

FUNDAÇÃO GETULIO VARGAS
ESCOLA DE MATEMÁTICA APLICADA
CURSO DE GRADUAÇÃO EM
MATEMÁTICA APLICADA

**Dinâmica de Disseminação de Notícias em
Redes Complexas**

por

Elisa Mussumeci

Rio de Janeiro
2015

FUNDAÇÃO GETÚLIO
VARGAS

FUNDAÇÃO GETULIO VARGAS
ESCOLA DE MATEMÁTICA APLICADA
CURSO DE GRADUAÇÃO EM
MATEMÁTICA APLICADA

Dinâmica de Disseminação de Notícias em
Redes Complexas

”Declaro ser o único autor do presente projeto de monografia que refere-se ao plano de trabalho a ser executado para continuidade da monografia e ressalto que não recorri a qualquer forma de colaboração ou auxílio de terceiros para realizá-lo a não ser nos casos e para os fins autorizados pelo professor orientador”

Elisa Mussumeci

Orientador: Flavio Codeço Coelho

**Rio de Janeiro
2015**

ELISA MUSSUMECI

**Dinâmica de Disseminação de Notícias em
Redes Complexas**

“Monografia apresentada à Escola de Matemática Aplicada
como requisito parcial para obtenção do grau de Bacharel
em Matemática Aplicada”

Aprovado em ____ de _____ de ____ .
Grau atribuído ao Projeto de Monografia: ____ .

Professor Orientador: Flávio Codeço Coelho
Escola de Matemática Aplicada
Fundação Getulio Vargas

Conteúdo

1	Introdução	5
1.1	O Projeto Media Cloud Brasil	5
1.2	Referencial Teórico	6
1.2.1	O espalhamento de notícias como um processo de contágio	6
1.2.2	Processos Epidemiológicos em Redes Complexas	6
1.2.3	Processamento de Linguagem Natural	7
1.3	Objetivo	8
2	Metodologia	9
2.1	Rede de Disseminação Real	9
2.1.1	Matriz de Documentos	10
2.1.2	Construção da Rede	11
2.2	Simulação Rede de Disseminação	12
2.2.1	Criação Rede Completa	13
2.2.2	Simulação	14
3	Resultados	17
3.1	Rede de Disseminação Real	17
3.1.1	Definindo Parâmetros	18
3.1.2	Visualizações da Rede Real	19
3.2	Simulação Rede de Disseminação	21
3.2.1	Resultado Simulação	22
3.2.2	Rede Simulada	23
3.2.3	Visualizações da Rede Simulada	23
4	Conclusão	25

Lista de Figuras

1	Rede Real Genérica	9
2	Relação de Influência	11
3	Rede Completa	12
4	Disseminação na Rede Completa	12
5	Pesos na Rede Completa	13
6	Diagrama Definição de Influência	16
7	Notícias x Tempo	17
8	Distribuição de Similaridades Máximas	18
9	Distribuição Out-degrees Rede Real	19
10	Rede Disseminação Real x Passo	20
11	Rede Disseminação Real x Tempo	21
12	Simulação Modelo	22
13	Rede Simulada x Tempo	23
14	Rede Simulada x Passo	24

Resumo

O processo de formação de opinião é fortemente influenciado pela mídia digital. Entretanto pouco se sabe sobre o processo de disseminação de notícias e os fatores que determinam o alcance de cada notícia.

A disseminação de uma notícia se dá por meio de um ou mais caminhos em uma rede desconhecida de influência entre formadores de opinião (produtores de notícias). Este padrão pode ser recuperado, com algum grau de incerteza, a partir de dados da sequência temporal das publicações sobre um mesmo tema, e dos links nelas contidos.

Este projeto tem como objetivo caracterizar as redes de interligação de veículos de mídia e modelar a dinâmica do espalhamento de notícias, a fim de prever tendências e mapear questões de interesse.

1 Introdução

Atualmente a internet é um dos principais meios de veiculação de notícias e informação do país. Com o crescente número de pessoas aderindo à redes sociais, o compartilhamento de notícias aumentou significativamente, o que tornou fundamental o papel das mídias digitais no acesso à informação.

Consideramos como mídia digital todo e qualquer veículo difusor de informação contido na internet brasileira, como jornais, revistas e blogs independentes. Cada mídia presente na internet possui sua própria periodicidade, alcance, público e credibilidade, o que afeta diretamente no processo de disseminação da informação.

Utilizando do fato que essas mídias digitais são fundamentais na disseminação da informação, e que com isso possuem um forte papel influenciador no processo de formação de opinião, podemos definir como importante entender como as notícias se formam e se espalham na internet brasileira.

Um dos métodos de se estudar a disseminação da informação é utilizando modelos epidemiológicos. Bettencourt et al. [2006] conseguiu modelar a disseminação de (tabela de alguma coisa) no tempo através de modelos epidemiológicos, como SIR, SIS entre outros. Esse tipo de abordagem vem sendo utilizada também para entender redes de contatos em redes sociais como (incluir referencia). MELHORAR PARAGRAFO

Neste trabalho utilizaremos redes complexas e modelos epidemiológicos para modelar a disseminação de notícias na mídia brasileira. Para isso estudaremos os caminhos percorridos em uma rede de disseminação criada através de modelos de recuperação de informação utilizados em cima da base de dados do Projeto MediaCloud Brasil¹.

1.1 O Projeto Media Cloud Brasil

O MediaCloud Brasil é um projeto concebido através de uma parceria do NAMD/EMAp da Fundação Getúlio Vargas, do Massachusetts Institute of Technology (MIT) e Harvard. O projeto

é mantido pelo NAMD/EMAp da Fundação Getúlio Vargas, e vem ao longo dos últimos três anos monitorando mais de cem mil veículos de mídia da internet brasileira. O projeto consiste na captura automatizada de dados de todos os feeds da mídia digital brasileira, captura-se páginas de blogs, jornais, revistas,

Possui em sua base de dados mais de 1.6 milhão de artigos capturados. **falar sobre o que e como o Media cloud captura os artigos** O projeto utiliza como banco de dados o MongoDB, um banco de dados de documentos open-source de alta performance. O MongoDB é classificado como um banco de dados 'NoSQL', uma vez que evita a tradicional estrutura baseada em tabela relacional e utiliza documentos JSON com esquemas dinâmicos para armazenamento dos documentos. A vantagem de utilizar o JSON é realizar a integração de dados em certos tipos de aplicações de forma mais fácil e mais rápida. **Falar que uma vez armazenado, o banco de artigos é indexado para permitir buscas textuais**

Que tipo de aplicações?

¹https://github.com/NAMD/mediacloud_backend

1.2 Referencial Teórico

1.2.1 O espalhamento de notícias como um processo de contágio

Uma epidemia é caracterizada pela incidência de grande número de casos de uma doença em um curto período de tempo. Sabemos que as doenças se espalham através do contágio entre infectados, mas como podemos definir esse contágio?

Cada doença possui uma forma própria de transmissão, por exemplo, a gripe é uma doença viral e se transmite a partir de vias orais, já a dengue é transmitida a partir da picada de um mosquito, assim como a febre amarela. Saber a forma de contágio de uma doença é fundamental para que possamos entender sua disseminação e modelar sua epidemia.

Ao observar o comportamento de notícias, assuntos e histórias na mídia, podemos ver o surgimento de memes e histórias 'virais'. Esses tipos de notícias são chamadas de virais por se espalharem muito rápido e obterem um alcance grande na população. Algumas dessas notícias se sustentam por um longo tempo na mídia, e outras são esquecidas rapidamente.

Se pensarmos que estamos lidando com um processo de disseminação, que possui uma taxa de espalhamento e uma taxa de esquecimento, podemos facilmente fazer uma comparação à modelos epidemiológicos, principalmente ao modelo SIR, que possui uma taxa de infecção e uma taxa de recuperação. Dessa forma, podemos estudar como uma notícia se espalha da mesma forma que modelamos uma epidemia.

Para modelar a disseminação de notícias da mesma forma que modelamos a de doenças, precisamos que nosso modelo seja compatível com o de uma epidemia, ou seja, precisamos definir os infectados, suscetíveis e o método de contágio.

Em nosso modelo, uma notícia/assunto é a doença, e os infectados são todos os artigos que falam sobre essa notícia. Para definir o contágio da nossa notícia, consideramos que um artigo infecta o outro quando ele influencia o outro. Ou seja, definimos que quando um artigo exerceu influência sobre um outro, ele infectou esse outro artigo.

Dessa forma, podemos ver a similaridade entre ambas modelagens, o que deixa claro a possibilidade de modelar a disseminação de notícias através de modelos epidemiológicos. O que nos falta descobrir são os parâmetros que utilizaremos para realizar essa modelagem de forma que fique o mais verídica possível.

1.2.2 Processos Epidemiológicos em Redes Complexas

O termo Redes Complexas se refere a um grafo que apresenta uma estrutura topográfica não trivial, composto por um conjunto de vértices (nós) que são interligados por meio de arestas ([Ravasz and Barabási \[2003\]](#)). A teoria das Redes Complexas está relacionada com a modelagem de redes reais, através da análise de dados empíricos. As redes complexas não são estáticas (evoluem no tempo alterando sua estrutura), e constituem estruturas onde processos dinâmicos (como disseminação de vírus ou opiniões) podem ser simulados.

Um processo dinâmico comumente modelado como rede complexa é a disseminação de uma doença na população. Através da rede conseguimos observar o processo de infecção dinamicamente, e realizar análises que trazem resultados significativos para o modelo epidêmico, como estudar os caminhos percorridos e definir parâmetros de infeciosidade.

1.2.3 Processamento de Linguagem Natural

Processamento de Linguagem Natural (*Natural Process Language - NLP*) é uma área de Inteligência Artificial que tem como principal objetivo desenvolver métodos computacionais que consigam traduzir a linguagem natural para a máquina. A utilização desses métodos é fundamental em tarefas que envolvem análises conjuntos de dados, pois a partir deles podemos traduzir os textos para uma linguagem compreensível ao computador.

Pré-Processamento

O primeiro passo ao lidar com documentos em texto em computadores é realizar um pré-processamento no conjunto de dados. Esse pré-processamento consiste em tokenizar, limpar e stemmatizar cada documento do corpus linguístico.

O processo de tokenização consiste em decompor o documento em termos, também chamado de tokens. Para cada documento, cria-se uma lista de tokens presentes nele. Após tokenizar, limpamos o documento retirando as *stop-words* (lista de termos não representativos para documentos, geralmente composta por preposições, pronomes, pontuação, advérbios e artigos) e realizamos o processo de *stemming*, que consiste em radicalizar cada termo do documento. Dessa forma, palavras que possuem o mesmo radical são consideradas iguais, já que representam a mesma definição.

Temos como exemplo as duas frases a seguir:

1. João gosta de jogar futebol. Maria prefere jogar xadrez.
2. Maria não gosta de futebol.

Após o pré-processamento temos:

1. (joão, gost, jog, futebol, maria, prefer, jog, xadrez)
2. (maria, não, gost, futebol)

Bag-of-Words

O modelo bag-of-words é uma forma de representar vetorialmente um texto muito usada em Processamento de Linguagem Natural e em Recuperação de Informação. O modelo consiste em aprender um vocabulário baseado em todos os documentos pré processados do conjunto de dados, e então modelar o documento a partir número de vezes em que cada palavra do vocabulário aparece. Dessa forma, cada documento é representado por um 'saco de palavras', o que, apesar de manter multiplicidade, causa uma perda de informação, visto que não leva em consideração a ordem e posição das palavras.

Exemplificamos o processo a seguir, considerando as frases pré processadas na seção anterior:

1. (joão, gost, jog, futebol, maria, prefer, jog, xadrez)
2. (maria, não, gost, futebol)

Cria-se então o vocabulário do conjunto de dados:

{joão, gost, jog, futebol, maria, prefer, xadrez, não}

A partir desse vocabulário, criamos então, o vetor referente a cada palavra:

1. (1,1,2,1,1,1,1,0)
2. (0,1,0,1,1,0,0,1)

Skip-gram

O *Skip-gram* é um modelo deep-learning que tem como objetivo representar palavras com o maior valor semântico possível, ou seja uma representação que leva em consideração o contexto. Intruduzido por Mikolov et al. [2013], o skip-gram é uma rede neural que ao treinar um corpus linguístico, retorna para cada palavra um vetor representativo de atributos.

A maior vantagem desse modelo é que palavras semelhantes possuem vetores próximos no espaço, ou seja, através dos atributos calculados pelo modelo conseguimos definir a semelhança semântica entre duas palavras. Esse fator faz do skip-gram um modelo essencial para este trabalho que tem na semelhança entre textos uma de suas principais eurísticas.

TF-IDF

O modelo Tf-Idf (*term frequency-inverse document frequency*), é uma medida estatística que tem o intuito de indicar a importância de uma palavra de um documento em relação a um corpus linguístico. O Tf-Idf é definido através do produto entre as estatísticas $Tf_{d,t}$ e Idf_t .

Dado um conjunto de N documentos, $Tf_{d,t}$ é a frequência do termo t no documento d , ou seja, o número de vezes em que t ocorre em d . Sendo assim, o Tf nos dá a frequência absoluta dos termos, o que faz com que um termo que possui frequência alta seja relevante.

O termo Idf_t tem como função aumentar o peso relativo de termos raros e diminuir o peso relativo de termos muito frequentes, uma vez que termos raros são mais informativos do que termos frequentes. Calculamos a medida TF-IDF $W_{t,d}$ do termo t no documento d da seguinte forma:

$$W_{t,d} = (1 + \log Tf_{t,d}) \cdot \log \frac{N}{Df_t}$$

Onde Df_t é uma medida inversa da informatividade do termo t , e é definida como o número de documentos em que t ocorre. O peso TF-IDF aumenta com o número de ocorrências dentro de um documento e com a raridade do termo na coleção.

1.3 Objetivo

Esse trabalho tem como principal objetivo caracterizar e modelar a dinâmica de disseminação das notícias no país através de redes complexas.

2 Metodologia

2.1 Rede de Disseminação Real

Nesta primeira parte do trabalho, iremos construir uma rede complexa que descreva o processo epidemiológico de uma notícia. O objetivo da construção dessa rede, é conseguir aproximar a rede de interligação real entre os artigos selecionados do banco MediaCloud, de forma a torná-la observável.

Primeiramente, selecionamos todos os artigos do banco referentes a uma determinada notícia, criando assim, nosso corpus linguístico. Depois, a partir desse corpus, utilizamos métodos de PLN para representar matricialmente nosso conjunto de dados. A partir da matriz criada construímos nossa rede direcionada de disseminação, onde cada vértice ou nó da rede é um artigo.

Para construir a rede de disseminação precisamos definir a relação de contágio entre artigos, ou seja, o que conecta um artigo ao outro na rede. No nosso caso, definimos que o contágio se dá através de uma relação de influência, porém como definir essa relação? Partindo do pressuposto de que se um artigo influenciou outro então esses artigos são necessariamente similares, consideramos a **similaridade** entre os artigos o fator principal para definir se eles possuem uma conexão em nossa rede.

Possuindo uma conexão, ainda precisamos saber quem influenciou e quem foi o influenciado. Para definir isso, utilizamos o **tempo**. Assim, dado que dois artigos são similares, definimos como influenciador o artigo que foi publicado primeiro e influenciado o artigo publicado depois. No caso de um artigo ter mais de um similar publicado anteriormente a ele, definimos como influenciador o artigo mais similar à ele. Dessa forma todos os vértices de nossa rede possuem um grau de entrada igual à 1 e podem possuir um grau de saída maior do que 1.

A seguir, podemos ver um exemplo do que esperamos da nossa rede de disseminação real:

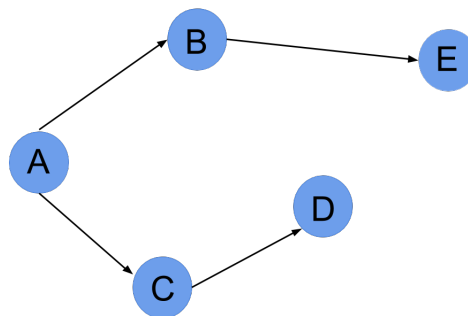


Figura 1: Rede Real Genérica

No exemplo acima, temos que o artigo A é similar aos artigos B e C e possui uma data de publicação mais antiga do que os dois. Sendo assim, foi definido como o influenciador dos dois, ou seja, foi o artigo que os contagiou com a notícia.

2.1.1 Matriz de Documentos

Selecionamos no banco de dados do MediaCloud todos os artigos referentes a uma determinada notícia, formando o corpus linguístico que utilizaremos para criar a matriz de documentos, representação vetorial dos artigos selecionados.

Para representar vetorialmente nosso conjunto de dados utilizamos o *Word2Vec*¹, ferramenta que promove uma implementação eficiente do modelo *skip-gram* e *bag-of-words* contínuo.

Para gerar nosso modelo *Word2Vec*, fornecemos como entrada nosso corpus linguístico e o modelo nos retorna uma matriz, onde as linhas p_1, p_2, \dots, p_n são as palavras contidas no corpus e as colunas t_1, t_2, \dots, t_n são os atributos gerados pelo modelo para cada palavra:

$$\begin{matrix} & t_1 & t_2 & \dots & t_n \\ \begin{matrix} p_1 \\ p_2 \\ \vdots \\ p_n \end{matrix} & \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{31} & a_{32} & \dots & a_{nn} \end{bmatrix} \end{matrix}_{n \times n}$$

preciso explicar mais sobre o word2vec?

preciso explicar que tem que haver um conjunt trainado com todos os artigos do media-cloud etc?

Dessa forma, temos uma matriz de palavras, ou seja, uma representação de todas as palavras presentes em nosso corpus. Porém, queremos representar documentos e não apenas palavras. Para criar uma matriz que represente documentos, fazemos, para cada documento, uma soma dos vetores referentes a cada palavra presente no documento, criando assim um vetor representativo do documento em questão.

Podemos exemplificar da seguinte forma: dado um documento D , sabemos que ele é composto pelo seguinte conjunto de palavras $P: \{1, 2, 3, 4, 5\}$. Para cada termo, buscamos o seu vetor representativo $p_i, i = \{1, 2, 3, 4, 5\}$ na matriz de palavras e somamos-os, criando o vetor v_D que representa o vetor referente ao documento D :

$$v_D = p_1 + p_2 + p_3 + p_4 + p_5 \quad (1)$$

Ao representar um documento dessa forma, não levamos em consideração a relevância de cada palavra para o documento, o que faz da nossa representação pouco eficiente. Para melhorar nossa eficiência, antes de somar os vetores de palavras p_i , iremos multiplicar cada um deles pelo valor *TF-IDF* da palavra ao qual ele representa.

Sendo assim, sabendo que para cada palavra i temos um vetor p_i que a representa na matriz de palavras, e um valor $w_{i,d}$ referente ao valor *TF-IDF* da palavra i no documento D , representamos o vetor v_D da seguinte forma:

$$v_D = \sum_{i=1}^5 p_i \cdot w_{i,D} \quad (2)$$

Sendo assim, ficamos com a matriz de documentos abaixo, onde cada linha v_i é um documento, e cada coluna t_i é a soma do atributo i em todas as palavras do documento i :

$$\begin{matrix} & t_1 & t_2 & \dots & t_n \\ \begin{matrix} v_1 \\ v_2 \\ \vdots \\ v_n \end{matrix} & \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{31} & a_{32} & \dots & a_{nn} \end{bmatrix} \end{matrix}_{n \times n}$$

¹<https://code.google.com/p/word2vec/>

2.1.2 Construção da Rede

Construímos a rede de disseminação a partir da matriz de documentos gerada na sessão anterior, onde cada vetor consiste em um documento/artigo.

Ao construir a rede dos nossos documentos, consideramos como contágio sofrer influência de outro artigo, ou seja, um artigo A contamina um artigo B se o artigo A influenciou o artigo B.

Dessa forma, em nossa rede, os vértices representam os artigos, e as arestas a relação de influência entre eles, isto é, dado um nó e_i e um nó e_j , existe uma aresta a_{ij} , que sai de i e vai para j , se o artigo i influenciou o artigo j .

Para definir quando um artigo influencia outro em nossa rede, foram usadas duas heurísticas: *similaridade* e *temporalidade*.

Similaridade

Para definir similaridade entre dois artigos, utilizamos a *Similaridade de Cosseno*. A similaridade de cosseno mede a semelhança entre dois vetores através da distância de cosseno do ângulo que eles formam. Para calcular a distância de cosseno $d(A, B)$ entre um vetor A e B , fazemos:

$$d(A, B) = \cos(\theta) = \frac{A \cdot B}{\|A\| \|B\|} \quad (3)$$

A similaridade de cosseno se dá por $1 - d(A, B)$. Utilizando essa equação, calculamos a similaridade de cosseno para cada par de vetor de nossa matriz de documentos, criando assim, uma matriz de similaridades. Se similaridade entre dois vetores for grande o suficiente, os definimos como similares.

Para saber o quão grande a similaridade de cosseno entre dois artigos deve ser para considerarmos-os similares, fazemos a distribuição das similaridade e realizamos um corte em suas extremidades.

Temporalidade

Após definir dois artigos como similares, inferimos uma relação de influência entre eles. Porém, ainda precisamos saber quem influenciou e quem foi influenciado. Para descobrir isso, olhamos a data de publicação de cada artigo. Se dois artigos A e B foram considerados similares, e o artigo A foi publicado antes do artigo B , então definimos que A influenciou B , logo temos em nossa rede:

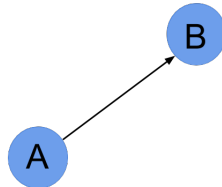


Figura 2: Relação de Influência

Porém, pode acontecer de termos um outro artigo C , também considerado similar ao artigo B e que também publicado anteriormente. Em nosso modelo, definimos que um artigo só pode ter sido influenciado por um único artigo. Dessa forma, como definir quem de fato influenciou B ? Para resolver esse impasse utilizamos a similaridade de cosseno. Consideramos influenciador o artigo publicado anteriormente, e que possui a **maior** similaridade com B .

Consideramos, também, que um artigo deixa de ser influente com o passar do tempo, assim como uma pessoa infectada por um vírus deixa de ser infecciosa após se curar da doença. Dessa forma, precisamos calcular o tempo máximo em que um artigo é influente em nossa rede. Para isso, calculamos a diferença entre as datas de publicação de cada par de artigos e fazemos a distribuição dessas diferenças. Assim como na distribuição de similaridades, realizamos um corte nas extremidades e pegamos os valores limites.

2.2 Simulação Rede de Disseminação

Na segunda parte do trabalho, temos como objetivo simular a disseminação de nossa rede real utilizando um modelo epidemiológico, com o intuito de validarmos nossa rede e conseguirmos caracterizá-la.

Queremos então, conseguir modelar a disseminação de uma notícia em uma rede genérica e conseguir um resultado semelhante à nossa rede de disseminação real. Dessa forma, dado uma rede completa com N nós, queremos traçar o espalhamento da infecção nesses N nós, ou seja, definir as relações de influencia entre eles, utilizando os resultados de uma simulação epidemiológica:

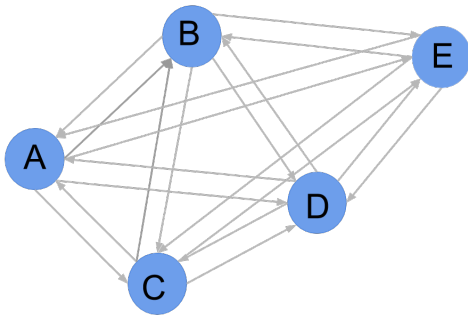


Figura 3: Rede Completa

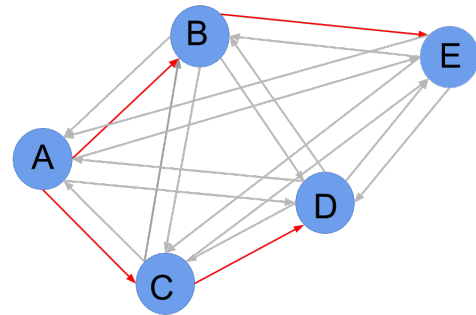


Figura 4: Disseminação na Rede Completa

2.2.1 Criação Rede Completa

Para simular nossa rede, construímos primeiramente uma rede completa com os mesmos nós de nossa rede real, onde os pesos das arestas são as probabilidades dessa aresta existir no caminho de disseminação da notícia, ou seja, a probabilidade de um artigo influenciar um outro. Podemos exemplificar da seguinte forma, dado duas vértices de nossa rede completa, A e B . Teremos como peso para as duas arestas que conectam esses nós a probabilidade de cada uma existir:

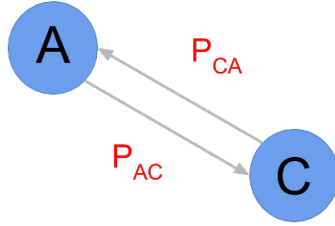


Figura 5: Pesos na Rede Completa

Onde P_{CA} e P_{AC} são as probabilidades do nó C ter influenciado o nó A , e do nó A ter influenciado o nó C , respectivamente.

Calculamos essas probabilidades através da informação do **veículo** responsável por cada artigo. Para cada nó presente em nossa rede completa, buscamos o veículo que publicou o artigo referente ao nó (exemplos de veículos são oglobo.com, folha.com.br). Possuindo o veículo de cada nó presente na rede, podemos saber qual a chance de um artigo do veículo X ser influenciado por um artigo do veículo Y . Dessa forma, considerando X e Y , os veículos dos nós A e C , respectivamente, calculamos a probabilidade P_{AC} da seguinte forma:

$$P_{AC} = \frac{N_{XY}}{NY} \quad (4)$$

Onde N_{XY} é o número de vezes que um artigo do veículo X influenciou um artigo do veículo Y , e NY representa o número de vezes em que um artigo do veículo Y foi influenciado em nossa rede de disseminação real.

Construímos então uma matriz de pesos, onde calculamos as probabilidades para cada par de veículos. É esperado em um conjunto de dados possuímos mais de um artigo publicado pelo mesmo veículo, sendo assim, considerando que temos n artigos em nosso corpus linguístico, possuímos nele m veículos responsáveis pela publicação desses n artigos.

Dessa forma, nossa matriz de pesos possui tamanho mxm , e cada posição a_{ij} temos a probabilidade do veículo i ser influenciado pelo veículo j . Considerando que um veículo não influencia a si mesmo, teremos a matriz a seguir:

$$\begin{array}{c} \begin{array}{cccc} & x & y & \dots & z \\ \begin{array}{c} x \\ y \\ \vdots \\ z \end{array} & \begin{bmatrix} 0 & a_{12} & \dots & a_{1m} \\ a_{21} & 0 & \dots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{31} & a_{32} & \dots & 0 \end{bmatrix} \end{array} \end{array} \Big]_{mxm}$$

Utilizando a matriz de pesos acima, atribuiremos os valores para cada aresta de a rede completa criada acima. Para isso, criamos uma matriz de adjacência, onde cada valor a_{ij} , contém a probabilidade dessa aresta existir. Sendo assim, para cada par de artigos d_i, d_j de nossa rede completa, descobrimos seus veículos x e y , e buscamos a probabilidade de conexão entre eles na matriz de pesos. Dessa forma, ficamos com uma matriz $n \times n$ documentos, que descreve a probabilidade da conexão entre cada par de vértices da rede completa, e que será essencial para a simulação da disseminação da notícia na rede.

$$\begin{matrix} & d_1 & d_2 & \dots & d_n \\ \begin{matrix} d_1 \\ d_2 \\ \vdots \\ d_n \end{matrix} & \begin{bmatrix} 0 & a_{12} & \dots & a_{1n} \\ a_{21} & 0 & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{31} & a_{32} & \dots & 0 \end{bmatrix} \end{matrix}_{n \times n}$$

2.2.2 Simulação

Nesta seção, iremos simular a disseminação da notícia em nossa rede completa. Para isso, simularemos um modelo epidemiológico de forma que possamos obter o estado de nossa rede em cada passo de contágio, ou seja, queremos saber quais artigos foram infectados em cada passo q de nossa simulação, e quais artigos foram responsáveis por infectá-los.

Modelo utilizado

O modelo epidemiológico que iremos simular foi proposto por [Pastor-Satorras et al. \[2014, p. 940, eq. \(33\)\]](#). Nele, simulamos a probabilidade $\rho_i^I(t)$ de cada artigo i estar infectado em um tempo t utilizando a matriz de adjacência criada na seção anterior:

não sei o quanto preciso explicar com minas palavras do modelo, talvez tenha ficado meio confuso

$$\frac{d\rho_i^I}{dt} = -\rho_i^I(t) + \lambda[1 - \rho_i^I(t)] \sum_{j=1}^N a_{ij}\rho_j^I(t) \quad (5)$$

Nossa matriz de adjacências é iterada para cada artigo i , como podemos ver no fator $\sum_{j=1}^N a_{ij}\rho_j^I(t)$ da equação diferencial acima. Nela, somamos as probabilidades de cada artigo j ter influenciado o artigo i em questão, vezes a probabilidade de cada artigo j estar infectado no tempo t , ou seja, a chance de um artigo já infectado ter infectado o artigo i .

Multiplicamos esse resultado pela taxa de transmissão λ e pela possibilidade dele ainda não estar infectado (proporção de documentos não infectados). Assim obtemos a probabilidade do artigo i se infectar.

O parâmetro λ é obtido através da divisão da taxa de infecção β pela taxa de recuperação μ :

$$\lambda = \frac{\beta}{\mu} \quad (6)$$

A equação (5) teve o tempo reescaldado em $\frac{1}{\mu}$, o que torna nosso modelo adimensional, como explicado em [Pastor-Satorras et al. \[2014, p. 939\]](#). Podemos interpretar que o tempo de cada passo é o inverso do tempo de recuperação média dos artigos.

Para tornar o resultado da simulação mais fiel ao nosso conjunto de dados, onde possuímos um decaimento da infecção, ou seja, nossos artigos se recuperam (deixam de ser influentes), incrementamos a equação abaixo ao modelo:

$$\frac{d\rho_i^S}{dt} = -\lambda\rho_i^S(t) \sum_{j=1}^N a_{ij}\rho_j^I(t) \quad (7)$$

Essa equação nos diz que um artigo que já foi infectado não pode ser infectado novamente. Modificando o modelo para o incremento da equação (7), ficamos com um modelo final utilizado nas simulações:

$$\begin{cases} \frac{d\rho_i^I}{dt} = -\rho_i^I(t) + \lambda\rho_i^S(t) \sum_{j=1}^N a_{ij}\rho_j^I(t) \\ \frac{d\rho_i^S}{dt} = -\lambda\rho_i^S(t) \sum_{j=1}^N a_{ij}\rho_j^I(t) \end{cases} \quad (8)$$

Temos como resultado da simulação, uma matriz de estados. Cada posição a_{ij} da matriz se refere à probabilidade do artigo d_j estar contaminado no passo t_i da simulação. Para determinar os artigos infectados no passo t_i , utilizamos a distribuição de bernoulli para cada probabilidade p_{ij} . Assim, ficamos com uma matriz booleana de estados, onde temos bem definidos quais vértices estão infectados em cada passo da simulação.

Após definir quais nós estão infectados em cada passo da simulação, precisamos definir as relações de influências entre eles, ou seja, definir quais arestas existem em nossa rede completa.

Para definir as relações de influência, a cada novo artigo contaminado no passo t_i , consideramos como possíveis influenciadores todos os artigos infectados no passo t_{i-1} . Por exemplo, possuindo a matriz booleana abaixo:

$$\begin{array}{ccccc} & d_1 & d_2 & d_3 & d_4 \\ \begin{array}{c} t_1 \\ t_2 \\ t_3 \end{array} & \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 1 \end{bmatrix} \end{array}$$

Temos que, em t_1 apenas o artigo d_2 está infectado. Logo, ele obrigatoriamente influenciará qualquer novo artigo infectado no tempo t_2 .

No tempo t_2 possuímos apenas d_3 como um novo artigo infectado, sendo assim, sabemos que d_2 influenciou d_3 , e logo, teremos em nossa rede simulada uma aresta a_{23} .

No próximo passo t_3 podemos observar que o artigo d_3 continua infectado, e que possuímos dois novos artigos infectados: d_1 e d_4 . Porém no passo anterior temos dois artigos infectados, como saber qual deles influenciou os novos infectados? Para definir o influenciador de cada novo infectado, buscaremos em nossa matriz de pesos, a probabilidade de cada artigo infectado no passo anterior ter infectado novos artigos. A partir dessas probabilidades, utilizamos novamente a *Bernoulli* e definimos as relações de influência.

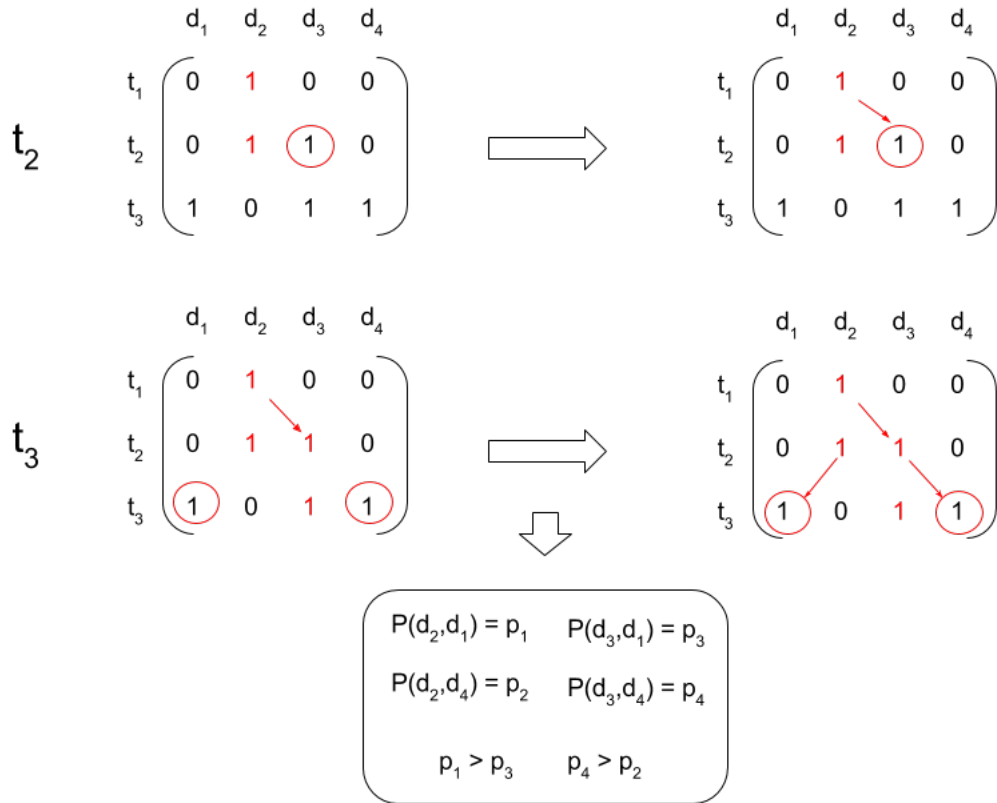


Figura 6: Diagrama Definição de Influência

No diagrama abaixo podemos ver como o algoritmo se comporta em t_3 , utilizando $P(d_i, d_j)$ como a probabilidade do artigo d_i influenciar o artigo d_j , e supondo uma ordem de grandeza entre eles.:

Nesse caso, utilizando os valores do exemplo acima, encontramos que o artigo d_2 influenciou d_1 e que d_3 influenciou d_4 .

Realizando esse procedimento para cada passo da simulação, construímos a disseminação na rede completa.

3 Resultados

Todos os resultados a seguir foram obtidos utilizando o conjunto de dados referente às notícias do atentado ao Charlie Hebdo na mídia brasileira. Esse conjunto de dados contém 2129 artigos, abaixo temos um gráfico da quantidade de artigos pelo tempo:

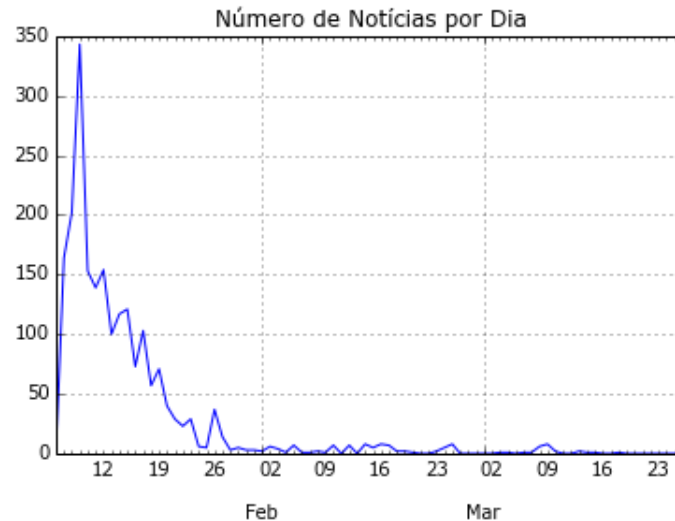


Figura 7: Notícias x Tempo

Observamos que o crescimento de artigos por tempo dessa notícia foi bem alto durante o mês de janeiro, e que após atingir seu máximo, começou a cair gradativamente.

3.1 Rede de Disseminação Real

Utilizando os dados descritos na seção anterior, construímos a matriz de documentos, que obteve o tamanho 2115x300, onde 2115 são as linhas que representam cada artigo e o 300 se refere às colunas, que são os atributos definidos pelo modelo *skipgram*.

Podemos observar que o número de artigos na matriz é inferior ao número de artigos totais do conjunto de dados. Isso aconteceu pois alguns artigos não obtiveram resultados satisfatórios durante o processo de tokenização, e por isso, foram descartados do conjunto de dados.

Possuindo a matriz de documentos, calculamos então, a matriz de similaridades para podermos construir a rede de disseminação.

3.1.1 Definindo Parâmetros

Após calcular a matriz de documentos e a matriz de similaridades, antes de construir a rede de disseminação, precisamos definir o tempo máximo de influência de um artigo, e a similaridade de cosseno mínima para definir se dois artigos são similares. Para isso calculamos a distribuição de cada uma dessas métricas:

incluir
distribuição
de tempo

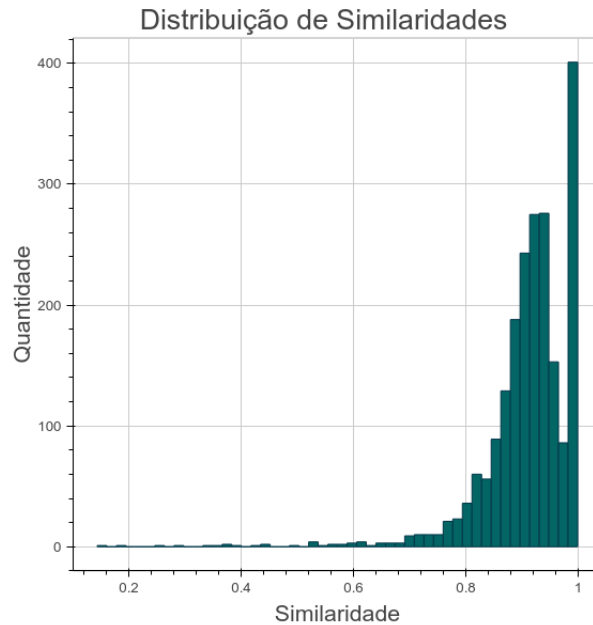


Figura 8: Distribuição de Similaridades Máximas

Observando as distribuições acima, definimos como similaridade mínima x e tempo máximo de influência de um artigo y .

3.1.2 Visualizações da Rede Real

Utilizando os parâmetros definidos acima, construímos finalmente nossa rede de disseminação do conjunto de dados escolhido. A rede encontrada possui 1786 vértices, onde cada vértice é um artigo.

Durante a criação da rede, artigos que não possuíam uma relação de influência com nenhum outro foram descartados, o que explica a diferença do número de nós para o número de artigos presentes na matriz de documentos.

Abaixo temos a distribuição de *out-degrees* da rede:

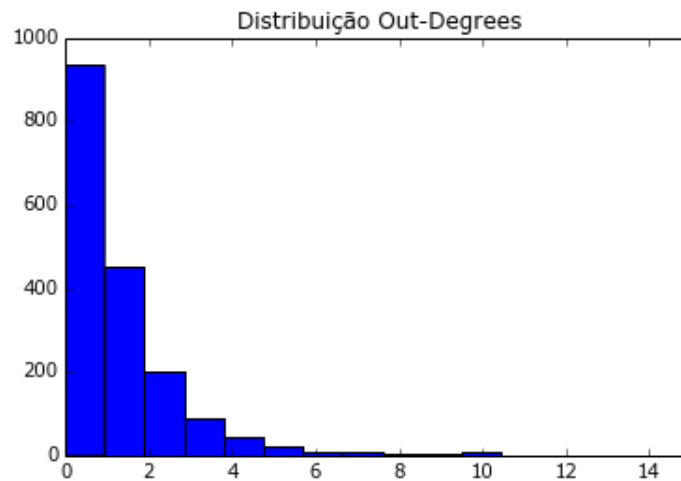


Figura 9: Distribuição Out-degrees Rede Real

Podemos reparar que, como esperado, possuímos poucos nós com grande grau de influência, isso é, poucos artigos são muito influentes na mídia.

Abaixo podemos ver uma visualização da rede criada. Nela, o eixo x representa o passo da infecção, ou seja, os vértices que estão na segunda coluna foram influenciados pelos que estão na primeira coluna. Os que estão na terceira coluna foram influenciados pelos da segunda, assim sucessivamente.

O tamanho e cor do vértice indicam o valor do out-degree, quanto maior e mais escuro maior seu out-degree. O mesmo se aplica ao tamanho do veículo de cada vértice, ou seja, o veículo que publicou o artigo referente ao vértice em questão.

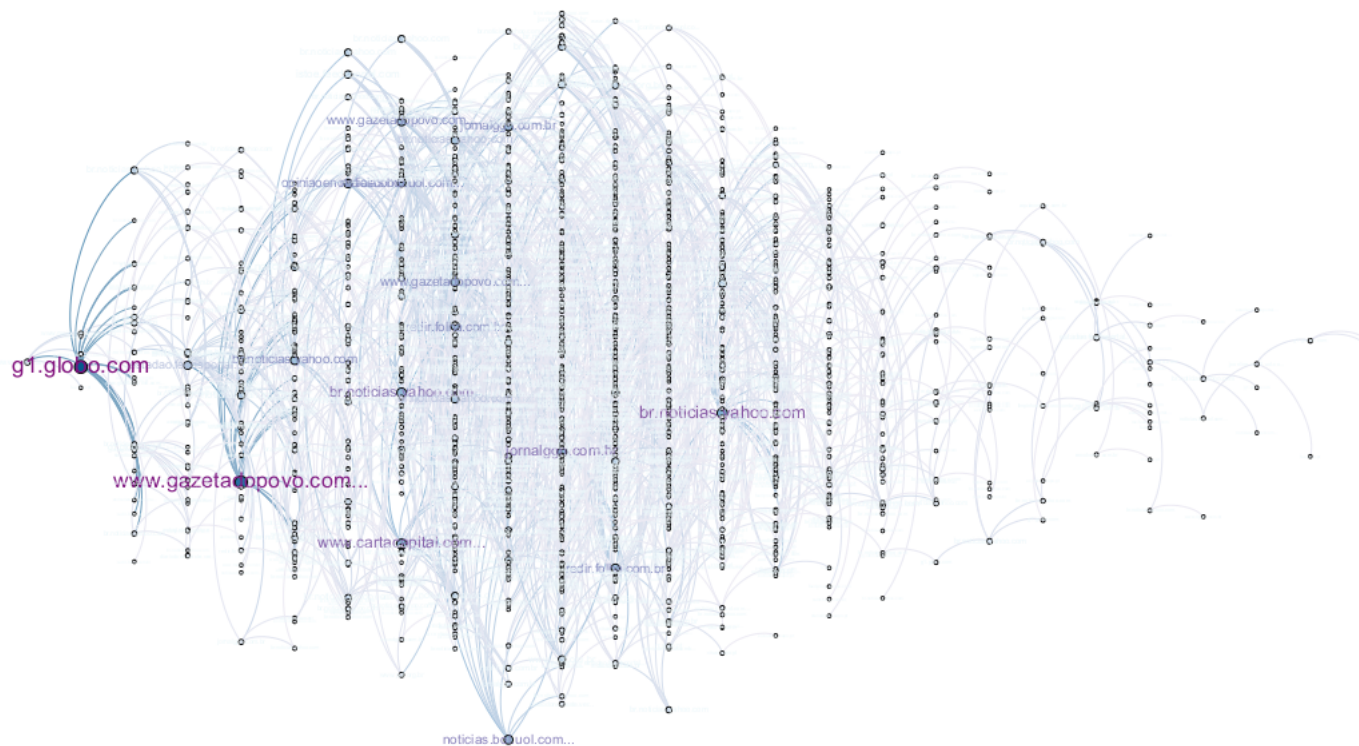


Figura 10: Rede Disseminação Real x Passo

Observando a visualização, vemos que a epidemia acontece rapidamente. Podemos visualizar também alguns veículos como principais influenciadores em nossa rede, os dois mais influentes são o g1.globo.com e o gazetadopovo.com.br.

Outra forma de visualizar nossa rede é organizar os vértices em função do tempo ao invés do passo de contágio. Abaixo vemos a visualização onde o eixo x é o tempo, e cada coluna é o tempo t de publicação dos artigos presente nela. Para construir a visualização, definimos como tempo t_0 a data de publicação do primeiro artigo publicado. Como espaçar t de hora em hora nos resultou em uma visualização muito confusa, resolvemos espaçar de 3 em 3 horas. Dessa forma, na segunda coluna t_1 encontramos todos os artigos publicados 3 horas após a publicação do primeiro artigo, e assim por diante.

Encontramos assim, uma visualização bem longa, com um formato semelhante de um cone deitado. Na medida que o tempo vai passando, nossas colunas vão diminuindo até não termos mais nenhum artigo. Abaixo mostramos um corte da parte inicial da visualização:

3.2.1 Resultado Simulação

Possuindo a matriz de adjacência da rede completa, podemos realizar a simulação do modelo. Porém, precisamos primeiro definir o valor do parâmetro λ . Para definir esse valor, realizamos a simulação para valores variados de λ , construímos as redes a partir desses resultados e calculamos a diferença de *Kullback-Leibler*. O λ que minimizasse essa diferença, seria o escolhido. Como sabemos que o λ crítico do modelo é dado por:

$$\lambda_c = \frac{1}{\Lambda}$$

Onde Λ é o maior autovalor da matriz de adjacência (Wang et al. [2003], Chakrabarti et al. [2008]), então testamos valores próximos ao λ_c . Vemos na tabela abaixo alguns valores de diferenças encontrados:

λ	KL
$\lambda_c + 0.1$	y
$\lambda_c + 0.01$	0.014
$\lambda_c + 0.5$	y

rodar o código para pegar esses valores

O valor de λ que minimizou a diferença de Kullback-Leibler foi $\lambda = 0.1$, logo, foi o λ utilizado para obter os resultados a seguir. Abaixo vemos o resultado da simulação feita com 3000 passos e utilizando o parâmetro escolhido:

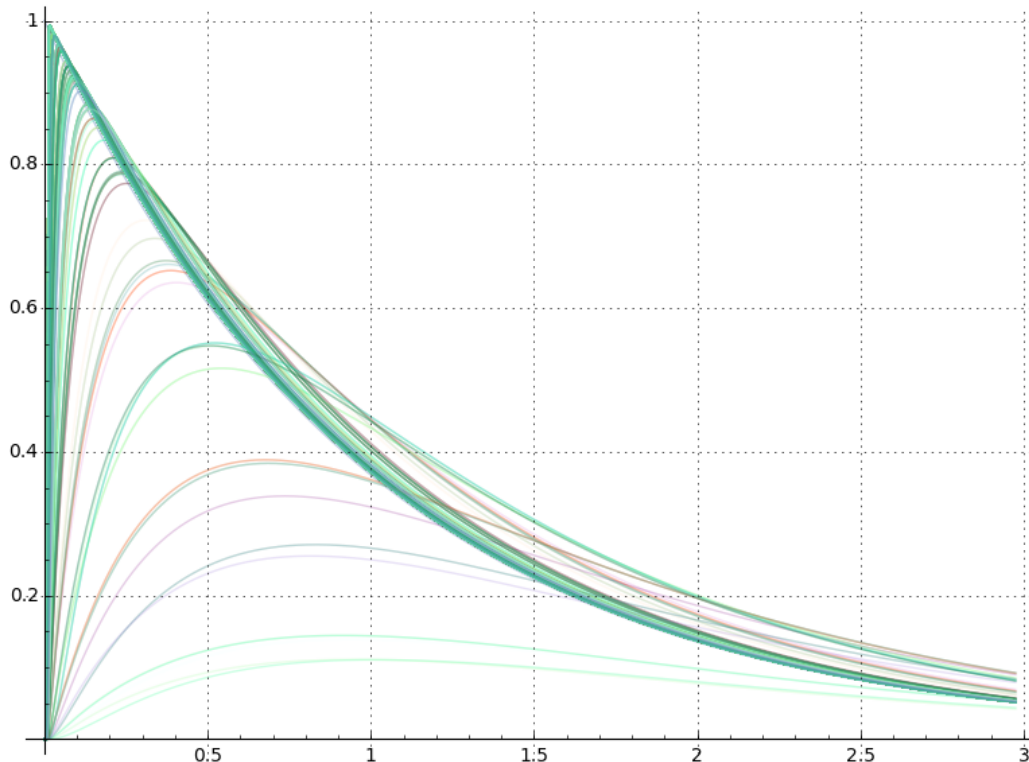


Figura 12: Simulação Modelo

Na figura acima possuímos 500 curvas, onde cada curva representa curva de infecção de um artigo. Cada ponto do gráfico diz respeito à probabilidade do artigo estar infectado no tempo t do modelo.

O resultado da simulação é um típico resultado para um modelo epidemiológico SIR, o que mostra que nossa matriz de adjacências foi coerente para a modelagem do sistema e que nossa rede de disseminação real pode ser descrita sim como um processo epidemiológico.

3.2.2 Rede Simulada

A simulação do modelo nos retorna uma matriz de estados para cada passo t . Utilizando essa matriz, construímos a rede de disseminação simulada.

3.2.3 Visualizações da Rede Simulada

Para visualizar a rede simulada, recriamos as duas visualizações feitas para a rede real de dados. Abaixo podemos ver a visualização onde o eixo x é o tempo t , ou seja, cada passo dado para realizar a simulação. Como nosso modelo é adimensional, não foi possível traduzir o passo da simulação para uma unidade de tempo comparável à visualização da rede original.

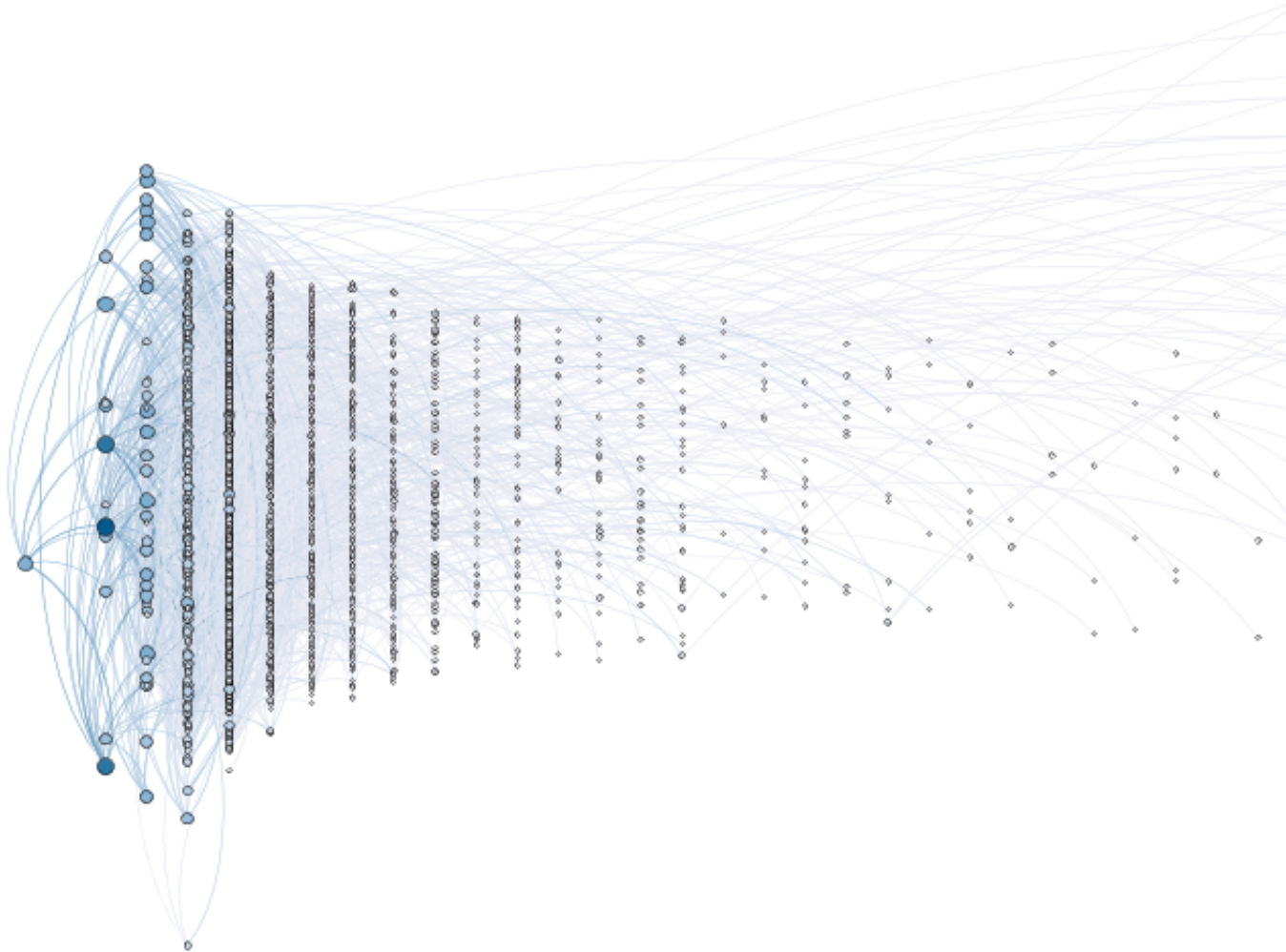


Figura 13: Rede Simulada x Tempo

O resultado dessa visualização tem um formato parecido com a visualização por tempo da rede real, e por também ser longa, selecionamos apenas seu início como mos-

trado na figura acima. O primeiro fator que observamos ao analisar a visualização é o fato de todos os vértices que possuem um out-degree maior do que um estarem concentrados nos três primeiros passos da simulação. Isso nos mostra que a infecção ocorreu mais rápido na simulação do que na rede real. Para chegarmos a mais conclusões observamos a rede abaixo, onde o eixo x consiste no passo da infecção, assim como fizemos para a rede real:

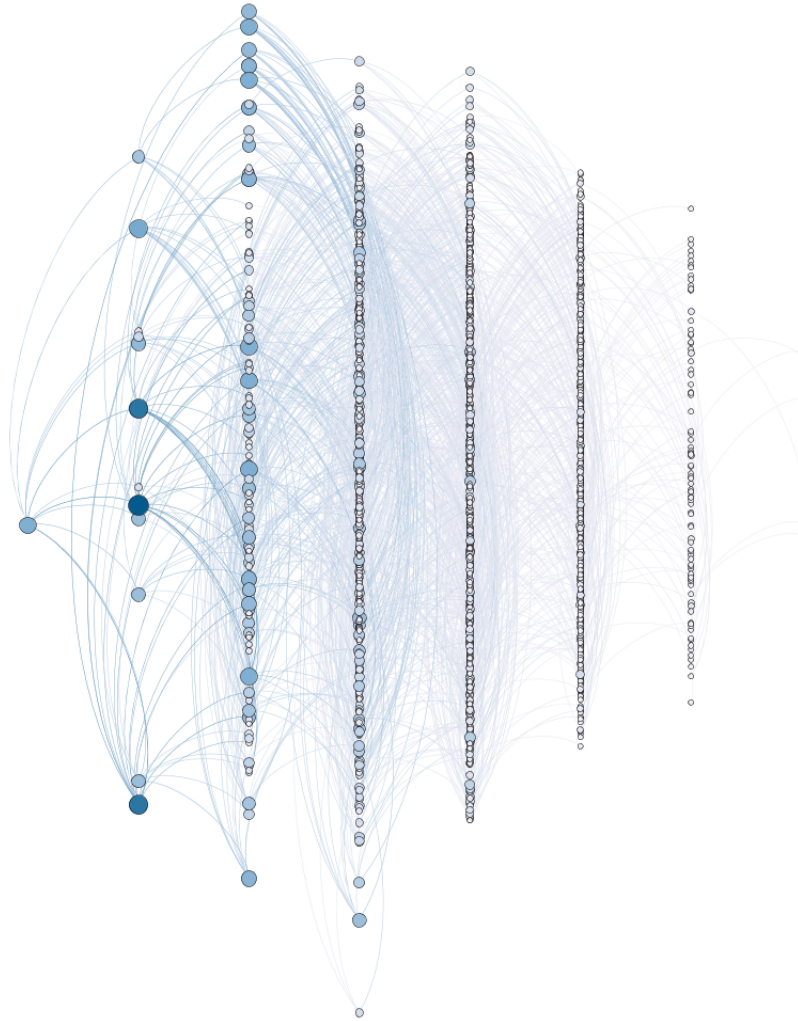


Figura 14: Rede Simulada x Passo

Essa visualização intensifica o observado na figura anterior, em poquíssimo tempo a epidemia se espalhou, necessitando de apenas alguns passos para o ciclo da epidemia ocorrer.

O motivo da epidemia na rede simulada ter ocorrido muito mais rápido do que na rede real é basicamente pelo fato de estarmos lidando com uma taxa de infecção (λ) constante. Como no mundo real a taxa de infecciosidade varia de acordo com fatores externos, a infecção demora mais para atingir seu máximo. Ao utilizar uma taxa constante, estamos desconsiderando qualquer fator externo e supondo que não existem variações na infecciosidade da doença com o passar do tempo, o que explica a epidemia ocorrer tão rapidamente em nosso modelo.

4 Conclusão

Referências

- Luís MA Bettencourt, Ariel Cintrón-Arias, David I Kaiser, and Carlos Castillo-Chávez. The power of a good idea: Quantitative modeling of the spread of ideas from epidemiological models. *Physica A: Statistical Mechanics and its Applications*, 364:513–536, 2006.
- Deepayan Chakrabarti, Yang Wang, Chenxi Wang, Jurij Leskovec, and Christos Faloutsos. Epidemic thresholds in real networks. *ACM Transactions on Information and System Security (TISSEC)*, 10(4):1, 2008.
- Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013.
- Romualdo Pastor-Satorras, Claudio Castellano, Piet Van Mieghem, and Alessandro Vespignani. Epidemic processes in complex networks. *arXiv preprint arXiv:1408.2701*, 2014.
- Erzsébet Ravasz and Albert-László Barabási. Hierarchical organization in complex networks. *Physical Review E*, 67(2):026112, 2003.
- Y Wang, D Chakrabarti, C Wang, and C Faloutsos. Proceedings of the 22nd international symposium on reliable distributed systems. 2003.