
# PUC

ISSN 0103-9741

Monografias em Ciência da Computação
n° 19/10

# Sistemas de Recomendação: uma abordagem por filtro colaborativo baseado em modelos

**Dárlinton Barbosa Feres Carvalho**
**Ruy Luiz Milidiú**
**Carlos José Pereira de Lucena**

Departamento de Informática

**PONTIFÍCIA UNIVERSIDADE CATÓLICA DO RIO DE JANEIRO**
**RUA MARQUÊS DE SÃO VICENTE, 225 - CEP 22453-900**
**RIO DE JANEIRO - BRASIL**

Monografias em Ciência da Computação, No. 19/10 ISSN: 0103-9741
Editor: Prof. Carlos José Pereira de Lucena Dezembro, 2010

# Sistema de recomendação: uma abordagem por filtro colaborativo baseado em modelos [1]

**Dárlinton Barbosa Feres Carvalho, Ruy Luiz Milidiú, Carlos José Pereira de Lucena**

{darlinton, milidiu, lucena}@inf.puc-rio.br

**Resumo.** Sistemas de recomendação estão cada vez mais presentes em aplicações na Internet e seus usuários mais dependentes de suas recomendações. Esses sistemas são bem diversos, com diferentes objetivos e baseados em técnicas variadas. Este trabalho apresenta um estudo exploratório do uso de algoritmos de aprendizado de máquina para o desenvolvimento de sistemas de recomendação conhecidos como sistemas de recomendação por filtro colaborativo. Os experimentos são realizados considerando um problema clássico de aprendizado de máquina e conjunto de dados da literatura de sistemas de recomendação por filtro colaborativo. São apresentados quatro modelos para representar o conjunto de dados no experimento e avaliados três algoritmos de aprendizado de máquina. Os resultados mostram que os algoritmos produzem resultados melhores a partir de modelos com mais dados, embora demandem mais recursos. Fica evidenciado que é necessário um estudo de diversas técnicas e configurações ao se implementar um sistema de recomendação por filtro colaborativos.

**Palavras-chave:** Sistemas de Recomendação, Filtro Colaborativo, Aprendizado de Máquina, Filtro Colaborativo baseado em modelos.

**Abstract.** On the Internet, an increasing number of applications are using recommender; consequently, the system users are becoming increasingly reliant on their recommendations. The present work explores the use of machine learning algorithms to implement a recommendation system based on the collaborative filtering technique. The experiment uses a classic machine-learning problem based on a dataset from the literature of collaborative filtering systems. Four models are used to represent the dataset are defined and solved by three machine-learning algorithms. The results show that the machine-learning algorithms produces better results from richer models, but requires more computational resources. The trade-off analysis shows evidences of the benefits for trying different techniques and configurations on the development of new recommendation systems.

**Keywords:** Recommender System, Collaborative Filtering, Machine Learning, Model-based Collaborative Filtering.

[1] Trabalho patrocinado pelo Ministério de Ciência e Tecnologia da Presidência da República Federativa do Brasil através da Bolsa de Doutorado CNPq 142620/2009-2.

**In charge of publications:**
Rosane Teles Lins Castilho
Assessoria de Biblioteca, Documentação e Informação
PUC-Rio Departamento de Informática
Rua Marquês de São Vicente, 225 - Gávea
22451-900 Rio de Janeiro RJ Brasil
Tel. +55 21 3527-1516 Fax: +55 21 3527-1530
E-mail: bib-di@inf.puc-rio.br
Web site: http://bib-di.inf.puc-rio.br/techreports/

# Sumário

# 1 Introdução

Aplicativos modernos, principalmente os que funcionam na Internet, possuem mecanismos para aprender as preferências de seus usuários. Alguns usuários já acreditam na capacidade dos computadores de anteciparem suas vontades, antes mesmo delas existirem (GROSSMAN 2010). Outros já se apoiam quase que exclusivamente nas recomendações disponibilizadas por estes sistemas na hora de tomar decisões, principalmente para compras. Estas aplicações quase mágicas são conhecidas como sistemas de recomendação. E a proposta deste trabalho é estudar e experimentar técnicas usadas na construção de um sistema de recomendação.

O foco deste trabalho é em aplicar técnicas conhecidas como filtros colaborativos (FC), especialmente os baseados em modelo, que utilizem técnicas de aprendizado de máquina (Herlocker, Konstan, Borchers & Riedl 1999). Este termo foi cunhado em um dos trabalhos pioneiros em sistemas de recomendação e é amplamente utilizado, até mesmo por sistemas que não utilizam explicitamente informações de colaboração de outros usuários (Su & Khoshgoftaar 2009). A ideia básica do FC é que se usuários $X$ e $Y$ qualificam $n$ itens similares, ou tem comportamentos similares (comprando, assistindo, ouvindo), então eles irão classificar ou agir em outros itens similarmente.

Filtros colaborativos se baseiam em um banco de dados de experiências dos usuários com itens para tentar adivinhar itens adicionais de interesse para seus usuários. Em um cenário típico há $m$ usuários $\{u_1, u_2, ..., u_m\}$ e $n$ items $\{i_1, i_2, ..., i_n\}$, e cada usuário $u_1$ tem uma lista de itens $I_{ui}$, com uma classificação do usuário para cada item. As classificações podem ser indicadores explícitos de gosto e não gosto ou em escala de *Likert* (1 a 5), ou até mesmo indicações implícitas de interação como compras de itens, visualizações e *click*.

A construção de sistemas de recomendação baseados em filtro colaborativos possui muitos desafios, conforme apresentado em (Su & Khoshgoftaar 2009). As interações de usuários e itens na prática formam um conjunto de dados esparso. Esse problema ainda é agravado em situações extremas como quando o sistema tem que recomendar itens para um usuário novo, que não possui histórico de interação. Este problema como o início frio e vale tanto para um novo usuário quanto para um novo item. Outro problema recorrente é em relação à ocorrência de sinônimos, que é a tendência de um mesmo item ou outros muito similares terem nomes diferentes ou até mesmo entradas duplicadas. A maioria dos recomendadores é incapaz de descobrir esta associação latente e trata os itens como sendo diferentes. A questão sobre segurança também traz diversos problemas a serem tratados por esses sistemas. Dado que qualquer um pode prover recomendações em um sistema, também deixa em aberto para que usuários maliciosos façam muitas interações com o propósito apenas de influenciar no sistema de recomendação, no intuito de se beneficiar desse viés. Além disso, há também questões éticas e culturais, pois muitas pessoas não querem seus hábitos e experiências sejam escrutinadas. Os sistemas de recomendação tem que respeitar o direito de privacidade de seus usuários.

Outros métodos utilizados no desenvolvimento de sistemas de recomendação são os métodos baseados em memória, os baseados em modelos de aprendizado de máquina ou mineração de dados, e ainda os métodos híbridos que combinam diferentes técnicas (Herlocker et al. 1999). A finalidade dos sistemas de recomendação varia bastante de acordo com o domínio da aplicação. Ainda vale citar que existem trabalhos que se baseiam em somente conhecimento específico de domínio, regras de associação, e preferência de usuários ao fazer recomendação de itens aos usuários (Su & Khoshgoftaar 2009, Chen & Pu. 2010).

O problema a ser resolvido neste trabalho é o de predizer a qualificação de um usuário para um item não conhecido. Esse problema é fundamental na construção de sistemas de recomendação de itens a um usuário. O método proposto para resolver o problema é um filtro colaborativo definido por um método de aprendizado de máquina baseado em modelo. Exploram-se técnicas de aprendizado de máquina sobre dados de usuários para construir um sistema de recomendação. Para isso, são avaliados diversos modelos resolvidos por diferentes técnicas de aprendizado de máquina.

O restante deste artigo está organizado como segue. Na Seção 2, detalha-se o problema de predição de uma qualificação e o conjunto de dados utilizado. A Seção 3 descreve as técnicas e os modelos utilizados para o aprendizado de máquina, bem como os resultados os resultados obtidos. O artigo é concluído na Seção 4.

# 2 Metodologia

Sistemas de recomendação definem uma classe de sistemas que podem ser desenvolvidos utilizando diversas técnicas e com diferentes propósitos, mas com a característica em comum de prover recomendações de itens a seus usuários. Neste trabalho, são considerados os sistemas de recomendação que aproveitam informações de diversos usuários para realizar recomendações específicas, sendo que este tipo de técnica é conhecida como filtro colaborativo (Su & Khoshgoftaar 2009). Há diferentes técnicas para programar um filtro colaborativo e neste trabalho consideram-se os métodos baseados em modelos de aprendizado de máquina. Ainda assim, esses métodos baseados em modelos podem ser definidos com propósitos distintos, por isso considerando diferentes problemas a serem resolvidos. Nesta seção, é apresentado o conjunto de dados utilizado para exercitar os modelos e técnicas de aprendizado, bem como também é definido formalmente o problema a ser resolvido com a métrica de avaliação.

## 2.1 Conjunto de Dados

A escolha do conjunto de dados foi pautada por uma busca a um conjunto clássico da literatura. O conjunto de dados escolhido foi o MovieLens (Herlocker et al. 1999), disponibilizado pelo Projeto de Pesquisa GroupLens Research Project[2] da University of Minnesota.

Os dados deste conjunto foram coletados a partir do site MovieLens (`movielens.umn.edu`)[3] durante um período de sete meses de 19 de Setembro de 1997 a 22 de Abril de 1998. Esses dados foram tratados e disponibilizados em três versões:

- 100 mil avaliações para 1.682 filmes por 943 usuários
- 1 milhão de avaliações para 3.900 filmes por 6.040 usuários
- 10 milhões de avaliações e 100.000 tags para 10.681 filmes por 71.567 usuários

Foi escolhida a versão com 100 mil avaliações. Os dados disponíveis neste conjunto são listados a seguir:

- 100 mil avaliações, variando de 1 a 5, a 1.682 filmes por 943 usuários

[2]Disponível em `http://www.grouplens.org/`

[3]Atualmente está disponível no endereço `http://www.movielens.org/`

- Cada usuário avaliou pelo menos 20 filmes
- Informações demográficas simplificadas sobre os usuários (idade, sexo, ocupação e CEP)
- Informações sobre o filme, como título, data de lançamento, data de lançamento em vídeo, IMDb URL, e o gênero (pode ser de vários gêneros ao mesmo tempo).

Vale acrescentar que este conjunto de dados também já com arquivos particionados para desenvolvimento e avaliação. São dois arquivos para desenvolvimento (a,b) e 5 arquivos para avaliação.

## 2.2 Problema

O problema a ser resolvido é o de se determinar a avaliação de um usuário para um item ainda não qualificado. Considera-se nesta predição todas as qualificações já realizadas pelos usuários para os itens. A Figura 1 ilustra este problema com todos os dados envolvidos (Herlocker et al. 1999). Este problema pode ser definido como um problema de classificação ou de regressão. O problema de classificação é definido considerando que a qualificação é um atributo categórico que pode assumir os valores inteiros de 1 a 5. O problema de regressão é similar ao de classificação, mas definindo o atributo classificação como numérico que pode assumir valores reais variando de 1 a 5. A escolha do problema a ser resolvido foi realizada ponderando sobre a utilização final deste preditor em um sistema de recomendação. Um sistema de recomendação pode utilizar o preditor para gerar uma lista com $n$ itens que obtiveram a melhor qualificação, e oferecer esta lista como recomendação para o usuário. Para ajudar no desempate na ordenação dos itens ao gerar esta lista, optou-se pela versão do problema de regressão, que considera a qualificação como um número real.

| | Star Wars | Hoop Dreams | Contact | Titanic |
|---|---|---|---|---|
| **Joe** | 5 | 2 | 5 | 4 |
| **John** | 2 | 5 | | 3 |
| **Al** | 2 | 2 | 4 | 2 |
| **Nathan** | 5 | 1 | 5 | ? |

Figura 1: O problema consiste em predizer a qualificação de um usuário para um item, considerando outras qualificações já conhecidas

A avaliação das técnicas de aprendizado de máquina para o problema de regressão (preditores) é feita pela raiz do erro médio ao quadrado (REMQ) (Su & Khoshgoftaar 2009). REMQ é uma medida usual da diferença entre valores preditos por um modelo e os valores de fato observados do que está sendo modelado. Esta é uma boa medida de precisão, pois as diferenças individuais das predições, que também são chamadas de residuais, são agregadas em uma única medida do poder de predição do modelo. A fórmula para cálculo do REMQ de $n$ avaliações, sendo $p_{i,j}$ a avaliação predita do usuário $i$ para o item $j$ e $c_{i,j}$ o valor correto para essa avaliação, é:

$$REMQ = \sqrt{\frac{1}{n}\sum_{i,j}(p_{i,j} - c_{i,j})^2}$$

# 3 Experimento

Para experimentar diferentes algoritmos de aprendizado de máquina, foram utilizados diferentes métodos para resolver o problema de regressão. Além disso, também foram variadas as características do conjunto de dados disponíveis aos métodos. Nesta seção são apresentadas as técnicas consideradas neste trabalho, com os diferentes modelos para os dados e os resultados obtidos por essas técnicas resolvendo-se o problema de regressão nos diferentes modelos para conjunto de dados.

Optou-se por usar o aplicativo Weka[4] para exercitar os algoritmos de aprendizagem de máquina. Este software é uma coleção de algoritmos de aprendizado de máquina para tarefas de mineração de dados. Os algoritmos podem tanto ser aplicados diretamente a um conjunto de dados através da interface da aplicação, como ser chamados a partir de código Java. O Weka contém ferramentas para o pré-processamento de dados, classificação, regressão, clusterização, regras de associação e visualização. Ele também é apropriado para o desenvolvimento de novos esquemas de aprendizado de máquina. O principal motivo para a escolha deste software é o fato dele prover boas implementações dos algoritmos de aprendizado, além de uma fácil integração com outras aplicações.

## 3.1 Técnicas

A escolha dos algoritmos utilizados como estudo de caso neste trabalho foi realizada de modo a exercitar as principais técnicas relacionadas com o problema de regressão. Para se determinar um nível inferior de desempenho foi utilizado o preditor `0-R`, que consiste em fazer a predição pelo valor médio das avaliações.

O preditor principal considerado é o REPTree, que é uma implementação para árvore de decisão rápida. O algoritmo basicamente constrói uma árvore de regressão usando informação de ganho/variância e realiza poda através poda por *reduced-error pruning (with backfitting)*. Um aprimoramento foi alcançado aplicando um comitê que combina diversas árvores geradas pelo REPTree. Para combinar as diversas árvores foi utilizado o método Bagging (Breiman 1996). Vale dizer que foram avaliados outros métodos disponíveis no Weka, mas não são reportados por não terem obtido resultados significativos.

A configuração de execução dos algoritmos é apresentada através dos esquemas utilizados no Weka.

- `0-R`: "`weka.classifiers.rules.ZeroR`".
- `REPTree`: "`weka.classifiers.trees.REPTree -M 2 -V 0.0010 -N 3 -S 1 -L -1`".
- `Bagging`$_i$: "`weka.classifiers.meta.Bagging -P 100 -S 1 -I` $i$ `-W weka.classifiers.trees.REPTree - -M 2 -V 0.0010 -N 3 -S 1 -L -1`", onde $i$ é o número de preditores considerados (5, 10, 20, 40).

## 3.2 Modelos

A partir do conjunto de dados apresentado na Seção 2.1, são definidos quatro modos de usar esses dados variando o tipo de informação disponível. Esses diferentes conjuntos de

[4] http://www.cs.waikato.ac.nz/ml/weka/

dados são chamados de modelos. O primeiro modelo considera apenas dados da avaliação de um filme por um usuário. O segundo modelo além dos dados do primeiro modelo, agrega informações sobre o usuário. O terceiro modelo segue por analogia e também possui os mesmos dados do primeiro modelo, mas agrega informações sobre o filme. O quarto modelo agrega informações sobre os usuários e filmes com a avaliação.

A seguir é apresentado um esquema da definição desses modelos:

- $M_0$: `<user_id, movie_id, rating>` - 3 atributos
  - `@ATTRIBUTE user_id NUMERIC`
  - `@ATTRIBUTE movie_id NUMERIC`
  - `@ATTRIBUTE rating NUMERIC`
- $M_u$: `<user_id, [use_data]*, movie_id, rating>` - 6 atributos

  `[user_data]` =
  - `@ATTRIBUTE user_age NUMERIC`
  - `@ATTRIBUTE user_gender {M,F}`
  - `@ATTRIBUTE user_occupation {...}` - uma lista de ocupações
- $M_m$: `<user_id, movie_id, [movie_data]*, rating>` - 22 atributos

  `[user_data]` =
  - `@ATTRIBUTE genre_xxx {0,1}` - um atributo para cada gênero
- $M_{u+m}$: `<user_id, [user_data]*, movie_id, [movie_data]*, rating>` - 25 atributos

### 3.3 Resultados

Nesta seção, são apresentados os resultados obtidos pelos algoritmos de aprendizado de máquina de acordo para os conjuntos de dados utilizados. Primeiramente, são mostrados dois gráficos com o desempenho considerando conjunto de dados de desenvolvimento. Na sequência está um gráfico consolidando o desempenho das técnicas de acordo com os modelos definidos para os conjuntos de dados.

Para avaliar os experimentos deste trabalho, utilizamos a técnica de validação cruzada (Geisser 1993). Validação cruzada é uma técnica para medir como os resultados de uma análise estatística vão ser generalizados para um conjunto de dados independente. Ela é principalmente usada em configurações onde o objetivo é a predição, e alguém deseja estimar o quão correto um modelo preditivo irá ser executado na prática. Uma rodada da validação cruzada envolve o particionamento de uma amostra de dados em subconjuntos complementares, executando a análise de um subconjunto (chamado de conjunto de treinamento), e validando a análise em outro subconjunto (chamado de conjunto de validação ou teste). Para reduzir a variabilidade, múltiplas rodadas do validação cruzada são executadas usando diferentes partições, e os resultados de validação são a média das rodadas.

O conjunto de dados MovieLens, conforme apresentado na Seção 2.1, vem com arquivos já divididos em conjunto de treinamento e teste, sendo dois conjuntos para desenvolvimento

(a,b) e cinco para validação cruzada. As 100 mil avaliações disponíveis estão divididas nos conjuntos de dados para desenvolvimento com 90.570 pra treinamento e 9.430 para teste. Nos arquivos com o conjunto de dados para a validação cruzada, a divisão é de 80.000 avaliações para treinamento e 20.000 para teste.

Para os conjuntos de dados de desenvolvimento (a,b), os resultados obtidos pelos algoritmos, apresentados na Seção 3.1, sobre os diferentes modelos (conjuntos de dados), definidos na Seção 3.2 são apresentados na Figura 2 e na Figura 3. No eixo X à esquerda do gráfico estão as medidas de desempenho dos algoritmos em relação à acurácia (quanto menor melhor), calculada pela raiz do erro médio ao quadrado conforme definido na Seção 2.2. No eixo X à direita está o tempo em segundos (quanto menor melhor) que os algoritmos gastaram para treinar os modelos de predição (treino<*algoritmo*>) e para avaliar (teste<*algoritmo*>). Vale ressaltar que a medida de tempo de execução serve apenas para ter uma noção do esforço computacional requerido. No eixo Y estão os diferentes modelos utilizados.

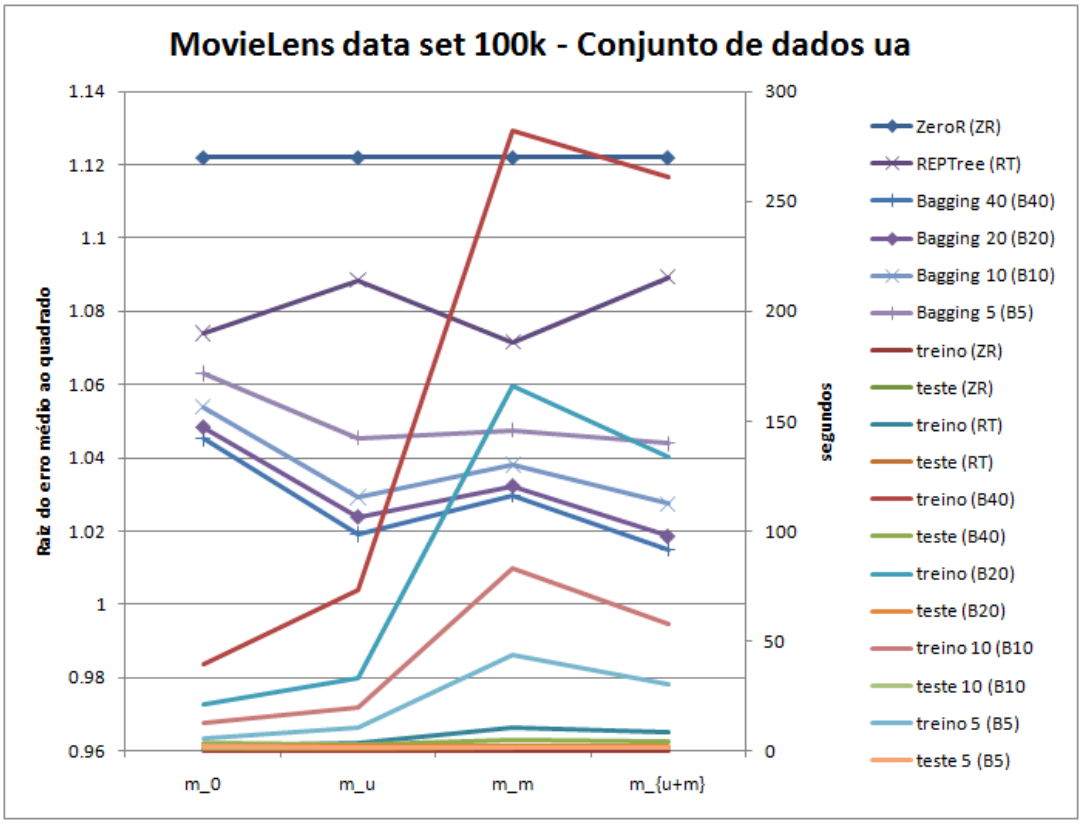


Figura 2: Resultado da execução sobre o conjunto de dados ua

Os resultados nos conjuntos de dados de desenvolvimento mostram que os algoritmos conseguem aprender a partir dos dados disponíveis e fazer predições melhores do que a média, e confirmam a expectativa de que modelos de dados mais ricos melhoram o desempenho dos algoritmos. Em contrapartida, modelos de dados maiores e algoritmos mais complexos requerem mais esforço computacional. Os gráficos permitem uma visualização consolidada do ganho de acurácia dos métodos versus o custo computacional envolvido.

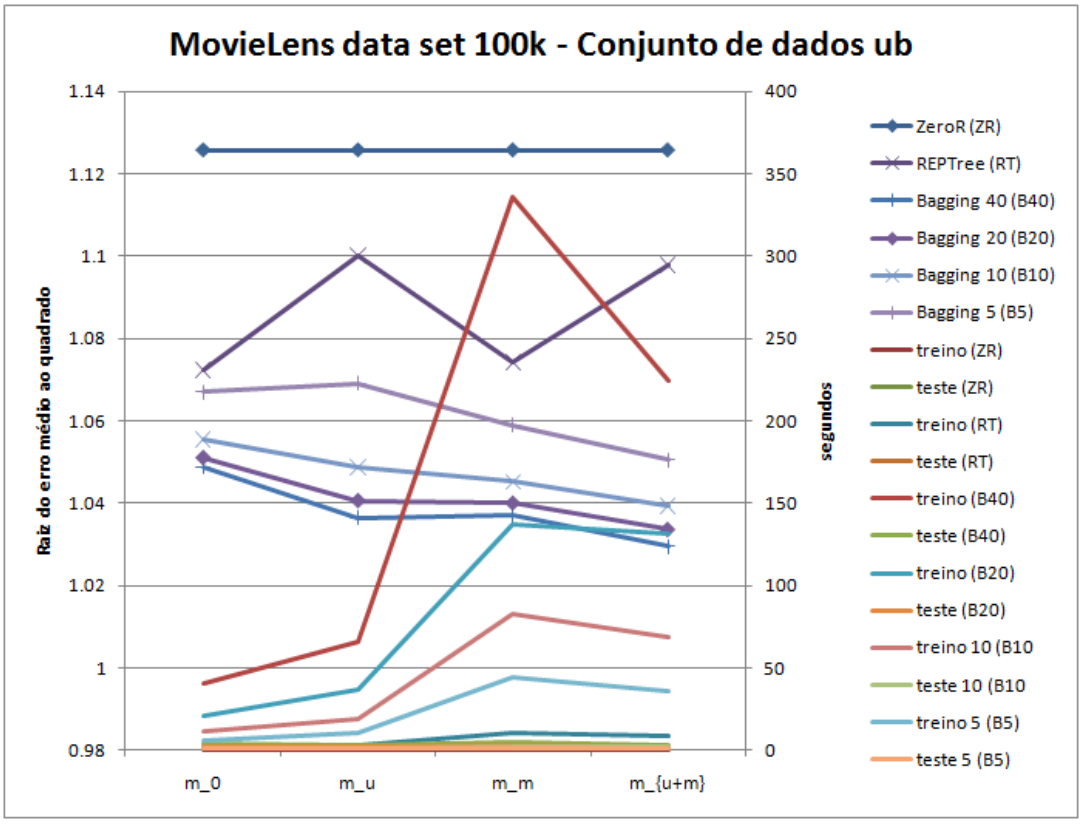


Figura 3: Resultado da execução sobre o conjunto de dados ub

Para avaliação final, utiliza-se a validação cruzada 5-fold aproveitando o particionamento oferecido junto com o conjunto de dados. A Figura 2 apresenta o resultado dessa validação para quatro técnicas. A medida do erro da técnica utilizada como base de referência, a ZeroR (média das avaliações), é de 1,125. O algoritmo Bagging que considera 40 árvores REPTree no modelo $M_{u+m}$ conseguiu a melhor acurácia, embora o custo computacional seja bem elevado conforme verificou-se no desenvolvimento. Esse algoritmo conseguiu uma melhoria de 11,84%, diminuindo o erro para 0,992.

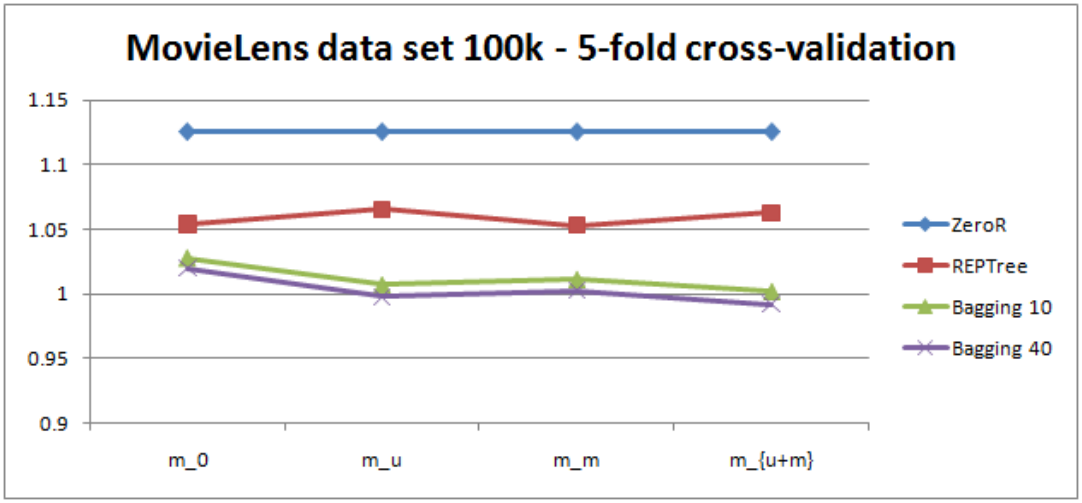


Figura 4: Resultado da execução com validação cruzada 5-fold

## 4 Conclusão

Este trabalho apresenta um estudo sobre soluções para um problema fundamental na construção de um sistema de recomendação, o problema de predizer a avaliação de um usuário para um item desconhecido. O método utilizado é classificado como sistema de recomendação por filtro colaborativo, e é baseado em algoritmos de aprendizado de máquina. O experimento utiliza um conjunto de dados clássico da literatura de sistemas de recomendação por filtro colaborativo, com um problema formalmente definido e métrica para avaliação. São avaliadas três técnicas de aprendizado de máquina e quatro modelos para o conjunto de dados no experimento. Utilizou-se a ferramenta Weka para a execução desse experimento e avaliação do desempenho. Os resultados foram satisfatórios com uma melhoria de quase 12% da técnica com menor erro em relação à técnica de avaliação de referência, embora que para aplicação comercial ainda seja necessário mais pesquisa e desenvolvimento.

## Referências

Breiman, L. (1996), 'Bagging predictors', *Mach. Learn.* **24**(2), 123–140.

Chen, L. & Pu., P. (2010), User evaluation framework of recommender systems, *in* 'Proceedings of 2010 Workshop on Social Recommender Systems (SRS'10) at the ACM

International Conference on Intelligent User Interfaces (IUI'10)', IEEE Computer Society, Hong Kong, China, p. 13.

Geisser, S. (1993), *Predictive Inference*, Chapman and Hall.

GROSSMAN, L. (2010), 'How computers know what we want – before we do', *Available online on http://www.time.com/time/magazine/article/0,9171,1992403-1,00.html* .

Herlocker, J. L., Konstan, J. A., Borchers, A. & Riedl, J. (1999), An algorithmic framework for performing collaborative filtering, *in* 'Proceedings of the 22nd Annual international ACM SIGIR Conference on Research and Development in information Retrieval', ACM, New York, NY, pp. 230–237.

Su, X. & Khoshgoftaar, T. M. (2009), 'A survey of collaborative filtering techniques', *Adv. in Artif. Intell.* **2009**, 2–2.