sucinta, algumas das técnicas e algoritmos mais usados no âmbito desta área de estudo, incluindo-se as técnicas mais usadas na validação de sistemas de recomendação.

Posteriormente, são também apresentados vários trabalhos relacionados com o domínio dos sistemas de recomendação. Particular destaque é dado aos sistemas que foram aplicados no contexto da televisão interactiva.

### 2.1 Conceitos

De uma forma genérica, os sistemas de recomendação são sistemas de informação capazes de recomendar itens potencialmente interessantes aos utilizadores, usando técnicas estatísticas ou da área da inteligência artificial, por forma a inferir o interesse dos utilizadores em itens com base nos padrões de comportamento dos mesmos e na similaridade entre os itens e/ou entre os utilizadores (Adomavicius & Tuzhilin, 2005).

Existem diferentes abordagens para a construção de sistemas de recomendação, as quais têm sido implementadas em diferentes domínios. Para cada contexto específico existem técnicas mais apropriadas, dado que todas apresentam vantagens e desvantagens. Deste modo, para cada sistema de recomendação e dependendo das suas características, deve-se deliberar qual a abordagem mais adequada para a sua concretização.

De acordo com o estudo de Adomavicius & Tuzhilin (2005), os sistemas de recomendação são tipicamente classificados segundo as seguintes categorias:

- **Sistemas baseados no conteúdo** – Nestes sistemas são recomendados itens com características e conteúdos similares aos dos itens em que o utilizador manifestou interesse no passado. A similaridade entre os itens é medida, na maior parte dos casos, através da análise de informação textual ou através da regularidades nos atributos associados aos itens (Su & Khoshgoftaar, 2009). Estes sistemas utilizam habitualmente técnicas da área de *information retrieval* (Manning *et al.*, 2008).
- **Sistemas de filtragem colaborativa** – Com base nas avaliações dos utilizadores

aos itens, estes sistemas inferem-se semelhanças entre utilizadores ou entre itens para depois recomendarem itens potencialmente interessantes aos utilizadores Herlocker *et al.* (2000). Este tipo de sistemas são os mais usados, apesar de necessitarem de um vasto conjunto de utilizadores e uma elevada taxa de participação destes para poderem fornecer recomendações úteis. Este tipo de sistemas subdivide-se em:

- **Filtragem colaborativa *user-to-user*** – Com base nas preferências dos utilizadores, são recomendados itens da preferência de utilizadores com preferências similares (Wang *et al.*, 2006).
- **Filtragem colaborativa *item-to-item*** – Com base nas preferências dos utilizadores, infere-se a similaridade dos itens e sugerem-se itens similares àqueles que os utilizadores preferiram no passado (Wang *et al.*, 2006).
- **Sistemas híbridos** – Os sistemas híbridos procuram tirar partido das mais-valias de ambos os métodos anteriores (i.e. baseados no conteúdo e filtragem colaborativa), combinando abordagens dos dois tipos de sistemas com vista a melhorar o desempenho (Su & Khoshgoftaar, 2009) e a colmatar algumas das lacunas das abordagens clássicas (Adomavicius & Tuzhilin, 2005).

Alguns autores referem-se ainda a outros tipos de sistemas de recomendação, tais como sistemas demográficos, sistemas baseados em utilidade e sistemas baseados em conhecimento (Burke, 2002). Contudo, estes podem ser considerados especializações dos casos anteriores.

Os sistemas de recomendação são também classificados de acordo com a técnica usada para analisar os dados que estão na base das recomendações. Existem dois grupos genéricos de tipos de técnicas para estes sistemas:

- **Técnicas *memory-based*** – Também conhecidas como técnicas *heuristic-based* (Adomavicius & Tuzhilin, 2005), assentam em heurísticas e fazem previsões baseadas na colecção completa de itens previamente classificados pelos utilizadores, de modo a calcular a similaridade entre utilizadores ou itens e, a partir daí, fazer recomendações (Su & Khoshgoftaar, 2009).
- **Técnicas *model-based*** – Usam a colecção de classificações dos utilizadores para aprender um modelo de forma *offline*, que será posteriormente usado para fazer as recomendações *online* (Adomavicius & Tuzhilin, 2005). Modelos probabilísticos tais como *Latent Semantic Analysis* (Hofmann, 2004), as redes Bayesianas (de Campos *et al.*, 2006) ou as máquinas de Boltzmann (Salakhutdinov *et al.*, 2007) são frequentemente usados neste contexto.

As classificações dos itens por parte dos utilizadores podem ser explícitas (i.e., indicadas directamente pelos utilizadores) ou implícitas (i.e., o sistema infere a classificação do utilizador aos itens, monitorizando a sua actividade). Estas classificações permitem calcular a similaridade entre itens e/ou utilizadores (Su & Khoshgoftaar, 2009). Existem diversas formas de calcular estas similaridades, muito

embora a correlação de Pearson e a medida do cosseno sejam as mais utilizadas (na Secção 3.3.1 serão descritas estas e outras métricas de similaridade).

Os diferentes tipos de sistemas de recomendação estão associados a problemas específicos na sua construção. Os problemas principais são:

- **Incompletude dos itens** – Este é um problema específico dos sistemas baseados no conteúdo que consiste na existência de informação insuficiente nos itens por forma a realizar o cálculo da similaridade entre estes (Pazzani & Billsus, 2007).
- **Itens sinónimos** – Este problema refere-se à existência de itens iguais ou muito semelhantes, mas com atributos diferentes. Deste modo, estes itens serão tratados como distintos, reduzindo-se assim o desempenho das recomendações do sistema (Su & Khoshgoftaar, 2009). Este problema manifesta-se sobretudo nos sistemas de recomendação baseados no conteúdo.
- **Arranque a frio** – Este problema ocorre quando um novo utilizador ou item entra no sistema e existe pouca informação associada a ele. O problema manifesta-se na dificuldade de encontrar itens similares a um dado item na filtragem colaborativa *item-to-item*, ou na dificuldade em encontrar utilizadores similares a um novo utilizador, no caso da filtragem colaborativa *user-to-user*. Nos sistemas baseados no conteúdo não se consegue recomendar itens a um novo utilizador enquanto este não indicar itens da sua preferência. Contudo, os novos itens não são sujeitos a este problema nos sistemas baseados no conteúdo, uma vez que desde a inclusão de um novo item o sistema já contém toda a informação que necessita sobre ele para recomendá-lo, porque a informação necessária é o próprio conteúdo do item (Su & Khoshgoftaar, 2009).
- **Ovelhas cinzentas** – Este problema relaciona-se com a existência de alguns utilizadores cujas classificações são inconsistentes, dificultando assim a correlação com os restantes utilizadores (Burke, 2002). Este problema manifesta-se nos sistemas de filtragem colaborativa.
- **Transitividade da vizinhança** – Este problema ocorre quando utilizadores com gostos semelhantes não são identificados pelo algoritmo, devido ao facto destes utilizadores não terem classificado os mesmos itens (Su & Khoshgoftaar, 2009). Este problema ocorre apenas na filtragem colaborativa.
- **Cobertura reduzida** – Este problema está relacionado com o arranque a frio e consiste na existência de um número de classificações muito baixo comparativamente com o número total de itens. Como este problema se relaciona apenas com as classificações dos itens, só se aplica à filtragem colaborativa, dado que os itens em sistemas baseados no conteúdo não necessitam de ter classificações para serem recomendados (Su & Khoshgoftaar, 2009).
- **Classificações falsas** – Este problema ocorre quando as classificações dadas pelos utilizadores não correspondem às suas preferências reais. Este facto pode-se dever a comportamentos não éticos (i.e., ataques) que visam classificar positivamente os próprios itens e negativamente os outros, ou a informações incorrectas fornecidas pelos utilizadores com o intuito de preservarem a sua

privacidade. Os sistemas baseados no conteúdo não são afectados por estes problemas porque as recomendações a um utilizador não são influenciadas pelas classificações dos restantes utilizadores (Su & Khoshgoftaar, 2009).

- **Escalabilidade** – Um sistema de recomendação necessita de muitos utilizadores e/ou itens para apresentar melhores recomendações. Por outro lado, estes sistemas podem tornar-se pouco eficientes aquando da presença de uma quantidade de dados elevada (Su & Khoshgoftaar, 2009), principalmente se a maioria do processamento for realizado *online*. O problema da escalabilidade é assim mais grave nas técnicas *memory-based*.

### 2.1.1 Avaliação de sistemas de recomendação

A validação de sistemas de recomendação depende do contexto em que estes são aplicados, bem como dos seus objectivos. Shani & Gunawardana (2009) apresentaram uma lista de quinze propriedades dos sistemas de recomendação que devem ser levadas em consideração aquando da avaliação. Estas propriedades são: preferências dos utilizadores, precisão das predições, cobertura do espaço dos itens, cobertura do espaço dos utilizadores, crença dos utilizadores nas recomendações, confiança nas recomendações, novidade nos itens recomendados, suporte de *serendipity* (i.e. a capacidade dos utilizadores fazerem descobertas interessantes ao acaso e de serem surpreendidos pelas recomendações), diversidade nas recomendações, utilidade das recomendações, risco associado às más recomendações, robustez, privacidade, capacidade de adaptação e escalabilidade. Estas propriedades podem ter maior ou menor relevância consoante os objectivos do sistema, podendo incorrer em *trade-offs*, ou seja, garantir totalmente uma propriedade pode implicar abdicar da parte ou da totalidade de outra. Nesta secção serão descritos métodos para avaliar sistemas de recomendação, levando em conto as principais propriedades referidas.

#### 2.1.1.1 Avaliação de sistemas com classificações binárias

A grande maioria da literatura relacionada descreve experiências de validação usando conjuntos de dados contendo classificações dadas por utilizadores a itens, de forma a aferir da qualidade e desempenho das diferentes técnicas. O uso de classificações binárias é muito comum em sistemas de recomendação (i.e., é atribuída igual relevância a todas as recomendações e os utilizadores só podem indicar se gostam ou não de um item), sendo que, mesmo que as classificações sejam *n*-árias, estas podem ser convertidas para binárias. No que diz respeito à qualidade das recomendações, existem várias métricas de desempenho que podem ser calculadas com base numa matriz de contingência entre itens classificados pelos utilizadores e itens recomendados pelo sistema – Veja-se a Tabela 2.1 (Shani & Gunawardana, 2009).

Algumas destas métricas são descritas de seguida, tendo-se que as mesmas se encontram detalhadas nos estudos de Shani & Gunawardana (2009) e de Herlocker *et al.* (2004):

- **Precisão** – Indica a percentagem de recomendações acertadas. Esta medida permite aferir a precisão do processo de predição das preferências do utilizador.

	Recomendado	Não recomendado
Classificado como relevante	Verdadeiro-Positivo (vp)	Falso-Negativo (fn)
Classificado como não relevante	Falso-Positivo (fp)	Verdadeiro-Negativo (vn)

**Tabela 2.1:** Matriz de contingência com recomendações de itens a utilizadores.

$$Precisao = \frac{\#vp}{\#vp + \#fp} \quad (2.1)$$

- **Cobertura ou rácio dos verdadeiros-positivos** – Indica quantos dos itens relevantes para o utilizador foram recomendados pelo sistema, permitindo perceber a utilidade das recomendações apresentadas do utilizador.

$$Cobertura = RVP = \frac{\#vp}{\#vp + \#fn} \quad (2.2)$$

- **Rácio dos verdadeiros-negativos** – Indica a percentagem de itens não recomendados que não são da preferência do utilizador, tendo por isso sido correctamente avaliados.

$$RVN = \frac{\#vn}{\#vn + \#fp} \quad (2.3)$$

- **Rácio dos falsos-positivos** – Indica a percentagem de itens não relevantes aos utilizadores que foram recomendados pelo sistema.

$$RFP = \frac{\#fp}{\#fp + \#vn} \quad (2.4)$$

- **Exactidão** – Indica a percentagem dos itens que são bem classificados, permitindo apresentar uma noção geral da exactidão do sistema.

$$Exactidao = \frac{\#vp + \#vn}{\#vp + \#vn + \#fp + \#fn} \quad (2.5)$$

- **Medida-F** – Corresponde à média harmónica da precisão e da cobertura, representando ambos os aspectos numa única métrica.

$$F = 2 \cdot \frac{Precisao \cdot Cobertura}{Precisao + Cobertura} \quad (2.6)$$

Além das métricas atrás referidas, as curvas ROC (*receiver operating characteristic*) são também um método muito utilizado para analisar o desempenho de um sistema. Estas curvas são representadas num gráfico de duas dimensões, correspondendo o eixo do  $y$  à cobertura (rácio dos verdadeiros-positivos) e o eixo do  $x$  ao rácio dos falsos-positivos. É também possível obter um número que sintetiza o desempenho do sistema, graças ao cálculo da área por baixo da curva, no qual se obtém um valor entre 0 e 1, sendo que

quanto maior o valor, melhor será o sistema. Esta medida é conhecida por AUC (*area under ROC curve*) (Herlocker *et al.*, 2004).

As métricas acima, apesar de não aferirem directamente a satisfação dos utilizadores, são bons indicadores para o facto das recomendações estarem ou não de acordo com as preferências dos utilizadores. Muito embora estas métricas de qualidade sejam populares, elas têm algumas limitações quando usadas em experiências *offline*. Uma das mais importantes relaciona-se com o facto dos utilizadores típicos apenas fornecerem avaliações explícitas para um conjunto muito reduzido de itens, normalmente aqueles que são da sua preferência. Desta forma, os resultados da avaliação apenas indicam o quão preciso é um sistema a retornar itens que os utilizadores já conhecem, tipicamente itens da sua preferência, e não a capacidade do sistema sugerir itens aleatórios que sejam do interesse dos utilizadores, pelo que não aferem algumas das propriedades referidas como a novidade e o suporte de *serendipity*. As avaliações *online*, com potenciais utilizadores do sistema, podem oferecer melhores indicadores para a utilidade de um sistema, sendo no entanto mais dispendiosas e difíceis de realizar.

Outros factores relacionados com a satisfação dos utilizadores relacionam-se com os graus de diversidade e novidade presentes nas recomendações, assim como com a cobertura do conjunto de itens pelas recomendações geradas pelo sistema. Importa no entanto referir que, se por um lado é importante apresentar recomendações que o utilizador não esperava e com itens que ele desconhecia, a excessiva diversidade e novidade nas recomendações poderá afectar a satisfação dos utilizadores (Shani & Gunawardana, 2009).

O índice de Gini permite medir a diversidade presente nas recomendações (Shani & Gunawardana, 2009). Sendo  $i$  um item, e  $p(i)$  a proporção de escolhas desse item pelos utilizadores aquando das recomendações, o índice Gini é dado por:

$$G = \frac{1}{n-1} \sum_{j=1}^n (2j - n - 1)p(i_j) \quad (2.7)$$

onde  $i_1, \dots, i_n$  é a lista de itens ordenados de forma crescente de acordo com o valor  $p(i)$ . Nos casos limite, temos que:

$$G = \begin{cases} 0 & \text{se todos os itens são escolhidos de igual forma por todos os utilizadores} \\ 1 & \text{se um só item é escolhido sempre por todos os utilizadores} \end{cases}$$

A entropia de Shannon permite também medir a diversidade nas recomendações (Shani & Gunawardana, 2009), sendo definida de acordo com a fórmula:

$$H = - \sum_{i=1}^n p(i) \log p(i) \quad (2.8)$$

Nos casos limite, temos que:

$$H = \begin{cases} \log n & \text{se } n \text{ itens são sempre escolhidos ou recomendados de igual forma} \\ 0 & \text{se um só item é sempre escolhido ou recomendado} \end{cases}$$

A novidade nas recomendações pode ser medida de acordo com o número de itens recomendados que o utilizador não conhecia (e.g., os itens previamente não classificados pelo utilizador). Finalmente, a cobertura do espaço de itens pelas recomendações geradas pelo sistema pode ser medida através do rácio dos itens recomendados aos utilizadores face ao número de itens totais.

### 2.1.1.2 Avaliação de sistemas com classificações $n$ -árias

Além das métricas para classificações binárias, existem também métricas para classificações  $n$ -árias que permitem aferir o quão próximo estiveram as classificações inferidas pelo sistema, das classificações atribuídas pelos utilizadores. Uma das métricas mais utilizadas é o Erro Médio Absoluto (EMA) que indica o erro médio associado às recomendações (Equação 2.9) (Herlocker *et al.*, 2004).

$$EMA = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i| \quad (2.9)$$

O Erro Médio Quadrático (EMQ) e Erro Médio Absoluto Percentual (EMAP) são outras variantes do EMA frequentemente utilizadas (Herlocker *et al.*, 2004), que são calculadas pelas Equações 2.10 e 2.11 respectivamente.

$$EMQ = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2 \quad (2.10)$$

$$EMAP = \frac{1}{n} \sum_{i=1}^n \left| \frac{y_i - \hat{y}_i}{y_i} \right| \quad (2.11)$$

### 2.1.1.3 Avaliação de sistemas com classificações $n$ -árias em listas ordenadas

O objectivo principal de um sistema de recomendação é recomendar um conjunto de itens potencialmente interessantes. Como tal, não é essencial que o sistema consiga prever bem todas as classificações, sendo mais interessante inferir quais os itens mais relevantes para um dado utilizador. Neste contexto, a métrica CG (*Comulative Gain* ou, em português, ganho acumulado) e as métricas relacionadas permitem avaliar a qualidade das listas de itens apresentadas aos utilizadores, de acordo com a preferência demonstrada por estes. O CG é uma métrica que consiste no acumular do valor de relevância dos itens recomendados (i.e., o valor real da classificação atribuída pelo utilizador), ou seja, quanto maior for esse ganho acumulado, mais interessante será a lista de itens para o utilizador (Jarvelin & Kekalainen, 2002). Sendo  $p$  o número de itens considerados na lista de itens recomendados, e  $rel_i$  a classificação real (i.e. a relevância)

dada pelo utilizador ao item da posição  $i$  da lista ordenada apresentada, CG é dado por:

$$CG_p = \sum_{i=1}^p rel_i \quad (2.12)$$

Este ganho acumulado ignora a ordem dos  $p$  primeiros itens, considerando essa sub-lista como uma colecção de valores não ordenados, pelo que o primeiro item da lista é tão valorizado como o último. De modo a considerar a ordem desta sub-lista, foi proposta a métrica DCG (ganho acumulado com desconto) que reduz a relevância de cada item de acordo com uma função logarítmica, valorizando mais os primeiros itens e valorizando menos os últimos, de acordo com a seguinte equação:

$$DCG_p = rel_1 + \sum_{i=2}^p \frac{rel_i}{\log_2 i} \quad (2.13)$$

O DCG' (ganho acumulado com desconto com maior ênfase na ordem) é uma formulação alternativa do DCG que valoriza ainda mais a ordem e atribui um maior peso a cada classificação (Manning *et al.*, 2008), tal como apresentado de seguida:

$$DCG'_p = \sum_{i=1}^p \frac{2^{rel_i} - 1}{\log_2(1 + i)} \quad (2.14)$$

Contudo, o valor destes ganhos é o resultado de um somatório, pelo que um valor isolado de ganho não permite aferir o quão boa é uma lista de itens. Para se avaliar devidamente uma lista recomendada, é necessário comparar o seu ganho com o ganho ideal (i.e., o valor de ganho acumulado da lista de  $p$  itens preferidos do utilizador). É neste sentido que surgem outras variantes destas métricas, que consistem no rácio entre o ganho acumulado e o ganho ideal, resultando num valor entre 0.0 e 1.0 (Jarvelin & Kekalainen, 2002). Como tal, sendo o numerador o valor resultante da Equação 2.12 e sendo o denominador o valor ideal de cada um destes ganhos (i.e., o valor acumulado da lista ordenada de acordo com a classificação real), o nCG (ganho acumulado normalizado) é dado por:

$$nCG_p = \frac{CG_p}{ICG_p} \quad (2.15)$$

Os outros dois ganhos (nDCG e nDCG') são determinados de forma análoga, tal como indicados de seguida:

$$nDCG_p = \frac{DCG_p}{IDCG_p} \quad (2.16)$$

$$nDCG'_p = \frac{DCG'_p}{IDCG'_p} \quad (2.17)$$



## 2.2 Trabalho relacionado

Nesta secção são apresentadas as técnicas e algoritmos utilizados em diversos sistemas de recomendação propostos na literatura relacionada, dando-se especial ênfase a sistemas especificamente desenvolvidos para a recomendação de programas de televisão.

### 2.2.1 Sistemas de recomendação baseados no conteúdo

Autores como Pazzani & Billsus (2007) apresentaram revisões detalhadas sobre sistemas de recomendação baseados na similaridade entre os conteúdos descritivos dos itens (vulgo sistemas baseados no conteúdo) desenvolvidos no contexto de diversos domínios de aplicação. De acordo com estes estudos, os sistemas de recomendação baseados no conteúdo assentam sobretudo em perfis de utilizadores. Normalmente, estes perfis assumem um de dois tipos:

- **Um histórico das interacções do utilizador com o sistema**, que pode incluir o conjunto de itens a que um utilizador acedeu no passado (Fu *et al.*, 2000), ou os itens para os quais o utilizador forneceu uma revisão explícita (Mooney & Roy, 2000). Estes históricos podem ser usados directamente (*memory-based*), por exemplo através de uma abordagem estilo vizinhos mais próximos aplicado aos itens (Billsus *et al.*, 2000) usando métricas de similaridade como a correlação de Pearson (O'Connor & Herlocker, 2001) ou a medida do cosseno (Pazzani & Billsus, 2007), ou indirectamente através da construção de um modelo de classificação com técnicas de aprendizagem automática supervisionada (*model-based*), tais como os classificadores Naive Bayes (Pazzani & Billsus, 1997). Autores como Kotsiantis (2007) fornecem uma revisão detalhada dos métodos de aprendizagem automática que podem ser usados neste contexto.
- **Um modelo das preferências dos utilizadores**, i.e. uma descrição dos tipos de itens em que o utilizador está interessado. Existem várias alternativas possíveis para esta descrição, mas uma das mais comuns consiste numa função que, para cada item, retorna uma estimativa da probabilidade do utilizador estar interessado nele (Pazzani & Billsus, 2007). Esta função pode ser usada para retornar os  $n$  itens potencialmente mais interessantes para o utilizador. Estudos anteriores usaram modelos baseados em regras (Adomavicius & Tuzhilin, 2001) (e.g., o perfil do utilizador corresponde a um conjunto de regras sobre os atributos dos itens, que permite discriminar os itens do seu interesse) ou baseados em técnicas da área de *information retrieval* (Balabanovic & Shoham, 1997) (e.g., o perfil do utilizador é um vector de palavras-chave que expressa o seu interesse, sendo esse vector usado para retornar itens relevantes através da aplicação da medida do cosseno para com os vectores dos itens). Para ponderação das palavras extraídas dos itens nas actividades de *information retrieval* é habitualmente usado o peso TF-IDF (Manning *et al.*, 2008), que é uma estimativa da importância de uma palavra num documento ou numa colecção de documentos. Estes modelos de preferências seguem uma abordagem *model-based*, uma vez que as previsões são realizadas com base em modelos previamente construídos.

### 2.2.2 Sistemas de filtragem colaborativa

Os sistemas de recomendação baseados em filtragem colaborativa são os mais usados, apesar de necessitarem de um vasto conjunto de utilizadores e uma elevada taxa de participação destes para poderem fornecer recomendações úteis. Alguns estudos, tais como os de Adomavicius & Tuzhilin (2005) e de Resnick & Varian (1997), analisaram diversos sistemas de filtragem colaborativa.

Na filtragem colaborativa *memory-based* são habitualmente utilizadas técnicas de *clustering* ou de classificação baseada nos  $k$  vizinhos mais próximos (Adomavicius & Tuzhilin, 2005). Nestes sistemas, costumam ser utilizadas matrizes utilizador-item (i.e., os utilizadores  $U$  são representados nas linhas e os itens  $I$  nas colunas, ou vice-versa, sendo que a célula da linha  $n$  e coluna  $m$  indica a classificação do utilizador  $u_n$  ao item  $i_m$ ) à qual é aplicada uma função de similaridade entre utilizadores ou itens. Nas funções de similaridade vectoriais (tal como a função cosseno), os vectores correspondem a linhas ou colunas da matriz referida. Ao nível dos sistemas *model-based*, estes costumam ser implementados usando redes Bayesianas, *clustering* ou redes neuronais. A aprendizagem dos modelos pode ser feita através de algoritmos de aprendizagem automática (e.g., *expectation-maximization* ou regressão linear) (Witten & Frank, 1999).

A abordagem *item-to-item* é mais eficiente e eficaz que a *user-to-user* (Adomavicius & Tuzhilin, 2005), tendo sido por isso utilizada no sistema de recomendação da loja *online* Amazon.com para suportar a sua vasta quantidade de artigos e de utilizadores (Linden *et al.*, 2003). No entanto, os sistemas *user-to-user* revelam uma boa precisão quando a maioria dos utilizadores tem uma forte correlação de interesses com outros utilizadores. Esta abordagem foi seguida pelo GroupLens, utilizando como função de similaridade a correlação de Pearson (Resnick *et al.*, 1994).

No caso dos utilizadores terem diversas preferências não relacionadas entre elas (e.g., gostar de futebol e de economia não têm necessariamente uma relação) estes sistemas terão dificuldade em recomendar itens, dado que dificilmente encontrarão utilizadores cujo conjunto de interesses seja igual. Se a pesquisa fosse feita utilizando apenas um subconjunto de interesses do utilizador, seria mais fácil encontrar similaridades entre utilizadores (i.e., um utilizador que gosta de futebol e economia teria similaridade, em parte, com outros utilizadores que gostam de economia, mas também com outros que gostam de futebol). Um exemplo da utilização desta técnica é o SiteSeer, um sistema de recomendação para páginas Web que tira partido da lista de favoritos dos utilizadores e da sua respectiva estrutura (i.e., da forma como estão organizados os links através de pastas) para recomendar outras páginas (Boardman *et al.*, 2003). Este sistema relaciona cada pasta dos favoritos do utilizador com as pastas dos outros utilizadores recomendando páginas Web presentes em pastas similares à pasta do utilizador alvo. As recomendações visam satisfazer os interesses apresentados por uma dada pasta, e não pelos favoritos do utilizador no seu todo. Como tal, pode-se apelidar este sistema como filtragem colaborativa *folder-to-folder*.

Para relacionar os itens e/ou os utilizadores, pode-se recorrer também a técnicas baseadas em modelos, nomeadamente utilizando redes Bayesianas e máquinas de Boltzmann. As redes Bayesianas são uma abordagem para raciocínio probabilístico que combina teoria de grafos, para o estabelecimento de relações entre variáveis, e teoria de

probabilidades para atribuir níveis de confiabilidade (Russell *et al.*, 1995). As redes Bayesianas são grafos direccionados acíclicos definidos por  $G = (V, E)$ , onde os nós  $V$  representam variáveis aleatórias com probabilidades determinadas *a priori* e os arcos  $E$  representam relações de dependência entre as variáveis, aos quais estão associadas distribuições de probabilidade condicional (Heckerman *et al.*, 1999; Neapolitan, 2003).

Zhang & Koren (2007) utilizaram um modelo Bayesiano, com a estrutura exemplificada na Figura 2.1, em que cada utilizador é representado pelo perfil  $W_i$  e o sistema prevê a classificação  $y$  de um utilizador  $i$  a um item  $x$ , dada a distribuição  $w_u$ . Neste modelo os utilizadores são relacionados através de um nó raiz que permite ainda inicializar o perfil de novos utilizadores de acordo com uma distribuição  $P(w|\Phi)$ , solucionando o problema do arranque a frio.

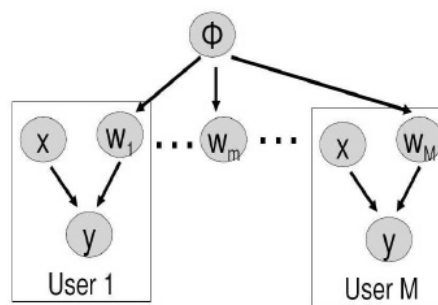


Figura 2.1: Topologia da rede Bayesiana de Zhang & Koren (2007)

Outro exemplo é o projecto TV Genius<sup>1</sup> onde foi utilizada uma rede Bayesiana para representar os programas de televisão e as relações entre eles, sendo estas últimas medidas através da monitorização da actividade dos utilizadores.

Outro dos modelos probabilísticos usados são as máquinas de Boltzmann, um tipo de rede neuronal estocástica (i.e., na rede existe uma distribuição de probabilidade associada ao facto dos neurónios estarem activos ou não) proposto por Geoffrey Hinton e Terry Sejnowski (Aarts & Korst, 1988; Hinton *et al.*, 1984). O formalismo das máquinas de Boltzmann é inspirado no processo que ocorre fisicamente com um determinado material aquando da recepção de uma determinada quantidade de calor (i.e., o material dilata-se, aumentando a sua superfície, e posteriormente contrai-se por forma a equilibrar as trocas energéticas) (Ackley *et al.*, 1985). Uma máquina de Boltzmann é uma rede de unidades individuais binárias estocásticas (i.e., os neurónios), em que cada unidade  $i$  tem uma probabilidade  $p_i$  de estar activa. Existem dois subconjuntos de unidades individuais nesta estrutura, nomeadamente um subconjunto formado por neurónios visíveis e outro formado por neurónios escondidos, sendo que este último não possui contacto com o exterior.

Os trabalhos de Truyen *et al.* (2009) e de Salakhutdinov *et al.* (2007) são exemplos da utilização de máquinas de Boltzmann em sistemas de recomendação. Nestes trabalhos, este modelo foi usado de modo a implementar um sistema de filtragem colaborativa *item-to-item* onde os itens classificados representam as unidades visíveis da rede com a respectiva classificação (que consistem nos elementos da camada de baixo de cada uma

<sup>1</sup><http://www.tvgenius.net/>

das topologias). No trabalho de Truyen *et al.* (2009) (Figura 2.2), as unidades escondidas ( $h_1, \dots, h_d$ ) representam os itens não classificados e no trabalho de Salakhutdinov *et al.* (2007) (Figura 2.3) as unidades escondidas correspondem às características de cada utilizador ( $h$ ). Através das ligações entre as unidades escondidas e visíveis é possível inferir as classificações aos itens ainda não classificados, directamente (como no primeiro caso) ou indirectamente através do perfil inferido do utilizador (segundo caso).

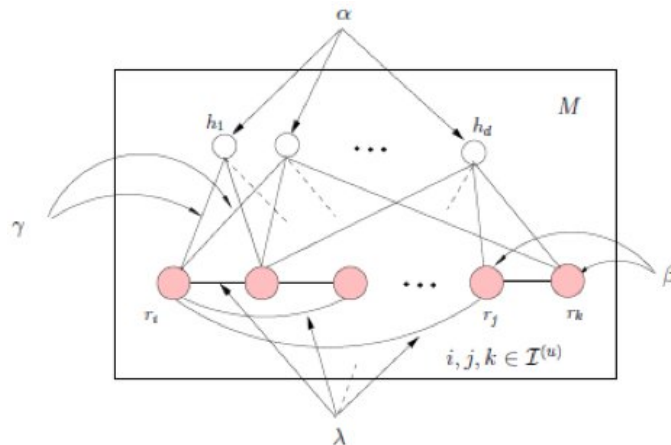


Figura 2.2: Topologia da máquina de Boltzmann de Truyen *et al.* (2009).

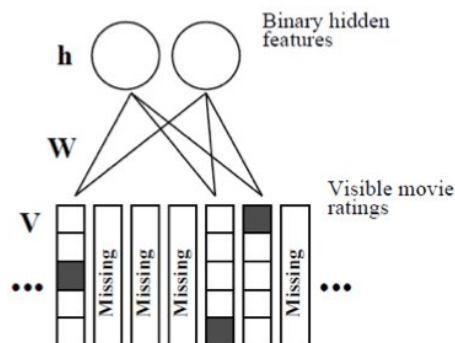


Figura 2.3: Topologia da máquina de Boltzmann de Salakhutdinov *et al.* (2007).

### 2.2.3 Sistemas de recomendação híbridos

Os sistemas híbridos, combinando técnicas baseadas no conteúdo com filtragem colaborativa, surgiram para tirar partido das mais-valias de ambos os tipos de sistemas referidos anteriormente, tentando ainda suprir as lacunas individuais dos mesmos.

Burke (2002) apresentou diferentes métodos de implementação de sistemas híbridos, sendo os mais comuns descritos de seguida:

- **Ponderação** – As recomendações de cada técnica são ponderadas, através de pesos e combinadas para produzir um único conjunto de recomendações.

- **Troca** – Para cada recomendação a ser efectuada utiliza-se apenas uma técnica, escolhendo-a de acordo com o contexto actual.
- **Misto** – As recomendações provenientes de diferentes técnicas são apresentadas ao utilizador ao mesmo tempo, ou seja, é apresentada a união das recomendações.
- **Cascata** – Uma técnica de recomendação refina as recomendações de outra.
- **Meta-nível** – O modelo correspondente a uma técnica de recomendação é utilizado como forma de providências os dados de entrada para uma outra técnica.

A *framework* RACOFI, que será descrita em mais detalhe na Secção 2.3.1, apresenta uma arquitectura híbrida, seguindo o modelo de cascata. As recomendações são primeiro calculadas pela componente de filtragem colaborativa, intitulada Cofi, e as mesmas são depois usadas como *input* para o RALOCA, um sistema baseado no conteúdo usando regras, que refina as recomendações (Anderson *et al.*, 2003).

O sistema Fab é um exemplo de um modelo híbrido meta-nível. Este sistema procura sugerir páginas Web de acordo com as preferências dos utilizadores. O objectivo do sistema passa por criar *clusters* de interesses contendo páginas Web, que são disponibilizadas aos utilizadores que tenham aquele tipo de perfil (Balabanovic & Shoham, 1997). Um dos elementos do Fab são os agentes de selecção que estão associados aos utilizadores e que realizam filtragem baseada no conteúdo, actualizando o perfil dos utilizadores. Outro dos elementos são os agentes de colecção que estão associados a um *cluster* e, por cada *cluster*, representam um tópico de interesse. Estes agentes usam os modelos dos perfis dos utilizadores mantidos pelos agentes de selecção para realizarem as suas operações de busca de novas páginas relevantes ao tópico de interesse que representam, através de técnicas de filtragem colaborativa, sendo portanto um modelo meta-nível. Quando os agentes de colecção obtêm novas páginas para os *clusters*, estas recomendações servem de *input* aos agentes de selecção que os distribuem aos utilizadores. Como tal, o Fab também pode ser considerado um sistema baseado num modelo em cascata, muito embora a componente baseada no conteúdo (agentes de selecção) não altere as recomendações dos agentes de colecção.

Nos sistemas baseados em modelos é possível combinar componentes baseadas no conteúdo com a filtragem colaborativa, relacionando os itens com o seu conteúdo, bem como relacionando os itens e/ou os utilizadores entre eles, directa ou indirectamente. Um exemplo desta técnica é apresentado no trabalho (de Campos *et al.*, 2006), que combina diferentes componentes por ponderação numa rede Bayesiana de quatro camadas relacionadas entre elas: atributos dos itens (representada pelos nós  $F$ ), itens (que correspondem aos nós  $I$ ), utilizadores (indicados pelos nós  $U$ ) e camada de relação entre utilizadores (que são os nós  $V$  que permitem ligar cada utilizador a outros utilizadores) – Veja-se a Figura 2.4.

A utilidade da combinação de diferentes técnicas de recomendação é comprovada perante a utilização de sistemas híbridos pela maioria das equipas que participam no maior concurso anual de sistemas de recomendação promovido pela Netflix. The Ensemble, uma das equipas melhor classificadas na edição de 2009, é um dos exemplos disso tendo os seus autores Sill *et al.* (2009) apresentando a abordagem em detalhe.

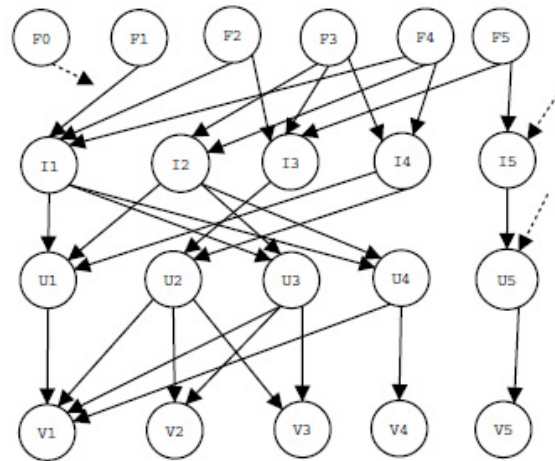


Figura 2.4: Topologia da Rede Bayesiana de de Campos *et al.* (2006)

### 2.2.4 Aplicações no domínio da recomendação de programas de televisão

A televisão digital abriu portas aos ambientes televisivos interactivos e com a capacidade de se adaptarem às preferências dos utilizadores. Contudo, ainda existem poucas aplicações de sistemas de recomendação na televisão interactiva.

Nestes sistemas, as classificações implícitas são mais utilizadas através, por exemplo, do registo das visualizações dos programas (por exemplo, no TV Genius<sup>1</sup>) ou através das gravações que são agendadas (Van Slobbe, 2005). Nesta área, os itens são geralmente muito ricos em termos de conteúdo (i.e., na maioria dos casos é possível saber o tipo de programa, a descrição do mesmo, a duração, etc.), pelo que não é necessário recorrer a técnicas complexas de *information retrieval* para obter informações úteis sobre os itens. O TiVo (popular sistema de recomendação colaborativo *item-to-item* descrito por Ali & Van Stam (2004)), o TV Genius (já descrito na Secção 2.2.2) e o Lighthouse (Van Slobbe, 2005) são dois exemplos de sistemas de recomendação aplicados ao domínio da televisão.

De acordo com trabalhos anteriores, a aplicação de sistemas de recomendação neste domínio envolve, entre outros, três grandes desafios:

1. **Quem é o utilizador?** Se o terminal onde é aplicado o sistema de recomendação não incluir sistema de autenticação/identificação, o sistema estará potencialmente a interagir com mais do que um utilizador (e.g., um agregado familiar) tendo-se que as preferências televisivas dos diferentes indivíduos diferem.
2. **As preferências relacionam-se com contexto temporal?** Provavelmente as preferências variam de acordo com a hora do dia, dia da semana e época do ano, pelo que o ignorar deste factor poderá conduzir a piores resultados.
3. **Onde realizar o processamento das recomendações?** Numa arquitectura cliente-servidor é importante decidir onde será realizada cada fase do

<sup>1</sup><http://www.tvgenius.net/>

processamento, tendo em conta aspectos como a eficiência, escalabilidade e privacidade do processo de geração das recomendações.

No projecto Lighthouse, a preocupação principal foi a de criar um sistema de filtragem colaborativa eficiente e que não violasse a privacidade dos utilizadores. Como tal, sugere-se uma implementação de um sistema distribuído, onde o processamento computacional fica em grande parte a cargo dos terminais clientes e as informações privadas não são partilhadas.

Outra abordagem foi seguida pelos criadores do DynamicTV (Simeoni *et al.*, 2009), uma vez que a componente interactiva consiste numa plataforma Web, intitulada iDynamicTV (Carmagnola *et al.*, 2009), em que os utilizadores podem interagir com os conteúdos (i.e., etiquetá-los, votar, etc.). De acordo com esta interacção e através de classificações explícitas, são apresentadas recomendações no DynamicTV.

## 2.3 Software existente e conjuntos de dados

A implementação e teste de diferentes técnicas de recomendação foi feita com base em *frameworks* de software existentes, tendo-se tentado ainda usar conjuntos de dados semelhantes aos normalmente utilizados em experimentos anteriores. Nesta secção são analisadas diversas as *frameworks* existentes para sistemas de recomendação e caracterizados alguns conjuntos de dados.

### 2.3.1 Frameworks para desenvolvimento de sistemas de recomendação

Têm sido criadas nos últimos anos várias *frameworks* que facilitam o desenvolvimento de sistemas de recomendação, e que já incluem os algoritmos mais comuns. Algumas das mais conhecidas apresentam-se de seguida. Estudos como o de Turati *et al.* (2008) já procuraram analisar e comparar estas *frameworks*.

**Taste**<sup>1</sup> *Framework open-source*, escrita em Java, que permite a criação de sistemas de recomendação. A *framework* foi desenhada por forma a garantir um bom desempenho, escalabilidade e flexibilidade, tendo-se que a mesma se encontra actualmente integrada no Apache Mahout, um projecto que contém a implementação de algoritmos de *data mining* distribuídos suportados pela arquitectura Map/Reduce. A bancada de software suporta os dois tipos básicos de filtragem colaborativa (*item-to-item* e *user-to-user*), através de algoritmos de *clustering*, e.g. o *k-means* e o *fuzzy k-means*, e de classificação. O Taste disponibiliza ainda mecanismos para a avaliação de um dado algoritmo de recomendação, com base em conjuntos de dados tais como os disponibilizados pelo GroupLens.

**COFI**<sup>2</sup> Acrónimo para "*Collaborative Filtering*", é uma *framework* de filtragem colaborativa escrita em Java, contendo inúmeros algoritmos que são divididos em quatro classes: *memory-based*, *eigentaste* (i.e., algoritmos que partem do

---

<sup>1</sup><http://lucene.apache.org/mahout/taste.html>

<sup>2</sup><http://www.nongnu.org/cofi/>

pressuposto de que existe um conjunto de dados *standard* classificados por todos os utilizadores), *learning-free* (i.e., algoritmos que não necessitam de saber nada sobre o utilizador) e *item-to-item* (Turati *et al.*, 2008). Este projecto parece estar descontinuado, não tendo actualizações desde 2005. A documentação é também praticamente inexistente. Os criadores desta *framework* disponibilizam ainda uma implementação muito simples de um algoritmo de recomendação denominado Slope One<sup>1</sup>, sob a forma de uma única classe Java (Lemire & Maclachlan, 2005). Esta implementação simples tem a vantagem de ser facilmente extensível ou integrada com outros componentes.

**RACOFI**<sup>2</sup> Acrónimo para "*A Rule-Aplying Collaborative Filtering System*", é um sistema de recomendação que resulta da integração do COFI com a bancada de software RALOCA, acrónimo para "*Rule Applying Learning Object Comparison Agent*". O componente COFI faz a filtragem colaborativa enquanto que o componente RALOCA filtra as recomendações apresentadas pelo COFI de acordo com técnicas de recomendação baseadas no conteúdo, suportadas na utilização de regras (Anderson *et al.*, 2003).

**CoFE**<sup>3</sup> Acrónimo para "*Collaborative Filtering Engine*", é um conjunto de algoritmos de filtragem colaborativa que permite processar classificações de múltiplas dimensões (i.e., a classificação divide-se em mais do que um critério), sendo estas valores numéricos ou binários. O CoFE permite prever a classificação de um dado utilizador a um item, bem como prever os *Top-N* itens. Sendo um mecanismo de predição, a manutenção do conteúdo da base de dados está a cargo da aplicação que usa o sistema de recomendação. O CoFE é implementado em Java segundo um modelo cliente-servidor. Esta *framework* engloba vários algoritmos, muito embora os criadores garantam apenas a fiabilidade do algoritmo *user-to-user* baseado na correlação de Pearson (Turati *et al.*, 2008).

**ColFi**<sup>4</sup> Acrónimo para "*Collaborative Filtering*", esta *framework* implementa em Java um conjunto de algoritmos de filtragem colaborativa (*user-to-user* e *item-to-item*) (Brozovsky & Petricek, 2007). Este sistema é flexível, permitindo acrescentar novos algoritmos colaborativos de uma forma relativamente fácil.

**Duine Framework**<sup>5</sup> Consiste num conjunto de bibliotecas de software *open-source*, escritas em Java, que permitem ao utilizador criar mecanismos de predição e combinar técnicas existentes, com vista a prever o interesse num item por parte de um dado utilizador. O sistema de recomendação Duine regista as preferências dos utilizadores face aos itens, bem como eventuais preferências genéricas que estes tenham indicado (e.g., categorias, géneros, etc.), associando as mesmas ao perfil de utilizador. O sistema é dinâmico, adaptando permanentemente o perfil dos utilizadores de acordo com as suas avaliações. O sistema suporta recomendações baseadas no conteúdo e ainda filtragem colaborativa *user-to-user*.

<sup>1</sup><http://www.daniel-lemire.com/fr/abstracts/SDM2005.html>

<sup>2</sup><http://www.daniel-lemire.com/fr/abstracts/COLA2003.html>

<sup>3</sup><http://eeces.oregonstate.edu/iis/CoFE/>

<sup>4</sup><http://colfi.wz.cz/>

<sup>5</sup><http://www.duineframework.org/>



Na Tabela 2.2 é apresentada uma comparação das bancadas de software existentes. Para utilização no contexto do meu trabalho, procurei escolher uma entre as *frameworks* cuja documentação fosse detalhada, que tivesse sido actualizada nos últimos meses, e que incluísse algoritmos de vários dos tipos de sistemas de recomendação mencionados anteriormente. A escolha recaiu assim na *framework* Duine.

Nome	Taste	COFI	RACOFI	CoFE	ColFi	Duine
Linguagem	Java	Java	Java	Java	Java	Java
Actualizações	Nov 2009	2005	2003	2004	Dez 2009	Jan 2009
Versão	0.2			0.4		4.4-RC1
Documentação	Detalhada	Simples	Simples	Simples	Detalhada	Detalhada
Bas. no Conteúdo			X			X
FC <i>user-to-user</i>	X	X	X	X		X
FC <i>item-to-item</i>	X	X	X	X	X	
Híbrido			X		X	X
Memory-based	X	X	X	X	X	X
Model-based		X	X		X	X
Tipo de Software	Sistema completo	Sistema completo	Sistema completo	Técnicas predição	Sistema completo	Técnicas predição

**Tabela 2.2:** *Frameworks* para desenvolvimento de sistemas de recomendação.

### 2.3.2 Conjuntos de dados públicos

A existência de dados de teste credíveis é um pré-requisito essencial para a validação de algoritmos de recomendação, ou para a aplicação de técnicas de aprendizagem automática. A necessidade de grandes volumes de dados pode colocar dificuldades na obtenção dos mesmos (Herlocker *et al.*, 2004), pelo que alguns esforços anteriores se focaram na construção e disponibilização de conjuntos de dados para experiências de avaliação na área dos sistemas de recomendação.

De entre os conjuntos de dados existentes, salientam-se os seguintes quatro pela sua ampla utilização:

**MovieLens** Criado pelo grupo de investigação GroupLens, consiste em três conjuntos de dados que diferem substancialmente no número de itens, utilizadores e classificações. Estes conjuntos referem-se ao domínio dos filmes e, tanto os itens como os utilizadores, têm diversos dados descritivos associados aos mesmos.

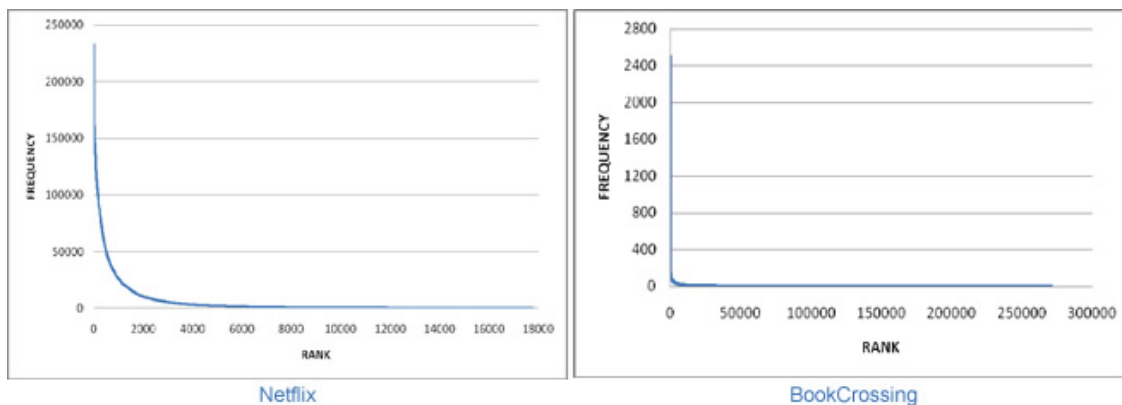
**Netflix** Esta empresa, conhecida por organizar anualmente um concurso de desenvolvimento de sistemas de recomendação<sup>1</sup>, disponibiliza também um conjunto de dados no domínio da recomendação de filmes. Este conjunto destaca-se pela quantidade de classificações (mais de cem milhões) e associa a cada classificação a data da mesma.

<sup>1</sup><http://www.netflixprize.com/>

**BookCrossing** Cai-Nicolas Ziegler disponibiliza um conjunto de dados construído com base em informação do conhecido site de partilha de livros BookCrossing<sup>1</sup>. Este conjunto de dados tem mais itens que os restantes (mais de 270 mil) e inclui classificações implícitas, além das habituais classificações explícitas. Existem também vários campos de metadados associados aos itens e aos utilizadores.

**Jester** Ken Goldberg disponibiliza quatro conjuntos de dados que correspondem a classificações de textos humorísticos do site Jester<sup>2</sup>. Ao contrário dos restantes, estes conjuntos de dados têm poucos utilizadores (um máximo de 150). No entanto, a distribuição das classificações dos utilizadores difere daquelas que estão associadas aos restantes conjuntos de dados. Enquanto que as classificações dos conjuntos de dados da MovieLens, Netflix e CrossBooking seguem uma distribuição de Lei de Potência (i.e., poucos utilizadores classificam muitos itens e muitos utilizadores classificam poucos itens (Zaier *et al.*, 2008)), as classificações do Jester revelam uma distribuição mais densa.

A distribuição de classificações dos utilizadores em cada um destes conjuntos de dados é apresentada nas Figuras 2.5 e 2.6 (Zaier *et al.*, 2008), originalmente apresentadas por Zaier *et al.* (2008).



**Figura 2.5:** Distribuição das classificações nos conjuntos de dados da Netflix e BookCrossing.

A Tabela 2.3 apresenta uma caracterização estatística dos quatro conjuntos de dados. Os três conjuntos de dados da MovieLens e outros três da Jester foram agrupados, uma vez que as características dos mesmos variam apenas no número de classificações, itens e utilizadores. Para esses três parâmetros, são apresentados os intervalos mínimo e máximo respectivos desses conjuntos de dados.

Importa referir que nenhum dos conjuntos de dados listados anteriormente contempla o domínio específico dos programas de televisão, muito embora os conjuntos do MovieLens e do Netflix contemplem um domínio relacionado, i.e. os filmes. Como tal, foi desenvolvido um novo conjunto de dados, específico para o domínio dos programas de televisão, que foi utilizado para avaliar a qualidade dos algoritmos de recomendação

<sup>1</sup><http://www.bookcrossing.com/>

<sup>2</sup><http://eigentaste.berkeley.edu/user/index.php>

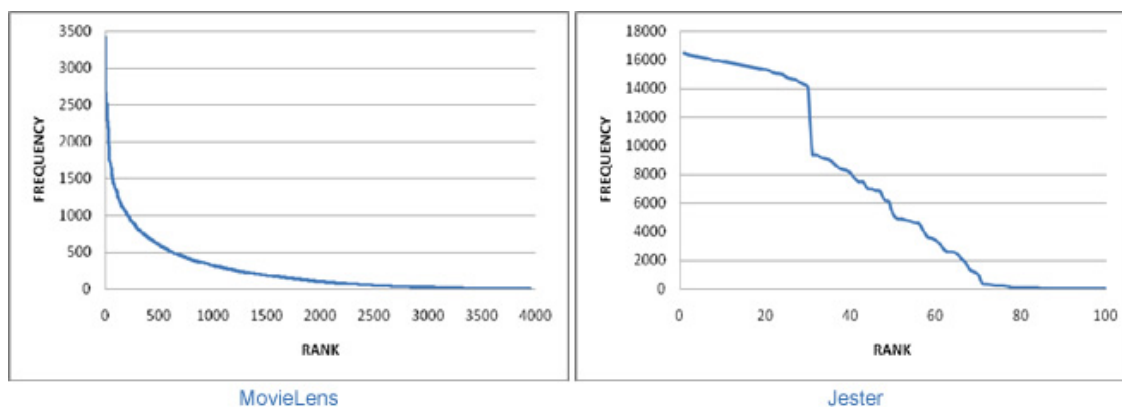


Figura 2.6: Distribuição das classificações nos conjuntos de dados do MovieLens e Jester.

que foram desenvolvidos. Este conjunto de dados deveria cumprir os seguintes requisitos:

- O conjunto de dados deve ter volumes de itens e de classificações comparáveis aos dos conjuntos de dados apresentados anteriormente.
- O conjunto de dados deve conter informação rica sobre os itens (i.e., programas de televisão), permitindo a utilização de várias métricas de similaridade entre os itens (e.g., similaridade textual nas descrições associadas aos programas, sobreposições nos conjuntos de actores ou criativos envolvidos no programa, etc.), tais como aquelas que já foram descritas na Secção 2.1.

Para a construção deste conjunto de dados, utilizei técnicas de extracção de informação, sobre os seguintes sites Web: TV.com<sup>1</sup>, IMDb<sup>2</sup> e Living Social<sup>3</sup>. A Secção 5.2 descreve a construção desta colecção de dados.

## 2.4 Sumário e discussão crítica

Apesar da popularidade dos sistemas de recomendação, estes têm sido relativamente pouco explorados no âmbito da recomendação de programas de televisão. A massificação da televisão interactiva proporcionou o surgimento de mais trabalhos nesta área, aproveitando o conhecimento adquirido na aplicação destes sistemas noutros domínios. O desenvolvimento destes sistemas envolve alguns problemas que dificilmente podem ser resolvidos totalmente, pelo que é necessário definir os objectivos e prioridades do sistema, de modo a tentar solucionar pelo menos os problemas mais críticos.

Tanto os sistemas baseados no conteúdo como colaborativos apresentam algumas vantagens distintas, pelos que se complementam bem. A literatura relacionada tem

<sup>1</sup><http://www.tv.com/>

<sup>2</sup><http://www.imdb.com/>

<sup>3</sup><http://tv.livingsocial.com/>

Nome	MovieLens 100m, 1M, 10M	Jester 1a, 1b, 1c	Jester 2	Netflix	BookCrossing
Domínio	Filmes	Piadas	Piadas	Filmes	Livros
Classificações	100m a 10M	> 374.070	< 9.596.100	100.480.507	1.149.780
Itens	1.682 a 10.681	100	150	17.770	271.379
Utilizadores	943 a 71.567	23.500 a 24.983	63.974	> 480.000	278.858
Distribuição de Lei de Potência	X			X	X
Dom. das Classific.	{1, ..., 5}	[-10.0, 10.0]	[-10.0, 10.0]	{1, ..., 5}	{1, ..., 10}
Classific. Explícita	X	X	X	X	X
Classific. Implícita					X
Informações dos Itens	Título, Ano, Género	-	-	Título, Ano,	Título, Autor, Ano, Editor
Informações dos Utilizadores	Idade, Sexo, Ocupação, CP	-	-	-	Idade, Localização
Informações das Classificações	-	-	-	Data	-

**Tabela 2.3:** Conjuntos de dados para sistemas de recomendação.

demonstrado as vantagens desta combinação, através da apresentação de diversas abordagens híbridas que tiram partido de conteúdos e de filtragem colaborativa.

Existem algumas *frameworks* que ajudam o desenvolvimento de sistemas de recomendação, muito embora não tenham todas elas as mesmas funcionalidades e algumas destas *frameworks* estarem já algo desactualizadas. Ao nível dos conjuntos de dados, verificou-se a inexistência de um conjunto específico para o domínio dos programas de televisão.

Perante esta análise, surge uma boa oportunidade para estudar os sistemas de recomendação num contexto ainda pouco explorado mas cujo domínio está cada vez mais em voga. Além do teste e discussão crítica de algoritmos já existentes, esta dissertação apresenta algumas alternativas, nomeadamente ao nível da aprendizagem automática como forma de determinação dos pesos para uma abordagem híbrida por combinação por ponderação.





## Capítulo 3

# Técnicas de Recomendação

Neste capítulo serão descritas diversas técnicas de recomendação implementadas neste trabalho. A Secção 3.1 começa por descrever algumas técnicas de predição básicas, que dependem apenas de uma das variáveis envolvidas numa predição (i.e., utilizador ou item), onde se seguem as técnicas baseadas no conteúdo descritivo dos programas na Secção 3.2 e terminando com as técnicas que usam filtragem colaborativa através do cruzamento de dados entre diversos utilizadores e itens, na Secção 3.3.

### 3.1 Técnicas de predição básicas

Para prever classificações existem duas técnicas básicas que têm a vantagem de considerar apenas dados agregados, contornando eventuais problemas de privacidade e de protecção de dados. As técnicas básicas utilizadas nos meus testes foram:

- **Média de classificações do utilizador** – Atribui uma previsão correspondente à média das classificações fornecidas pelo utilizador. Dado que muitos utilizadores utilizam uma pequena gama de classificações, esta técnica poderá apresentar bons resultados, uma vez que a avaliação de um sistema de recomendação realiza-se maioritariamente com base nas classificações já conhecidas (que estão dentro dessa pequena gama de valores). Sendo  $J$  o conjunto de itens já classificados pelo utilizador e  $r_{u,j}$  a classificação dada pelo utilizador  $u$  ao item  $j$ , a equação é dada por (Van Setten, 2005):

$$p_{u,i} = \bar{r}_u = \frac{\sum_{j \in J} r_{u,j}}{|J|} \quad (3.18)$$

- **Média de classificações do item** – Atribui uma previsão correspondente à média das classificações fornecidas para um dado item. Apesar de muito simplista, esta técnica reflecte bem a opinião geral face a um item. Esta técnica apresenta a mesma predição para um dado item para utilizadores diferentes e é muito útil para resolver problemas de arranque a frio dos utilizadores, apresentando recomendações genericamente credíveis. Sendo  $V$  o conjunto de utilizadores que já classificaram o item  $i$ , esta técnica é calculada do seguinte modo:

$$p_{u,i} = \bar{r}_i = \frac{\sum_{v \in V} r_{v,i}}{|V|} \quad (3.19)$$

## 3.2 Técnicas de recomendação baseadas no conteúdo

As técnicas de recomendação baseadas em conteúdo tiram partido do conteúdo dos itens, baseando-se na estimação da similaridade entre os itens com base nos conteúdos. No domínio da recomendação de programas de televisão, cada item é caracterizado com os seguintes atributos: título do programa, géneros, companhias responsáveis, criadores, realizadores, escritores, actores ou apresentadores e descrição.

O título e a descrição são atributos textuais. Por outro lado, as companhias, criadores, realizadores e escritores podem ser vistas como listas ordenadas de sequências de caracteres, sendo que a ordem se refere à relevância de cada elemento dessa lista ou ao histórico da participação de cada elemento no programa (e.g., na lista de realizadores, o realizador actual do programa é o primeiro da lista, o segundo foi o realizador anterior, etc.). O atributo dos géneros corresponde a um conjunto de cadeias de caracteres.

Nesta secção são descritas as métricas de similaridade utilizadas, seguindo-se uma explicação das técnicas implementadas na recomendação de programas com base nos conteúdos.

### 3.2.1 Métricas de similaridade

Para implementar técnicas de recomendação baseadas no conteúdo, é necessário calcular a similaridade entre os itens, ou seja, entre o conteúdo destes. Para isso são utilizadas diferentes métricas, consoante a informação que se pretende comparar.

#### 3.2.1.1 Métricas de similaridade entre conjuntos

Para medir a similaridade entre dois conjuntos  $A$  e  $B$  são habitualmente utilizadas as métricas descritas de seguida.

- **Coefficiente de Jaccard** – Indica a percentagem de elementos semelhantes nos dois conjuntos (Manning *et al.*, 2008). A equação deste coeficiente é dada por:

$$Jaccard(A, B) = \frac{|A \cap B|}{|A \cup B|} \quad (3.20)$$

- **Coefficiente de sobreposição** – Indica a percentagem de elementos semelhantes tendo por base o número mínimo de elementos<sup>1</sup>. O coeficiente de sobreposição é dado por:

$$sobreposicao(A, B) = \frac{|A \cap B|}{\min(|A|, |B|)} \quad (3.21)$$

<sup>1</sup><http://www.dcs.shef.ac.uk/~sam/stringmetrics.html>



Comparativamente com o coeficiente de Jaccard, este coeficiente apenas difere no denominador, sendo que:

$$\forall_{A,B}, |A \cup B| \geq \min(|A|, |B|) \Rightarrow J(A, B) \leq \text{sobreposicao}(A, B)$$

- **Coeficiente de Dice** – Indica a percentagem de elementos semelhantes nos dois conjuntos, contabilizando a repetição de elementos (ao contrário do que acontece no coeficiente de Jaccard) (Manning *et al.*, 2008). O cálculo deste coeficiente é feito através da seguinte equação:

$$\text{Dice}(A, B) = 2 \cdot \frac{|A \cap B|}{|A| + |B|} \quad (3.22)$$

### 3.2.1.2 Métricas de similaridade entre sequências de cadeias de caracteres

As métricas descritas de seguida podem ser aplicadas a qualquer tipo de sequências, ou seja, listas ordenadas, muito embora neste caso o foco seja dado a sequências de cadeias de caracteres (i.e., cada elemento da sequência é ele próprio uma sequência de caracteres, e.g., uma palavra). Este tipo de informação existe em cinco dos atributos dos itens considerados neste trabalho. Sendo  $A$  e  $B$  duas sequências de caracteres, três possíveis métricas de similaridade são as seguintes:

- **Distância de Levenshtein** – Esta métrica contabiliza o número mínimo de operações necessárias para transformar a sequência  $A$  na sequência  $B$ , utilizando operações de inserção, eliminação ou substituição (Manning *et al.*, 2008). A cada uma destas operações está associado um custo, no nosso caso igual para as três operações, cuja soma corresponderá à distância de edição.

Para calcular a similaridade em termos de um valor entre 0.0 e 1.0, temos que:

$$\text{Levenshtein}(A, B) = \frac{1 - \min(\text{DistanciaLevenshtein}(A, B))}{\max(\text{DistanciaLevenshtein}(A, B))} \quad (3.23)$$

onde  $\text{DistanciaLevenshtein}(A, B)$  é o conjunto das distâncias correspondentes a todas as transformações possíveis de  $A$  para  $B$ .

- **Métrica da posição inversa** – Esta métrica corresponde à similaridade entre duas sequências tendo em conta a ordem dos elementos, atribuindo pesos diferentes de acordo com a posição de cada elemento (i.e., atribui mais relevância aos primeiros elementos) (Bar-Ilan *et al.*, 2007).

Tendo duas sequências  $A$  e  $B$  de  $k_A$  e  $k_B$  elementos respectivamente, as permutações  $\sigma_A$  e  $\sigma_B$  são vectores contêm ambos  $|A \cup B|$  elementos, sendo que cada posição representa um elemento diferente indicando a posição desse elemento nos conjuntos  $A$  e  $B$  respectivamente. Nas situações em que um dado elemento não está presente num determinado conjunto  $I$ , a posição é artificialmente definida como  $k_I + 1$ .

Sendo  $Z$  o conjunto de elementos presentes simultaneamente nos conjuntos  $A$  e  $B$ ,  $S$  o conjunto de elementos presentes apenas no conjunto  $A$  e sendo  $T$  o conjunto de elementos presentes apenas no conjunto  $B$ , temos que:

$$N^{(k_A, k_B)}(\sigma_A, \sigma_B) = \sum_{i \in Z} \left| \frac{1}{\sigma_A(i)} - \frac{1}{\sigma_B(i)} \right| \quad (3.24)$$

$$+ \sum_{j \in S} \left| \frac{1}{\sigma_A(j)} - \frac{1}{k_B + 1} \right| \quad (3.25)$$

$$+ \sum_{l \in T} \left| \frac{1}{\sigma_B(l)} - \frac{1}{k_A + 1} \right| \quad (3.26)$$

Esta métrica é normalizada do seguinte modo:

$$M^{(k_A, k_B)} = 1 - \frac{N^{(k_A, k_B)}}{\max N^{(k_A, k_B)}} \quad (3.27)$$

onde

$$\max N^{(k_A, k_B)} = \sum_{i=1}^{k_A} \left( \frac{1}{i} - \frac{1}{k_B + 1} \right) + \sum_{i=1}^{k_B} \left( \frac{1}{i} - \frac{1}{k_A + 1} \right) \quad (3.28)$$

### 3.2.1.3 Métricas de similaridade entre sequências de números

Apesar de nenhum dos atributos dos itens ser uma sequência de números, este tipo de métricas pode ser relevante, no sentido em que uma sequência de palavras pode ser convertida numa sequência de números, sendo que cada número indica a frequência de uma dada palavra numa frase (i.e., uma sequência de palavras). Ou seja, tendo duas sequências de palavras (e.g., uma frase separada por espaços e caracteres de pontuação), calcula-se o número de ocorrências de cada palavra em cada uma das sequências, registando os resultados nos vectores  $\vec{A}'$  e  $\vec{B}'$ . Cada um destes vectores tem  $|A \cap B|$  elementos, sendo que, por exemplo,  $A_1$  e  $B_1$  indicam o número de vezes em que a mesma palavra está presente nas sequências de palavras  $A$  e  $B$  respectivamente. Tendo isto, a similaridade do cosseno das duas sequências é calculada através do cosseno do ângulo formado entre os dois vectores (Manning *et al.*, 2008), tal como demonstrado na equação seguinte:

$$\cos(\vec{A}', \vec{B}') = \frac{\vec{A}' \cdot \vec{B}'}{\|\vec{A}'\| \cdot \|\vec{B}'\|} \quad (3.29)$$

## 3.2.2 Técnicas implementadas

Sendo  $J$  o conjunto de itens classificados pelo utilizador  $u$  cuja similaridade com o item  $i$  é superior a um determinado valor (*threshold*), a predição para a classificação do

utilizador  $u$  ao programa  $i$  é calculada da seguinte forma (Van Setten, 2005):

$$p_{u,i} = \frac{\sum_{j \in J} r_{u,j} * w_{i,j}^2}{\sum_{j \in J} w_{i,j}^2} \quad (3.30)$$

onde  $w_{i,j}$  é o factor de ponderação, que neste caso corresponde exclusivamente ao valor de similaridade (de 0 a 1) entre os itens  $i$  e  $j$ , e  $r_{u,j}$  é a classificação do utilizador  $u$  ao item  $j$ . O quadrado aplicado ao valor de similaridade (que representa uma diferença face à fórmula indicada na referência citada) permite diferenciar mais as similaridades, uma vez que a similaridade baseada no conteúdo representa valores maioritariamente muito baixos.

Dos oito atributos associados a cada programa, apenas o título não será considerado para efeitos de comparação ao nível da similaridade com outros programas. Como já foi referido, existem atributos que são sequências, um que é um conjunto e outro que é uma simples cadeia de caracteres. Para cada um destes três casos serão utilizadas técnicas diferentes. Deste modo, obtém-se um valor de similaridade entre os itens que estão a ser comparados para cada um destes atributos, sendo que estes são ponderados por factores de peso  $p_i$  diferentes de forma a obter um único valor de similaridade. Como tal, a similaridade é calculada do seguinte modo:

$$w_{i,j} = \frac{\sum_{a \in A} sim_a(i,j) * p_a}{\sum_{a \in A} p_a} \quad (3.31)$$

Ao nível das sequências, costuma ser muito utilizada a distância de Levenshtein para calcular a similaridade. Contudo, esta métrica valoriza demasiado a ordem dos itens, pelo que se duas sequências  $A$  e  $B$  têm um dado elemento  $k$  mas em posições diferentes pode ser equivalente a um dos conjuntos não ter sequer esse elemento. No limite, tome-se como exemplo as sequências de actores  $A = \{a, b, c, d\}$  e  $B = \{d, c, b, a\}$ . A similaridade entre estas duas sequências usando a distância de Levenshtein dá  $1/4 = 0.25$ , quando na realidade as sequências são muito parecidas e os itens em questão aparentam ter uma grande similaridade e uma eventual correlação ao nível das preferências dos utilizadores.

Para solucionar o problema mencionado, será utilizada a Métrica da Classificação Inversa que mede a distância entre a posição de cada elemento numa sequência e na outra, além de pesar os elementos de acordo com a sua posição.

No que diz respeito ao cálculo da similaridade entre conjuntos de géneros, a utilização do coeficiente de sobreposição teria o problema de dar uma similaridade máxima, por exemplo, aos conjuntos  $A = \{Comedia, Romance\}$  e  $B = \{Romance\}$ , sendo que  $i_A$  era uma comédia romântica e  $i_B$  era um romance. Quanto às outras duas métricas (Jaccard e Dice), foi utilizado o coeficiente de Dice pelo facto de valorizar mais as semelhanças existentes nos conjuntos comparados.

A descrição dos programas, ao contrário dos restantes atributos, é uma simples cadeia de caracteres. Para calcular a similaridade entre duas descrições, foi utilizada a similaridade do cosseno, sendo cada elemento do vector correspondente a uma palavra, indicando a frequência dessa palavra na descrição do item. Não foi utilizada uma

métrica de conjuntos uma vez que essas métricas ignoram os elementos repetidos.

A Figura 3.7 sintetiza o modo como funcionam os cálculos associados ao método de recomendação baseado no conteúdo entre dois itens.

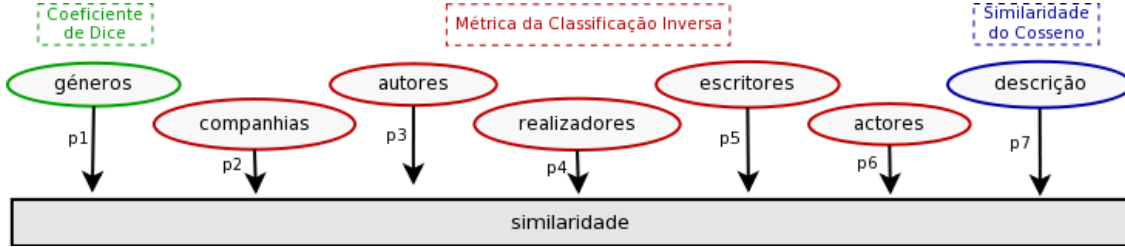


Figura 3.7: Diagrama da metodologia baseada no conteúdo.

### 3.3 Técnicas de recomendação colaborativas

Neste trabalho, foram testadas técnicas de recomendação colaborativas *user-to-user* e *item-to-item*, de forma a comparar as duas abordagens. As diferentes técnicas implementadas são descritas nesta secção

#### 3.3.1 Técnicas de recomendação colaborativas *user-to-user*

Sendo  $V$  o conjunto de utilizadores excepto o utilizador  $u$ , sendo  $r_{v,i}$  a classificação atribuída pelo utilizador  $v$  ao item  $i$  e sendo  $w_{u,v}$  o valor de similaridade entre os utilizadores  $u$  e  $v$ , a predição da classificação do utilizador  $u$  ao item  $i$  é dada por:

$$p_{u,i} = \frac{\sum_{v \in V} r_{v,i} * w_{u,v}}{\sum_{v \in V} w_{u,v}} \quad (3.32)$$

Contudo, a gama de classificações dos utilizadores é variável, ou seja, há utilizadores cuja classificações máximas e mínimas variam muito, pelo que é mais importante verificar qual a classificação relativa de cada utilizador a um item (i.e., de que forma se enquadra a classificação na gama de classificações desse utilizador), em vez de considerar os valores absolutos das classificações. Note-se que a interpretação face a cada classificação varia de utilizador para utilizador, pelo que uma classificação de 7 para um utilizador pode ter o mesmo significado que uma classificação de 9 de outro utilizador. A equação 3.32 apresenta este problema.

A equação 3.33 resolve este problema tomando em consideração a classificação relativa de cada utilizador, resultando num valor percentual (de -1.0 a 1.0) que é aplicado ao desvio padrão de  $u$  (i.e.,  $\sigma_u$ ) e adicionado à média das suas classificações ( $\bar{r}_u$ ), de modo a ser adaptado à gama de classificações de  $u$  (Herlocker, 2000). Ou seja:

$$p_{u,i} = \bar{r}_u + \sigma_u \frac{\sum_{v \in V} \left[ \left( \frac{r_{v,i} - \bar{r}_v}{\sigma_v} \right) w_{u,v} \right]}{\sum_{v \in V} w_{u,v}} \quad (3.33)$$

onde  $\sigma_u$  é o desvio padrão das classificações do utilizador  $u$ .

O estudo de Herlocker (2000) demonstra que a precisão da previsão melhora se for multiplicado ao valor de similaridade um factor de ponderação associado ao número de itens que tais utilizadores classificaram em conjunto. Na prática, se o número de itens classificados por ambos ( $itensComuns$ ) for menor do que um determinado valor ( $minItensComuns$ ) é aplicada a ponderação  $itensComuns/minItensComuns$ , ou seja:

$$w_{u,v} = \begin{cases} sim_{u,v} * \frac{itensComuns}{minItensComuns} & \text{se } itensComuns < minItensComuns \\ sim_{u,v} & \text{caso contrário} \end{cases} \quad (3.34)$$

Neste trabalho será testada a Equação 3.34.

Na Equação 3.33 são contabilizados todos os utilizadores que tenham classificado algum item em comum com o utilizador em causa. Contudo, assim contabiliza-se também utilizadores cuja similaridade pode ser demasiado baixa e, portanto, inconclusiva, podendo mesmo prejudicar as predições. Como tal, será testada também a inclusão de um *threshold* (i.e., um valor mínimo de similaridade que, caso não seja cumprido, leva a que a mesma não seja considerada), de forma a que apenas os utilizadores cuja similaridade esteja acima de um determinado valor sejam considerados (Herlocker, 2000).

A similaridade  $w_{u,v}$  entre os utilizadores  $u$  e  $v$  pode ser calculada com recurso a diferentes métricas de correlação, analisadas e descritas no livro de Myers & Well (2003). Todas estas métricas devolvem um valor entre -1.0 e 1.0. As métricas testadas neste trabalho são descritas de seguida

- **Correlação de Pearson** – Esta correlação é calculada do seguinte modo:

$$\rho_{u,v} = \frac{\sum_{i \in I} (r_{u,i} - \bar{r}_u)(r_{v,i} - \bar{r}_v)}{\sqrt{\sum_{i \in I} (r_{u,i} - \bar{r}_u)^2 \sum_{i \in I} (r_{v,i} - \bar{r}_v)^2}} = \frac{Cov(u, v)}{\sqrt{Var(u) \cdot Var(v)}} \quad (3.35)$$

Foi também testada uma variante da Correlação de Pearson, à qual apelidamos de Correlação de Pearson Fixa, onde a média de cada utilizador é substituída pelo valor médio da gama de classificações possíveis (sendo de 0.0 a 1.0, o valor médio é 0.5) (Herlocker, 2000). Assim, a polaridade das classificações não depende da média das classificações do utilizador, mas sim da gama das classificações em termos absolutos. A equação é dada por:

$$\rho_{u,v}^{fixo} = \frac{\sum_{i \in I} (r_{u,i} - 0.5)(r_{v,i} - 0.5)}{\sqrt{\sum_{i \in I} (r_{u,i} - 0.5)^2 \sum_{i \in I} (r_{v,i} - 0.5)^2}} \quad (3.36)$$

- **Coefficiente de correlação de Spearman** – Este coeficiente ordena as classificações de cada utilizador de acordo com a classificação por este atribuída aos vários itens e, em vez de utilizar para efeitos de cálculo de similaridade as classificações, utiliza as posições deles. Sendo  $x$  e  $y$  as listas de posições das classificações de  $u$  e  $v$  respectivamente, temos que:

$$\rho_{u,v} = \frac{\sum_{i \in I} (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i \in I} (x_i - \bar{x})^2 \sum_{i \in I} (y_i - \bar{y})^2}} = \frac{cov(u, v)}{\sqrt{var(u) \cdot var(v)}} \quad (3.37)$$

Tendo estas posições, o coeficiente de correlação de Spearman utiliza a correlação de Pearson, considerando a posição de cada elemento.

- **Coeficiente de correlação de Kendall** – Este coeficiente, tal como o de Spearman, aplica-se às posições de cada classificação entre todas as classificações de cada utilizador. Sendo  $R$  e  $S$ , respectivamente, os conjuntos de posições de classificações de  $u$  e  $v$ , tendo-se que existem outras classificações com o mesmo valor, o coeficiente de Kendall é calculada deste modo:

$$\tau_{u,v} = \frac{(\text{numero de pares concordantes}) - (\text{numero de pares não concordantes})}{\sqrt{\left(\frac{n(n-1)}{2} - \sum_{r \in R} \frac{r(r-1)}{2}\right) \left(\frac{n(n-1)}{2} - \sum_{s \in S} \frac{s(s-1)}{2}\right)}} \quad (3.38)$$

- **Covariância** – A equação é definida de seguida:

$$Cov_{u,v} = E[(u - E[u])(v - E[v])] \quad (3.39)$$

Nesta equação,  $E[x]$  é o valor esperado e é calculado do seguinte modo:

$$E[x] = \sum_{i=1}^{|x|} x_i p(x_i) \quad (3.40)$$

### 3.3.2 Técnicas de recomendação colaborativas *item-to-item*

Ao contrário das técnicas apresentadas na Secção 3.3.1, as técnicas colaborativas *item-to-item* calculam a similaridade entre os itens, e essa similaridade será usada para aferir a classificação que um dado utilizador  $u$  atribuirá a um item  $i$ , sabendo que ele já classificou itens similares a  $i$  (Linden *et al.*, 2003). A forma de cálculo é análoga à das técnicas de recomendação colaborativas *user-to-user*. Como tal, a predição do utilizador da classificação do utilizador  $u$  ao item  $i$  é dada por:

$$p_{u,i} = \bar{r}_i + \sigma_i \frac{\sum_{j \in J} \left[ \left( \frac{r_{u,j} - \bar{r}_u}{\sigma_u} \right) w_{i,j} \right]}{\sum_{j \in J} w_{i,j}} \quad (3.41)$$

Para cálculo de similaridade, a Correlação de Pearson Fixa é dada por:

$$\rho_{i,j}^{fixo} = \frac{\sum_{u \in U} (r_{u,i} - 0.5)(r_{u,j} - 0.5)}{\sqrt{\sum_{u \in U} (r_{u,i} - 0.5)^2 \sum_{u \in U} (r_{u,j} - 0.5)^2}} \quad (3.42)$$

### 3.3.3 Outras técnicas de recomendação colaborativas

Ao nível das técnicas colaborativas, existem também outras menos complexas e mais eficientes. A literatura da área dos sistemas de recomendação demonstra que estas técnicas conseguem apresentar resultados muitas vezes tão bons como as técnicas mais complexas referidas anteriormente (Lemire & Maclachlan, 2005). De seguida descrevem-se duas dessas técnicas, as quais também foram testadas no contexto deste trabalho.

- **Desvio Top-N** – Sendo  $V$  o conjunto de utilizadores que classificaram  $i$ , a predição é dada por (Van Setten, 2005):

$$p_{u,i} = \bar{r}_u + \frac{\sum_{v \in V} (r_{v,i} - \bar{r}_v)}{|V|} \quad (3.43)$$

- **Slope One** – Formalmente, o Slope One (Lemire & Maclachlan, 2005) produz recomendações em dois passos. Primeiro, calcula a matriz *item-to-item*  $D_{n \times n}$ , para  $n$  itens, com a variação média dos dois itens. Para cada dois itens  $i$  e  $j$  na matriz, o algoritmo calcula o desvio médio  $D_{i,j}$  das classificações do item  $i$  face às classificações feitas ao item  $j$  de acordo com a fórmula seguinte, onde  $S_{i,j}$  se refere ao conjunto de utilizadores que classificaram tanto o item  $i$  como o  $j$ .

$$D_{i,j} = \sum_{u \in S_{i,j}} \frac{r_{u,j} - r_{u,i}}{|S_{i,j}|} \quad (3.44)$$

Tendo a matriz  $D_{n \times n}$ , existem duas formas de calcular a predição ao item  $i$  por um utilizador  $u$ . Sendo  $S_u$  o conjunto de itens classificados pelo utilizador  $u$  e sendo  $R_i = \{j | j \in S_u, j \neq i, |S(i,j)| > 0\}$ , o Slope One Básico é dado por:

$$p_{u,i} = \frac{1}{|R_i|} \sum_{j \in R_i} (D_{i,j} + u_j) \quad (3.45)$$

Por outro lado, o **Slope One Ponderado** considera o número de utilizadores que classificaram cada par de itens para ponderar a predição obtida por esse par. Como tal, esta abordagem do Slope One tem a seguinte fórmula:

$$p_{u,i} = \frac{\sum_{j \in R_i} (D_{i,j} + u_j) |S_{i,j}|}{\sum_{j \in R_i} |S_{i,j}|} \quad (3.46)$$

Além destas duas técnicas apresentadas previamente na literatura, apresentamos de seguida duas variantes que visam ter em conta o desvio padrão de classificações de cada utilizador. Neste sentido, a técnica proposta **Desvio Top-N Relativo** é apresentada na Equação 3.47.

$$p_{u,i} = \bar{r}_u + \sigma_u \frac{\sum_{v \in V} \left( \frac{r_{v,i} - \bar{r}_v}{\sigma_v} \right)}{|V|} \quad (3.47)$$

Por outro lado, o  $D_{i,j}$  do **Slope One Ponderado Relativo** é calculado pela Equação 3.48, sendo que o valor da predição é dado pela Equação 3.49.

$$D_{i,j} = \sum_{u \in S_{i,j}} \frac{\left( \frac{r_{u,j} - r_{u,i}}{\sigma_u} \right)}{|S_{i,j}|} \quad (3.48)$$

$$p_{u,i} = \frac{\sum_{j \in R_i} (dev_{i,j} * \sigma_u + u_j) |S_{i,j}|}{\sum_{j \in R_i} |S_{i,j}|} \quad (3.49)$$

Neste trabalho serão testadas as técnicas Desvio Top-N, Desvio Top-N Relativo, Slope One Ponderado e Slope One Ponderado Relativo.

### 3.4 Sumário

Neste capítulo foram apresentados essencialmente dois tipos de técnicas de recomendação: baseadas no conteúdo e colaborativas. Além destas, foram mencionadas duas técnicas básicas (média de classificações do utilizador e do item) que permitem prever uma classificação perante um contexto de pouca informação disponível e assegurando a privacidade dos utilizadores. Por um lado, as técnicas baseadas no conteúdo exigem a existência de informação rica sobre os itens, o que nem sempre se verifica, além de que a semelhança dos conteúdos descritivos pode não ser suficiente para aferir a relação entre diferentes itens. Por outro lado, as técnicas colaborativas exigem a existência de uma elevada quantidade de classificações, nomeadamente do utilizador a quem se destina a recomendação. Como tal, estas limitações dificultam a inferência das classificações, reduzindo eventualmente a precisão e a cobertura. Neste sentido, a combinação de diferentes técnicas pode servir para tirar partido das vantagens de cada uma, atenuando as limitações.







## Capítulo 4

# Estratégias Híbridas

Neste capítulo são apresentadas algumas estratégias que visam combinar diferentes técnicas descritas no capítulo anterior. Primeiro são referidas estratégias que combinam técnicas através de heurísticas, e depois é apresentado um exemplo de uma combinação utilizando técnicas de aprendizagem automática, procurando otimizar listas ordenadas de recomendações.

### 4.1 Ponderação por heurísticas

De modo a combinar diferentes técnicas, foram utilizadas as abordagens de troca e de ponderação, bem como a combinação destas duas. As estratégias sugeridas e que foram seguidas, são as seguintes:

#### 4.1.1 Combinação por troca por ordem fixa

Esta estratégia consiste em percorrer uma lista ordenada de técnicas (que se mantém inalterável no sistema), perguntando a cada uma qual a sua predição para uma dada classificação. Quando uma técnica consegue apresentar a sua predição (há situações em que a ausência de dados impossibilita o cálculo da predição), essa predição é usada, terminando o algoritmo sem percorrer as restantes técnicas. O esquema desta estratégia é apresentado na Figura 4.8 onde são indicadas as técnicas incluídas nesta combinação.



Figura 4.8: Diagrama da metodologia de combinação por troca por ordem fixa.

#### 4.1.2 Combinação por troca por ordem variável

Esta estratégia é semelhante à anterior. Contudo, a ordem das técnicas varia para cada predição, consoante um dado critério. Neste caso, o critério utilizado foi o valor do

Erro Médio Absoluto (EMA) acumulado de acordo com os itens já classificados pelo utilizador. Ou seja, para cada classificação dada pelo utilizador, é inferida a predição através de cada técnica (ignorando a classificação real já conhecida), e de acordo com os resultados determina-se o EMA de cada técnica. Quando é necessário realizar uma predição para uma eventual recomendação, as técnicas são ordenadas ascendentemente de acordo com o EMA de cada técnica para o utilizador em causa. Como tal, para cada utilizador existe um registo do EMA para cada uma das técnicas. Esta é uma forma de ir adaptando a estratégia de acordo com os resultados obtidos, procurando adaptar-se a cada utilizador (isto porque uma técnica que seja boa para um utilizador, pode não o ser para outro). Quando o utilizador ainda não classificou nenhum item, a ordem é a mesma da estratégia de combinação por troca por ordem fixa. O esquema desta estratégia é apresentado na Figura 4.9.

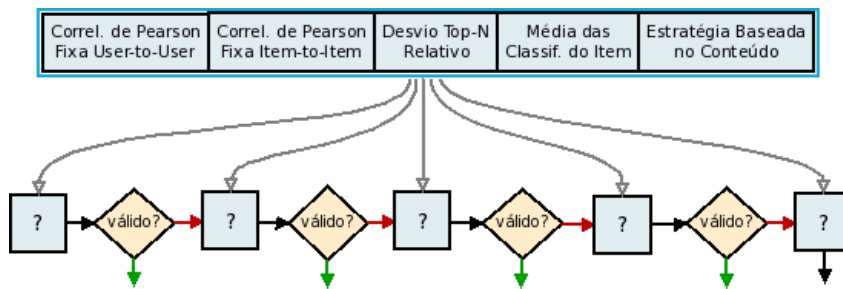


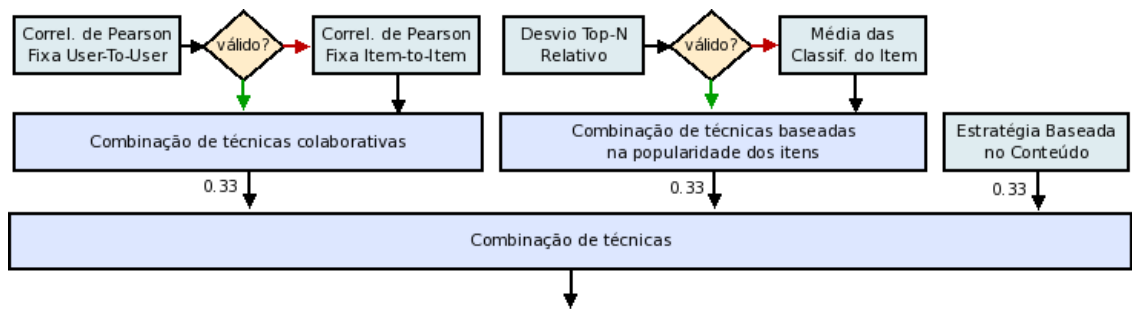
Figura 4.9: Diagrama da metodologia de combinação por troca por ordem variável.

### 4.1.3 Combinação por ponderação fixa com três conjuntos de técnicas combinadas por troca

Nesta estratégia são considerados três tipos de predições, nomeadamente (i) filtragem colaborativa, (ii) baseada na popularidade dos itens, e (iii) baseado no conteúdo. Cada um destes três tipos contém uma ou duas técnicas que são combinados por ponderação fixa, ou seja, em cada um destes tipos de técnicas apenas uma apresenta a sua predição, tendo-se que posteriormente são combinadas as três através de uma média aritmética. Esta estratégia visa garantir que estes três tipos de técnicas são ponderados de igual forma em todas as predições, para que os três tipos possam contribuir para as mesmas. A Figura 4.10 ilustra o funcionamento desta estratégia.

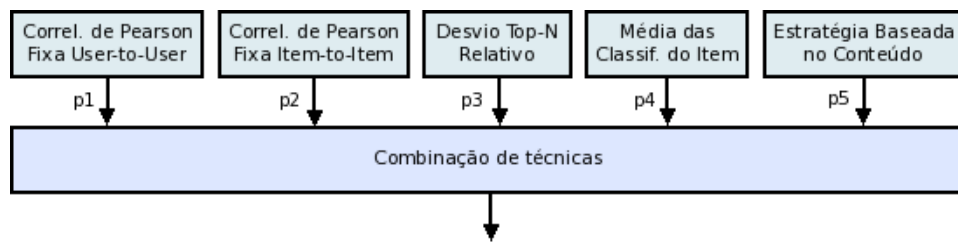
### 4.1.4 Combinação por ponderação variável

Ao contrário das duas estratégias mencionadas anteriormente, todas as técnicas contidas nesta estratégia são invocadas sempre. Todavia, o peso de cada técnica para a predição final varia consoante um dado critério variável. O critério utilizado foi o mesmo da estratégia anterior, ou seja, o EMA. Deste modo, quanto menor for o EMA de uma dada técnica de acordo com o histórico, maior será o peso dessa técnica na predição. Procura-se assim diluir o eventual erro de algumas técnicas através de melhores predições obtidas com outras técnicas, aproveitando as diferenças no tipo cálculo de cada técnica, ou seja, é uma forma de combinar sempre a popularidade dos



**Figura 4.10:** Diagrama da metodologia de combinação por ponderação fixa com três conjuntos de técnicas combinadas por troca.

itens (Média das Classificações dos Itens e Desvio Top-N Relativo), a similaridade entre as classificações dos utilizadores (Correlação de Pearson Fixa *User-to-User*), a similaridade entre as classificações dos itens (Correlação de Pearson Fixa *Item-to-Item*), a similaridade entre os conteúdos dos itens (Estratégia Baseada no Conteúdo). Se o utilizador ainda não classificou nenhum item, é atribuído igual peso a todas as técnicas. As técnicas que não consigam apresentar a sua predição são ignoradas. Esta abordagem está ilustrada na Figura 4.11.



**Figura 4.11:** Diagrama da metodologia de combinação por ponderação com pesos variáveis.

#### 4.1.5 Combinação por ponderação fixa

Nas estratégias de combinação ponderadas, surge o desafio de encontrar os pesos ideais. Anteriormente referimos uma estratégia por ponderação variável onde os pesos de cada técnica são ajustados ao longo do tempo para cada utilizador. Contudo, para a criação de uma estratégia por combinação por ponderação fixa é necessário determinar quais os melhores pesos para cada técnica. Para cumprir esse objectivo, as técnicas de aprendizagem automática supervisionada, nomeadamente os algoritmos de *learning-to-rank*, são uma solução frequentemente utilizada. Através destas técnicas, é possível utilizar dados de treino por forma a aprender um modelo de combinação para diferentes técnicas de recomendação, optimizando-se os resultados de acordo com métricas como o nDCG (Liu, 2009). Este processo realiza-se tal como ilustrado na Figura 4.12.

O processo de *learning-to-rank* recebe como input o conjunto de treino que consiste em  $n$  utilizadores  $q_i (i = 1, \dots, n)$ , em que cada um deles está associado a  $m^{(i)}$  vectores de técnicas de recomendação  $x^{(i)} = \{x_j^{(i)}\}_{j=1}^{m^{(i)}}$ , cada um deles correspondendo a um item

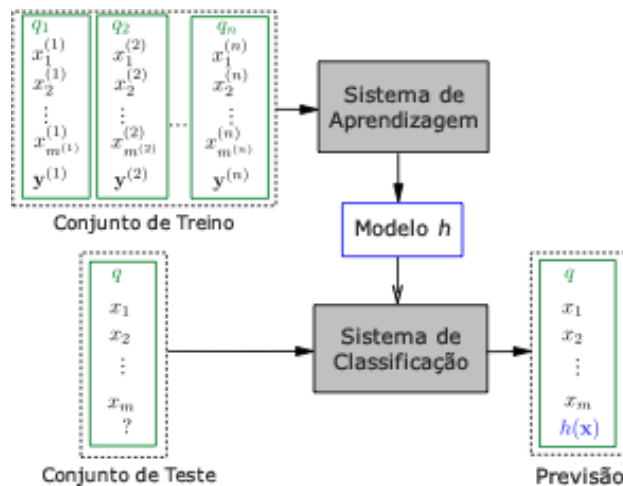


Figura 4.12: Framework do processo de *learning-to-rank*.

associado ao utilizador  $q_i$ . Para cada utilizador existe ainda a ordenação dos documentos através de  $y^{(i)}$  (Liu, 2009). O sistema de aprendizagem ao receber este conjunto de treino treina o modelo  $h$  que será recebido como input no sistema de classificação e permitirá aplicá-lo num conjunto de teste idêntico ao conjunto de treino. Como resultado, os itens associados a cada utilizador de input são ordenados, podendo assim comparar essa ordenação com a ordenação ideal.

A utilização de técnicas de *learning-to-rank* pressupõe (onde se procura determinar os pesos ideais) e um conjunto de teste (onde se testam esses pesos e se avalia esta estratégia). Neste sentido, procedeu-se da seguinte forma:

1. Divide-se o conjunto de dados em dois: um conjunto de treino e um conjunto de teste. Para isso, procede-se da seguinte forma:
  - (a) Avalia-se todo o conjunto de classificações usando as técnicas simples e estratégias híbridas mencionadas anteriormente. É necessário realizar este passo de modo a obter os conjuntos de treino e de teste.
  - (b) Ordenam-se de forma aleatória os utilizadores.
  - (c) Divide-se os utilizadores em dois grupos, de acordo com o tamanho pretendido para o conjunto de treino e de teste.
  - (d) Agrupam-se os triplos  $\langle \text{utilizador}, \text{item}, \text{tecnica} \rangle$  por cada utilizador, no grupo respectivo.
2. Percorrem-se as classificações do conjunto de treino procurando construir o modelo de classificação  $h$ , utilizando o algoritmo  $SVM^{ndcg}$  (Chakrabarti *et al.*, 2008).
3. Percorre-se as classificações do conjunto de teste, do seguinte modo:
  - (a) Para cada utilizador:

- i. Para cada par  $\langle \text{utilizador}, \text{item} \rangle$  infere-se a classificação utilizando o modelo calculado no passo anterior.
  - ii. Depois de percorridos todos os itens do utilizador, ordenam-se os itens de acordo com a classificação inferida.
  - iii. Calcula-se o nDCG para os primeiros  $n$  itens da lista de itens recomendados.
- (b) Calcula-se o nDCG médio.

Esta abordagem foi testada com conjuntos de treino de 25%, 50% e 75% do tamanho total, de modo a analisar a variação do nDCG resultante consoante o tamanho do conjunto de treino.

## 4.2 Sumário

As estratégias híbridas apresentadas neste capítulo visam adaptar a estratégia utilizada de acordo com cada situação, por exemplo, privilegiando técnicas baseadas no conteúdo quando existem poucas correlações entre utilizadores, e técnicas colaborativas aquando da ausência de conteúdos descritivos sobre os itens ou de similaridade entre estes. Assim, é possível aumentar a cobertura das técnicas e melhorar a precisão da estratégia. No caso das estratégias de combinação por ponderação, é difícil determinar quais os pesos ideais para cada factor ponderado, pelo que a aprendizagem automática supervisionada, tal como apresentada na Secção 4.1.5, surge como uma solução para este problema.





## Capítulo 5

# Avaliação Experimental

Na Secção 2.1.1 foram apresentadas as principais métricas de avaliação usadas no contexto dos sistemas de recomendação. Os resultados de avaliação são reportados neste capítulo de acordo com essas métricas, e ainda de acordo com algumas novas métricas aqui propostas, resultantes de extensões das métricas anteriormente propostas como forma a abordar algumas das suas limitações.

### 5.1 Metodologia de avaliação

A avaliação das técnicas implementadas fez-se com recurso ao Duine Framework, que já contém uma componente de validação. Contudo, essa componente consiste num método de validação diferente daquele que habitualmente é usado em trabalhos semelhantes, embora análogo à situação real de um sistema (i.e., inicialmente nenhum utilizador classificou qualquer item e, à medida que o tempo vai passando, os números de utilizadores e de classificações vão aumentando). Na prática, a validação do Duine Framework percorre todos os itens e para cada um deles infere a classificação usando todas as técnicas em análise, só depois registando essa classificação. Como tal, a validação tem início sem que haja qualquer classificação registada e termina quando todas as classificações ficarem registadas no sistema. Este processo de avaliação beneficia no início as técnicas baseadas nos conteúdos, uma vez que as mesmas são menos susceptíveis ao problema do arranque a frio. Por outro lado, o processo de avaliação beneficia no final as técnicas que tiram melhor partido do vasto conjunto de classificações, nomeadamente as técnicas de filtragem colaborativa.

Neste trabalho procura-se avaliar as técnicas em contextos com um vasto conjunto de utilizadores, itens e classificações. Como tal, optou-se por validar as técnicas de acordo com a estratégia de validação mais comum. Nesta estratégia, registam-se todas as classificações no sistema; depois percorrem-se todas as classificações e, para cada uma delas, infere-se a classificação utilizando cada técnica, ignorando a classificação real existente. Para realizar a avaliação de acordo com esta metodologia, foi necessário implementá-la no Duine, dado que este *framework* não a suporta. O output deste processo de predição é uma tabela de uma base de dados com os seguintes campos relevantes:

- *id* – Identificador unívoco para cada tuplo, representando a tríade  $\langle \text{utilizador}, \text{item}, \text{tecnica} \rangle$ .
- *id<sub>utilizador</sub>* – Identificador do utilizador.
- *id<sub>item</sub>* – Identificador do item (programa de TV).
- *tecnica* – Técnica utilizada para prever a classificação.
- *predicao* – Valor da predição de uma dada técnica *t*, relativamente a um dado par  $\langle \text{utilizador}, \text{item} \rangle$ .
- *classificacao – real* – Valor da classificação real.

Após completar o processo de predição e ter a base de dados preenchida, são feitos dois tipos de avaliações:

- **Avaliação da distância entre predições e valores reais** – Nesta avaliação são aplicadas as métricas *n*-árias descritas na Secção 2.1.1.2, ou seja, a avaliação baseia-se na diferença entre o valor das predições e o valor das classificações reais.
- **Avaliação das listas de itens recomendadas** – Nesta avaliação os itens são inicialmente ordenados em duas listas distintas:
  - Numa lista os itens são ordenados de acordo com o seu valor de predição;
  - Na outra lista são ordenados de acordo com o seu valor de classificação real.

As duas listas resultantes são comparadas consoante as técnicas referidas na Secção 2.1.1.3.

## 5.2 Conjunto de dados de validação

Perante a inexistência de um conjunto de dados público para o domínio dos programas de televisão, foi construído neste trabalho um conjunto de dados, com os requisitos referidos na Secção 2.3.2. O conjunto de dados foi construído utilizando as três seguintes fontes (que correspondem a páginas Web onde os utilizadores humanos descrevem, comentam e manifestam preferências sobre programas de televisão): TV.com, IMDb e Living Social.

Para cada uma das três fontes foi escrito um script XQuery para aceder às páginas com informação dos itens mais populares. Através de expressões XPath, o script acede a cada um dos nomes dos programas e posteriormente acede à página de cada um dos programas para obter todas as informações necessárias sobre os programas e as classificações atribuídas pelos utilizadores. Como cada site tem um layout diferente, cada um deles necessitou de um script específico. Além disso, cada uma destas fontes tem diferentes conjuntos de informações para os itens. Esta extracção resultou em três ficheiros XML para cada uma das fontes. A caracterização de cada um destas fontes de dados, é apresentada na Tabela 5.5.

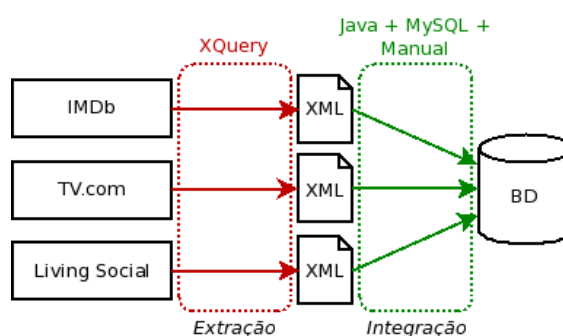


Figura 5.13: Construção do conjunto de dados para o domínio dos programas de televisão.

Nome	TV.com	IMDb	Living Social
Domínio	Programas TV	Programas TV e Filmes	Programas TV
Classificações	107.942	19.292	15.569
Itens Considerados	1.000	1.000	1.000
Utilizadores	46.032	14.299	5.862
Domínio das Classific.	{1.0, 1.5, ..., 9.5, 10.0}	{1, ..., 10}	{1.0, 1.5, ..., 4.5, 5.0}
Principais Informações dos Itens	Título, Géneros, Tema, Descrição, Autores, Realizadores, Actores/Apresentadores	Título, Géneros, Duração, País, Língua, Descrição, Companhias, Autores, Realizadores, Produtores, Actores/Apresentadores	Título, Género, Descrição
Principais Informações dos Utilizadores	Nome, Biografia	Nome, Localidade	Nome
Informações das Classificações	Data, Comentário textual	Data, Comentário textual	Data, Comentário textual

Tabela 5.4: Caracterização das fontes de dados.

Como se verifica na tabela, o TV.com destaca-se no número de classificações e utilizadores, pelo que para cada item tem um maior número de classificações, em média. Contudo, no sentido de criar um conjunto de dados maior e com um volume de dados mais próximo daqueles que foram descritos na Secção 2.3.2, e ainda de modo a garantir maior heterogeneidade nas classificações e perfil de utilizadores, optou-se por se integrar os dados destas três fontes.

Ao nível do domínio de classificações, o IMDb e o Living Social apresentam o mesmo número de hipóteses de classificação, sendo que o TV.com apresenta o dobro. Como tal as suas classificações devem ser convertidas. Ao nível das informações existentes, verifica-se também que cada uma das fontes tem informações diferentes, em especial ao nível dos itens. Apenas nas classificações todos eles apresentam os mesmos dois dados: data e comentário textual.

Após a obtenção dos três ficheiros XML, foi necessário integrá-los na base de dados utilizada no trabalho. Esta integração foi feita utilizando um XML de cada vez, começando no IMDb, passando pelo TV.com e terminando com o Living Social. Antes de realizar a integração foi necessário decidir qual o tipo de dados que seriam registados no conjunto de dados. Ao nível dos itens só foram considerados os seguintes

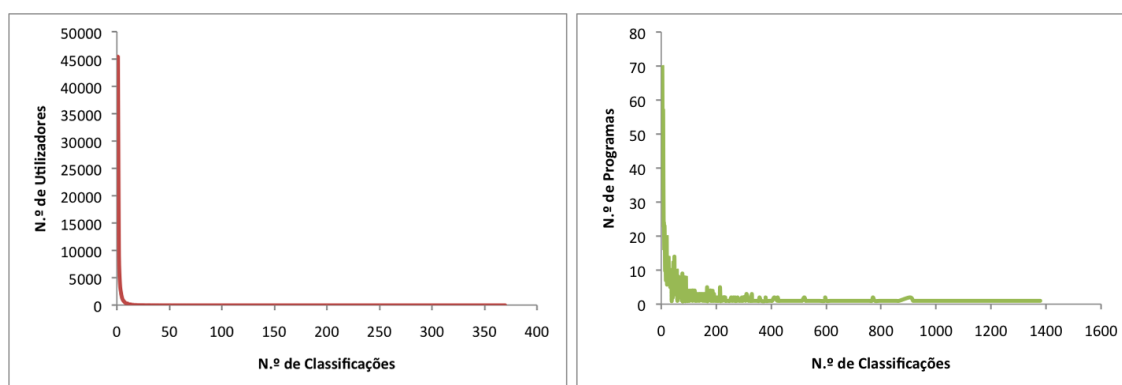
atributos: título, companhias, autores, realizadores, escritores, actores/apresentadores, géneros e descrição. Relativamente aos utilizadores, de modo a preservar a sua privacidade, não foi registado qualquer dado pessoal, tendo-se que os mesmos são apenas distinguidos por um identificador unívoco. No que diz respeito às classificações, também não foi registado nenhum dado adicional além dos dados obrigatórios: identificador do utilizador, identificador do item e valor da classificação. Todas as classificações foram normalizadas para terem o domínio  $\{0.1, 0.2, \dots, 0.9, 1.0\}$ .

O ficheiro XML do IMDb, sendo o primeiro, é iterado por um código escrito em Java que, usando expressões XPath, percorre os dados obtidos que são relevantes (i.e., os dados que vão ser considerados pelo sistema de recomendação) e insere-os directamente na base de dados. No caso dos outros dois ficheiros XML, estes são lidos por um código Java, uma vez que é necessário integrar com os dados já existentes na base de dados. Como são fontes de dados diferentes, não existe partilha de utilizadores entre as fontes de dados. Contudo, ao nível dos itens, existem itens transversais a duas ou três fontes de dados, sendo portanto necessário identificar os duplicados e juntá-los num só item. Como forma de realizar esta junção, para cada novo item adicionado o sistema verificava se já existia algum item com um título exactamente igual. Em caso positivo, era feito um mapeamento entre esse item e o já existente. Caso contrário, são identificados os itens cujo distância de edição é inferior ou onde um dos títulos faz *match* completo com uma parte do outro. Esses itens com um grau de semelhança elevado são analisados manualmente, decidindo-se ou não pelo mapeamento.

Um dos desafios encontrados aquando da junção dos dados foi o problema de cada elemento de cada atributo ser identificado por um ID que o identifica dentro dessa fonte de dados, mas que não tem qualquer significado nas outras fontes. Para identificar os elementos iguais, decidiu-se utilizar o nome do elemento, em vez de utilizar o seu ID. Esta abordagem tem a desvantagem de considerar como identificador uma cadeia de caracteres que não garante a identificação unívoca de um elemento. Contudo, antes de executar este passo, os atributos dos conjuntos de dados foram analisados e concluiu-se que apenas 2% dos elementos têm pelo menos um outro elemento com o mesmo nome mas ID diferente, dentro do mesmo conjunto de dados. Logo, considerou-se que apesar de 2% dos mapeamentos entre elementos ficarem incorrectos, tal não iria, em princípio, afectar muito os resultados finais, uma vez que tais elementos só seriam utilizados pelas estratégias baseadas no conteúdo, que procuram não só um factor de semelhança, mas sim vários factores que tornem os itens com um nível de similaridade não desprezável.

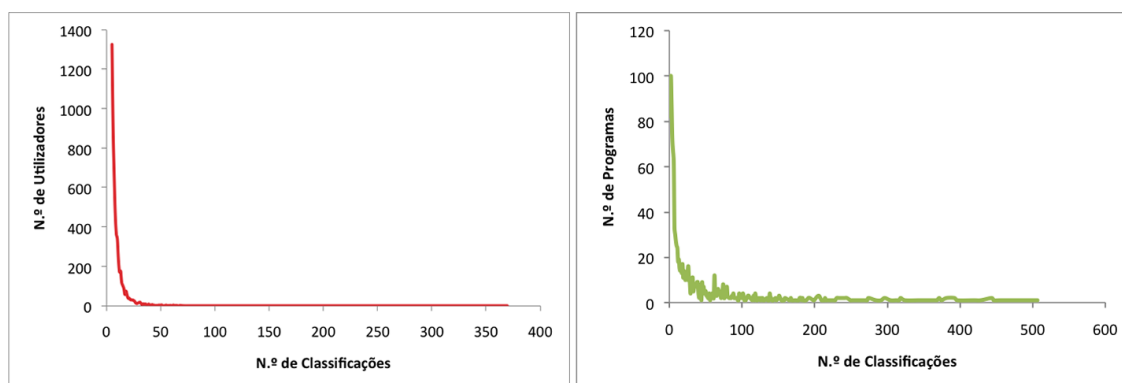
Apesar da fonte de dados do TV.com ter mais classificações e utilizadores que o conjunto de dados mais pequeno do MovieLens, tanto os dados do TV.com como das outras duas fontes pecam pelo facto da maioria dos utilizadores ter apenas classificado poucos itens. No conjunto de dados do MovieLens só são incluídos os utilizadores que já classificaram pelo menos 20 itens, uma vez que só se conseguem testar as abordagens colaborativas quando os utilizadores do sistema já classificaram vários itens. O requisito de um maior número de classificações por utilizador garante uma maior credibilidade dos utilizadores e, conseqüentemente, das suas classificações. Note-se que cerca de 76% dos utilizadores do conjunto de dados criado só classificaram um ou dois itens mas só um terço das classificações pertence a esses utilizadores. Tal aspecto revela a distribuição de Lei de Potência nestas fontes de dados e a necessidade de filtrar os dados.

O conjunto de dados apresenta a distribuição de classificações por utilizadores e por itens apresentado na Figura 5.14. Os gráficos demonstram uma acentuada distribuição de lei da potência relativamente aos utilizadores o que dificulta a execução de testes, ao nível da eficácia e eficiência, uma vez que um grande número de utilizadores não classificou itens suficientes para que possa ser calculada a sua similaridade com outros utilizadores, nomeadamente ao nível de técnicas de filtragem colaborativa. Além disso, a excessiva proporção de utilizadores com poucas classificações pode prejudicar predições a outros utilizadores, uma vez que o reduzido número de classificações pode representar falta de credibilidade e coerência nas classificações.



**Figura 5.14:** Distribuição das classificações dos utilizadores e itens antes da filtragem.

Para colmatar os problemas referidos, após o tratamento de todos os dados, estes foram filtrados de modo a terem apenas utilizadores que classificaram cinco ou mais itens, apresentando as distribuições apresentadas na Figura 5.15. Após esta filtragem, a distribuição de classificações dos utilizadores ficou com uma distribuição de lei da potência menos acentuada. Contudo, esta filtragem conduziu, em parte, ao efeito contrário na distribuição das classificações dos itens. Ou seja, após a filtragem existe um maior número de itens com poucas classificações, mas tal não se revela tão crítico como no caso dos utilizadores.



**Figura 5.15:** Distribuição das classificações dos utilizadores e itens após realização da filtragem.

A proporção de cada classificação é demonstrada na Figura 5.16 onde se verifica que

mais de metade das classificações são de 0.8 ou superior. Isto sugere que os utilizadores classificam maioritariamente os seus itens preferidos.

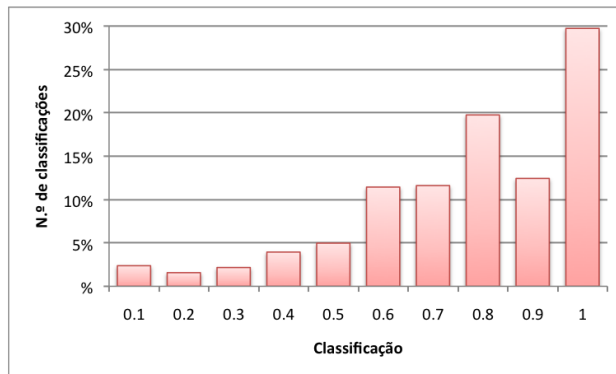


Figura 5.16: Proporção de cada classificação no conjunto de dados.

É importante analisar outros aspectos que foram tidos em conta aquando da implementação das técnicas de recomendação, bem como na análise dos resultados que estas obtiveram no momento da avaliação. Convém analisar a distribuição dos valores de classificação de cada utilizador. Na Figura 5.17 pode-se observar que o valor mais elevado de classificação dos utilizadores é maioritariamente 1, todavia o valor mínimo varia muito, o que demonstra que os utilizadores utilizam gamas de classificações diferentes apesar de terem quase sempre o mesmo majorante (que corresponde à classificação máxima).

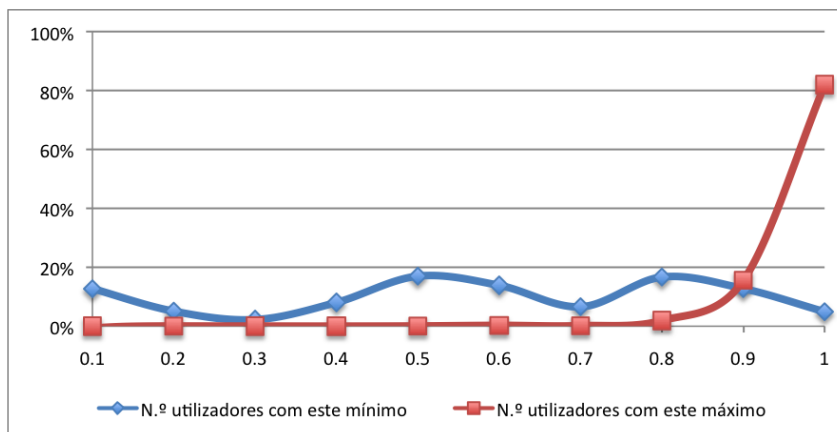
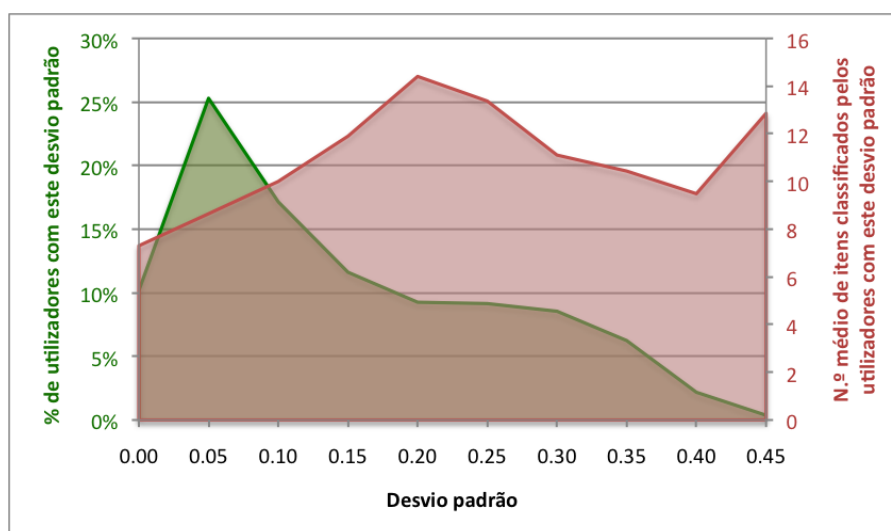


Figura 5.17: Valores mínimos e máximos das classificações dos utilizadores.

Analisando o desvio padrão das classificações (veja-se o gráfico verde da Figura 5.18), verifica-se que o desvio padrão é maioritariamente baixo, apesar de mais de um quarto do número de utilizadores ter desvios padrões superiores a 0,25. Importa por isso analisar se existe alguma relação entre o desvio padrão e o número de itens classificados por um utilizador, ou seja: será que o baixo desvio padrão das classificações de alguns utilizadores deve-se às suas poucas classificações ou será que efectivamente cada utilizadores utiliza uma gama de classificações própria,

independentemente do seu número de classificações? Observando a Figura 5.18 constata-se que apesar de haver alguma relação de proporcionalidade inversa entre os dois aspectos, esta não é significativa, pelo que revela que de facto cada utilizador utiliza uma gama de classificações diferente (i.e., há utilizadores que tendem a concentrar as suas classificações num pequeno subconjunto de valores, enquanto outros dispersam mais as classificações e utilizam um conjunto maior).



**Figura 5.18:** Comparação entre a percentagem de utilizadores que tem cada desvio padrão com o número médio de classificações desses utilizadores.

Em suma, o conjunto de dados resultante (ao qual se apelidou de TV Programs Dataset) tem as características indicadas na Tabela 5.5.

Nome	TV Programs Dataset
Domínio	Programas TV
Classificações	58.878
Itens Considerados	1.302
Utilizadores	5.638
Mínimo de classificações por utilizador	5
Distribuição de Lei de Potência	X
Domínio das Classificações	{0.1, ..., 1.0}
Informações dos Itens	Título, Géneros, Companhias, Descrição, Realizadores, Autores, Actores/Apresentadores
Informações dos Utilizadores	-
Informações das Classificações	-

**Tabela 5.5:** Caracterização do conjunto de dados TV Programs Dataset

### 5.3 Métricas de avaliação consideradas

Ao nível da avaliação da distância entre previsões e valores reais, serão utilizadas as métricas Erro Médio Absoluto (EMA) e Erro Médio Quadrático (EMQ). O EMQ dá mais ênfase à consistência das técnicas, enquanto o EMA é uma média aritmética simples, sem valorizar em demasia os erros. Note-se que bastam algumas previsões com erros significativos para que o valor do EMQ suba consideravelmente, pelo que uma técnica com baixo EMQ terá sempre uma margem de erro sempre baixa.

Quanto à avaliação das listas de itens preferenciais, apesar destas serem normalizadas, elas ignoram a gama de classificações do utilizador. Imagine-se que temos dois utilizadores  $A$  e  $B$  com as classificações indicadas na Tabela 5.6.

Util./Item	$i_1$	$i_2$	$i_3$	$i_4$	$i_5$
$A$	0.4	0.7	0.7	0.6	0.5
$B$	1.0	1.0	0.7	0.9	0.7
$C$	0.7	1.0	0.4	0.7	1.0

Tabela 5.6: Exemplo de uma matriz utilizadores/itens.

Supondo que o sistema de recomendação recomenda os itens que estão indicados a cinzento na tabela para cada utilizador, ou seja,  $R_A = \{i_2, i_3\}$  e  $R_B = \{i_3, i_5\}$ , sendo  $R_u$  a lista de itens recomendados a um utilizador  $u$ , os valores das métricas são apresentados na Tabela 5.8.

Utilizador	$A$	$B$	$C$
Itens recomendados	$\{i_2, i_3\}$	$\{i_3, i_5\}$	$\{i_1, i_4\}$
Relevância dos itens recomendados	$\{0.7, 0.7\}$	$\{0.7, 0.7\}$	$\{0.7, 0.7\}$
Itens mais relevantes	$\{i_2, i_3\}$	$\{i_3, i_5\}$	$\{i_2, i_5\}$
Relevância dos itens mais relevantes	$\{0.7, 0.7\}$	$\{1.0, 1.0\}$	$\{1.0, 1.0\}$
$CG_2$	1.4	1.4	1.4
$ICG_2$	1.4	2.0	2.0
$nCG_2$	1.0	0.7	0.7
$DCG'_2$	1.019	1.019	1.019
$IDCG'_2$	1.019	1.631	1.631
$nDCG'_2$	1.0	0.625	0.625

Tabela 5.7: Valores das métricas para listas ordenadas para o caso de exemplo.

Note-se que na Tabela 5.8 não foram apresentados os valores do nDCG, uma vez que esta métrica para  $p$  igual ou inferior a 2 produz os mesmos resultados do nCG. Perante os dados obtidos, verifica-se que nenhuma das métricas distingue o resultado das recomendações apresentadas aos utilizadores  $B$  e a  $C$ . Contudo, verifica-se que a recomendação dos itens  $\{i_1, i_4\}$  ao utilizador  $C$  deveria ser mais valorizado do que a recomendação dos itens  $\{i_3, i_5\}$  ao utilizador  $B$ , uma vez que ao utilizador  $B$  estão a ser recomendados os dois itens aos quais ele deu pior classificação, enquanto que no caso do utilizador  $C$  os itens recomendados não são aqueles que têm pior relevância, dado que o item  $i_3$  tem relevância 4. Como tal, os valores normalizadores deveriam ser superiores para  $C$  comparativamente com  $B$ .



Observando as métricas para  $A$  e  $B$  verifica-se ainda que apesar de em  $A$  terem sido recomendados os itens preferidos deste, e em  $B$  terem sido recomendados os itens com pior relevância (de entre os itens já classificados pelo utilizador) a diferença do nCG, nDCG e nDCG' é pequena (1.0 e 0.7 respectivamente), não evidenciando a diferença da qualidade patente entre as duas classificações. Aliás, de entre os itens passíveis de recomendação num contexto de validação, a recomendação a  $B$  foi a pior possível, o que deveria reflectir-se em nCG, nDCG e nDCG' muito baixos (inferiores a 0.5 de forma a ter um valor negativo).

Após esta análise, é importante realçar que as classificações atribuídas pelos utilizadores não podem ser vistas como valores absolutos, devem ser enquadradas no contexto do utilizador, ou seja, devem ser vistas como classificações relativas consoante a distribuição de classificações de cada utilizador. Para colmatar esta lacuna destas métricas, neste trabalho é apresentada uma forma alternativa para avaliar a relevância de cada classificação no contexto de uma avaliação de um sistema de recomendação. Deste modo, sendo  $r_u$  o conjunto de classificações de  $u$ , sugere-se que  $rel_i$  seja calculado do seguinte modo:

$$rel_i = \begin{cases} \frac{rel_i - \min(r_u)}{\max(r_u) - \min(r_u)} & \text{se } \max(r_u) \neq \min(r_u) \\ 1 & \text{caso contrário} \end{cases}$$

Consequentemente, ao alterar o modo de cálculo de  $rel_i$  surgem três métricas análogas àquelas que foram apresentadas anteriormente:

- nRCG (ganho acumulado normalizado com valores relativos)
- nDRCG (ganho acumulado com desconto normalizado com valores relativos)
- nDRCG' (ganho acumulado com desconto normalizado com maior ênfase na ordem com valores relativos)

Voltando ao exemplo referido no início desta subsecção, com estas novas métricas obtemos os seguintes valores:

Utilizador	$A$	$B$	$C$
$\min(r_u)$	0.4	0.7	0.4
$\max(r_u)$	0.7	1.0	1.0
Itens recomendados	$\{i_2, i_3\}$	$\{i_3, i_5\}$	$\{i_1, i_4\}$
Relevância dos itens recomendados	$\{1.0, 1.0\}$	$\{0.0, 0.0\}$	$\{0.5, 0.5\}$
Itens mais relevantes	$\{i_2, i_3\}$	$\{i_3, i_5\}$	$\{i_2, i_5\}$
Relevância dos itens mais relevantes	$\{1.0, 1.0\}$	$\{1.0, 1.0\}$	$\{1.0, 1.0\}$
$RCG_2$	2.0	0.0	1.0
$IRCG_2$	2.0	2.0	2.0
<b><math>nRCG_2</math></b>	<b>1.0</b>	<b>0.0</b>	<b>0.5</b>
$DRCG'_2$	1.631	0.0	0.676
$IDRCG'_2$	1.631	1.631	1.631
<b><math>nDRCG'_2</math></b>	<b>1.0</b>	<b>0.0</b>	<b>0.414</b>

**Tabela 5.8:** Valores das métricas para listas ordenadas para o caso de exemplo.

Como se observa na Tabela 5.8, os valores de nRCG, nDRCG e nDRCG' avaliam as recomendações de acordo com as classificações dos utilizadores, reflectindo maiores diferenças entre as três recomendações apresentadas. Como tal, será também utilizada nos testes a métrica nDRCG.

A proposta apresentada para considerar as classificações de forma relativa e não absoluta foi ao nível do cálculo do valor do  $rel_i$ . Contudo, apresentamos também outra abordagem alternativa que consiste em não alterar o cálculo do  $rel_i$ , alterando em vez disso a equação da métrica, de modo a não considerar apenas o valor ideal, mas também o pior valor para os ganhos acumulados. Ou seja, no caso do nCG, sendo WCG o valor de CG para o pior caso (i.e., para a lista de  $p$  itens com piores classificações), o nARCG (ganho acumulado normalizado relativo alternativo) poderia ser calculado do seguinte modo:

$$nARCG_p = \frac{CG_p - WCG_p}{ICG_p - WCG_p} \quad (5.50)$$

$$nARCG_p = \begin{cases} \frac{CG_p - WCG_p}{ICG_p - WCG_p} & \text{se } ICG_p \neq WCG_p \\ 1 & \text{caso contrário} \end{cases}$$

De forma análoga poderiam ser calculadas as seguintes métricas: nADRCG e nADRCG'.

## 5.4 Avaliação da distância entre predições e valores reais

Nesta secção são apresentados os resultados obtidos após a realização da avaliação da distância entre as predições de cada técnica e os valores reais das classificações atribuídas pelos utilizadores. Depois, é apresentada uma análise a estes resultados.

### 5.4.1 Resultados

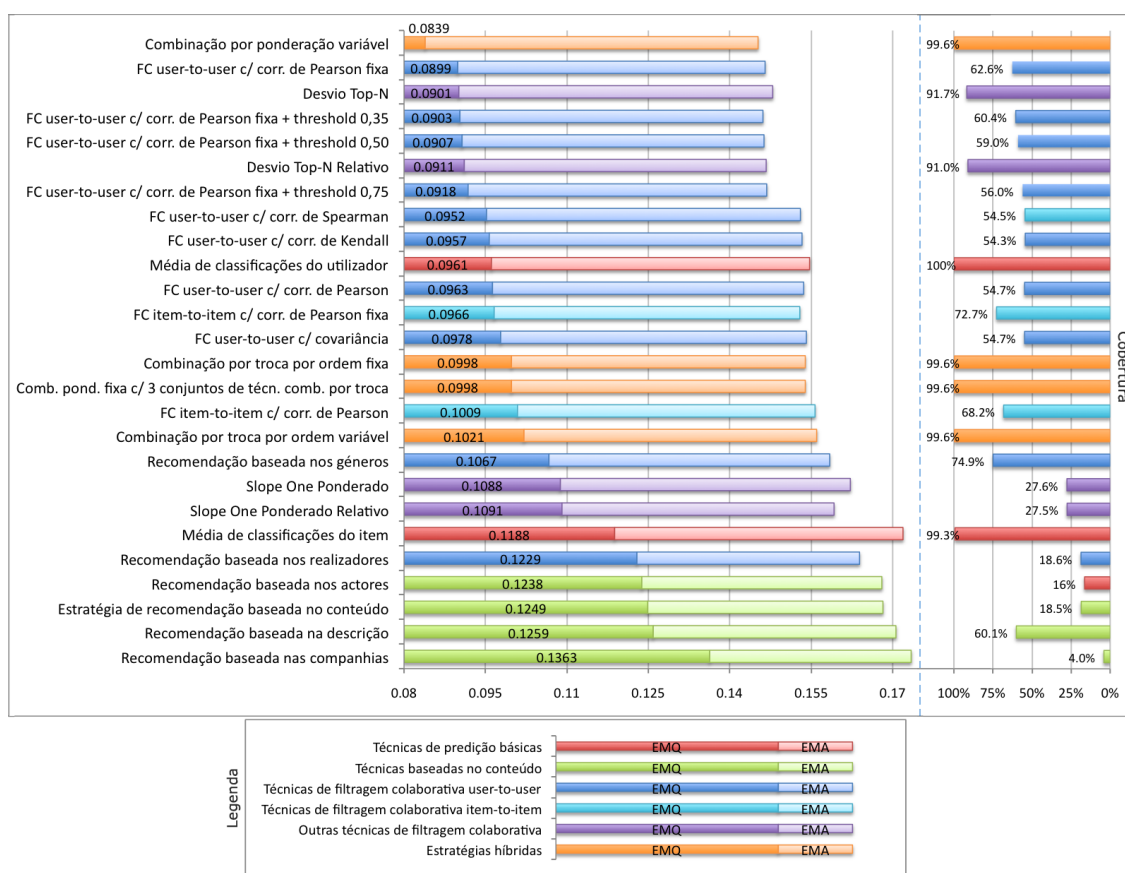
A Figura 5.19 apresenta os resultados obtidos para a avaliação da distância entre o valor das predições e as classificações reais.

### 5.4.2 Discussão

Nas secções seguintes serão analisados os resultados obtidos através da avaliação realizada. Começa-se por discutir os resultados das técnicas de predição básicas, seguidas das técnicas baseadas no conteúdo, técnicas de filtragem colaborativa e terminando com a análise das estratégias híbridas que combinam diferentes tipos de técnicas.

#### 5.4.2.1 Técnicas de predição básicas

Os razoáveis resultados da média de classificações dos utilizadores (Equação 3.18), demonstram que pode servir como um valor de referência na inferência das



**Figura 5.19:** Resumo dos resultados obtidos de acordo com a avaliação da distância entre predições e valores reais, bem como da cobertura de cada técnica de predição.

classificações. Note-se que muitas das técnicas apresentadas neste trabalho utilizam a média de classificações do utilizador como valor de inferência base, à qual é somado um outro valor (positivo ou negativo), dependendo da técnica. Veja-se, por exemplo, as técnicas de filtragem colaborativa *user-to-user* (Equação 3.33).

Contudo, por outro lado, a média de classificações dos itens (Equação 3.19) apresenta um erro mais elevado, o que se justifica por duas razões. Por um lado, como já foi verificado na Figura 5.17, a gama de classificações utilizada por cada utilizador varia, pelo que a classificação de 5 de um utilizador pode representar o mesmo do que um valor de classificação de 7 para outro utilizador. Esta discrepância afecta técnicas como esta que ignoram esta variação de classificações, calculando apenas a média geral sem considerar que as classificações dadas por cada utilizador dependem do seu perfil de classificação. Por outro lado, estes resultados demonstram que não existe muito consenso ao nível das preferências dos utilizadores pelos programas, daí o interesse em aplicar técnicas de filtragem colaborativa que visam encontrar apenas os utilizadores com preferências semelhantes.

Ambas as técnicas de predição básicas têm a vantagem de ter uma cobertura muito elevada.

#### 5.4.2.2 Técnicas de recomendação baseadas no conteúdo

A estratégia de recomendação baseada no conteúdo (descrita na Secção 3.2.2) não se mostrou competitiva em relação às outras técnicas, uma vez que apresenta elevados valores de EMQ e EMA. Isto demonstra que a maioria dos utilizadores não rege os seus interesses apenas por alguns atributos de um programa, mas acima de tudo pela qualidade percebida que não está explícita nos conteúdos descritivos. Além disso, note-se a dificuldade em aferir a similaridade entre itens com base nos seus conteúdos, apesar do conjunto de dados criado ser rico em informações descritivas. Como tal, esta estratégia pode servir de complemento a outras, nomeadamente quando o utilizador tem poucos itens classificados em comum com outros utilizadores ou quando o item em causa tiver sido sujeito a poucas ou nenhuma classificações.

#### 5.4.2.3 Técnicas de recomendação por filtragem colaborativa

Várias métricas de similaridade foram utilizadas na filtragem colaborativa, tendo-se também obtido diferentes resultados. No que diz respeito às técnicas de filtragem colaborativa *user-to-user*, a correlação de Pearson (Equação 3.35), a correlação de Spearman (Equação 3.37), a correlação de Kendall (Equação 3.38) e a covariância (Equação 3.39) apresentam resultados muito parecidos. Apesar da correlação de Pearson ser mais utilizada, estes resultados demonstram que as outras técnicas de similaridade são boas alternativas à correlação de Pearson, até porque as correlações de Spearman e de Kendall apresentam melhores resultados que a correlação de Pearson. Os resultados revelam ainda que a escolha da técnica de cálculo de similaridade poderá apenas melhorar as previsões numa proporção inferior a 3%. Como tal, a qualidade das técnicas de filtragem colaborativa dependem acima de tudo no algoritmo de predição propriamente dito, independentemente do cálculo das similaridades entre os utilizadores.

No caso da filtragem colaborativa *item-to-item* (Equação 3.41), verifica-se que esta técnica apresenta um resultado 5% pior que a técnica análoga *user-to-user*, embora tendo uma cobertura muito superior.

As variantes correlação de Pearson fixa *user-to-user* e *item-to-item* permitiram uma redução de 8% e 5% no EMQ respectivamente, o que comprova que no cálculo da similaridade é preferível comparar as classificações com o valor intermédio (neste caso 0.5), do que com a média de classificações do utilizador. Isto demonstra que uma classificação abaixo da média não tem necessariamente conotação negativa, pelo que é mais interessante considerar apenas como classificações negativas aquelas que estão abaixo do valor intermédio.

Foi também testada a inclusão de um *threshold*, de forma a que apenas os utilizadores cuja similaridade está acima de um determinado valor é que são considerados. Como seria de esperar, a cobertura diminui quanto maior for o *threshold* aplicado. Contudo, mesmo com um *threshold* de 75%, a diminuição de cobertura foi apenas de cerca de 14%. A inclusão de *threshold* não melhorou de forma considerável os resultados, o que demonstra que a filtragem de similaridades serve principalmente para reduzir a cobertura. Verifica-se que os utilizadores com menores similaridades não prejudicam

significativamente os resultados, até porque o cálculo das predições em filtragem colaborativa é ponderados de acordo com a similaridade, tendo-se que os valores de utilizadores com similaridades baixas pesam de forma proporcional (i.e., pesam pouco) no resultado final.

Além das técnicas referidas, foram implementadas também outras duas técnicas mais simples: desvio Top-N (Equação 3.43) e Slope One (Equação 3.45). A cobertura do Slope One é relativamente baixa, uma vez que esta técnica necessita da correspondência de pelo menos um item entre o utilizador que é alvo da predição e o utilizador sobre o qual se calcula a similaridade, além de ser necessário que o utilizador similar tenha classificado o item ao qual se pretende realizar a predição. Este último requisito é o único necessário para o desvio Top-N. O desvio Top-N surge como uma das melhores técnicas, sendo uma excelente alternativa às técnicas mais complexas de filtragem colaborativa. O desvio Top-N permite ainda tornar o sistema mais eficiente, dada a sua simplicidade de cálculo, e tem uma cobertura muito elevada.

As variantes desvio Top-N relativo e Slope One relativo apresentaram resultados muito semelhantes às técnicas originais.

#### 5.4.2.4 Estratégias híbridas

É difícil comparar as estratégias híbridas com as outras técnicas uma vez que estas estratégias têm uma cobertura muito superior. Perante os resultados, conclui-se que a participação de várias técnicas (no caso da estratégia de ponderação variável) numa predição permite obter resultados muito melhores do que a predição recorrendo apenas à técnica potencialmente melhor. De realçar os óptimos resultados da estratégia híbrida por ponderação variável, a qual obteve uma melhoria superior a 20% no erro médio quadrático face às restantes estratégias híbridas, excluindo a estratégia de ponderação fixa. Como tal, esta estratégia consegue ter ao mesmo tempo uma cobertura total, combinada com óptimos resultados. A participação colectiva de várias técnicas permite que o erro de algumas técnicas possa ser atenuado pela precisão de outras, em vez de atribuir a responsabilidade apenas a uma técnica.

As restantes estratégias apresentaram resultados semelhantes entre elas e, como visam abranger todas as classificações, acabam por perder alguma precisão, uma vez que em muitos casos são obrigadas a recorrer a técnicas piores (nomeadamente a estratégia baseada no conteúdo e a média de classificações do item) por serem as únicas que podem fazer essas predições. A estratégia por ponderação variável ultrapassa este problema por combinar várias técnicas ao mesmo tempo, diluindo eventuais erros de algumas técnicas em determinadas situações.

## 5.5 Avaliação das listas de itens recomendadas

O principal foco do trabalho foi implementar e comparar técnicas que melhor inferem a classificação de um utilizador a um item. Num contexto real, esta abordagem pode ser aplicada num sistema de televisão digital em que para cada programa, é apresentada a classificação inferida, dando informação ao utilizador se este gostará ou não do

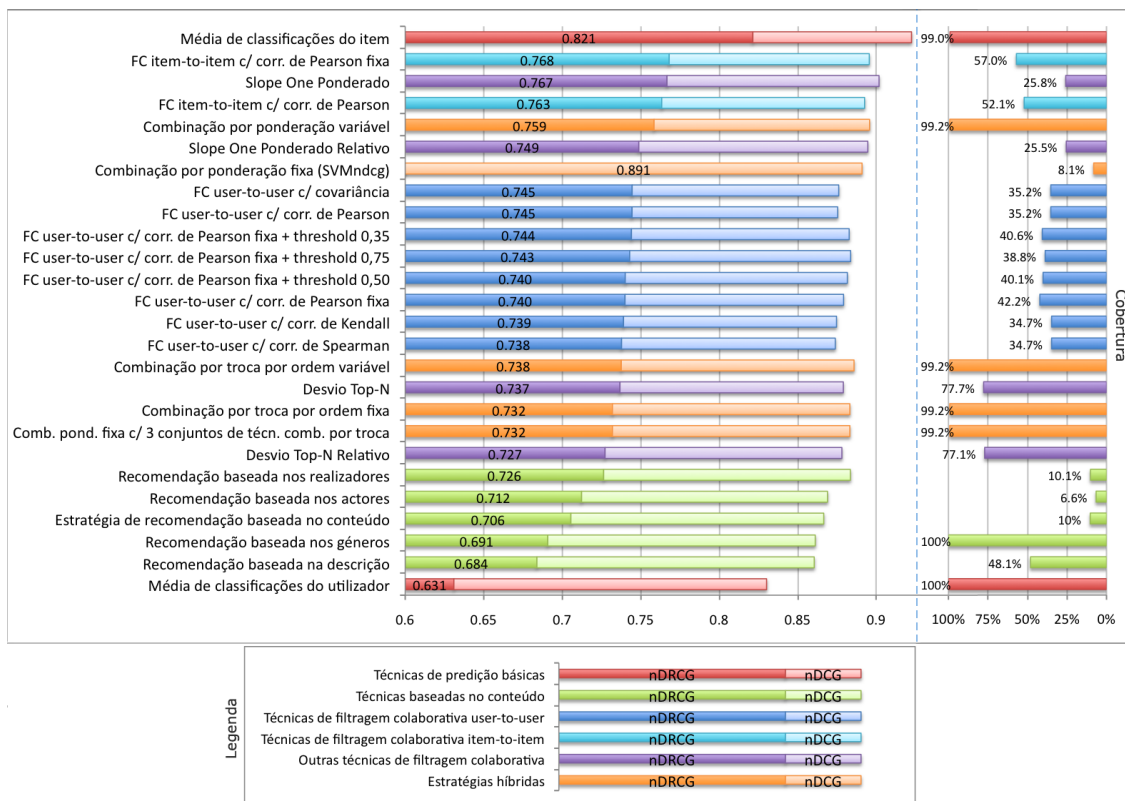
programa em causa, entre outras possíveis soluções de aplicação.

Todavia, foi feita também uma análise num contexto de recomendação de listas de itens. Neste caso, o objectivo consiste em recomendar uma lista limitada de itens ordenados de acordo com a preferência do utilizador, pelo que os itens cujo utilizador tem menos interesse não são avaliados. Ou seja, esta abordagem foca-se exclusivamente nos itens potencialmente de maior interesse para o utilizador. Estes dois tipos de avaliação têm objectivos diferentes e, de acordo com a aplicação em causa, deve-se dar mais ênfase a uma ou outra estratégia de avaliação, sendo que neste trabalho foi dada maior relevância à primeira abordagem, uma vez que esta é a estratégia de avaliação que a maioria dos trabalhos na área usam.

Nesta secção serão apresentados os resultados desta estratégia de avaliação, bem como uma pequena análise dos resultados.

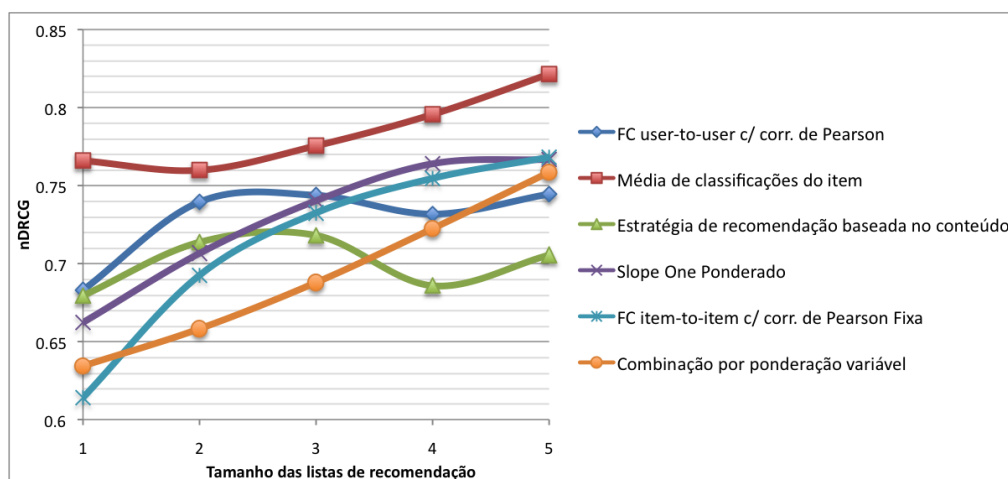
### 5.5.1 Resultados

A Figura 5.20 apresenta os resultados da avaliação realizada às listas de itens recomendados por cada técnica a cada utilizador. Neste caso, foram utilizadas listas de 5 elementos. Na Figura 5.21 é apresentada a variação do nDRCG consoante o número de itens considerados na avaliação das listas recomendadas aos utilizadores.



**Figura 5.20:** Resumo dos resultados obtidos de acordo com a avaliação das listas de 5 itens recomendadas.

Repare-se que a Figura 5.20 inclui também os resultados estratégia híbrida de



**Figura 5.21:** Variação do valor de nDRCG consoante o número de itens considerados nas listas de recomendação.

combinação por ponderação fixa de *learning-to-rank*, tal como descrito na Secção 4.1.5, usando 25% do conjunto de dados para treino. Esta estratégia foi implementada apenas usando o algoritmo  $SVM^{nDCG}$ , pelo que só foram calculados os valores da métrica nDCG (que são consideravelmente superiores aos da métrica nDRCG).

### 5.5.2 Discussão

No processo de avaliação das listas recomendadas por cada técnica, a média de classificações dos itens destaca-se com os melhores resultados. Este resultado demonstra que ao nível dos itens preferenciais dos utilizadores, existe um elevado consenso, sendo que o mesmo consenso não existe nos itens com classificações mais baixas, daí o mau resultado desta técnica perante a avaliação do erro.

Verifica-se ainda que as técnicas de filtragem colaborativa *item-to-item* são melhores neste tipo de abordagem, significando que existe uma elevada correlação entre os vectores de classificação dos itens mais populares, daí que a similaridade entre itens seja preferível neste caso em detrimento do cálculo da similaridade entre utilizadores.

Ao contrário dos resultados da outra estratégia de avaliação, o Slope One e o Slope One Relativo desta vez apresentaram melhores resultados que o Desvio Top-N e o Desvio Top-N Relativo.

Mais uma vez constata-se que a estratégia de combinação por ponderação variável diminui o erro das outras técnicas, apesar de neste caso ter piores resultados que a média de classificações dos itens. O algoritmo de *learning-to-rank* implementado permitiu uma melhoria de menos de 1% face à estratégia de combinação por ponderação variável, utilizando 25% do conjunto de classificações para treino.

Na Figura 5.21 verifica-se que quando as listas têm um maior número de elementos, o erro associado à inclusão de determinadas técnicas é atenuado perante a correta inclusão de outras. Os melhores resultados da avaliação com listas de 5 elementos são também justificados pelo facto do número mínimo de itens classificados por utilizador ser de

cinco, pelo que para esses utilizadores, que são a maioria, a dificuldade consiste apenas em descobrir a ordenação óptima dos cinco elementos, não havendo qualquer elemento fora da lista.

## 5.6 Conclusões

Os resultados deste capítulo revelam que a combinação de diferentes tipos de técnicas de recomendação pode melhorar a qualidade das recomendações. Apesar das técnicas de filtragem colaborativa com métricas de similaridade apresentarem bons resultados, o Desvio Top-N e as suas variantes apresentam resultados igualmente bons, com a vantagem de serem mais simples de calcular e exigirem menos dados históricos.

A estratégia híbrida por ponderação fixa usando aprendizagem automática apresentou resultados muito diferentes na fase de treino e na fase de teste, revelando que não existe uma distribuição de pesos ideal, pelo que a combinação por ponderação variável que adapta o método de inferência a cada utilizador apresenta resultados aproximadamente iguais, sem necessitar de uma divisão em conjunto de treino e conjunto de teste.







## Capítulo 6

# Conclusões

Neste trabalho procurou-se estudar a área dos sistemas de recomendação, nomeadamente através de uma aplicação no contexto da recomendação de programas de televisão, domínio que até hoje tem sido relativamente pouco explorado. Estes sistemas têm de lidar com a subjectividade e dependência do contexto associada às opiniões dos seres humanos, sendo esse um dos principais desafios na construção de sistemas que procuram ir de encontro aos interesses dos utilizadores. Para aumentar a certeza nas recomendações, é muito importante a contribuição (inconsciente) de todos os utilizadores que com as suas classificações permitem que os outros utilizadores e eles próprios sejam alvo de melhores recomendações e, deste modo, têm uma melhor experiência com a aplicação (que no caso deste trabalho poderia ser uma televisão).

Neste trabalho demonstrou-se que as estratégias híbridas por ponderação fixa pecam por assumir que existem pesos ideais para a inferência de classificações para todos os utilizadores, ignorando que cada utilizador tem um perfil de classificação diferente, e cada um identifica-se mais com algumas técnicas de recomendação, e outras identificam-se com outras. É neste sentido que a estratégia de recomendação híbrida por ponderação variável apresentada neste trabalho permite adaptar a estratégia a cada utilizador ao longo do tempo.

Foram apresentadas duas abordagens diferentes para avaliação de sistemas de recomendação: inferir a classificação a todos os itens e posteriormente calcular o erro ou calcular a predição de listas limitadas de itens recomendados. Comprovou-se que estas abordagens têm resultados diferentes e existem técnicas que têm resultados muito bons perante uma abordagem e muito maus perante outra. Como tal, é importante avaliar os objectivos do sistema que será implementado para depois dar mais ênfase a uma das duas abordagens.

O domínio da recomendação de programas de televisão continua ainda pouco explorado, contudo a massificação da televisão digital abrirá portas para um maior desenvolvimento deste tipo de sistemas neste contexto.

## 6.1 Sumário das contribuições

As principais contribuições deste trabalho foram as seguintes:

- Foi criada uma colecção de testes para o domínio específico da recomendação programas de televisão. Esta colecção tem características semelhantes às de outras colecções de teste existentes na área, tendo-se que estas outras são focadas em domínios de aplicação diferentes. A colecção desenvolvida neste trabalho permite a realização de experiências futuras no domínio concreto de aplicação à recomendação de programas de televisão.
- Foi feita uma análise comparativa de diferentes técnicas de recomendação existentes no domínio concreto da recomendação de programas de televisão, tendo-se obtido resultados diferentes consoante a estratégia de avaliação considerada:
  - Ao avaliar da distância entre predições e valores reais, concluiu-se que as técnicas de filtragem colaborativa *user-to-user* usando correlação de Pearson fixa e o Desvio Top-N produzem os melhores resultados, sendo que as técnicas baseadas no conteúdo, além de terem uma baixa cobertura, apresentam os piores resultados.
  - Perante a outra estratégia de avaliação, ou seja, a avaliação das listas de  $n$  itens preferenciais, a média de classificações dos itens destacou-se como sendo a melhor técnica e os piores resultados foram apresentados pelas técnicas baseadas no conteúdo e pela média de classificações dos utilizadores.
- Após a análise de diferentes técnicas, estas foram combinadas através de estratégias híbridas, utilizando diferentes abordagens, tendo-se concluído que a combinação por ponderação variável e a combinação por ponderação fixa apresentaram os melhores resultados. De notar que ambas as estratégias utilizam aprendizagem automática para melhorar as predições, sendo que no caso da ponderação variável a aprendizagem é feita incrementalmente, enquanto que na ponderação fixa a aprendizagem é feita *offline* uma única vez utilizando um conjunto de treino específico através de algoritmos de *learning-to-rank*.
- Foram apresentadas novas métricas de avaliação que consideram a gama de classificações de cada utilizador. As métricas referidas são as seguintes: o nRCCG, o nDRCCG, o nDRCCG', o nARCCG, o nDARCCG e o nDARCCG'.

## 6.2 Trabalho futuro

Dada a limitação de tempo associada à dissertação de mestrado bem como à necessidade de prioritização das tarefas realizadas, não foi possível aprofundar todos os aspectos mencionados. A maioria das técnicas propostas exige um processamento elevado dos dados, percorrendo a matriz de classificações utilizador/item. Este tipo de

técnicas tem problemas de eficiência perante conjuntos de classificações elevados, podendo consequentemente afectar a qualidade de experiência dos utilizadores perante um sistema de recomendação. Os modelos probabilísticos, tais como as redes Bayesianas e máquinas de Boltzmann, são alternativas eficientes a esta abordagem, uma vez que o processamento mais pesado é realizado *offline*.

Ao nível das estratégias híbridas, e dados os bons resultados da estratégia de combinação por ponderação variável, fica para trabalho futuro a adaptação desta estratégia ao contexto da recomendação de listas de itens, ou seja, considerando os valores de nDCG ou nDRCG antigos para cada utilizador, em vez do EMA ou EMQ. Pela razão oposta, poderia ser desenvolvida uma variante da estratégia de ponderação fixa com aprendizagem automática descrita neste trabalho de modo a maximizar o EMA ou EMQ, em substituição do nDCG.

Os resultados apresentados utilizam apenas o conjunto de dados criado, contudo, para efectivamente avaliar as novas métricas de avaliação de sistemas com classificações *n*-árias em listas ordenadas bem como as novas técnicas e estratégias propostas seria necessário experimentar com outros conjuntos de dados e, deste modo, comparar os resultados.

Além disso, a validação realizada limita-se a considerar os itens já avaliados pelos utilizadores presentes conjunto de dados, ignorando os restantes itens, pelo que deveria ser realizada uma sessão de testes em tempo real com utilizadores presenciais. Ou seja, perante uma situação real, à medida que o utilizador interagisse com o sistema, ser-lhe-iam apresentadas recomendações às quais o utilizador apresentaria a sua opinião. Perante a interacção e feedback do utilizador, seria possível comparar os valores de predição com os valores reais de classificação atribuídos pelo utilizador. Este tipo de avaliação consiste numa simulação da realidade, pelo que obteria resultados mais credíveis.

Neste trabalho não foi considerado nenhum dado adicional referente às classificações. Seria interessante considerar a data das classificações, procurando desenvolver sistemas para este domínio que ponderassem as classificações do utilizador de acordo com a sua antiguidade (i.e., dando mais relevância às classificações mais recentes e menor relevância às antigas, consoante uma dada função ao longo do tempo).



# Bibliografia

- AARTS, E. & KORST, J. (1988). *Simulated Annealing and Boltzmann Machines: A Stochastic Approach to Combinatorial Optimization and Neural Computing*. Wiley New York.
- ACKLEY, D., HINTON, G. & SEJNOWSKI, T. (1985). A learning algorithm for Boltzmann machines. *Cognitive science*.
- ADOMAVICIUS, G. & TUZHILIN, A. (2001). Expert-driven validation of rule-based user models in personalization applications. *Data Mining and Knowledge Discovery*.
- ADOMAVICIUS, G. & TUZHILIN, E. (2005). 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*.
- ALI, K. & VAN STAM, W. (2004). TiVo: making show recommendations using a distributed collaborative filtering architecture. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*.
- ANDERSON, M., BALL, M., BOLEY, H., GREENE, S., HOWSE, N., LEMIRE, D. & MCGRATH, S. (2003). Racofi: A rule-applying collaborative filtering system. *Proceedings of Collaboration Agents Conference (COLA)*.
- BALABANOVIC, M. & SHOHAM, Y. (1997). Fab: content-based, collaborative recommendation. *Communications of the ACM*.
- BAR-ILAN, J., KEENOY, K., YAARI, E. & LEVENE, M. (2007). User rankings of search engine results. *Journal of the American Society for Information Science and Technology*.
- BILLSUS, D., PAZZANI, M. & CHEN, J. (2000). A learning agent for wireless news access. In *Proceedings of the 5th international conference on Intelligent user interfaces*.
- BOARDMAN, P., OLIVER, S. & HUBBARD, S. (2003). SiteSeer: visualisation and analysis of transcription factor binding sites in nucleotide sequences. *Nucleic acids research*.
- BROZOVSKY, L. & PETRICEK, V. (2007). Recommender system for online dating service. In *Proceedings of Znalosti Conference*.
- BURKE, R. (2002). Hybrid recommender systems: Survey and experiments. *User Modeling and User-Adapted Interaction*.
- CARMAGNOLA, F., CENA, F., CONSOLE, L., GRILLO, P., VERNERO, F., SIMEONI, R. & PERRERO, M. (2009). iDYNamicTV: a social adaptive television experience. In *Proceedings of the 20th ACM conference on Hypertext and hypermedia*.

- CHAKRABARTI, S., KHANNA, R., SAWANT, U. & BHATTACHARYYA, C. (2008). Structured learning for non-smooth ranking losses. In *Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*.
- CHAMBERLIN, D. (2003). XQuery: A query language for XML. In *Proceedings of the 2003 ACM SIGMOD international conference on Management of data*.
- DE CAMPOS, L., FERNANDEZ-LUNA, J. & HUETE, J. (2006). A Bayesian network approach to hybrid recommending systems. In *Proceedings of the 11th International Conference of Information Processing and Management of Uncertainty in Knowledge-Based Systems*.
- FU, X., BUDZIK, J. & HAMMOND, K. (2000). Mining navigation history for recommendation. In *Proceedings of the 5th international conference on Intelligent user interfaces*.
- HECKERMAN, D. *et al.* (1999). A tutorial on learning with Bayesian networks. In *In Learning in Graphical Models*.
- HERLOCKER, J. (2000). Understanding and improving automated collaborative filtering systems.
- HERLOCKER, J., KONSTAN, J. & RIEDL, J. (2000). Explaining collaborative filtering recommendations. In *Proceedings of the 2000 ACM conference on Computer supported cooperative work*.
- HERLOCKER, J., KONSTAN, J., TERVEEN, L. & RIEDL, J. (2004). Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems (TOIS)*.
- HINTON, G., SEJNOWSKI, T. & ACKLEY, D. (1984). Boltzmann machines: Constraint satisfaction networks that learn. *Cognitive Science*.
- HOFMANN, T. (2004). Latent semantic models for collaborative filtering. *ACM Transactions on Information Systems (TOIS)*.
- HSU, S., WEN, M., LIN, H., LEE, H. & LEE, C. (2007). Amed - a personalized tv recommendation system. In *Proceedings of the 5th European Interactive TV Conference*.
- JARVELIN, K. & KEKALAINEN, J. (2002). Cumulated gain-based evaluation of IR techniques. *ACM Transactions on Information Systems (TOIS)*.
- KOTSIANTIS, S. (2007). Supervised machine learning: A review of classification techniques.
- LEMIRE, D. & MACLACHLAN, A. (2005). Slope One Predictors for Online Rating-Based Collaborative Filtering.
- LINDEN, G., SMITH, B. & YORK, J. (2003). Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet computing*.
- LIU, T. (2009). Learning to rank for information retrieval. *Foundations and Trends in Information Retrieval*.



- MANNING, C., RAGHAVAN, P. & SCHATZ, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press, 1st edn.
- MOONEY, R. & ROY, L. (2000). Content-based book recommending using learning for text categorization. In *Proceedings of the 5th ACM conference on Digital libraries*.
- MYERS, J. & WELLS, A. (2003). *Research design and statistical analysis*. Lawrence Erlbaum.
- NEAPOLITAN, R. (2003). *Learning bayesian networks*. Pearson Prentice Hall.
- O'CONNOR, M. & HERLOCKER, J. (2001). Clustering items for collaborative filtering. In *Proceedings of SIGIR-2001 Workshop on Recommender Systems, New Orleans, LA*.
- PAZZANI, M. & BILLISUS, D. (1997). Learning and revising user profiles: The identification of interesting web sites. *Machine learning*.
- PAZZANI, M. & BILLISUS, D. (2007). Content-based recommendation systems.
- RESNICK, P. & VARIAN, H. (1997). Recommender systems. *Communications of the ACM*.
- RESNICK, P., IACOVOU, N., SUCHAK, M., BERGSTROM, P. & RIEDL, J. (1994). GroupLens: An open architecture for collaborative filtering of netnews. In *Proceedings of the ACM conference on Computer supported cooperative work*.
- RUSSELL, S., NORVIG, P., CANNY, J., MALIK, J. & EDWARDS, D. (1995). *Artificial intelligence: a modern approach*. Prentice hall, 2nd edn.
- SALAKHUTDINOV, R., MNIN, A. & HINTON, G. (2007). Restricted Boltzmann machines for collaborative filtering. In *Proceedings of the 24th international conference on Machine learning*.
- SHANI, G. & GUNAWARDANA, A. (2009). Evaluating Recommendation Systems. Tech. rep., Microsoft Research.
- SILL, J., TAKACS, G., MACKAY, L. & LIN, D. (2009). Feature-weighted linear stacking.
- SIMEONI, R., GEYMONAT, M., ANTONINI, R., MONTANARI, R. & PERRERO, M. (2009). DynamicTV: a new Inter-tainment paradigm for television.
- SU, X. & KHOSHGOFTAAR, T.M. (2009). A Survey of Collaborative Filtering Techniques. *Advances in Artificial Intelligence*.
- SWEARINGEN, K. & SINHA, R. (2001). Beyond algorithms: An hci perspective on recommender systems. In *Proceedings of the ACM SIGIR. Workshop on Recommender Systems*, 5-6.
- TRUYEN, T.T., PHUNG, D.Q. & VENKATESH, S. (2009). Ordinal boltzmann machines for collaborative filtering. In *Proceedings of the 25th Conference on Uncertainty in Artificial Intelligence (UAI)*.
- TURATI, A., DELL'AGLIO, D., CERIZZA, D., CELINO, I., DELLA VALLE, E., ASSESSOR, Q. & STEINMETZ, N. (2008). D5-1: User and Service Clustering Research Report - Version1.

- VAN SETTEN, M. (2005). Supporting people in finding information: hybrid recommender systems and goal-based structuring.
- VAN SLOBBE, M. (2005). Lighthouse: A distributed recommendations system for television programs. *Communications of the ACM*.
- WANG, J., DE VRIES, A. & REINDERS, M. (2006). Unifying user-based and item-based collaborative filtering approaches by similarity fusion. In *Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval*.
- WITTEN, I. & FRANK, E. (1999). *Data mining: practical machine learning tools and techniques with Java implementations*. Morgan Kaufmann.
- ZAIER, Z., GODIN, R. & FAUCHER, L. (2008). Evaluating Recommender Systems. In *Proceeding of Automated solutions for Cross Media Content and Multi-channel Distribution*.
- ZHANG, Y. & KOREN, J. (2007). Efficient bayesian hierarchical user modeling for recommendation system. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*.