UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
INSTITUTO DE INFORMÁTICA
CURSO DE CIÊNCIA DA COMPUTAÇÃO

ADRIANO CARNIEL BENIN

A comparison of recommender systems for crowdfunding projects.

Work presented in partial fulfillment
of the requirements for the degree of
Bachelor in Computer Science

Advisor: Prof. Dr. Bruno Castro da Silva

Porto Alegre
March 2018

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL
Reitor: Prof. Rui Vicente Oppermann
Vice-Reitora: Profª. Jane Fraga Tutikian
Pró-Reitor de Graduação: Prof. Wladimir Pinheiro do Nascimento
Diretora do Instituto de Informática: Profª. Carla Maria Dal Sasso Freitas
Coordenador do Curso de Ciência de Computação: Prof. Raul Fernando Weber
Bibliotecária-chefe do Instituto de Informática: Beatriz Regina Bastos Haro

ABSTRACT

Este documento é um exemplo de como formatar documentos para o Instituto de Informática da UFRGS usando as classes LaTeX disponibilizadas pelo UTUG. Ao mesmo tempo, pode servir de consulta para comandos mais genéricos. *O texto do resumo não deve conter mais do que 500 palavras.*

Keywords: UFRGS. recommender systems. AI. crowdfunding.

Using LaTeX to Prepare Documents at II/UFRGS

RESUMO

This document is an example on how to prepare documents at II/UFRGS using the LaTeX classes provided by the UTUG. At the same time, it may serve as a guide for general-purpose commands. *The text in the abstract should not contain more than 500 words.*

Palavras-chave: Electronic document preparation. LaTeX. ABNT. UFRGS.

LIST OF FIGURES

LIST OF TABLES

LIST OF ABBREVIATIONS AND ACRONYMS

SMP	Symmetric Multi-Processor
CBF	Content-based filtering
CF	Collaborative filtering
SPMD	Single Program Multiple Data
ABNT	Associação Brasileira de Normas Técnicas

CONTENTS

1 INTRODUÇÃO

No início dos tempos, Donald E. Knuth criou o TEX. Algum tempo depois, Leslie Lamport criou o LATEX. Graças a eles, não somos obrigados a usar o Word nem o LibreOffice.

1.1 Figuras e tabelas

Esta seção faz referência às Figuras 1.1, 1.2 e 1.3, a título de exemplo. A primeira figura apresenta a estrutura de uma figura. A *descrição* deve aparecer **acima** da figura. Abaixo da figura, deve ser indicado a origem da imagem, mesmo se essa for apenas os autores do texto.

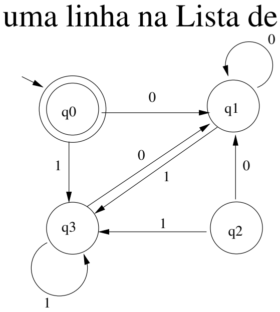
A Figura 1.2 representa o caso mais comum, onde a figura propriamente dita é importada de um arquivo (neste exemplo o formato é `eps` ou `pdf`. Veja a seção 1.1.1). A Figura 1.3 exemplifica o uso do environment `picture`, para desenhar usando o próprio LATEX.

Figure 1.1: Descrição da Figura deve ir no topo

Fonte: Os Autores

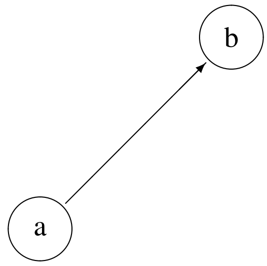
Tabelas são construídas com praticamente os mesmos comandos. Ver a tabela 1.1.

Figure 1.2: Exemplo de figura importada de um arquivo e também exemplo de caption muito grande que ocupa mais de uma linha na Lista de Figuras



Fonte: Os Autores

Figure 1.3: Exemplo de figura desenhada com o environment `picture`.



Fonte: Os Autores

Table 1.1: Uma tabela de Exemplo

Col 1	*Col 2*	*Col 3*
Val 1	Val 2	Esta coluna funciona como um parágrafo, tendo uma margem definida em 5cm. Quebras de linha funcionam como em qualquer parágrafo do tex.
Valor Longo	Val 2	Val 3

Fonte: Os Autores

1.1.1 Formato de Figuras

O LaTeX permite utilizar vários formatos de figuras, entre eles *eps*, *pdf*, *jpeg* e *png*. Programas de diagramação como Inkscape (e mesmo LibreOffice) permitem gerar arquivos de imagens vetoriais que podem ser utilizados pelo LaTeX sem dificuldade. Pacotes externos permitem utilizar SVG e outros formatos.

Dia e xfig são programas utilizados por dinossauros para gerar figuras vetoriais. Se possível, evite-os.

1.1.2 Classificação dos etc.

O formato do instituo de informática define 5 níveis: capítulo, seção, subseção e outros 2 sem nome.

1.1.2.1 Subsubseção

Exemplo de uma subsubseção.

1.1.2.1.1 Parágrafo Exemplo de um parágrafo.

1.2 Sobre as referências bibliográficas

A classe *iiufrgs* faz uso do pacote *abnTeX2* com algumas alterações feitas por Sandro Rama Fiorini. Culpe ele se algo der errado. Agradeça a ele pelo que der certo. As modificações dão uma camada de tinta NATBIB-style, já que o abntex2 usa uns comandos de citação feitos para alienígenas de 5 braços wtf. Exemplos de citação:

- *cite*: Unicórnios são verdes (ADAMS; RAUBAL, 2009);
- *citep*:Unicórnios são verdes (ADAMS; RAUBAL, 2009);
- *citet*: Segundo Adams and Raubal (2009), unicórnios são verdes.
- *citen or citenum*: Segundo Adams and Raubal (2009), unicórnios são verdes.
- *citeauthor e citeyearpar*: Segundo artigos de ADAMS; RAUBAL , unicórnios são azuis (2009).

O estilo abnt fornecido antigamente pelo UTUG não é mais recomendado, pois não produz saída de acordo com as exigências da biblioteca.

Recomenda-se o uso de bibtex para gerenciar as referências (veja o arquivo biblio.bib).

2 RECOMMENDATION ALGORITHMS

Recommendation Algorithms are widely used in the industry today to provide useful suggestions to end-users in a completely automated manner. They are ubiquitous in modern e-commerce Web sites (SCHAFER; KONSTAN; RIEDL, 2001), where new products can be recommended based on a customer's interests and preferences, and in many other fields such as movies (Netflix) and music (Spotify). Its importance can't be overstated: the effectiveness of targeted recommendations, as measured by click-through and conversion rates, far exceed those of untargeted content (LINDEN; SMITH; YORK, 2003).

The basic idea behind any recommender system is to obtain a utility function to estimate a user preferences towards an item. The meaning of this function will differ for each context; it could mean how likely a user will want to watch a specific movie or listen to a song, or the likelihood of buying a particular product. In our case, the goal is to find projects the customer is most likely to back given his backing history and other characteristics.

Recommender systems can be broadly divided into two categories: content-based methods and collaborative filtering methods (RAKESH; LEE; REDDY, 2016). The former utilizes the content features of users or items in order to recommend items to users. In collaborative filtering methods, user ratings and other data are used in order to calculate similarities between users or items, which are then ranked to show the most relevant recommendations. These two methods are sometimes combined into what is known as Hybrid Recommender Systems.

2.1 Collaborative filtering

Collaborative filtering is based on the principle that similar users will share similar interests. In the traditional approach, each customer is represented by a N-dimensional vector of items, where N stands for the number of available items, and each vector component corresponds to the user rating of the item. For large databases, CF can be prohibitively computationally expensive. Its worst case performance is $O(MN)$ where M is the number of customers and N is the number of items (LINDEN; SMITH; YORK, 2003), but this problem can be generally alleviated since in most cases the customer vector is extremely sparse. Item-to-item CF is another method proposed by Amazon that tries to minimize these scaling issues by focusing on item similarity instead of user similarity. Each item purchased by the customer is compared to other items in the dataset in order to calculate a similarity metric. The algorithm is shown bellow:

```
for each item in product catalog, I1 do
    for each customer C who purchased I1 do
        for item I2 purchased by customer C do
            Record that a customer purchased I1 and I2;
        end
    end
    for each item I2 do
        Compute the similarity between I1 and I2;
    end
end
```

Algorithm 1: Item-to-item CF

There are many ways to compute the similarity between items, one common method is the cosine measure defined as follows:

$$similarity(x, y) = cos(x, y) = \frac{x \cdot y}{||x|| \cdot ||y||}$$

In either case, for this to work large amounts of data about the user is required. This is known as the cold start problem: new users who haven't rated many items yet will have a reduced recommendation quality. On the other hand, no information about the item itself is needed, making CF specially applicable to collections of hard-to-analyze items such as movies.

2.2 Content-based filtering

In this method a description of each item is constructed using some kind of item presentation algorithm such as Latent Dirichlet Allocation or TF-IDF. These representations are then compared to items previously liked by the user and the best-matching items are recommended. Since this approach focus on item rather than user similarity, it doesn't suffer from the cold start problem as no user preferences information is required. However, since only similar items to those already rated by the user will be considered, CBF approaches tend to suffer from over-specialization (IAQUINTA et al., 2008). This is known as the serendipity problem.

3 ABOUT CATARSE

Launched in January 2011, Catarse was the first crowdfunding platform for creative projects in Brazil. With over 7000 successfully financed projects raising R$77 millions from 480.000 people, it's currently the largest national platform of its kind. It works similarly to most crowdfunding platforms: the project owner presents his or her idea and specifies the required investment as well as the cutoff date for the project, while offering rewards for those who back it. Projects are divided in 3 main categories: all-or-nothing, flexible and recurrent. In the first type, projects are available for backing up to 60 days and the project owner only receives the raised amount if the project's goal is met, otherwise all the money is returned to its original backers. On flexible projects the owner receives the raised amount whether the goal is reached or not. Recurrent projects are subscription based and the owner can collect the money monthly. This work will only focus on the first two types of projects.

REFERENCES

ADAMS, B.; RAUBAL, M. Conceptual space markup language (csml): Towards the cognitive semantic web. In: IEEE INTERNATIONAL CONFERENCE ON SEMANTIC COMPUTING, 2009, Berkeley, USA. **Proceedings...** Washington, USA: IEEE, 2009. p. 253–260.

IAQUINTA, L. et al. Introducing serendipity in a content-based recommender system. In: **2008 Eighth International Conference on Hybrid Intelligent Systems**. [S.l.]: IEEE, 2008.

LINDEN, G.; SMITH, B.; YORK, J. Amazon.com recommendations: item-to-item collaborative filtering. **IEEE Internet Computing**, Institute of Electrical and Electronics Engineers (IEEE), v. 7, n. 1, p. 76–80, jan 2003.

RAKESH, V.; LEE, W.-C.; REDDY, C. K. Probabilistic group recommendation model for crowdfunding domains. In: **Proceedings of the Ninth ACM International Conference on Web Search and Data Mining - WSDM 16**. [S.l.]: ACM Press, 2016.

SCHAFER, J. B.; KONSTAN, J. A.; RIEDL, J. E-commerce recommendation applications. In: **Applications of Data Mining to Electronic Commerce**. [S.l.]: Springer US, 2001. p. 115–153.