

# Ruído Natural na Filtragem Colaborativa

[Trabalho Final]

Paulo Xavier  
contato.pauloxavier@gmail.com

Gabriel Segobia  
segobia.gos@gmail.com

Fellipe Bravo  
fellipe.bravo@gmail.com

## ABSTRACT

Foi mostrado recentemente que os usuários podem ser inconsistentes quando eles elicitam avaliações para itens, expondo os dados usados nos SR à inconsistências. Esse trabalho busca quantificar o impacto do ruído na predição por modelos, e o desempenho dos algoritmos de detecção de ruído propostos por *O'Mahony* e *Toledo*.

## 1. INTRODUÇÃO

Sistemas de Recomendações(SR) são ferramentas de software e técnicas que fornecem sugestões de itens que sejam de uso ao usuário. Existem diversas abordagens que podem ser usadas em SR, e uma das mais populares é a *Collaborative Filtering(CF)*, que produz recomendações específicas a um usuário baseado em padrões de avaliação ou uso de itens.

Em SR é assumido que as avaliações nos *datasets* estão livres de irregularidades. Foi mostrado recentemente que os usuários podem ser inconsistentes quando eles elicitam avaliações para itens, expondo os dados usados nos SR à inconsistências. Esse tipo de inconsistência é conhecida como *ruído natural*. Abordagens que buscam *detectar* e *corrigir* essas avaliações inconsistentes surgiram para resolver esse problema em aberto.

O artigo será organizado da seguinte forma: Seção 3 faz uma revisão de filtragem colaborativa. Seção 4 elicit a o problema e faz uma proposta para quantificar o impacto do ruído nos modelos e qual o desempenho dos algoritmos *Mahony* e *Toledo* na detecção de ruído natural. Seção 5 faz uma discussão sobre os experimentos feitos para avaliar o impacto do ruído nos modelos, apresenta os resultados e mostra o *dataset* utilizado para o artigo, além da metodologia usada para obtenção dos resultados obtidos. A seção 5 apresenta uma conclusão sobre os resultados obtidos e propostas para trabalhos futuros sobre o tema.

## 2. FILTRAGEM COLABORATIVA

Collaborative Filtering é um algoritmo de recomendação popular que baseia suas predições e recomendações nas avali-

ações ou comportamento de outros usuários no sistema. A principal suposição por trás desse método é que as opiniões de outros usuários podem ser selecionadas e agregadas de maneira que é possível ter uma predição razoável das preferências um usuário alvo.

Intuitivamente, assume-se também, que se os usuários concordam sobre a qualidade ou relevância de alguns itens, então eles também vão provavelmente concordar sobre outros itens.

A grande maioria dos algoritmos de CF usados atualmente, operam inicialmente gerando predições de preferência usuário e então produzindo as recomendações deles ranqueando os itens candidatos por preferências previstas.

Os CFs possuem algumas vantagens notáveis:

- Não necessitam de informações sobre conteúdo ou usuários. As *abordagens puras* utilizam apenas as avaliações para fazer a predição;
- Conseguem avaliar a experiência, qualidade e ponto de vista de outras pessoas na predição;
- Conseguem sugerir *serendipitous items* através da análise de pessoas do comportamento de usuários semelhantes

Possui também algumas desvantagens, dentre elas:

- Baixa acurácia quando possui poucas informações sobre as avaliações dos usuários(*Cold Start*);
- Ratings são dados explicitamente por usuários, gerando uma base de dados ruidosa por natureza (*ruído natural* e/ou *ruído malicioso*).

## 3. PROBLEMA/PROPOSTA

Em SR é assumido que as avaliações nos *datasets* estão livres de irregularidades. Foi mostrado recentemente que os usuários podem ser inconsistentes quando eles elicitam avaliações para itens, expondo os dados usados nos SR à inconsistências.

Ruído em SR pode ser classificado em 2 categorias principais:

1. *Ruído Malicioso*, associado ao ruído intencionalmente introduzido por um agente externo para influenciar os resultados de um recomendador, e
2. *Ruído Natural*, involuntariamente introduzido por usuários, e que também poderia afetar o resultado da recomendação.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Copyright 20XX ACM X-XXXXX-XX-X/XX/XX ...\$15.00.

O *ruído natural* é inserido sem intenção maliciosa, e, ao contrário do *ruído malicioso*, que já é estudado na literatura há tempos, é um tópico recente.

A identificação do *ruído natural* é mais difícil, pois ele tende a aparecer de diversas formas (ao contrário do malicioso, que costuma estar associado à alguns padrões nos perfis dos usuários).

Neste artigo serão usados dois algoritmos para quantificação do impacto do ruído no modelo, *O'Mahony* e *Toledo*.

O algoritmo de *O'Mahony* considera o quão consistente o rating atual para um par usuário-item é em respeito a avaliação prevista que é feita por algum algoritmo de recomendação G.

A *consistência*  $c$  de uma avaliação  $r_{u,v}$  como o Erro Médio Absoluto (MAE) entre a avaliação atual e a prevista:

$$c(G, T)_{u,v} = \frac{|u, v - p_{u,v}|}{r_{max} - r_{min}}$$

onde  $p_{u,v}$  é a avaliação prevista para o par usuário-item  $(u, v)$  e  $r_{min}/r_{max}$  são o mínimo e máximo dos ratings permitidos, respectivamente. Uma avaliação  $r_{u,v}$  é considerada como ruído e excluída do processo de recomendação se:

$$c(G, T)_{u,v} > th$$

onde  $th$  é um valor *threshold*.

O algoritmo proposto por *Toledo* inicialmente classifica avaliações, usuários e itens, e que cada item também sua própria tendência de receber avaliações. São identificadas quatro tendências possíveis para usuários e itens:

**Usuário:**

- *Positivo*: usuário tende a avaliar todos os itens positivamente.
- *Médio*: tende a avaliar com valores médios.
- *Negativo*: tende a avaliar com valores baixos.
- *Hesitante*: usuário oscila entre as categorias anteriores e não pertence a uma específica.

**Item:**

- *Bem Avaliado*
- *Médio Avaliado*
- *Mal Avaliado*
- *Duvidoso*

**Avaliações:**

- *Forte*: se  $r(u, i) \geq v$ .
- *Médio*: se  $k \leq r(u, i) < v$ .
- *Fraco*: se  $r(u, i) < k$ .

O processo para detecção do ruído assume que para um rating  $r(u, i)$  de um dataset, se as classes associadas ao usuário  $u$  e o item  $i$  pertencem ao meso grupo, então a avaliação deve pertencer a classe de avaliação no mesmo grupo. Se a avaliação não estiver de acordo com a condição, poderia ser uma avaliação ruidosa, e sua transformação poderia mitigar o ruído natural do dataset.

Após detectar as possíveis avaliações ruidosas a próxima fase lida com essas avaliações, corrigindo a anomalia ao invés de removê-la para evitar perda de informação.

## 4. EXPERIMENTOS

### 4.1 Dataset

Para este artigo, foi utilizado o dataset fornecido pelo *Movie Lens Research Project*. *MovieLens* é um SR *web-based* de filmes que começou a operar em 1997.

Consiste de 943 usuários, 1682 filmes e contém 100,000 avaliações no total. As avaliações são baseadas numa escala de 1 a 5.

### 4.2 Metodologia

10% treinamento -> gerar ruidos aqui

90% teste

Usando Precision recall f1 (comparando a previsão sem ruído/com ruído e checar a % de ruidos)

Algoritmos: Mahony e Toledo

### 4.3 Resultados

## 5. CONCLUSÃO E TRABALHOS FUTUROS