Propagação em grafos bipartidos para extração de tópicos em fluxo de documentos textuais

Thiago de Paulo Faleiros

Thiago de Paulo Faleiros

Propagação em grafos bipartidos para extração de tópicos em fluxo de documentos textuais

Tese apresentada ao Instituto de Ciências Matemáticas e de Computação - ICMC-USP, como parte dos requisitos para obtenção do título de Doutor em Ciências - Ciências de Computação e Matemática Computacional. *VERSÃO REVISADA*

Área de Concentração: Ciências de Computação e Matemática Computacional

Orientador: Prof. Dr. Alneu de Andrade Lopes

USP – São Carlos
Agosto de 2016

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados fornecidos pelo(a) autor(a)

F187p

Faleiros, Thiago de Paulo
Propagação em grafos bipartidos para extração de tópicos em fluxo de documentos textuais / Thiago de Paulo Faleiros; orientador Alneu de Andrade Lopes. -- São Carlos, 2016.
160 p.

Tese (Doutorado - Programa de Pós-Graduação em Ciências de Computação e Matemática Computacional) -- Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 2016.

1. Aprendizado em grafos bipartidos. 2. Extração de tópicos. 3. Fluxo de dados textuais. 4. Redução de Dimensionalidade. I. Lopes, Alneu de Andrade , orient. II. Título.

Thiago de Paulo Faleiros

Propagation in bipartite graphs for topic extraction in stream of textual data

Doctoral dissertation submitted to the Instituto de Ciências Matemáticas e de Computação – ICMC-USP, in partial fulfillment of the requirements for the degree of the Doctorate Program in Computer Science and Computational Mathematics. *FINAL VERSION*

Concentration Area: Computer Science and Computational Mathematics

Advisor: Prof. Dr. Alneu de Andrade Lopes

USP – São Carlos
August 2016

Dedico esta tese a memória de meu irmão, Wesley Faleiros de Paulo,
que se foi tão jovem, deixando muita saudade...

AGRADECIMENTOS

À Deus pelo amparo naquela hora de dificuldade.

Aos meus pais Vilma Faleiros de Paulo e João Batista de Paulo, pelo grande apoio e incentivo durante o desenvolvimento de meus estudos. A Juliana Aliques, pela paciência, por acreditar na minha capacidade e por estar ao meu lado.

A todos os amigos e colegas do LABIC, pela convivência diária e pelas "hora do café", que foram uma excelente forma de desestressar. Em especial, gostaria de agradecer pelo companheirismo e colaboração nos trabalhos realizados aos até então alunos Jorge Valverde-Rebaza, Lilian Berton, Alan Valejo, Rafael Rossi e Ricardo Puma.

Ao meu orientador prof. Dr. Alneu de Andrade Lopes, por me orientar e acreditar no meu trabalho. Não me recordo se alguma vez ele gastou mais de uma semana para revisar algum texto meu. Mas isso só mostra o seu comprometimento em auxiliar os seus alunos.

Ao prof. Dr. Jordan Boyd-Graber e a University of Maryland por aceitarem o meu projeto de estágio no exterior.

À Fundação de Amparo à Pesquisa do Estado de São Paulo (processo nº 2011/23689-9, FAPESP), pela concessão da bolsa de doutorado, pelo apoio financeiro para a realização desta pesquisa e pelo financiamento do estágio sanduíche. À USP pela infra-estrutura oferecida.

"...Quem gosta de viver não tem preguiça de reinventar, nem medo de ousar.
Quem gosta de viver não tem medo de ternura, da gentileza, do amor.
Quem gosta de viver, educa!..."
(Gabriel Chalita)

RESUMO

FALEIROS, T. P.. **Propagação em grafos bipartidos para extração de tópicos em fluxo de documentos textuais**. 2016. 160 f. Tese (Doutorado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação (ICMC/USP), São Carlos – SP.

Tratar grandes quantidades de dados é uma exigência dos modernos algoritmos de mineração de texto. Para algumas aplicações, documentos são constantemente publicados, o que demanda alto custo de armazenamento em longo prazo. Então, é necessário criar métodos de fácil adaptação para uma abordagem que considere documentos em fluxo, e que analise os dados em apenas um passo sem requerer alto custo de armazenamento. Outra exigência é a de que essa abordagem possa explorar heurísticas a fim de melhorar a qualidade dos resultados. Diversos modelos para a extração automática das informações latentes de uma coleção de documentos foram propostas na literatura, dentre eles destacando-se os modelos probabilísticos de tópicos. Modelos probabilísticos de tópicos apresentaram bons resultados práticos, sendo estendidos para diversos modelos com diversos tipos de informações inclusas. Entretanto, descrever corretamente esses modelos, derivá-los e em seguida obter o apropriado algoritmo de inferência são tarefas difíceis, exigindo um tratamento matemático rigoroso para as descrições das operações efetuadas no processo de descoberta das dimensões latentes. Assim, para a elaboração de um método simples e eficiente para resolver o problema da descoberta das dimensões latentes, é necessário uma apropriada representação dos dados. A hipótese desta tese é a de que, usando a representação de documentos em grafos bipartidos, é possível endereçar problemas de aprendizado de máquinas, para a descoberta de padrões latentes em relações entre objetos, por exemplo nas relações entre documentos e palavras, de forma simples e intuitiva. Para validar essa hipótese, foi desenvolvido um arcabouço baseado no algoritmo de propagação de rótulos utilizando a representação em grafos bipartidos. O arcabouço, denominado PBG (*Propagation in Bipartite Graph*), foi aplicado inicialmente para o contexto não supervisionado, considerando uma coleção estática de documentos. Em seguida, foi proposta uma versão semissupervisionada, que considera uma pequena quantidade de documentos rotulados para a tarefa de classificação transdutiva. E por fim, foi aplicado no contexto dinâmico, onde se considerou fluxo de documentos textuais. Análises comparativas foram realizadas, sendo que os resultados indicaram que o PBG é uma alternativa viável e competitiva para tarefas nos contextos não supervisionado e semissupervisionado.

Palavras-chave: Aprendizado em grafos bipartidos, extração de tópicos, fluxo de dados textuais, redução de dimensionalidade.

ABSTRACT

FALEIROS, T. P.. **Propagação em grafos bipartidos para extração de tópicos em fluxo de documentos textuais**. 2016. 160 f. Tese (Doutorado em Ciências – Ciências de Computação e Matemática Computacional) – Instituto de Ciências Matemáticas e de Computação (ICMC/USP), São Carlos – SP.

Handling large amounts of data is a requirement for modern text mining algorithms. For some applications, documents are published constantly, which demand a high cost for long-term storage. So it is necessary easily adaptable methods for an approach that considers documents flow, and be capable of analyzing the data in one step without requiring the high cost of storage. Another requirement is that this approach can exploit heuristics in order to improve the quality of results. Several models for automatic extraction of latent information in a collection of documents have been proposed in the literature, among them probabilistic topic models are prominent. Probabilistic topic models achieve good practical results, and have been extended to several models with different types of information included. However, properly describe these models, derive them, and then get appropriate inference algorithms are difficult tasks, requiring a rigorous mathematical treatment for descriptions of operations performed in the latent dimensions discovery process. Thus, for the development of a simple and efficient method to tackle the problem of latent dimensions discovery, a proper representation of the data is required. The hypothesis of this thesis is that by using bipartite graph for representation of textual data one can address the task of latent patterns discovery, present in the relationships between documents and words, in a simple and intuitive way. For validation of this hypothesis, we have developed a framework based on label propagation algorithm using the bipartite graph representation. The framework, called PBG (Propagation in Bipartite Graph) was initially applied to the unsupervised context for a static collection of documents. Then a semi-supervised version was proposed which need only a small amount of labeled documents to the transductive classification task. Finally, it was applied in the dynamic context in which flow of textual data was considered. Comparative analyzes were performed, and the results indicated that the PBG is a viable and competitive alternative for tasks in the unsupervised and semi-supervised contexts.

Key-words: Learning in bipartite graphs, topic extraction, text data stream, dimensionality reduction.

LISTA DE ILUSTRAÇÕES

LISTA DE QUADROS

LISTA DE ALGORITMOS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

AN ..... artigos de notícias
DC ..... documentos científicos
DM ..... documentos médicos
DTM ..... *Dynamic Topic Model*
ELBO .... *Evidence Lower Bound*
EM ..... *Expectation Maximization*
ES ..... *e-mails*
GFHF .... *Gaussian fields and Harmonic Functions*
GM ..... *GNetMine*
HDP ..... *Hierarchical Dirichlet Processes*
HLC ..... *Hierarchical Link Clustering*
IMBHN .. *Inductive Model Based on Bipartite Heterogeneous Network*
KL ..... Kullback-Leibler
LDA ..... *Latent Dirichlet Allocation*
LLGC .... *Learning with Local and Global Consistency*
LP ..... *Label Propagation with Gaussian Fields and Harmonic Functions*
LPBHN .. *Label Propagation using Bipartite Heterogeneous Networks*
LSI ..... *Latent Semantic Indexing*
MNB ..... *Multinomial Naive Bayes*
NMF ..... *Nonnegative Matrix Factorization*
NPMI .... *Normalized Pointwise Mutual Information*
oLDA .... versão *online* do LDA
oPBG .... *online Propagation on Bipartite Graph*
PBG ..... *Propagation in Bipartite Graph*
pLSI ..... *probabilistic Latent Semantic Indexing*
PMI ..... *Pointwise Mutual Information*
PW ..... páginas *web*
SA ..... análise de sentimentos
SSD ..... *Statistically Significant Differences*
ST ..... *Multinomial Nave Bayes with Self-Training*
SVD ..... *Singular Value Decomposition*

SVM ..... *Support Vector Machine*

TB ..... *Tag-based Model*

TPBG .... *Transductive Propagation in Bipartite Graph*

TSVM ... *Transductive Support Vector Machines*

LISTA DE SÍMBOLOS

K — Número de tópicos.

Γ — Função Gamma.

Ψ — Função Digama.

m — Número de documentos em um corpus.

n — Número de palavras distintas de um corpos de documentos.

d_j — j-ésimo documento da coleção

w_i — i-ésima palavra do vocabulário da coleção.

n_{d_j} — Número de palavras em um documento d_j.

θ — Distribuição de tópicos por documentos no modelo LDA.

ϕ — Distribuição de tópicos por palavras no modelo LDA.

α — Parâmetro de concentração no algoritmo PBG. Priori da distribuição de Dirichlet no modelo LDA.

β — Priori da distribuição de Dirichlet no modelo LDA.

γ — Distribuição variacional de tópicos por documentos no modelo LDA.

λ — Distribuição variacional de tópicos por palavras no modelo LDA.

φ — Distribuição variacional de tópicos de uma palavra no documento do modelo LDA.

$\mathbb{R}$ — Conjunto dos números reais.

G — Grafo

$\mathscr{C}$ — Conjunto de rótulos de classes.

$\mathscr{D}$ — Conjunto de vértices representando os documentos.

$\mathscr{D}^l$ — Conjunto de vértices representando os documentos rotulados.

$\mathscr{D}^u$ — Conjunto de vértices representando os documentos não rotulados.

$\mathscr{W}$ — Conjunto de vértices representando as palavras da coleção de documentos.

$\mathscr{E}$ — Conjunto de arestas.

A_j — Rótulo multidimensional, ou vetor de pesos, associado ao vértice representando um documento d_j.

Y_j — Rótulo multidimensional, ou vetor de pesos, associado ao vértice representando um documento rotulado d_j

$\mathscr{Y}$ — Conjunto de rótulos multidimensionais associados aos documentos rotulados

B_j — Rótulo multidimensional, ou vetor de pesos, associado ao vértice representando uma palavra w_i.

$C_{e_{j,i}}$ — Rótulo multidimensional, ou vetor de pesos, associado a uma aresta $e_{j,i}$.

$\mathscr{R}(\cdot)$ — Função de regularização.

$\pi(\cdot)$ — Função que atribui a cada documento um grupo.

$\mathscr{T}$ — Um pequeno conjunto de documentos, ou *mini-batch* de documentos que chegou no fluxo.

∇ — Operador para o cálculo do vetor gradiente.

ρ — Fator de aprendizado *online*, ou o passo em direção do gradiente estocástico.

κ — Fator de controle da velocidade com que novos vetores substituem os antigos no processo atualização incremental dos algoritmos *online*.

τ_0 — Fator que controla a inércia do aprendizado.

SUMÁRIO

CAPÍTULO

1

INTRODUÇÃO

Existe uma grande quantidade de dados no formato textual. De fato, a forma mais simples de armazenar informação é no formato de texto. Maneiras automáticas para auxiliar na organização e na extração de informações no formato textual é um tópico de pesquisa interessante e uma tarefa desafiadora pois envolve a manipulação de dados não estruturados. Além disso, mineração de texto é uma área interdisciplinar que faz o uso de técnicas avançadas de mineração de dados, aprendizado de máquinas, recuperação de informação, extração de informação, linguística computacional e processamento de linguagem natural (SUMATHY; CHIDAMBARAM, 2013).

A extração de informação em dados textuais envolve diretamente a tentativa de se extrair informações úteis em coleções de documentos textuais. O que se objetiva nessa abordagem é a criação de uma representação estruturada das informações retiradas dessas coleções de documentos (AGGARWAL; ZHAI, 2012). O modelo espaço vetorial é o modelo mais tradicional para obter uma representação estruturada dos documentos. Nesse modelo, documentos e textos são representados por um conjunto de atributos/termos[1] e pesos associados a esses atributos de acordo com a frequência desses termos nos documentos. Assim, a coleção pode ser representada por uma matriz documento-termo. Outras representações consideram o grafo como forma de representação da coleção, sendo que a representação dos documentos, termos ou ambos são os vértices e a relação entre esses objetos, medida por uma função de similaridade, são as arestas. Neste trabalho, esta última é adotada e será apresentada em detalhes posteriormente.

Estruturar automaticamente dados não estruturados está fortemente relacionado com as tarefas de agrupamento e classificação. Modelos de tópicos é uma abordagem que foi aplicada com sucesso nessas tarefas, sendo seu principal objetivo descobrir dimensões latentes de um corpus (BLEI, 2011). Nesta tese, a expressão **tópico** é usada levando-se em conta que o assunto tratado em uma coleção de documentos é extraído automaticamente, ou seja, tópico é definido

[1] Nesse trabalho, **termos** e **atributos** são utilizados com o mesmo significado

como um conjunto de palavras que frequentemente ocorrem em documentos semanticamente relacionados. Esses conjuntos de palavras (que definem os tópicos) são obtidos por um processo de pós-processamento realizado a partir das dimensões latentes descobertas pela aplicação dos métodos de modelo de tópicos.

Modelos de tópicos têm ganhado bastante atenção e sido alvo de várias pesquisas (BLEI; NG; JORDAN, 2003; BLEI, 2011). A ideia básica dos modelos de tópicos é descobrir, nas relações entre documentos e termos, padrões latentes que sejam significativos para o entendimento dessas relações. Por exemplo, tais modelos podem ranquear um conjunto de termos como importantes para um ou mais temas. Bem como ranquear documentos como tendo relevância para um ou mais temas. Se for associado um vetor A_j a um documento d_j e um vetor B_i para um termo w_i, sendo A_j e B_i K-dimensionais, pode-se considerar que cada uma dessas dimensões caracterizam um fator latente relacionado com um documento A_j e um termo B_i. Assim o produto interno $A_j \cdot B_i$ pode modelar a importância desses fatores na relação documento d_j e termo w_i. Para toda a coleção, o problema base dos modelos de tópicos é encontrar duas matrizes não negativas A e B, tal que cada linha do produto $(A \cdot B^t)$ aproxima-se da linha da matriz documento-termo. Esse problema é conhecido como fatoração de matrizes não negativas (*Nonnegative Matrix Factorization* (NMF)) (PAATERO; TAPPER, 1994; LEE; SEUNG, 1999). Caso as linhas das matrizes A e B sejam amostras independentes e identicamente distribuídas de uma distribuição de Dirichlet, esse problema é descrito pelo modelo probabilístico *Latent Dirichlet Allocation* (LDA) (ARORA; GE; MOITRA, 2012). A abordagem predominante para resolução dos modelos de tópicos é o uso de algoritmos iterativos nos quais se objetiva maximizar a verossimilhança do modelo. Nos trabalhos de Sontag e Roy (2011), Arora, Ge e Moitra (2012) são apresentadas provas de que estimadores de verossimilhança máxima para os problemas de modelos de tópicos são problemas NP-Difíceis, logo, encontrar soluções ótimas, ou mesmo aproximadas, para esse tipo de problema é um desafio computacional. Além disso, uma questão principal nas abordagem aplicadas em modelos de tópicos é que elas contam com procedimentos iterativos susceptíveis a ótimos locais. Por isso, soluções baseadas em heurísticas podem ser alternativas eficientes para resolver o problema de modelos de tópicos, e auxiliar na melhoria das soluções via técnicas que possam escapar das regiões de ótimos locais.

O objetivo de uma heurística é tentar encontrar uma solução "boa" de maneira simples e rápida. Apesar de não garantir a solução ótima do problema, técnicas heurísticas são capazes de retornar uma solução de qualidade em um tempo adequado para as necessidades da aplicação. Para isso, o problema deve ser modelado de tal forma que fique "fácil" de resolver. Porém, modelos de tópicos baseados na representação matricial da coleção de documentos são modelados como um problema de decomposição de matrizes, o que faz com que as soluções propostas tenham alto consumo de memória e tempo computacional. Ainda mais no contexto de mineração de texto, onde as matrizes que descrevem coleções de documentos são esparsas. Já nos modelos probabilísticos de tópicos, existe um tratamento matemático rigoroso para descrição das operações efetuadas no processo de descoberta de tópicos. Na perspectiva de um desenvolvedor

de aplicações práticas, criar um modelo generativo e derivá-lo a fim de obter um algoritmo de inferência implementável é uma tarefa difícil. O rigor matemático desafia uma rápida exploração de novas suposições, heurísticas, ou adaptações que podem ser úteis em vários cenários reais. Além disso, não é claro como incluir heurísticas dentro dos processos realizados por técnicas como o LDA e NMF.

Assim, para a elaboração de um método heurístico simples e eficiente que resolva o problema fundamental em modelos de tópicos, é necessário, inicialmente, uma forma de representação simples e intuitiva dos dados e da solução. E com essa representação, descrever as operações para a obtenção da solução do problema. Inspirado em métodos de busca local, a construção do algoritmo heurístico deve explorar o espaço de busca partindo de uma solução inicial e, iterativamente, realizar melhorias nessa solução corrente através de uma busca em sua vizinhança até não existir soluções melhores. Com base nisso, algumas questão fundamentais nortearam a pesquisa desenvolvida neste trabalho: 1) como representar os dados e a solução para o problema? 2) como definir a estratégia de busca para o problema de extração de tópicos em fluxo de documentos?

Os modelos mais populares em modelagem de tópicos, como o LDA e NMF, possuem uma característica muito importante, que é relacionar documentos e termos e agrupá-los simultaneamente. Os termos, portanto, não são apenas características que descrevem o documento, mas são também objetos que podem ser agrupados de acordo com fatores latentes das relações documentos-termos. Nesses modelos, o agrupamento é realizado para os objetos do tipo documento e para os objetos do tipo termo. Na forma tradicional de agrupamento, onde se considera apenas os objetos de um mesmo tipo, existem técnicas eficientes tanto na representação vetorial quanto em grafos (AGGARWAL; ZHAI, 2012). É possível encontrar os grupos de documentos aplicando técnicas tradicionais de agrupamento, e, em seguida, realizar o pós-processamento para obter os grupos de termos. Porém, essa abordagem não permite o "enriquecimento" do método com a inclusão de heurísticas que podem melhorar os resultados. Uma abordagem que representa documentos, termos e suas relações possibilita a inclusão de informações adicionais diretamente relacionadas aos objetos do tipo documento ou termo (ou ambos), enriquecendo o método de forma a obter melhores resultados. Assim, considerando o desejo de representar de forma apropriada coleções de documentos, neste trabalho é proposto uma abordagem simples e descritiva que utiliza representação em grafo bipartido. A representação por grafo bipartido é intuitiva, documentos e palavras são vértices, e a ocorrência da palavra no documento são as arestas.

Uma hipótese importante levantada neste trabalho é a de que, usando a representação da coleção de documentos em grafo bipartido, é possível endereçar problemas de aprendizado de máquina de forma simples. Para validar essa hipótese, todo um arcabouço teve que ser desenvolvido. Antes de atacar diretamente o problema de extração de tópicos em fluxo de documentos, é descrito o desenvolvimento desse arcabouço. Inicialmente, foi desenvolvido

um algoritmo de classificação supervisionado baseado na representação em grafos bipartidos (ROSSI *et al.*, 2014; ROSSI *et al.*, 2012). Esse algoritmo foi desenvolvido pelo grupo de pesquisa e não é a principal contribuição desta tese. Apesar disso, a formulação do algoritmo teve a participação do aluno de doutorado que desenvolveu esta tese, e também serviu de inspiração para a generalização do aprendizado baseado em grafos bipartidos considerando o contexto semissupervisionado e não supervisionado, apresentados aqui como o algoritmo *Propagation in Bipartite Graph* (PBG). O arcabouço teve o desenvolvimento inicial para a contexto não supervisionado, com a aplicação nas tarefas de redução de dimensionalidade e extração de tópicos. Não foi considerado que a coleção de documentos está em fluxo. Isso foi necessário pois é mais fácil validar o algoritmo no contexto estático. Aproveitando esse arcabouço, foram incluídas algumas heurísticas simples de inicialização para melhorar a qualidade dos resultados no contexto não supervisionado e também proposto uma versão paralelizada para melhorar o tempo de processamento. Em seguida, foi proposta uma versão semissupervisionada, que utiliza uma pequena quantidade de documentos rotulados para a tarefa de classificação transdutiva. E por fim, esse arcabouço foi aplicado no contexto dinâmico, onde se considera fluxo de documentos. Todos os algoritmos são fundamentados na otimização da divergência entre os vetores associados aos vértices do grafo bipartido, e a validação é realizada por meio de experimentos e análise comparativa com os algoritmos estado-da-arte.

O arcabouço desenvolvido neste projeto para o aprendizado de máquina utilizando a representação de documentos textuais em grafos bipartidos é baseado no algoritmo de propagação de rótulos. Porém, no método aqui proposto, os rótulos são vetores K-dimensionais (onde K é o número de tópicos, grupos ou classes) atribuídos a cada vértice do grafo bipartido. Diferentemente da técnica tradicional de propagação de rótulos, o que é propagado no método proposto neste trabalho são os valores contidos nos vetores associados a cada vértice. A estrutura de propagação é um grafo bipartido representando documentos, termos e suas relações. É demonstrado no capítulo posterior que o algoritmo de propagação proposto é de fato um procedimento de otimização da divergência entre os vetores associados a cada vértices e os vetores associados a seus vértices vizinhos. A premissa básica do algoritmo proposto é que vértices (que representam documentos e palavras) que estão altamente conectados devem possuir vetores com informações de tópicos similares, e que vértices distintos, ou vizinhança de vértices com poucas ligações, devem ter vetores com informações de tópicos distintos. Utilizando a abordagem baseada na representação em grafo bipartido proposta, foram encontrados resultados similares, em alguns casos superiores, as técnicas LDA e NMF.

No contexto não supervisionado, a propagação de rótulos assume que os rótulos atribuídos aos vértices são índices dos grupos. Para o problema de extração de tópicos, os rótulos são vetores K-dimensionais onde a posição k desses vetores correspondem ao grau de filiação do vértice ao tópico k. Os valores desses vetores podem ser iniciados aleatoriamente. Porém, é possível melhorar substancialmente a solução inicial aplicando uma heurística de agrupamento para encontrar bons rótulos. Essa simples aplicação de heurísticas na inicialização dos dados mostrou

uma significativa melhora nos resultados finais em comparação com as técnicas LDA e NMF. Outra melhora obtida pela representação da coleção como grafo bipartido foi a possibilidade de definir subestruturas do grafo. Com isso, foi possível dividir de forma simples o problema em subproblemas, de modo que a resolução de todos os subproblemas possam compor uma solução para o problema maior. Esses subproblemas foram identificados como subestruturas locais do grafo, e com isso os procedimentos de propagação foram divididos em propagação local e propagação global. Para tirar vantagem disso, foi desenvolvido uma versão paralela do algoritmo proposto, trazendo melhora no tempo de convergência total. A versão paralela aplica várias propagações locais simultaneamente em diferentes processos, e uni as soluções no procedimento de propagação global.

Com a utilização da heurística de propagação em grafo bipartido, foi possível estender o algoritmo do contexto não supervisionado para o semissupervisionado. Em específico, foram utilizados documentos previamente rotulados e documentos não rotulados para melhorar a tarefa de classificação transdutiva. Na abordagem transdutiva, os rótulos dos documentos não rotulados são estimados diretamente sem a criação de um modelo de classificação. Na versão semissupervisionada, presume-se que os documentos rotulados possuem a solução ótima, simplificando o problema fixando o valor dos vetores associados a documentos rotulados. Os resultados dos experimentos realizados em comparação com outros algoritmos de classificação transdutiva mostraram que essa abordagem é promissora, obtendo resultados superiores principalmente quando são considerados poucos documentos rotulados.

A estrutura local do grafo corresponde as ligações feitas por um único vértice do tipo documento para os vértices do tipo palavra que ocorrem no documento. Já a estrutura global são as ligações feitas para todos os vértices do tipo documento. Claramente, operações na estrutura global do grafo são dispendiosas, e se torna o gargalo em aplicações com grandes quantidades de documentos. Para superar esses problemas, foi proposta uma versão online do algoritmo de propagação em grafos bipartidos para o problema de extração de tópicos. Nessa versão online, a propagação é feita na estrutura local do grafo, e as propagações na estrutura global são alteradas para um esquema incremental. É mostrado que essa abordagem é semelhante a aplicação do método de gradiente estocástico no problema estabelecido pela otimização da divergência dos vetores associados aos vértices do grafo bipartido.

1.1 Objetivos e Hipótese

Motivado pelos desafios comentados anteriormente e pela necessidade de métodos para extração de tópicos que sejam simples, úteis nos cenários reais e de fácil inserção de conhecimento heurístico, este trabalho tem como objetivo investigar técnicas eficientes em aprendizado de máquinas e mineração de documentos textuais que permitam a extração de conhecimento temático. O problema de descoberta de tópicos está relacionado com o agrupamento de palavras

que ocorrem frequentemente em documentos correlacionados. Assim, acredita-se que a utilização de uma estrutura em grafos para relacionar documentos e palavras seja compensatória para a construção de modelos de extração de tópicos. Além disso, são investigadas as abordagem tradicionais de modelagem de tópicos e suas principais características.

A hipótese levantada neste trabalho é que a representação via grafos permite a geração de modelos de extração de tópicos eficazes, eficientes e simples de se adaptarem para trabalhar em outros domínios, como no caso de dados em fluxo.

O objetivo geral do projeto é investigar e desenvolver técnicas que combinem a representação expressiva possibilitada pela teoria de redes complexas com técnicas heurísticas que permitam tratar o problema de extração de tópicos em fluxo de textos. Os objetivos específicos que tratam de pontos de pesquisa ainda em aberto, são os seguintes:

- investigar os modelos de tópicos, estudar a formulação desses modelos, conhecer os detalhes dos algoritmos de inferência e os métodos computacionais aplicado para o problema de extração de tópicos, realizar comparações entre os métodos, e obter fundamentos para o desenvolvimento de novos algoritmos para extração de tópicos;
- desenvolver um arcabouço baseado na representação em grafo bipartido para a construção de algoritmos simples para o aprendizado e extração de informação em coleções de documentos textuais;
- aplicar o arcabouço desenvolvido em outros domínios, como o aprendizado transdutivo, realizar experimentos e a avaliação dos resultados;
- descobrir quais tópicos pertencem aos documentos textuais em tempo real por um processo automático, desenvolvendo, para isso, um algoritmo que descobre quais são esses tópicos e os incrementa gradualmente, à medida que novos documentos chegam;
- contribuir no desenvolvimento dos modelos para extração de tópicos em fluxo de documentos aplicando técnicas baseadas em grafos bipartidos.

1.2 Contribuições

O presente trabalho apresenta várias contribuições, as quais são descritas sucintamente a seguir.

Estudo detalhado sobre modelos probabilísticos de tópicos: No levantamento bibliográfico, encontrou-se vários trabalhos recentes que estendem ou modificam o modelo LDA. Isso revela que este modelo representa o estado da arte em extração de tópicos. Por isso foi conduzido um estudo detalhado dos modelos probabilísticos de tópicos, tendo os algoritmos de inferência do modelo LDA como base. Esse estudo foi importante para

perceber as nuances dos algoritmos e métodos para a extração de tópicos. Além disso, como contribuição dessa tese, foi realizada uma análise comparativa entre o LDA e NMF, onde foi possível concluir que a função objetivo do NMF com a divergência de Kullback-Leibler é uma aproximação da função objetivo estabelecida pelo método de inferência variacional aplicado no LDA com priori simétricas. Resultando no trabalho (FALEIROS; LOPES, 2016).

Aprendizado em grafos: Grafos são simplesmente vértices e arestas, porém é uma representação robusta, na qual as relações criadas entre os objetos representados são informações úteis para a tarefa de classificação. Aproveitando isso, foram investigados algoritmos de aprendizado em grafos e desenvolvidos os seguintes trabalhos em colaboração: (FALEIROS; BERTON; LOPES, 2012; VALVERDE-REBAZA *et al.*, 2015).

Elaboração de um algoritmo simples para a extração de tópicos: Grafo bipartido é uma das formas mais simples de se representar coleções de textos. E o projeto do algoritmo de propagação de rótulos é fácil de compreender. Assim, neste trabalho foi proposto um arcabouço baseado no algoritmo de propagação em grafos bipartidos para a aplicação em tarefas de aprendizado de máquinas.

Aprendizado em grafos bipartidos: Colaboração na proposta de um algoritmo de categorização de documentos textuais inspirado na estrutura de grafos bipartidos para a indução de um modelo de classificação. Resultando nos trabalhos (ROSSI *et al.*, 2012; ROSSI *et al.*, 2014). Proposta de um algoritmo semissupervisionado para a tarefa de transdução de coleções de documentos utilizando grafos bipartidos. Resultando no trabalho (FALEIROS; ROSSI; LOPES, 2016).

Elaboração do algoritmo *online* para a extração de tópicos: Extensão do algoritmo de propagação em grafos bipartidos para o ambiente de fluxo de documentos. Descrever a fundamentação do algoritmo *online* e como ele pode ser descrito considerando o arcabouço desenvolvido.

Todos os algoritmos desenvolvido nesta tese (desenvolvidos como trabalho principal ou coautoria) estão descritos na Tabela 1. Nessa tabela são descritas as tarefas principais nos quais os algoritmos foram aplicados e os procedimentos de iniciação. Para cada tipo de tarefa foi aplicado um método de avaliação adequado e foram realizadas comparações com um outro algoritmo na literatura que se adéqua a correspondente atividade.

1.3 Organização da Tese

Para melhor entendimento da pesquisa descrita nessa tese, no Capítulo 2 é revisado os modelos probabilísticos de tópicos, detalhando o modelo LDA, a formulação matemática, as

Tabela 1 – Algoritmos desenvolvidos como contribuição direta desta tese ou desenvolvidos como co-autoria. Nessa tabela são comentadas algumas diferenças entre esses algoritmos. Maiores detalhes são descritos no Capítulo 3

algoritmo	descrição	tarefa	iniciação
pbg	algoritmo não supervisionado	agrupamento, extração de tópicos, redução de dimensionalidade	aleatória
kmeans+pbg	algoritmo não supervisionado	agrupamento, extração de tópicos, redução de dimensionalidade	*k*-means
hcl+pbg	algoritmo não supervisionado	agrupamento, extração de tópicos, redução de dimensionalidade	algorithm hcl
pbg-threads	algoritmo não supervisionado paralelizado	agrupamento, extração de tópicos, redução de dimensionalidade	aleatória
imbhn (ROSSI *et al.*, 2012)	algoritmo supervisionado (não é contribuição principal desta tese)	classificação indutiva	aleatória
tpbg	algorithmo semissupervisionado	classificação transdutiva	aleatória
opbg	algoritmo não supervisionado *online*	agrupamento, extração de tópicos, redução de dimensionalidade	aleatória

Fonte: Elaborada pelo autor.

derivações do modelo, os algoritmos de inferência e a análise comparativa com o método NMF. No Capítulo 3 é apresentado o arcabouço de aprendizado de máquina usando a representação de documentos textuais via grafos bipartidos. Nesse capítulo, inicialmente, é apresentado a versão supervisionada, que, apesar de não ser uma contribuição direta dessa tese, teve participação do aluno e serviu como primeiro passo para a construção do arcabouço. Em seguida é apresentado a versão não-supervisionada, que é o arcabouço base do método aqui proposto. Os experimentos realizados mostraram a aplicabilidade desse algoritmo em problemas de redução de dimensionalidade e extração de tópicos. E por último, ainda no Capítulo 3, é apresentado a versão semissupervisionada, que é uma extensão da versão não supervisionada, mas com a presença de alguns documentos rotulados para auxiliar na classificação transdutiva. No Capítulo 4 é apresentado a versão *online* do algoritmo de propagação de rótulos em grafos bipartidos. O algoritmo é aplicado para o problema de extração de tópicos considerando documentos em fluxo. E por fim, no Capítulo 5, são apresentadas as conclusões, a descrição das limitações dos métodos propostos e os trabalhos futuros.

CAPÍTULO

2

MODELOS PROBABILÍTICOS DE TÓPICOS

Motivados pela necessidade de técnicas eficientes para extração de informações em texto, uma nova área de pesquisa surgiu em 2003, chamada de modelos probabilísticos de tópicos (*Probabilistic Topic Models*) (BLEI, 2011). O início dessa área se deu basicamente com a apresentação do LDA (*Latent Dirichlet Allocation*) (BLEI; NG; JORDAN, 2003) – LDA é o modelo base. Modelos probabilísticos de tópicos (BLEI; NG; JORDAN, 2003; GRIFFITHS; STEYVERS, 2004; STEYVERS; GRIFFITHS, 2007; HOFMANN, 1999) são um conjunto de algoritmos cujo objetivo é descobrir estruturas temáticas ocultas em grandes coleções de documentos. Inicialmente, esses modelos foram propostos para serem aplicados em documentos textuais, mas logo foram explorados em outros tipos de dados com atributos discretos, como imagens (LI; PERONA, 2005; SIVIC *et al.*, 2005; RUSSELL *et al.*, 2006; CAO; LI, 2007), grafos (HENDERSON; ELIASSI-RAD, 2009; BRONIATOWSKI; MAGEE, 2010; CHANG; BLEI, 2009; MEI *et al.*, 2008) e outros. Como a base do modelo e a descrição teórica é fundamentada em documentos e palavras, neste trabalho não é explorada a aplicação em outros tipos de dados. Porém, a transição para outros tipos de dados é direta uma vez que se entenda como o modelo é aplicado em texto.

A exploração de grandes volumes de dados é simplificada pelos modelos probabilísticos na descoberta dos *tópicos*. Os tópicos são estruturas com valor semântico e que, no contexto de mineração de texto, formam grupos de palavras que frequentemente ocorrem juntas. Esses grupos de palavras quando analisados, dão indícios a um tema ou assunto que ocorre em um subconjunto de documentos. Imagine vários discursos proferido por políticos e transcritos como documentos textuais. Se a cada dia vários documentos são gerados, ao longo dos anos essa coleção de documentos aumentará, inviabilizando o gerenciamento manual. Aplicando uma técnica como o LDA, é possível organizar e agrupar um subconjunto de discursos pelos seus respectivos temas.

Pesquisadores da área de recuperação de informação já propuseram várias técnicas

para reduzir o tamanho dos descritores de uma coleção de documentos. Entre as técnicas mais notáveis está o *Latent Semantic Indexing* (LSI) (DEERWESTER *et al.*, 1990; BERRY; DUMAIS; O'BRIEN, 1995). O LSI usa a decomposição em valores singulares de uma matriz documento-termo para identificar um subespaço linear que apresenta uma maior variação no espaço de características. Conhecendo a funcionalidade do LSI, é possível estender essa técnica para um modelo generativo probabilístico. Fazendo isso, Hofmann (1999) propôs o *probabilistic Latent Semantic Indexing* (pLSI). O pLSI é um modelo probabilístico com habilidade de recuperar aspectos de coleções de documentos. No modelo pLSI, cada palavra em um documento é amostrada de uma variável aleatória que representa um tópico. Assim, cada palavra em um documento é gerada por um tópico, e cada documento possui palavras geradas por diferentes tópicos. Isso faz com que um documento possua diferentes proporções de tópicos. Apesar do pLSI ser um modelo probabilístico, ele não é um modelo gerador de documentos completo pois não provê um modelo probabilístico no nível dos documentos. Ou seja, apesar das palavras serem geradas por variáveis aleatórias obedecendo uma distribuição multinomial, os documentos são apenas *bag-of-words*.

O modelo pLSI foi estendido para o modelo LDA (*Latent Dirichlet Allocation*). O LDA é um modelo bayesiano completo e se baseia na geração dos tópicos como distribuições de Dirichlet. Em comparação ao pLSI, o LDA descreve um modelo capaz de classificar documentos não conhecidos (documentos que não foram utilizados no treinamento), e utilizar informações *a priori*. Por essas características, o LDA tem influenciado uma grande quantidade de trabalhos e se tornado a base dos modernos modelos estatísticos de aprendizado de máquina, resultando em uma nova classe de modelos estatísticos chamados *Modelos Probabilísticos de Tópicos*.

O modelo LDA especifica um simples procedimento probabilístico no qual uma coleção de documentos pode ser gerada. Para criar um novo documento, inicialmente, escolhe-se uma distribuição de tópicos. Em seguida, para cada palavra nesse documento, escolhe-se um tópico aleatoriamente de acordo com essa distribuição. A palavra é amostrada de acordo com o tópico escolhido.

O processo inverso da geração de documentos é descobrir a distribuição de tópicos que geraram uma coleção de documentos. Esse processo está relacionada com a inferência do modelo probabilístico. Os algoritmos para inferência de modelos probabilísticos de tópicos são métodos estatísticos que analisam as palavras do texto original para descobrir os tópicos. Esses algoritmos são não supervisionados – os tópicos "emergem" da análise dos textos originais (STEYVERS; GRIFFITHS, 2007).

Neste capítulo é descrito o modelo base e referência para o desenvolvimento de modelos probabilístico de tópicos, o LDA. Inicialmente, é descrita a formulação do modelo e, em seguida, são apresentadas as principais técnicas de inferência probabilística desse modelo: método de amostragem de Gibbs e o método de inferência variacional. E por fim, é realizada uma análise comparativa do LDA com o NMF, onde são apontadas similaridades dessas duas técnicas.

2.1 *Latent Dirichlet Allocation* (LDA)

Quando se discute sobre modelos probabilísticos de tópicos, o que se encontra na literatura como estado da arte é o modelo LDA. O LDA é um modelo probabilístico generativo para coleções de dados discretos como corpus de documentos (BLEI; NG; JORDAN, 2003). Um modelo generativo é aquele que aleatoriamente gera os dados a partir das variáveis latentes. Assim, o LDA não é um algoritmo com descrições sequenciais de instruções para encontrar tópicos dada uma coleção de documentos. O LDA é um modelo probabilístico no qual é descrito como os documentos são gerados. Nesse modelo, as variáveis observáveis são os termos de cada documento e as variáveis não observáveis são as distribuições de tópicos. Os parâmetros das distribuições de tópicos, conhecidos como hiper-parâmetros, são dados *a priori* no modelo.

A distribuição utilizada para amostrar a distribuição de tópicos é a distribuição de Dirichlet. No processo generativo, o resultado da amostragem da Dirichlet é usado para alocar as palavras de diferentes tópicos e que preencherão os documentos. Assim, pode-se perceber o significado do nome *Latent Dirichlet Allocation*, que expressa a intenção do modelo de alocar os tópicos latentes que são distribuídos obedecendo a distribuição de Dirichlet.

A função de densidade da distribuição de Dirichlet, denotada como $Dir(z,\alpha)$, é a seguinte:

$$Dir(z,\alpha) = \frac{1}{B(\alpha)} \prod_{k=1}^{K} z_k^{\alpha_k - 1}, \tag{2.1}$$

onde $z = (z_1, \ldots, z_K)$ é uma variável K-dimensional, $0 \leq z_i \leq 1$ e $\sum_{i=1}^{K} z_i = 1$. Aqui, $\alpha = (\alpha_1, \ldots, \alpha_K)$ são os hiper-parâmetros da distribuição. A função $B(\alpha)$ é a função Beta, na qual pode ser expressa em termos da função gama Γ:

$$B(\alpha) = \frac{\prod_{k=1}^{K} \Gamma(\alpha_k)}{\Gamma\left(\sum_{k=1}^{K} \alpha_k\right)}. \tag{2.2}$$

A distribuição de Dirichlet tem algumas propriedades importantes (BISHOP, 2006), e por isso são comumente usadas em estatística Bayesiana [1].

O processo gerador do LDA é um processo imaginário e, inverso ao que é proposto em uma tarefa computacional de extração de informações, assume-se que os tópicos são especificados antes que qualquer dado seja gerado. Aqui, os tópicos são definidos como distribuições de probabilidade sobre um vocabulário fixo de palavras. Enquanto que os documentos, nada mais do que *bag-of-words*, surgem da escolha aleatória das palavras pertencentes a uma distribuições de tópicos.

Pode-se detalhar o processo gerador do modelo LDA. Para isso, deve-se assumir que um documento d_j é criado da seguinte forma:

[1] A distribuição de Dirichlet é a *priori* conjugada da distribuição multinomial

1. Crie as distribuições $\phi_k \sim Dir(\phi_k, \beta)$ para todo tópico k, como $0 \leq k \leq K$.
2. Crie uma distribuição $\theta_j \sim Dir(\theta, \alpha)$ para o documento d_j.
3. Para cada posição i das palavras no documento d_j,
   a) Escolha aleatoriamente um tópico $z_{j,i} \sim Multinomial(\theta_j)$.
   b) Escolha aleatoriamente uma palavra $w_{j,i}$ com probabilidade $p(w_{j,i}|\phi_{z_{j,i}})$.

No processo gerador, inicialmente, são utilizadas duas variáveis para representar as distribuições. A variável ϕ é uma variável n-dimensional, onde n é o número de palavras do vocabulário. A variável θ é uma variável K-dimensional, onde o valor de K é o número de tópicos. Essas variáveis descrevem distribuições de probabilidades, logo $\sum_j^n \phi_j = 1$, $\phi_i > 0$, e $\sum_i^K \theta_i = 1$, $\theta_i > 0$. Essas duas variáveis são geradas pela distribuição de Dirichlet (*Dir*) com seus respectivos hiper-parâmetros β e α.

Em seguida, com as distribuições ϕ e θ_j, é gerado o documento d_j. No modelo LDA um documento é simplesmente uma *bag-of-words*, com n_{d_j} termos em um documento d_j. Os termos de uma *bag-of-words* são palavras do vocabulário, e ocasionalmente podem ocorrer repetições de uma mesma palavra. Para cada posição i, das n_{d_j} posições de termos de uma *bag-of-words*, é escolhida uma palavra da distribuição de tópicos. Para isso, deve-se escolher um tópico k dos K tópicos existentes, e associar esse tópico a posição i do documento d_j. A variável $z_{j,i}$ armazenará o tópico escolhido. O tópico é escolhido obedecendo a distribuição θ_j, que informa a participação dos tópicos no documento d_j especificadamente. Em seguida, é escolhido da distribuição ϕ a palavra que irá preencher a posição i. A variável ϕ são K distribuições n-dimensionais, onde cada distribuição k, ϕ_k, corresponde a proporções de palavras que semanticamente descrevem o assunto do qual o tópico k trata. Assim, o termo $w_{j,i}$ deve ser escolhido do tópico $z_{j,i}$, obedecendo a distribuição de palavras $\phi_{z_{j,i}}$.

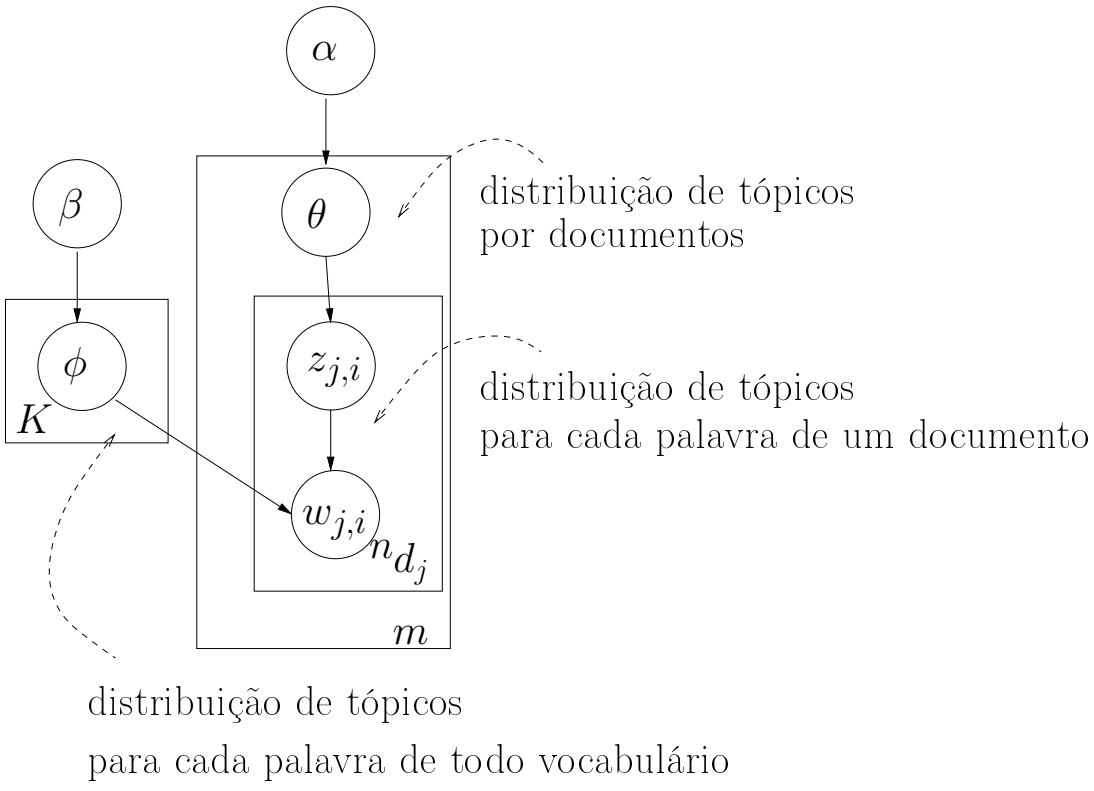
Uma característica importante do LDA é que cada documento possui sua própria distribuição de tópicos θ_j. Assim, um mesmo documento pode estar relacionado com vários tópicos com diferentes proporções de relevâncias. Pode-se perceber isso no modelo generativo pela escolha do tópico atribuído a variável $z_{j,i}$, onde ocasionalmente existirá a chance da escolha de diferentes tópicos segundo a distribuição θ_j.

Todo o processo generativo pode ser representado de forma gráfica por meio de uma rede Bayesiana. Essa rede é ilustrada na Figura 1. Nessa rede, cada vértice corresponde a uma variável e a aresta à relação de dependência. Na notação do modelo gráfico, em vez de ilustrar cada variável repetidas vezes, um retângulo é usado para agrupar variáveis em um subgrafo que se repete. O número de repetições é rotulado na parte inferior de cada retângulo. Os itens abaixo descrevem a notação utilizada na Figura 1.

- K – Número de tópicos

- n – Número de palavras do Vocabulário.
- m – Número de documentos.
- n_{d_j} – Número de palavras em um documento d_j, onde $1 \leq j \leq m$.
- θ – Distribuição de tópicos por documentos.
- ϕ – Distribuição dos tópicos sobre as palavras de todo o vocabulário.
- θ_j – Vetor com a proporção dos tópicos para o documento d_j, onde $1 \leq j \leq m$.
- ϕ_k – Vetor com a proporção das palavras do vocabulário para o tópico k, onde $1 \leq k \leq K$.
- α – Priore da distribuição de Dirichlet, relacionada a distribuição documento-termo.
- β – Priore da distribuição de Dirichlet, relacionada a distribuição tópico-palavra.
- w_i – i-ésima palavra do vocabulário, onde $1 \leq i \leq n$.
- $w_{j,i}$ – palavra w_i observada no documento d_j, onde $1 \leq j \leq m$ e $1 \leq i \leq n$.
- $z_{j,i}$ – Distribuição de tópicos associado a palavra $w_{j,i}$ no documento d_j, onde $1 \leq j \leq m$ e $1 \leq i \leq n$.

Figura 1 – Modelo Gráfico do LDA.



Fonte: Adaptada de Blei, Ng e Jordan (2003).

O modelo Bayesiano do LDA é um modelo hierárquico com três níveis (veja a Figura 1). O primeiro nível representa a distribuição de tópicos em toda a coleção de documentos. No segundo nível, tem-se a distribuição dos tópicos para cada documento. E o último nível, repete-se a distribuição dos tópicos internamente para as palavras em um documento. Com o último nível, tona-se possível representar um documento como uma mistura de tópicos.

Utilizando a Figura 1 para esclarecer a representação do modelo, percebe-se que no nível de toda a coleção de documentos estão os hiper-parâmetros α e β. De forma simples, sem o formalismo matemático, pode-se dar uma interpretação para esses parâmetros. Um alto valor de α significa que cada documento provavelmente conterá uma maior mistura de tópicos. Um valor baixo para α indica maior probabilidade dos documentos conterem mistura de poucos tópicos, fazendo uma maior concentração em poucos tópicos. Da mesma forma, um valor alto para β significa que cada tópico terá alta probabilidade de conter misturas de várias palavras. Enquanto que um valor baixo para β indica que o tópico será formado por poucas palavras.

No nível de todo o vocabulário de palavras está a variável ϕ_k, que é amostrada para cada tópico k. Cada vetor ϕ_k forma uma matriz de tamanho $n \times K$, onde cada linha corresponde às palavras do vocabulário e as colunas aos tópicos. O valor de $\phi_{k,i}$ é a proporção do tópico k para uma palavra w_i.

No nível dos documentos, está a variável θ_j, que é amostrada para cada documento da coleção. Pode-se interpretar essa distribuição de documentos por tópicos como uma matriz θ de tamanho $m \times K$, onde cada linha são os documentos e as colunas os tópicos. Uma linha dessa matriz, referenciada como θ_j, corresponde a proporção de tópicos para um documento d_j da coleção.

No nível das palavras estão as variáveis $z_{j,i}$ e $w_{j,i}$, essas variáveis são amostradas para cada palavra w_i em cada documento d_j. A variável $z_{j,i}$ é a atribuição de um tópico k ($1 \leq k \leq K$) para uma palavra w_i de um documento d_j.

Aqui, para dar um maior entendimento e também já conhecendo a descrição gráfica do LDA (Figura 1), vamos reescrever o processo generativo, só que dessa vez apresentando os passos para a geração de toda a coleção de documentos. Assim, tem-se o processo generativo do LDA com os seguintes passos:

1. Amostre K multinomiais $\phi_k \sim Dir(\phi_k, \beta)$, um para cada tópico k.
2. Amostre m multinomiais $\theta_j \sim Dir(\theta_j, \alpha)$, um para cada documento d_j.
3. Para cada documento d_j da coleção
    a) Para cada palavra w_i do documento d_j:
        i. Associe um tópico para $z_{j,i}$ amostrado da distribuição de Dirichlet θ_j.
        ii. Amostre uma palavra w_i da distribuição $\phi_{z_{j,i}}$.

Com base no processo generativo, e observando a relação de dependência existente entre as variáveis do modelo, é possível descrever a probabilidade de todas as variáveis latentes do modelo dado as informações *a priori* (BLEI, 2011). Transcrevendo essas probabilidades, tem-se a seguinte distribuição conjunta:

$$p(z,w,\phi,\theta|\alpha,\beta) = \prod_{k=1}^{K} p(\phi_k|\beta) \prod_{j=1}^{M} p(\theta_j|\alpha) \left(\prod_{i=1}^{V} p(z_{j,i}|\vec{\theta}_j) p(w_{i,j}|z_{i,j},\phi_{z_{j,i}}) \right). \tag{2.3}$$

Essa equação determina uma distribuição de probabilidade com um complexo número de dependências. Por exemplo, a atribuição de tópico $z_{j,i}$ depende da distribuição dos tópicos por documento θ_j, e a palavra observada $w_{j,i}$ depende da atribuição do tópico $z_{j,i}$ e da proporção dessa palavra na distribuição $\phi_{z_{j,i}}$.

Levando em consideração as variáveis observadas e não observadas, almeja-se descobrir as atribuições de tópicos para os documentos e as distribuições de documentos por tópicos e tópicos por termos. Ou seja, o grande problema computacional do LDA é inferir $p(z,\phi,\theta,|w,\alpha,\beta)$, onde w são todas as palavras observadas na coleção de documentos.

Pelo teorema de Bayes, pode-se formular a probabilidade de $p(z,\phi,\theta,|w,\alpha,\beta)$ como o cálculo da *a posteriori* do LDA. Dessa forma, tem-se

$$p(z,\phi,\theta|w,\alpha,\beta) = \frac{p(z,w,\phi,\theta|\alpha,\beta)}{p(w)}. \tag{2.4}$$

O numerador é a distribuição conjunta (Equação 2.3) do modelo e o denominador é a probabilidade marginal dos dados observados.

Logo, o problema computacional central pode ser resolvido inferindo a probabilidade *a posteriori* de todo o modelo, descrito na Equação 2.4. Isso pode ser pensado como o inverso do processo generativo. Teoricamente, esse cálculo de inferência pode ser feito pela soma da distribuição conjunta de todos os valores possíveis atribuídos as variáveis não observadas (todas as palavras da coleção). Entretanto, o número de atribuições possíveis é exponencialmente grande, fazendo esse cálculo intratável computacionalmente (BLEI, 2011). Apesar disso, existem vários métodos para aproximar a distribuição *a posteriori*. Entre os métodos mais utilizados na literatura para inferência do modelo LDA estão o *Gibbs Sampling* (Amostrador de Gibbs) (GRIFFITHS; STEYVERS, 2004) e *Variational Inference* (Inferência Variacional) (BLEI; NG; JORDAN, 2003).

Nas próximas seções serão discutidos os métodos de inferência. Inicialmente, será descrito o Amostrador de Gibbs e como aplicá-lo no caso do LDA. Em seguida será descrito o método de inferência variacional e sua aplicação no LDA.

2.1.1 Inferência do LDA via Amostrador de Gibbs

Entre os métodos de inferência, o Amostrador de Gibbs é o mais popular principalmente pela facilidade de implementação e sua aplicação em diversos problemas (GEMAN; GEMAN,

1984). O amostrador de Gibbs é um caso especial da simulação de Monte Carlo em Cadeia de Markov. Métodos de Monte Carlo em Cadeia de Markov podem emular distribuições de probabilidades com alta dimensionalidade por meio do comportamento estacionário da cadeia de Markov.

O processo realizado pelo amostrador de Gibbs se baseia na amostragem de cada dimensão alternadamente, uma de cada vez, condicionada ao valor de todas as outras dimensões. Suponha que há uma variável K-dimensional não observada, $z = \{z_1, \ldots, z_K\}$, onde z_i corresponde ao valor da i-ésima dimensão do vetor z e $z_{-i} = \{z_1, z_2, \ldots, z_{i-1}, z_{i+1}, \ldots, z_K\}$, suponha também a evidência dada pela variável observada, w. Nesse exemplo, a distribuição a ser inferida é $p(z|w)$. Assim, em vez de fazer amostras em toda a distribuição z para inferir $p(z|w)$, o amostrador de Gibbs faz escolhas separadas para cada dimensão i de z, onde a amostragem de z_i depende das outras dimensões em z_{-i} amostradas até o momento.

Com a intensão de descobrir $p(z|w)$, pode-se apresentar um algoritmo simples descrevendo o processo realizado pelo amostrado de Gibbs. Esse algoritmo está sumarizado no Algoritmo 1.

Algoritmo 1: Amostrador de Gibbs

Entrada : número de iterações T, variável não observada z, variável observada w.

início
  $z^{(0)} \leftarrow \left\{z_1^{(0)}, \ldots, z_K^{(0)}\right\}$;
  para $t = 1$ **to** T **faça**
    para $i = 1$ **to** K **faça**
      $z_i^{(t+1)} \sim P(z_i|z_1^{(t+1)}, \ldots, z_{i-1}^{(t+1)}, z_{i+1}^{(t)}, \ldots, z_K^{(t)}, w)$;
  retorne estimativa de $p(z|w)$;

No Algoritmo 1, a variável não observada no instante inicial, $z^{(0)}$, pode ser iniciada aleatoriamente. Em seguida, é realizado o processo de iteração onde é amostrado cada dimensão de z em relação a todas as outras dimensões amostradas até então. Esse processo é realizado T vezes, tendo no final desse processo a estimativa de $p(z|w)$.

O amostrador de Gibbs gera uma cadeia de Markov de amostras. Com base nisso, é possível demonstrar convergência do algoritmo. Aqui, não serão apresentadas as demonstrações de que o algoritmo alcança um estado estacionário de transições na cadeia de Markov, mas essas demonstrações podem ser encontradas no trabalho de Russell e Norvig (2003).

2.1.1.1 Integrando o LDA para o Amostrador de Gibbs

No caso do LDA, o amostrador de Gibbs deve fazer amostragem de três variáveis ocultas, z, θ e ϕ. Para simplificar, o método de Gibbs aplicado no LDA é colapsado de forma a amostrar apenas a variável z, e a partir de z encontrar os valores das variáveis θ e ϕ.

O processo de amostragem é realizado por meio de estatísticas obtidas pela contagem das atribuições de palavras para os tópicos e tópicos para documentos feitas após amostragem. Para manter essas estatísticas, serão introduzidas as seguintes variáveis contadoras:

- $c_{j,i,k}$ corresponde ao número de vezes que um termo w_i é atribuída ao tópico k no documento d_j,
- $c_{j,*,k}$ é o número de termos no documento d_j atribuídas ao tópico k,
- $c_{*,i,k}$ é o número de vezes que o termo w_i é atribuída ao tópico k em todos os documento,
- $c_{*,*,k}$ é o número de termos atribuídos ao tópico k considerando toda a coleção de documentos.

O amostrador colapsado de Gibbs computa a probabilidade do tópico $z_{a,b}$ ser atribuído para a posição do termo w_b no documento d_a, dada as atribuições realizadas anteriormente para as outras posições no documento d_b, denotada como $z_{-(a,b)}$,

$$p(z_{a,b}|z_{-(a,b)}, w, \alpha, \beta) \tag{2.5}$$

Pela definição de probabilidade condicional, tem-se

$$= \frac{p(z_{a,b}, z_{-(a,b)}, w|\alpha, \beta)}{p(z_{-(a,b)}, w|\alpha, \beta)}. \tag{2.6}$$

Removendo o denominador, que não depende de $z_{a,b}$, e unindo $z_{a,b}$ e $z_{-(a,b)}$ em z,

$$\propto p(z_{a,b}, z_{-(a,b)}, w|\alpha, \beta) = p(z, w|\alpha, \beta). \tag{2.7}$$

Usando a regra da probabilidade total, integrando a distribuição de documentos por tópicos θ, e a distribuição de tópicos por palavras ϕ,

$$= \int\int p(\theta, \phi, z, w|\alpha, \beta) d\theta d\phi. \tag{2.8}$$

Expandindo a integral dada pela propriedade da distribuição conjunta do LDA (Equação 2.3),

$$= \int p(z|\theta) p(\theta|\alpha) d\theta \times \int p(w|\phi, z) p(\phi|\beta) d\phi, \tag{2.9}$$

e então expandindo os termos,

$$= \int \prod_{j=1}^{m} p(z_j|\theta_j) p(\theta_j|\alpha) d\theta \times \int \prod_{k=1}^{K} p(\phi_k|\beta) \prod_{j=1}^{m} \prod_{i=1}^{n_{d_j}} p(w_{j,i}|\phi_{z_{j,i}}) d\phi \tag{2.10}$$

$$= \prod_{j=1}^{m} \int p(z_j|\theta_j) p(\theta_j|\alpha) d\theta_j \times \prod_{k=1}^{K} \int p(\phi_k|\beta) \prod_{j=1}^{m} \prod_{i=1}^{n_{d_j}} p(w_{j,i}|\phi_{z_{j,i}}) d_{\phi_k} \tag{2.11}$$

Desde que essas probabilidades obedeçam a distribuição de Dirichlet, elas podem ser substituídas pela fórmula usual (Equação 2.1),

$$
\begin{aligned}
&= \prod_{j=1}^{m} \int \frac{\Gamma(\sum_{k=1}^{K} \alpha_k)}{\prod_{k=1}^{K} \Gamma(\alpha_k)} \prod_{k=1}^{K} \theta_{j,k}^{\alpha_k - 1} \prod_{i=1}^{n_{d_j}} \theta_{j,z_{j,i}} d\theta_j \\
&\quad \times \prod_{k=1}^{K} \int \frac{\Gamma(\sum_{l=1}^{n} \beta_l)}{\prod_{l=1}^{n} \Gamma(\beta_l)} \prod_{l=1}^{n} \phi_{l,k}^{\beta_l - 1} \prod_{j=1}^{m} \prod_{i=1}^{n_{d_j}} \phi_{z_{j,i},w_{j,i}} d\phi_k
\end{aligned} \tag{2.12}
$$

Lembrando que $x^a x^b = x^{a+b}$, pode-se substituir o produto interno das distribuições θ e ϕ,

$$
\begin{aligned}
&= \prod_{j=1}^{m} \int \frac{\Gamma(\sum_{k=1}^{K} \alpha_k)}{\prod_{k=1}^{K} \Gamma(\alpha_k)} \prod_{k=1}^{K} \theta_{j,k}^{\alpha_k - 1} \prod_{k=1}^{K} \theta_{j,k}^{c_{j,*,k}} d\theta_j \\
&\quad \times \prod_{k=1}^{K} \int \frac{\Gamma(\sum_{l=1}^{n} \beta_l)}{\prod_{l=1}^{n} \Gamma(\beta_l)} \prod_{l=1}^{n} \phi_{l,k}^{\beta_l - 1} \prod_{l=1}^{n} \phi_{k,l}^{c_{*,l,k}} d\phi_k
\end{aligned} \tag{2.13}
$$

$$
= \prod_{j=1}^{m} \int \frac{\Gamma(\sum_{k=1}^{K} \alpha_k)}{\prod_{k=1}^{K} \Gamma(\alpha_k)} \prod_{k=1}^{K} \theta_{j,k}^{\alpha_k + c_{j,*,k} - 1} d\theta_j \times \prod_{k=1}^{K} \int \frac{\Gamma(\sum_{l=1}^{n} \beta_l)}{\prod_{l=1}^{n} \Gamma(\beta_l)} \prod_{l=1}^{n} \phi_{l,k}^{\beta_l c_{*,l,k} - 1} d\phi_k. \tag{2.14}
$$

Em seguida, multiplicar por uma constante formada por duas frações inversas e com valor igual a um,

$$
\begin{aligned}
&= \prod_{j=1}^{m} \frac{\Gamma(\sum_{k=1}^{K} \alpha_k)}{\prod_{k=1}^{K} \Gamma(\alpha_k)} \frac{\prod_{k=1}^{K} \Gamma(c_{j,*,k} + \alpha_k)}{\Gamma(\sum_{k=1}^{K} c_{j,*,k} + \alpha_k)} \int \frac{\Gamma(\sum_{k=1}^{K} c_{j,*,k} + \alpha_k)}{\prod_{k=1}^{K} \Gamma(c_{j,*,k} + \alpha_k)} \prod_{k=1}^{K} \theta_{j,k}^{\alpha_k + c_{j,*,k} - 1} d\theta_j \\
&\quad \times \prod_{k=1}^{K} \frac{\Gamma(\sum_{l=1}^{n} \beta_l)}{\sum_{l=1}^{n} \Gamma(\beta_l)} \frac{\prod_{l=1}^{n} \Gamma(c_{*,l,k} + \beta_l)}{\Gamma(\sum_{l=1}^{n} c_{*,l,k} + \beta_l)} \int \frac{\Gamma(\sum_{l=1}^{n} c_{*,l,k} + \beta_l)}{\prod_{l=1}^{n} \Gamma(c_{*,l,k} + \beta_l)} \prod_{l=1}^{n} \phi_{l,k}^{\beta_l c_{*,l,k} - 1} d\phi_k.
\end{aligned} \tag{2.15}
$$

O valor formado pelas integrais correspondem a densidade de uma distribuição de Dirichlet, consequentemente elas tem valor igual a 1 e podem ser removidas,

$$
= \prod_{j=1}^{m} \frac{\Gamma(\sum_{k=1}^{K} \alpha_k)}{\prod_{k=1}^{K} \Gamma(\alpha_k)} \frac{\prod_{k=1}^{K} \Gamma(c_{j,*,k} + \alpha_k)}{\Gamma(\sum_{k=1}^{K} c_{j,*,k} + \alpha_k)} \times \prod_{k=1}^{K} \frac{\Gamma(\sum_{l=1}^{n} \beta_l)}{\sum_{l=1}^{n} \Gamma(\beta_l)} \frac{\prod_{l=1}^{n} \Gamma(c_{*,l,k} + \beta_l)}{\Gamma(\sum_{l=1}^{n} c_{*,l,k} + \beta_l)}. \tag{2.16}
$$

Removendo as funções Γ que dependem apenas dos hiper-parâmetros (constantes) α e β, terá a seguinte proporcionalidade

$$
\propto \prod_{j=1}^{M} \frac{\prod_{k=1}^{K} \Gamma(c_{j,*,k} + \alpha_k)}{\Gamma(\sum_{k=1}^{K} c_{j,*,k} + \alpha_k)} \times \prod_{k=1}^{K} \frac{\prod_{l=1}^{V} \Gamma(c_{*,l,k} + \beta_l)}{\Gamma(\sum_{l=1}^{V} c_{*,l,k} + \beta_l)}. \tag{2.17}
$$

Em seguida, será separado esse produto evidenciando a posição b no documento d_a,

$$
\begin{aligned}
&= \prod_{j \neq a}^{m} \frac{\prod_{k=1}^{K} \Gamma(c_{j,*,k} + \alpha_k)}{\Gamma(\sum_{k=1}^{K} c_{j,*,k} + \alpha_k)} \times \frac{\prod_{k=1}^{K} \Gamma(c_{a,*,k} + \alpha_k)}{\Gamma(\sum_{k=1}^{K} c_{a,*,k} + \alpha_k)} \\
&\qquad \times \prod_{k=1}^{K} \frac{\prod_{l \neq w_{a,b}}^{n} \Gamma(c_{*,l,k} + \beta_l) \times \Gamma(c_{*,w_{a,b},k})}{\Gamma(\sum_{l=1}^{n} c_{*,l,k} + \beta_l)}.
\end{aligned} \tag{2.18}
$$

Removendo os termos da equação que não dependam da posição (a,b),

$$\propto \frac{\prod_{k=1}^{K}\Gamma(c_{a,*,k}+\alpha_k)}{\Gamma(\sum_{k=1}^{K}c_{a,*,k}+\alpha_k)} \times \prod_{k=1}^{K}\frac{\Gamma(c_{*,w_{a,b},k}+\beta_{w_{a,b}})}{\Gamma(\sum_{l=1}^{n}c_{*,l,k}+\beta_l)}. \tag{2.19}$$

Seja $c^{-(a,b)}$ o valor das contagens feitas como o contador c, mas desconsiderando a contagem da posições (a,b). Note que para contagens que não inclui o documento a, o valor de $c^{(a,b)} = c$, e para as contagens na posição b do documento a, é incrementado 1 mais o valor já contado em $c^{a,b}$,

$$\begin{aligned}\propto &\frac{\prod_{k\neq z_{a,b}}^{K}\Gamma(c_{a,*,k}^{-(a,b)}+\alpha_k)\times\Gamma(c_{a,*,z_{a,b}}^{-(a,b)}+\alpha_{z_{a,b}}+1)}{\Gamma(1+\sum_{k=1}^{K}c_{a,*,k}^{-(a,b)}+\alpha_k)}\\ &\times\prod_{k\neq z_{a,b}}^{K}\frac{\Gamma(c_{*,w_{a,b},k}^{-(a,b)}+\beta_{w_{a,b}})}{\Gamma(\sum_{l=1}^{n}c_{*,l,k}+\beta_l)}\times\frac{\Gamma(c_{*,w_{a,b},z_{a,b}}^{-(a,b)}+\beta_{w_{a,b}}+1)}{\Gamma(1+\sum_{l=1}^{n}c_{*,l,z_{a,b}}^{-(a,b)}+\beta_l)}.\end{aligned} \tag{2.20}$$

Desde que x é inteiro tem-se que $\Gamma(x+1) = x\times\Gamma(x)$, assim expande-se os termos dependentes da posição (a,b),

$$\begin{aligned}=&\frac{\prod_{k\neq z_{a,b}}^{K}\Gamma(c_{a,*,k}^{-(a,b)}+\alpha_k)\times\Gamma(c_{a,*,z_{a,b}}^{-(a,b)}+\alpha_{z_{a,b}})\times(c_{a,*,z_{a,b}}^{-(a,b)}+\alpha_{z_{a,b}})}{\Gamma(1+\sum_{k=1}^{K}c_{a,*,k}^{-(a,b)}+\alpha_k)}\\ &\times\prod_{k\neq z_{a,b}}^{K}\frac{\Gamma(c_{*,w_{a,b},k}^{-(a,b)}+\beta_{w_{a,b}})}{\Gamma(\sum_{l=1}^{n}c_{*,l,k}+\beta_l)}\times\frac{\Gamma(c_{*,w_{a,b},z_{a,b}}^{-(a,b)}+\beta_{w_{a,b}})\times(c_{*,w_{a,b},z_{a,b}}^{-(a,b)}+\beta_{w_{a,b}})}{\Gamma(\sum_{l=1}^{n}c_{*,l,z_{a,b}}^{-(a,b)}+\beta_l)\times\sum_{l=1}^{n}c_{*,l,z_{a,b}}^{-(a,b)}+\beta_l}.\end{aligned} \tag{2.21}$$

Unindo os produtos de $k\neq z_{a,b}$ e $k = z_{a,b}$,

$$\begin{aligned}=&\frac{\prod_{k=1}^{K}\Gamma(c_{a,*,k}^{-(a,b)}+\alpha_k)\times(c_{a,*,z_{a,b}}^{-(a,b)}+\alpha_{z_{a,b}})}{\Gamma(1+\sum_{k=1}^{K}c_{a,*,k}^{-(a,b)}+\alpha_k)}\\ &\times\prod_{k=1}^{K}\frac{\Gamma(c_{*,w_{a,b},k}^{-(a,b)}+\beta_{w_{a,b}})}{\Gamma(\sum_{l=1}^{n}c_{*,l,k}+\beta_l)}\times\frac{(c_{*,w_{a,b},z_{a,b}}^{-(a,b)}+\beta_{w_{a,b}})}{\sum_{l=1}^{n}c_{*,l,z_{a,b}}^{-(a,b)}+\beta_l}.\end{aligned} \tag{2.22}$$

Os produtórios indexados por tópicos resultam em valores constantes, logo podem ser removidos,

$$\propto \frac{(c_{a,*,z_{a,b}}^{-(a,b)}+\alpha_{z_{a,b}})\times(c_{*,w_{a,b},z_{a,b}}^{-(a,b)}+\beta_{w_{a,b}})}{\sum_{l=1}^{V}c_{*,l,z_{a,b}}^{-(a,b)}+\beta_l} \tag{2.23}$$

E por fim, pode-se simplificar o denominador de forma que $\sum_{l=1}^{K}c_{*,l,z_{a,b}}^{-(a,b)} = c_{*,*,k}^{-(a,b)}$,

$$\propto \frac{(c_{a,*,z_{a,b}}^{-(a,b)}+\alpha_{z_{a,b}})\times(c_{*,w_{a,b},z_{a,b}}^{-(a,b)}+\beta_{w_{a,b}})}{c_{*,*,k}^{-(a,b)}+\sum_{l=1}^{n}\beta_l}. \tag{2.24}$$

Assim, chega-se na equação de amostragem via algoritmo de Gibbs para o modelo LDA:

$$p(z_{a,b}=k|z_{-(a,b)},w,\alpha,\beta)\propto\frac{(c_{a,*,z_{a,b}}^{-(a,b)}+\alpha_{z_{a,b}})\times(c_{*,w_{a,b},z_{a,b}}^{-(a,b)}+\beta_{w_{a,b}})}{c_{*,*,k}^{-(a,b)}+\sum_{l=1}^{n}\beta_l}. \tag{2.25}$$

Uma vez estimado z, é possível encontrar os valores para as distribuições θ e ϕ, as respectivas distribuição de documentos por tópicos e tópicos por palavras. Elas podem ser obtidas pelo cálculo

$$\theta_{j,k} = \left(\frac{c_{j,*,k} + \alpha_k}{n_{d_j} + m\alpha_k} \right) \tag{2.26}$$

$$\phi_{k,i} = \left(\frac{c_{*,w_i,k} + \beta_k}{(c_{*,*,k} + n\beta_k)} \right). \tag{2.27}$$

Esses valores correspondem a estatísticas obtidas durante a amostragem e são normalizados levando-se em conta a relação de proporcionalidade.

Com base nas proporções e equações 2.25, 2.26 e 2.27 é apresentada na próxima seção o algoritmo completo para a mostragem.

2.1.1.2 *Algoritmo de inferência via amostrador de Gibbs*

O procedimento de amostragem do algoritmo de Gibbs para o LDA pode ser executado usando a Equação 2.25 para a amostragem. O algoritmo é descrito no Algoritmo 3. Ele utiliza apenas quatro grandes estrutura de dados: o contador $c_{j,*,k}$ que é uma matriz de dimensão $m \times K$, onde mantêm o contador do documento d_j para um tópico k; o contador $c_{*,i,k}$ que é uma matriz de dimensão $k \times n$, onde mantêm o contador do tópico k atribuído a uma palavra w_i; o contador $c_{*,*,k}$ que é um vetor K-dimensional, onde mantêm a quantidade de atribuições a um tópico k; e as atribuições z, que é uma matriz $m \times n$ onde mantêm a atribuição de um documento d_j para cada termo w_i. Com isso, o algoritmo executará em dois procedimentos, inicialização (Algoritmo 2) e amostragem (Algoritmo 3).

Algoritmo 2: Inicialização do Amostrador de Gibbs para LDA

Entrada : número de tópicos K, coleção de documentos

início
- todas as variáveis contadoras $c_{j,*,k}$, $c_{*,i,k}$, $c_{*,*,k}$ são iniciados com zero ;
- **para** *documento* d_j *com* $j \in [1,m]$ **faça**
  - **para** *palavra* w_i *com* $i \in [1,n_j]$ *no documento* d_j **faça**
    - amostre o índice do tópico $z_{j,i} \leftarrow Mult(\frac{1}{k})$ para palavra $w_{j,i}$;
    - incremente o contador de documento por tópico: $c_{j,*,z_{j,i}} + +$;
    - incremente o contador de tópico por palavra: $c_{*,i,z_{j,i}} + +$;
    - incremente a soma de palavras do tópico amostrado: $c_{*,*,z_{j,i}} + +$;

A amostragem envolve o cálculo das estatísticas obtidas pelos contadores. Note que na inicialização os contadores são incrementados com tópicos atribuídos aleatoriamente. Em seguida, para atribuir um tópicos para a variável $z_{j,i}$, é necessário decrementar a contagem já atribuída ao termo na posição i do documento d_j, fazer uma nova amostragem (via Equação 2.25), e atualizar os contadores. O novo tópico é atribuído para a variável $z_{j,i}$ e, em seguida, são

utilizadas para encontras as distribuições θ e ϕ (via equações 2.26 e 2.27). Veja o Algoritmo 3 para o procedimento completo de amostragem de Gibbs para o LDA.

Algoritmo 3: Amostrador de Gibbs para o LDA

Entrada: número de tópicos K, coleção de documentos, hiper-parâmetros α e β, número de iterações T

início
  Inicializa os contadores – Algoritmo 2 ;
  enquanto *não terminar o número de iterações* T **faça**
    para *documento* d_j *com* $j \in [1,m]$ **faça**
      para *palavra* w_i *com* $i \in [1,n_j]$ *no documento* d_j **faça**
        $c_{j,*,z_{j,i}} --$, $c_{*,i,z_{j,i}} --$, $c_{*,*,z_{j,i}} --$;
        para *tópico* k *com* $k \in [1,K]$ **faça**
          $p(z_{j,i} = k|\cdot) = \frac{(c^{-(a,b)}_{a,*,z_{a,b}} + \alpha_{z_{a,b}}) \times (c^{-(a,b)}_{*,w_{a,b},z_{a,b}} + \beta_{w_{a,b}})}{c^{-(a,b)}_{*,*,k} + \sum_{l=1}^{n} \beta_l}$
        topico = amostre de $p(z|\cdot)$;
        $z_{j,i}$ = *topico* ;
        $c_{j,*,z_{j,i}} ++$, $c_{*,i,z_{j,i}} ++$, $c_{*,*,z_{j,i}} ++$;
  Atualize o conjunto de parâmetros θ e ϕ de acordo com as equações (2.26) e (2.27) ;

A convergência desse algoritmo é alcançada quando não existe alterações na distribuição conjunta do modelo (Equação 2.3). Em termos práticos, é definido um número T de iterações.

2.1.2 Inferência do LDA via método variacional

Nessa seção é apresentado o algoritmo de inferência variacional para o LDA. Esse algoritmo tem uma abordagem diferente do amostrador de Gibbs. O método variacional não se baseia em amostragem, em vez disso, ele transforma o processo de inferência da distribuição *a posteriori* do LDA em um problema de otimização.

Antes de especificar o método variacional para o LDA, é apresentada a noção básica do método. Para isso, é utilizada uma notação genérica, considerando um modelo onde as variáveis latentes não observadas é denotado por z, e o conjunto de todas as variáveis observáveis e o conhecimento *a priori* é denotado por w. A probabilidade conjunta desse modelo é $p(w,z)$. Aqui, o objetivo é encontrar uma solução para a distribuição *a posteriori* $p(z|w)$, ou seja, descobrir o conhecimento oculto representado pela variável não observada z dada a observação em w. Nessa seção, é introduzido o método variacional (*Variational Bayes Inference*) para inferência da distribuição *a posteriori* $p(z|w)$.

Supõe-se que o cálculo da distribuição $p(z|w)$ seja intratável. Assim, no método variacional, uma solução aproximada para $p(z|w)$ é alcançada por meio de uma outra distribuição $q(z)$. Essa distribuição $q(z)$ é definida por uma família de distribuição mais "fácil" de calcular do que $p(z|w)$. Dessa forma, inicialmente, é necessário definir uma família de distribuições que se

aproxime da *a posteriori*. A relação de proximidade entre a distribuição *a posteriori* $p(z|w)$ e a distribuição variacional $q(z)$ é medida pela divergência de Kullback-Leibler (KL) (KULLBACK; LEIBLER, 1951) (veja a Definição 1). Quanto menor a divergência KL entre as distribuições q e a distribuição real p melhor será a aproximação. Logo, o método variacional transforma o problema de inferência em um problema de otimização onde o objetivo é minimizar $KL(q||p)$ (BISHOP, 2006).

Definição 1 (Divergência KL)**.** Para duas distribuições contínuas p e q, a divergência de Kullback-Leibler, ou divergência KL, é calculada da seguinte forma:

$$KL(p||q) = \int_x p(x) log \frac{p(x)}{q(x)} d_x, \tag{2.28}$$

onde x é uma variável aleatória contínua.

Assim como no cálculo da posteriori, minimizar diretamente a divergência $KL(q||p)$ é intratável. Porém, é possível limitar a distribuição marginal do modelo em função apenas da distribuição variacional. Existe uma relação entre as distribuições real p e a variacional q com o logaritmo da probabilidade marginal do modelo. Para alcançar essa relação, será estendido o logaritmo da probabilidade marginal do modelo da seguinte forma:

$$\begin{aligned} \log p(x) &= \log \int_z p(x,z) dz \\ &= \log \int_z p(x,z) \frac{q(z)}{q(z)} dz \\ &= \log \left(E_q \left[\frac{p(x,z)}{q(z)} \right] \right) \\ &\geq E_q \left[p(x,z) \right] - E_q \left[q(z) \right]. \end{aligned} \tag{2.29}$$

O último passo utiliza a desigualdade de Jensen (NEEDHAM, 1993) para encontrar um limite inferior para o logaritmo da probabilidade marginal do modelo.

Pela Desigualdade (2.29), tem se que $\log p(x)$ é no mínimo $E_q \left[p(x,z) \right] - E_q \left[q(z) \right]$. Essa expressão, chamada de *Evidence Lower Bound* (ELBO) $\mathscr{L}$, será denotada como

$$\mathscr{L} \triangleq E_q \left[p(x,z) \right] - E_q \left[q(z) \right]. \tag{2.30}$$

Agora, qual a relação do ELBO com a minimização da divergência KL? Para encontrar essa relação, basta calcular a diferença

$$\begin{aligned} & E_q \left[p(x,z) \right] - E_q \left[q(z) \right] - \log p(x) \\ &= \int_z q(z) p(x,z) - \int_z q(z) q(z) - \log p(x) \\ &= \int_z q(z) \log p(x,z) - \int_z q(z) \log q(z) - \int_z q(z) \log p(x) \\ &= \int_z q(z) \log \frac{p(x,z)}{p(x)} - \int_z q(z) \log q(z) \end{aligned}$$

$$
\begin{aligned}
&= \int_z q(z) \log p(Z|X) - \int_z q(z) \log q(z) \\
&= \int_z q(z) \log \frac{p(z|x)}{q(z)} \\
&= -\left(\int_z q(z) \log \frac{q(z)}{p(z|x)} \right) \\
&= -KL(q||p)
\end{aligned} \tag{2.31}
$$

Relembrando que o principal objetivo é minimizar a divergência de Kullback-Leibler, de forma que a distribuição $q(z)$ se aproxime de $p(z|x)$. Com isso, pela Desigualdade (2.29) tem-se que $\log p(x)$ é no máximo $E_q[p(x,z)] - E_q[q(z)]$. Já na Equação (2.31) tem-se a expressão para a divergência KL. Por essas duas equações, nota-se que para minimizar a divergência de Kullback-Leibler é preciso maximizar $E_q[p(x,z)] - E_q[q(z)]$. Assim, foi transformado o problema de inferência em um problema de otimização onde o objetivo é maximizar a Desigualdade 2.29. A Desigualdade 2.29 é o ponto principal no método de inferência variacional, pois todo o processo computacional desse método se baseia em otimizar o ELBO, aqui denotado como $\mathscr{L}$.

2.1.3 Integrando o LDA para o método de inferência variacional

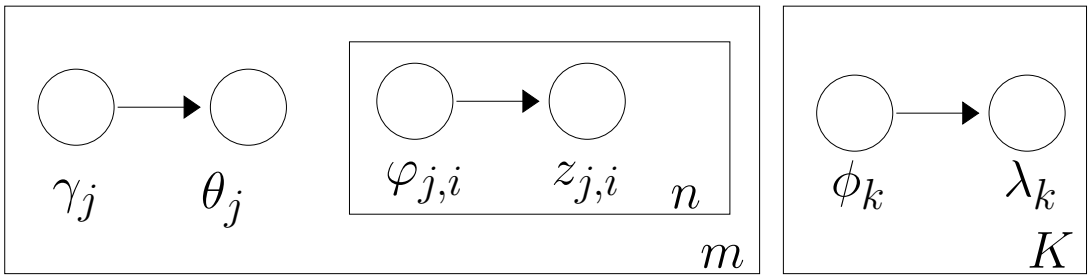
Para utilizar o método de inferência variacional no LDA, inicialmente, é necessário definir uma distribuição q, tal que essa distribuição se aproxime da distribuição *a posteriori* original do LDA. Veja a Figura 1 para a descrição gráfica do LDA e a Equação 2.4 com a descrição da distribuição *a posteriori*. A distribuição do LDA p é denotada nesse texto como a distribuição "original", e a distribuição q é chamada de distribuição variacional. Um modo simples de se obter a família de distribuição variacional q é considerar simples modificações na distribuição original. Removendo as arestas que ligam as variáveis θ, ϕ, z e w, obtêm um modelo simplificado sem a relação de dependência entre essas variáveis. Com as variáveis independentes, o número de combinações de valores atribuídos a elas se tornam computacionalmente viáveis. Cada variável da distribuição variacional, aqui denotada como variáveis variacionais, terão suas correspondentes na distribuição original. Na Figura 2 está o modelo gráfico da distribuição q, com suas variáveis variacionais e suas correspondentes originais. A atribuição dos tópicos $z_{j,i}$ tem como distribuição variacional $q(z_{j,i}|\varphi_{j,i}) = Mult(\varphi_{j,i})$. Note que cada palavra observada w_i terá uma distribuição variacional sobre os tópicos, isso permite que diferentes palavras sejam associadas para diferentes tópicos. A distribuição dos documentos por tópicos, θ_j, tem sua distribuição variacional gerada por uma distribuição de Dirichlet $q(\theta_j) = Dir(\gamma_j, \alpha)$, onde γ_j é um vetor K-dimensional. Existem diferentes distribuições de Dirichlet variacionais para cada documento, permitindo que diferentes documentos sejam atribuídos a diferentes tópicos com diferentes proporções. Por fim, tem-se a distribuição de tópicos por termos, ϕ, que tem distribuição variacional para cada tópicos $q(\phi_k) = Dir(\lambda_k, \beta)$, onde λ_k é um vetor n-dimensional com valores gerados pela distribuição de Dirichlet. Com base nessas simplificações, a família de

distribuições q é caracterizada pela seguinte distribuição

$$q(\theta, z, \phi) = \prod_{k=1}^{K} q(\phi_k|\lambda_k) \prod_{j=1}^{m} \left(q(\theta_j|\gamma_j) \prod_{i=1}^{n_{d_j}} q(z_{j,i}|\varphi_{j,i}) \right), \tag{2.32}$$

onde φ, γ e λ são as distribuições variacionais.

Figura 2 – Distribuição variacional aproximada para o modelo LDA.



Fonte: Adaptada de Blei, Ng e Jordan (2003).

O método variacional transforma o problema de inferência do LDA em um problema de otimização, onde o objetivo é

$$(\lambda^*, \gamma^*, \varphi^*) = \arg\min_{\lambda^*, \gamma^*, \varphi^*} KL(q(\theta, z, \phi)||p(\theta, z, \phi|w, \alpha, \beta)), \tag{2.33}$$

onde λ^*, γ^* e φ^* são os valores ótimos.

Otimizar a Equação 2.33 diretamente é inviável, mas como foi discutido na Seção 2.1.2, pode-se otimizar essa equação por meio do ELBO. Para encontrar o ELBO $\mathscr{L}$, o logaritmo da probabilidade marginal do modelo é estendido. Para isso, é descrito a Desigualdade (2.29) em relação a probabilidade marginal do LDA da seguinte forma:

$$\begin{aligned}
\log p(w|\alpha, \beta) &= \log \int \sum_{z} p(\theta, \phi, z, w|\alpha, \beta) d\theta \\
&= \log \int \sum_{z} \frac{p(\theta, \phi, z, w|\alpha, \beta) q(\theta, \phi, z)}{q(\theta, \phi, z)} d\theta \\
&\geq \int \sum_{Z} q(\theta, \phi, z) \log p(\theta, \phi, z, w|\alpha, \beta) d\theta - \int \sum_{Z} q(\theta, \phi, z) \log q(\theta, \phi, z) d\theta \\
&= E_q[\log p(\theta, \phi, z, w|\alpha, \beta)] - E_q[\log q(\theta, \phi, z)] \\
&\triangleq \mathscr{L}(\gamma, \varphi, \lambda|\alpha, \beta)
\end{aligned} \tag{2.34}$$

Como foi discutido na Seção 2.1.2, maximizar o ELBO é igual a minimizar a divergência KL entre a distribuição real do LDA e a distribuição variacional (Veja a Equação (2.31)). Os valores encontrados para os parâmetros variacionais γ, φ e λ pela minimização de $\mathscr{L}(\gamma, \varphi, \lambda|\alpha, \beta)$ são aproximações para os parâmetros da distribuição $p(\theta, \phi, z|w, \alpha, \beta)$. Então, o que é feito no método variacional é maximizar o ELBO – $\mathscr{L}(\gamma, \varphi, \lambda|\alpha, \beta)$.

As próximas subseções descrevem a expansão do ELBO ($\mathscr{L}$) segundo o modelo LDA (como descrito na Seção 2.1), e o procedimento de maximização do ELBO utilizando a técnica de gradiente descendente. Por fim, é descrito o algoritmo de inferência variacional.

Expandindo o ELBO

Relembrando alguns conceitos importantes para auxiliar na expansão do ELBO. Primeiro, resolvendo $E_q[\log p(\theta|\alpha)]$, onde a notação $E_q[\cdot]$ corresponde a esperança em relação a distribuição variacional q, e pela definição da distribuição de Dirichlet (Equação 2.1),

$$p(\theta|\alpha) = \frac{\Gamma(\sum_{k=1}^{K}\alpha_k)}{\prod_{k=1}^{K}\Gamma(\alpha_k)}\prod_{k=1}^{K}\theta_k^{\alpha_k-1}. \tag{2.35}$$

Aplicando o logaritmo em $p(\theta|\alpha)$,

$$\log p(\theta|\alpha) = \sum_k(\alpha_k-1)\log\theta_k + \log\Gamma(\sum_k\alpha_k) - \sum_k^K\log\Gamma(\alpha_k). \tag{2.36}$$

Agora, calculando a esperança em relação a distribuição q,

$$E_q[\log p(\theta|\alpha)] = \sum_k(\alpha_k-1)E_q[\log\theta_k] + \log\Gamma(\sum_k\alpha_k) - \sum_k^K\log\Gamma(\alpha_k). \tag{2.37}$$

A expressão $E[\log\theta]$ corresponde a esperança do logaritmo da distribuição θ e é calculada como

$$E[\log\theta_k] = \Psi(\gamma_i) - \Psi\left(\sum_j\gamma_j\right), \tag{2.38}$$

onde $\Psi(\cdot)$ é a função digama, e γ_i são os parâmetros variacionais da distribuição q correspondente a distribuição original θ (BLEI; NG; JORDAN, 2003).

Com isso, o que é feito agora é reescrever o ELBO. Pela Equação 2.34, tem-se a definição do ELBO

$$\mathscr{L}(\gamma,\varphi,\lambda|\alpha,\beta) \triangleq E_q[\log p(\theta,\phi,z,w|\alpha,\beta)] - E_q[\log q(\theta,\phi,z)] \tag{2.39}$$

Com base na probabilidade conjunta do LDA, descrita na Equação (2.3), pode-se reescrever a Equação (2.39) da seguinte forma:

$$\begin{aligned}
\mathscr{L}(\gamma,\varphi,\lambda|\alpha,\beta) =& E_q[\log p(\phi|\beta)] && (2.40)\\
&+ E_q[\log p(\theta|\alpha)] && (2.41)\\
&+ E_q[\log p(z|\theta)] && (2.42)\\
&+ E_q[\log p(w|z,\phi)] && (2.43)\\
&- E_q[\log q(\theta,\phi,z)] && (2.44)
\end{aligned}$$

O que será feito agora é estender cada um dos termos de $\mathscr{L}(\gamma,\varphi,\lambda|\alpha,\beta)$. Iniciando com o Termo 2.40:

$$\begin{aligned}
E_q[\log p(\phi|\beta)] &= E_q[\sum_{k=1}^{K}\log p(\phi_k|\beta)] \\
&= E_q\left[\sum_{k=1}^{K}\left(\sum_{i=1}^{n}(\beta_i-1)\log\phi_{k,i}+\log\Gamma(\sum_{i=1}^{n}\beta_i)-\sum_{i=1}^{n}\log\Gamma(\beta_i)\right)\right] \\
&= \sum_{k=1}^{K}\left(\log\Gamma(\sum_{i=1}^{n}\beta_i)-\sum_{i=1}^{n}\log\Gamma(\beta_i)+\sum_{i=1}^{n}(\beta_i-1)E_q\left[\log\phi_{k,i}\right]\right] \\
&= \sum_{k=1}^{K}\left(\log\Gamma(\sum_{i=1}^{n}\beta_i)-\sum_{i=1}^{n}\log\Gamma(\beta_i)+\sum_{i=1}^{n}(\beta_i-1)\left(\Psi(\lambda_{k,i})-\Psi(\sum_{u}^{n}\lambda_{k,u})\right)\right)
\end{aligned}$$

Da mesma forma, para o Termo 2.41 tem-se:

$$\begin{aligned}
E_q[\log p(\theta|\alpha)] &= E_q[\sum_{j=1}^{m}\left(\log p(\theta_j|\alpha)\right)] \\
&= E_q\left[\sum_{j=1}^{m}\left(\sum_{k=1}^{K}(\alpha_k-1)\log\theta_{j,k}+\log\Gamma(\sum_{i=1}^{K}\alpha_i)-\sum_{k=1}^{K}\log\Gamma(\alpha_k)\right)\right] \\
&= \sum_{j=1}^{m}\left(\log\Gamma(\sum_{k=1}^{K}\alpha_k)-\sum_{k=1}^{K}\log\Gamma(\alpha_i)+\sum_{k=1}^{K}(\alpha_k-1)E_q\left[\log\theta_{j,k}\right]\right) \\
&= \sum_{j=1}^{m}\left(\log\Gamma(\sum_{k=1}^{K}\alpha_k)-\sum_{k=1}^{K}\log\Gamma(\alpha_k)+\sum_{k=1}^{K}(\alpha_k-1)\left(\Psi(\gamma_{j,k})-\Psi(\sum_{k}^{K}\gamma_{j,k})\right)\right)
\end{aligned}$$

Para expandir o Termo 2.42, é necessário escrever $p(z|\theta)$ da seguinte forma:

$$\begin{aligned}
p(z|\theta) &= \prod_{j}^{m}\prod_{n}^{n_{d_j}}p(z_{j,i}|\theta_d) \\
&= \prod_{j}^{m}\prod_{n}^{n_{d_j}}\theta_d^{z_{j,i}}.
\end{aligned}$$

O vetor $z_{j,i}$ contêm a distribuição de tópicos atribuídos a palavra w_i no documento d_j. Quando atribuído a um tópico k, o valor de $z_{j,i,k}=1$, caso contrário, $z_{j,i,k}=0$. Logo, para o Termo 2.43, tem-se

$$\begin{aligned}
E_q[\log p(z|\theta)] &= E_q\left[\sum_{k}^{K}\sum_{j}^{m}\sum_{i}^{n_{d_j}}\log\theta_{j,k}^{z_{j,i,k}}\right] \\
&= E_q\left[\sum_{k}^{K}\sum_{j}^{m}\sum_{i}^{N_{d_j}}z_{j,i,k}\log\theta_{j,k}\right]
\end{aligned}$$

$$= \sum_{k}^{K}\sum_{j}^{m}\sum_{n}^{n_{d_j}} E_q\left[z_{j,i,k}\right] E_q\left[\log\theta_{j,k}\right]$$
$$= \sum_{k}^{K}\sum_{j}^{m}\sum_{n}^{n_{d_j}} \varphi_{j,i,k}\left(\Psi(\gamma_{j,k}) - \Psi(\sum_{l}^{K}\gamma_{l,k})\right) \tag{2.45}$$

Para expandir o Termo 2.43, é necessário escrever $p(w|z,\phi)$ da seguinte forma:

$$p(w|z,\phi) = \prod_{k}^{K}\prod_{j}^{m}\prod_{i}^{n_{d_j}} p(w_i|z_{j,i,k},\phi_k)$$
$$= \prod_{k}^{K}\prod_{j}^{m}\prod_{i}^{n_{d_j}} \phi_{k,w_i}^{z_{j,i,k}}$$

logo,

$$E_q[\log p(w|z,\phi)] = E_q\left[\sum_{k}^{K}\sum_{j}^{m}\sum_{n}^{n_{d_j}} \log\beta_{k,w_i}^{z_{j,i,k}}\right]$$
$$= \sum_{k}^{K}\sum_{j}^{m}\sum_{i}^{N_{d_j}} E_q\left[z_{j,i,k}\right] E_q\left[\log\phi_{k,w_i}\right]$$
$$= \sum_{k}^{K}\sum_{j}^{m}\sum_{i}^{n_{d_j}} \varphi_{j,i,k}\left(\Psi(\lambda_{k,i}) - \Psi(\sum_{l}^{n}\lambda_{k,l})\right)$$

No Termo 2.44, tem-se o correspondente da distribuição variacional (Equação 2.32)

$$-E_q[q(\theta,\phi,z)] = -\int_{k=1}^{K}\int_{j=1}^{m}\sum_{z} q(\theta_d,\phi_k,z)\log q(\theta_j,\phi_k,z)d\theta d\phi dz$$
$$= -\int_{k=1}^{K} q(\phi_k)\log q(\phi_k)d\phi - \int_{j=1}^{m} q(\theta_j)\log q(\theta_j)d\theta - \sum_{z} q(z)\log q(z).$$

Note que essa equação corresponde a entropia das distribuições variacionais $q(\theta)$, $q(\phi)$ e $q(z)$. Sabendo disso, basta substituir pela fórmula da entropias das distribuições de Dirichlet θ e ϕ, e distribuição Muntinomial z com parâmetros variacionais. Aplicando a definição de entropia (veja a Definição no trabalho de Frigg e Werndl (2011)) tem-se

$$-E_q[q(\theta,\phi,z)] = -\int_{j=1}^{m} q(\theta_j)\log q(\theta_j)d\theta - \sum_{z} q(z)\log q(z) - \int_{k=1}^{K} q(\phi_k)\log q(\phi_k)d\phi$$
$$= \sum_{j=1}^{m}\left(-\left(\sum_{k=1}^{K}(\gamma_{j,k}-1)\left(\Psi(\gamma_{j,k}) - \Psi(\sum_{r=1}^{K}\gamma_{j,r})\right)\right)\right.$$
$$- \log\Gamma(\sum_{k=1}^{K}\gamma_{j,k}) + \sum_{k=1}^{K}\log\Gamma(\gamma_{j,k})$$
$$\left. - \sum_{i=1}^{n_{d_j}}\sum_{k=1}^{K}\varphi_{j,i,k}\log\varphi_{j,i,k}\right)$$

$$+\sum_{k=1}^{K}\left(-\left(\sum_{i=1}^{n}(\lambda_{k,i}-1)\left(\Psi(\lambda_{k,i})-\Psi(\sum_{u=1}^{n}\lambda_{k,u})\right)\right)\right.$$
$$\left.-\log\Gamma(\sum_{i=1}^{n}\lambda_{k,i})+\sum_{i=1}^{n}\log\Gamma(\lambda_{k,i})\right)$$

Com as extensões detalhadas de todos os itens, tem-se a formulação expandida do ELBO

$$\begin{aligned}
\mathscr{L}(\gamma,\varphi,\lambda|\alpha,\beta) = &\sum_{k=1}^{K}\left(\log\Gamma(\sum_{i=1}^{n}\beta_i)-\sum_{i=1}^{n}\log\Gamma(\beta_i)+\sum_{i=1}^{n}(\beta_i-1)\left(\Psi(\lambda_{k,i})-\Psi(\sum_{u}^{n}\lambda_{k,u})\right)\right)\\
&+\sum_{j=1}^{m}\left(\log\Gamma(\sum_{k=1}^{K}\alpha_k)-\sum_{k=1}^{K}\log\Gamma(\alpha_k)+\sum_{k=1}^{K}(\alpha_k-1)\left(\Psi(\gamma_{j,k})-\Psi(\sum_{r}^{K}\gamma_{j,r})\right)\right)\\
&+\sum_{k}^{K}\sum_{j}^{m}\sum_{i}^{n_{d_j}}\varphi_{j,i,k}\left(\Psi(\gamma_{j,k})-\Psi(\sum_{l}^{K}\gamma_{j,l})\right)\\
&+\sum_{k}^{K}\sum_{j}^{m}\sum_{i}^{n_{d_j}}\varphi_{j,i,k}\left(\Psi(\lambda_{k,i})-\Psi(\sum_{v}^{n}\lambda_{k,v})\right)\\
&+\sum_{j=1}^{m}\left(-\left(\sum_{k=1}^{K}(\gamma_{j,k}-1)\left(\Psi(\gamma_{j,k})-\Psi(\sum_{r=1}^{K}\gamma_{j,r})\right)\right)\right.\\
&-\log\Gamma(\sum_{k=1}^{K}\gamma_{j,k})+\sum_{k=1}^{K}\log\Gamma(\gamma_{j,k})\\
&\left.-\sum_{i=1}^{n_{d_j}}\sum_{k=1}^{K}\varphi_{j,i,k}\log\varphi_{j,i,k}\right)\\
&+\sum_{k=1}^{K}\left(-\left(\sum_{i=1}^{n}(\lambda_{k,i}-1)\left(\Psi(\lambda_{k,i})-\Psi(\sum_{u=1}^{n}\lambda_{k,u})\right)\right)\right.\\
&\left.-\log\Gamma(\sum_{i=1}^{n}\lambda_{k,i})+\sum_{i=1}^{n}\log\Gamma(\lambda_{k,i})\right)
\end{aligned}\tag{2.46}$$

Otimizando o ELBO

Como discutido na Seção anterior, o objetivo do algoritmo de inferência variacional é encontrar os valores dos parâmetros variacionais resolvendo o problema de otimização na Equação 2.33. A especificação detalhada do ELBO serve para definir a equação a ser otimizada. Assim, deve-se maximizar a Equação (2.46) em relação a cada parâmetro variacional: γ, φ e λ

Primeiro, para maximizar a Equação (2.46) em relação a $\varphi_{d,n,k}$, definida como $\mathscr{L}_{\varphi_{d,n,k}}$, é necessário incluir a restrição $\sum_k^K \varphi_{d,n,k} = 1$. Essa restrição é imposta na equação incorporando o multiplicador de Lagrande $\rho_{j,i}$, tal que

$$\mathscr{L}_{\varphi} = \sum_{k}^{K}\sum_{j}^{m}\sum_{i}^{n_{d_j}}\varphi_{j,i,k}\left(\left(\Psi(\gamma_{j,k})-\Psi(\sum_{l}^{K}\gamma_{j,l})\right)\right.$$

$$+\left(\Psi(\lambda_{k,i})-\Psi(\sum_{r}^{n}\lambda_{k,r})\right)-\log\varphi_{j,i,k}\Bigg)+\rho_{j,i}\left(\sum_{l}^{K}\varphi_{j,i,l}-1\right)$$

Note que $\mathscr{L}_{\varphi}$ é o ELBO com apenas os termos dependentes de φ.

Para determinar o gradiente de $\mathscr{L}_{\varphi}$, deve-se calcular a derivada de $\mathscr{L}_{\varphi_{j,i,k}}$

$$\frac{d\mathscr{L}_{\varphi_{j,i,k}}}{d\varphi_{j,i,k}}=\Bigg(\left(\Psi(\gamma_{j,k})-\Psi(\sum_{r}^{K}\gamma_{j,r})\right)+\left(\Psi(\lambda_{k,i})-\Psi(\sum_{l}^{n}\lambda_{k,l})\right)-\log\varphi_{j,i,k}\Bigg)$$
$$-1+\rho_{j,i}$$

Colocando $\mathscr{L}_{\varphi}$ igual a zero e isolando $\varphi_{j,i,k}$, tem-se

$$\varphi_{j,i,k}=exp\left(\Psi(\gamma_{j,k})-\Psi(\sum_{r}^{K}\gamma_{j,r})+\Psi(\lambda_{k,i})-\Psi(\sum_{l}^{n}\lambda_{k,l})-1+\rho_{j,i}\right)$$

Não é necessário computar $\rho_{j,i}$ e $\Psi(\sum_{r}^{K}\gamma_{j,r})$, pois eles são os mesmos para todo k. Assim,

$$\varphi_{j,i,k}\propto exp\left(\Psi(\gamma_{j,k})+\Psi(\lambda_{k,i})-\Psi(\sum_{l}^{n}\lambda_{k,l})\right) \tag{2.47}$$

Em seguida, para maximizar a Equação (2.46) em relação a γ, define-se $\mathscr{L}_{\gamma}$ como

$$\begin{aligned}\mathscr{L}_{\gamma}=&\sum_{j=1}^{m}\sum_{k=1}^{K}(\alpha_k-1)\left(\Psi(\gamma_{j,k})-\Psi(\sum_{r}^{K}\gamma_{j,r})\right)\\&+\sum_{j}^{m}\sum_{i}^{n_{d_j}}\sum_{k=1}^{K}\phi_{j,i,k}\left(\Psi(\gamma_{j,k})-\Psi(\sum_{l}^{K}\gamma_{j,l})\right)\\&-\sum_{j=1}^{m}\sum_{k=1}^{K}\Bigg((\gamma_{j,k}-1)\left(\Psi(\gamma_{j,k})-\Psi(\sum_{r=1}^{K}\gamma_{j,r})\right)\\&-\log\Gamma(\sum_{k=1}^{K}\gamma_{j,k})+\sum_{k=1}^{K}\log\Gamma(\gamma_{j,k})\Bigg)\end{aligned}$$

Derivando $\mathscr{L}_{\gamma_{j,k}}$,

$$\frac{d\mathscr{L}}{d\gamma_{j,k}}=\left(\Psi'(\gamma_{j,k})-\Psi'(\sum_{r=1}^{K}\gamma_{j,r})\right)\left(\alpha-1+\sum_{i=1}^{n_{d_j}}\phi_{j,i,k}-(\gamma_{j,k}-1)\right)$$

Colocando $\mathscr{L}_{\gamma_{j,k}}$ igual a zero e isolando $\gamma_{j,k}$, tem-se

$$\gamma_{j,k}=\alpha+\sum_{i=1}^{n_{d_j}}\phi_{j,i,k} \tag{2.48}$$

E por fim, maximizar a Equação (2.46) em relação a λ

$$\begin{aligned}\mathscr{L}_{\lambda} = &\sum_{k=1}^{K}\sum_{i=1}^{n}(\beta_i - 1)\left(\Psi(\lambda_{k,i}) - \Psi(\sum_{l=1}^{n}\lambda_{k,l})\right) \\ &+ \sum_{k=1}^{K}\sum_{j=1}^{m}\sum_{i=1}^{n_{d_j}}\varphi_{j,i,k}\left(\Psi(\lambda_{k,i}) - \Psi(\sum_{l=1}^{n}\lambda_{k,l})\right) \\ &- \sum_{k=1}^{K}\left(\left(\sum_{i=1}^{n}(\lambda_{k,i} - 1)\left(\Psi(\lambda_{k,i}) - \Psi(\sum_{l=1}^{n}\lambda_{k,l})\right)\right)\right. \\ &\left. - \log\Gamma(\sum_{i=1}^{n}\lambda_{k,i}) + \sum_{i=1}^{n}\log\Gamma(\lambda_{k,i})\right)\end{aligned}$$

Derivando $\mathscr{L}_{\lambda}$

$$\frac{d\mathscr{L}_{\lambda_{k,i}}}{d\lambda_{k,i}} = \left(\Psi'(\lambda_{k,i}) - \Psi'(\sum_{l=1}^{n}\lambda_{k,l})\right)\left(\beta - 1 + \sum_{j=1}^{m}\sum_{l=1}^{n_{d_j}}1(w_{j,l} = w_i)\varphi_{j,l,k} - (\lambda_{k,i} - 1)\right)$$

Colocando $\mathscr{L}_{\lambda_{k,i}}$ igual a zero e isolando $\lambda_{k,i}$, tem-se

$$\lambda_{k,i} = \beta_k + \sum_{j=1}^{m}\sum_{l=1}^{n_{d_j}}1(w_{j,l} = w_i), \varphi_{j,l,k} \tag{2.49}$$

onde a expressão $1(w_{j,l} = w_i)$ é igual a 1 caso o termo na posição l no documento d_j for igual a palavra w_i, e 0 caso contrário.

Por fim, chegou-se nas operações de atualização do algoritmo – equações 2.49, 2.48 e 2.47. Na próxima seção será descrito o algoritmo de inferência variacional.

2.1.4 Algoritmo de inferência variacional

O método de inferência variacional transforma o problem de inferência probabilística em um problema de otimização. Esse problema de otimização pode ser resolvido iterando em direção do gradiente da função objetiva estabelecida por $\mathscr{L}(\gamma, \varphi, \lambda | \alpha, \beta)$. Derivando o ELBO $\mathscr{L}(\gamma, \varphi, \lambda | \alpha, \beta)$, são obtidas as atualizações referentes as equações 2.49, 2.48 e 2.47 que aproximam o valor da verossimilhança do modelo, $p(w|\alpha, \beta)$. A descrição desse método está no Algoritmo 4.

Algumas considerações devem ser feitas sobre o Algoritmo 4. Primeiro, a versão aqui apresentada não descreve como encontrar os hiperparâmetros α e β, é assumido que eles são simétricos e dado como entrada. O hiperparâmetro é simétrico quando é um valor constante para todas as componentes da distribuição. Em termos práticos, encontra-se na literatura valores padrões de $alpha = 50/K$ e $\beta = 0.01$ (STEYVERS; GRIFFITHS, 2007; WEI; CROFT, 2006), sendo esses valores sugerido para a maioria das aplicações.

Algoritmo 4: Algoritmo de inferência variacional para o LDA

Entrada : número de tópicos K, coleção de documentos, hiper-parâmetros α e β, número de iterações T

início
  Inicia $\gamma_j = \alpha + \frac{n}{K}$ para todo documento d_j ;
  Inicia aleatoriamente $\lambda_i = Dir(\beta)$ para toda palavra w_i do vocabulário ;
  $logverossimilhanca = 0$;
  enquanto *logverossimilhanca não convergiu* **faça**
    para *documento* d_j *com* $j \in [1,m]$ **faça**
      repita
        para *palavra* w_i *com* $i \in [1,n_{d_j}]$ *no documento* d_j **faça**
          para *tópico* $k \in [1,K]$ **faça**
            $\varphi_{j,i,k} = \exp\left(\Psi(\gamma_{j,k}) + \Psi(\gamma_{k,i} - \Psi(\sum_{l=1}^{m} \lambda_{k,l}))\right)$;
          Normaliza o vetor $\varphi_{j,i}$ para somar 1 ;
        para *tópico* $k \in [1,K]$ **faça**
          $\gamma_{j,k} = \alpha + \sum_{i=1}^{n_{d_j}} \varphi_{j,i,k}$;
      até *convergência do vetor* γ_j;
    para *palavra* w_i *com* $j \in [1,n]$ **faça**
      para *tópico* $k \in [1,K]$ **faça**
        $\lambda_{k,i} = \beta_k + \sum_{j=1}^{m} \sum_{l=1}^{n_{d_j}} 1(w_{j,l} = w_i)\varphi_{j,l,k}$;
      Normalize o vetor λ_k para somar 1 ;
    $logverossimilhanca = logverossimilhanca + \mathscr{L}(\gamma, \varphi, \lambda | \alpha, \beta)$;

2.2 Avaliação dos Modelos Probabilísticos de Tópicos

A forma mais comum de avaliar os modelos probabilísticos de tópicos é calculando o logaritmo da verossimilhança do modelo (WALLACH *et al.*, 2009). Isso é normalmente realizado separando a coleção de documentos em dois subconjuntos, um para treino e outro para teste. Com os documentos de treino cria-se o modelo. Em seguida, averígua-se o quão bom esse modelo descreve documentos não conhecidos, utilizando os documentos de testes. Quanto maior o valor do logaritmo da verossimilhança melhor é o modelo. Para o LDA, o modelo corresponde a distribuição de tópicos por palavras, ϕ. O algoritmo de inferência é aplicado na coleção de treino para encontrar as distribuições ϕ e θ_{treino}. Já o conjunto de teste corresponde aos documentos não conhecidos pelo modelo. A distribuição de documentos por tópicos, θ_{treino}, não é considerada na avaliação pois descreve apenas aos documentos de treino, logo será necessário computar uma nova distribuição θ_{teste} com apenas os documentos de teste.

Avaliar a efetividade do modelo de extração de tópicos está fortemente relacionada com a correta decomposição do conjunto de documentos em conceitos humanamente interpretáveis. O que se encontra na literatura para avaliar modelos de mistura de tópicos são métricas que verificam a capacidade do modelo aprendido em predizer dados não vistos (CHANG *et al.*,

2009). O modelo que descreve uma coleção de documentos será bom se a distribuição de tópicos por palavras ϕ também corresponder aos tópicos contidos na coleção de treino. Uma métrica bastante utilizada é a medida de perplexidade (*perplexity* measure) (WAAL; BARNARD, 2008). Para aplicar essa medida é necessário dividir todo o conjunto de dados em treinamento e teste. O modelo é criado com o conjunto de treino, então mede-se o quão "perplexo" está o modelo no conjunto de teste, ou seja, é medido o quão bem está a probabilidade das palavras do documento de teste representada pela distribuição de tópicos por palavras obtidas pelo modelo. Quanto menor o valor de perplexidade melhor será o modelo. A perplexidade é calculada da seguinte forma:

$$perplexidade(w) = \exp\left(-\frac{\log p(w|\alpha,\beta)}{\log \sum_{j=1}^{m} n_{d_j}}\right). \tag{2.50}$$

O logaritmo da verossimilhança do modelo, $p(w|\alpha,\beta)$, é obtido de forma diferente dependendo do algoritmo de inferência empregado. No algoritmo de amostragem de Gibbs é calculado da seguinte forma:

$$p(w|\alpha,\beta) = \sum_{j=1}^{m}\sum_{i=1}^{n_{d_j}} \log \sum_{k=1}^{K} \theta_{j,k}\phi_{k,i}. \tag{2.51}$$

Já no algoritmo de inferência variacional, o logaritmo da verossimilhança do modelo é aproximado pelo ELBO ($\mathscr{L}$ – Equação 2.34).

Essas medidas são boas para comparações entre os modelos probabilísticos, entretanto, os valores obtidos nas medições não necessariamente condizem com a correta relação entre os tópicos encontrados e os assuntos descritos na coleção (CHANG *et al.*, 2009).

Informalmente, avaliações desses modelos podem ser realizadas das seguintes formas: (1) Inspecionar cada tópico, fazendo a busca por palavras de maior probabilidade e verificar se essas palavras são coerentemente relacionadas a algum conceito presente na coleção. (2) Manter um conjunto de documentos escolhidos aleatoriamente e ver se os tópicos encontrados fazem sentido ou não. Baseado nisso, o trabalho de (CHANG *et al.*, 2009) propõe métodos quantitativos para mensurar o significado semântico dos tópicos inferidos pelo modelo. Um método, chamado *Word Intrusion*, mede a coerência desses tópicos. Na tarefa realizada por esse método, um tópico é escolhido aleatoriamente, em seguida, as cinco palavras mais relacionadas com esse tópico é selecionada, uma sexta palavra é escolhida do conjunto de palavras menos relacionadas ao tópico escolhido. Entre essas seis palavras, o usuário deve encontrar aquela que menos se relaciona com todas as outras palavras. Se o tópico não é coerente semanticamente, será difícil apontar qual é a palavra menos relacionada. Outro método proposto é chamado *Topic Intrusion*, nessa tarefa é mostrado o título e partes do texto de um documento. Junto com o documento são apresentados quatro tópicos (cada tópico contém oito palavras com maior probabilidade). Desses tópicos, três são altamente relacionados com o documento. O tópico restante é escolhido aleatoriamente do conjunto de tópicos poucos relacionados. O usuário deve encontrar o tópico que menos se relaciona ao documento apresentado.

Ainda com o objetivo de se obter a avaliação da interpretabilidade, no trabalho de Newman *et al.* (2010) foi proposto um método automático baseado na informação mútua entre pares de palavras que formam o tópico, chamado *Pointwise Mutual Information* (PMI), para simular a avaliação humana sobre a qualidade dos tópicos. No trabalho de Mimno *et al.* (2011) foi avaliado a metodologia para computar ao coerência, substituindo PMI pelo logaritmo da probabilidade condicional dos pares de palavras. Já no trabalho de Musat *et al.* (2011) foi incorporado a hierarquia do WordNet para capturar a relevância dos tópicos. No trabalho de Lau, Newman e Baldwin (2014), foram utilizadas diferentes técnicas de avaliação com o objetivo de encontrar a melhor, como resultado, a medida *Normalized Pointwise Mutual Information* (NPMI) (veja a Definição 2) foi apontada como a que mais se aproxima da avaliação feita por especialistas, e pode ser utilizada para automatizar a avaliação da coerência do tópico considerando os termos selecionados como descritores e sua coocorrência em relação a uma coleção de referência. A coleção de referência é utilizada para calcular a coocorrência entre os termos relacionados, e usualmente, os documentos da Wikipédia[2] são utilizados como referência externa.

Definição 2 (NPMI – *Normalized Pointwise Mutual Information*)**.** Seja $top_K^L = \{w_1, \ldots, w_L\}$ o conjunto das L palavras com maior probabilidade na distribuição de tópicos. Newman *et al.* (2010) assumem que quanto maior a similaridade média entre os pares das palavras em top_K^L, mais coerente é o tópico. Com isso, pode-se definir a função do NPMI como:

$$NPMI(top_K^L) = \sum_{i=1}^{L} \sum_{j=1}^{L-1} \frac{\log \frac{P(w_i,w_j)}{P(w_i)P(w_j)}}{-\log P(w_i,w_j)} \tag{2.52}$$

2.3 Comparando LDA com NMF

O método NMF (*Nonnegative Matrix Factorization*) (PAATERO; TAPPER, 1994; LEE; SEUNG, 1999) fatoriza aproximadamente a matriz com elementos não negativos em duas outras matrizes também com elementos não negativos (Veja a Definição 3).

Definição 3 (NMF – *Nonnegative Matrix Factorization*)**.** Dado uma matriz com valores não negativos $F \in \mathbb{R}^{m \times n}$, quando o número de dimensões reduzidas é K, o objetivo do NMF é encontrar duas matrizes $A \in \mathbb{R}^{m \times K}$ e $B \in \mathbb{R}^{K \times n}$ com apenas entradas não negativas tal que

$$F \approx A \cdot B \tag{2.53}$$

Os fatores A e B são obtidos pela minimização de uma função de custo definida por uma medida de "distância". Existem diferentes tipos de funções de custo (LEE; SEUNG, 2001).

[2] <https://www.wikipedia.org/>

A função que se relaciona com a formulação dos modelos probabilísticos de tópicos é aquela baseada na divergência KL. Essa função é definida como

$$Q_{NMF-KL} = \min \sum_{j,i} \left(F_{j,i} \log \frac{F_{j,i}}{(AB)_{j,i}} - F_{j,i} + (AB)_{j,i} \right), \tag{2.54}$$

onde $F_{j,i}$ é a entradas da linha j e coluna i matriz F, no caso de uma matriz documento-termo, o valor de $F_{j,i}$ pode ser a frequência do termo w_i no documento d_j. Note que o valor de $-F_{j,i} + (AB)_{j,i}$ será igual a zero caso $F_{j,i} = (AB)_{j,i}$.

A técnica mais simples de resolver a otimização da Equação 2.54 é aplicando o método de gradiente descendente. Fazendo as derivações, chega-se nas seguinte equações de atualização

$$A_{j,k} = A_{j,k} \frac{\sum_i B_{k,i} F_{j,i} / (AB)_{j,k}}{\sum_q B_{k,q}}, \tag{2.55}$$

$$B_{k,i} = B_{k,i} \frac{\sum_j A_{j,k} V_{j,i} / (AB)_{j,i}}{\sum_p A_{p,k}}. \tag{2.56}$$

Assim, interpolando as atualizações das equações 2.55 e 2.56 em várias iterações chega-se nos fatores que aproximam a matriz F. A convergência desse algoritmo não é apropriadamente demonstrada, entretanto, no trabalho de (LEE; SEUNG, 2001) é demonstrado que em cada iteração as atualizações sempre irão diminuir o valor resultante da Equação 2.54.

Apesar de não ser um método probabilístico, o NMF é descrito nesse capítulo pois apresenta similaridades com os modelos de tópicos. Além disso, o NMF e o LDA são duas técnicas popularmente aplicadas no problema de extração de tópicos em coleções de documentos. Nessa seção é realizada uma análise comparativa entre essas duas técnicas, demonstrando que NMF com divergência KL aproxima ao algoritmo de inferência variacional do LDA. Essa análise comparativa é útil para elucidar a implementação do algoritmo de inferência variacional e explorar as relações entre as diferentes técnicas.

A equivalência entre o NMF e o pLSI (*probabilistic Latent Semantic Indexing*) tem sido discutida em vários trabalhos (BUNTINE, 2002; GAUSSIER; GOUTTE, 2005). Ding, Li e Peng (2008) demonstraram que NMF e PLSI otimizam a mesma função objetivo. Apesar do LDA ser a contrapartida com fundamentação em probabilidade Bayesiana do pLSI (GIROLAMI; KABáN, 2003), a equivalência entre NMF e LDA não é bem definida. Entretanto, existem evidências que tal relação intrínseca também exista (J.; LIU; CAO, 2012; GERSHMAN; BLEI, 2012). Nessa seção, o objetivo é esclarecer essa relação em termos de formulação matemática, demonstrando que o NMF com divergência KL aproxima a solução obtida pelo algoritmo de inferência variacional do LDA. Além disso, será demonstrado a relação entre os dois algoritmos.

As correspondências entre NMF com divergência KL e o algoritmo de inferência variacional para o LDA seguem do fato de que ambos tentam minimizar a divergência entre as estatísticas que relacionam a frequência de palavras, documentos por tópicos e tópicos por

palavras. Para esclarecer essa relação, o NMF será descrito como uma relaxação do problema estabelecido no método de inferência variacional. A equivalência é alcançada quando as funções $\log\Gamma(\cdot)$ e $\Psi(\cdot)$ são aproximadas e substituídas nas derivações do LDA. Essa relação é demonstrada no Teorema 1.

Teorema 1. A função objetivo do NMF com a divergência KL é uma aproximação do ELBO (*Evidence Lower Bound*) do LDA com *priori* simétricas.

Demonstração. Inicialmente, expande-se o ELBO usando fatoração da distribuição conjunta do LDA, p (Equation 2.3), e a distribuição variacional, q (Equation 2.32):

$$\begin{aligned}
\mathscr{L} &\triangleq E_q[\log p(\theta,z,w|\alpha,\beta)] - E_q[\log q(\theta,z)] \\
&= E_q[\log p(\theta|\alpha)] + E_q[\log p(z|\theta)] + E_q[\log p(w|z,\beta)] - E_q[\log q(\theta)] - E_q[\log q(z)] \\
&= \prod_{j=1}^{m} \Bigg\{ \left[\log\Gamma\left(\sum_{k=1}^{K}\alpha_k\right) - \sum_{k=1}^{K}\log\Gamma(\alpha_k) + \sum_{k=1}^{K}(\alpha_k-1)\left(\Psi\left(\gamma_{j,k}\right) - \Psi\left(\sum_{r=1}^{K}\gamma_{j,r}\right)\right)\right] \\
&\quad + \left[\sum_{i=1}^{n_{d_j}}\sum_{k=1}^{K}\varphi_{j,i,k}\left(\Psi(\gamma_{j,k}) - \Psi\left(\sum_{r=1}^{K}\gamma_{j,r}\right)\right)\right] \\
&\quad + \left[\sum_{l=1}^{n_{d_j}}\sum_{k=1}^{K}\sum_{i=1}^{n}\varphi_{j,i}1(w_{j,l}=w_i)\log\beta_{k,i}\right] \\
&\quad + \left[-\log\Gamma\left(\sum_{k=1}^{K}\gamma_{j,k}\right) + \sum_{r=1}^{K}\log\Gamma\left(\gamma_{j,r}\right) - \sum_{k=1}^{K}(\gamma_{j,k}-1)\left(\Psi(\gamma_{j,k}) - \Psi\left(\sum_{r=1}^{K}\gamma_{j,r}\right)\right)\right] \\
&\quad + \left[-\sum_{i=1}^{n_{d_j}}\sum_{k=1}^{K}\varphi_{j,i,k}\log\varphi_{j,i,k}\right]\Bigg\}
\end{aligned} \tag{2.57}$$

Agora, aproxima-se a Equação 2.57 substituindo as funções Gamma $\Gamma(\cdot)$ e Digamma $\Psi(\cdot)$. A função Gamma é definida como $\Gamma(x) = \int_0^\infty u^{x-1}e^{-u}du$, para $x > 0$. Em geral, $\Gamma(x+1) = x\Gamma(x)$, e para argumentos inteiros, $\Gamma(x+1) = x!$. Para propósitos práticos, é considerado a aproximação de Stirlings da função $\Gamma(\cdot)$:

$$\log\Gamma(x) = \log x! = \sum_{i=1}^{n}\log i \approx \int_{i=1}^{x}\log(i)di \approx x\log x - x. \tag{2.58}$$

A função Digamma é definida como $\Psi(x) = \frac{d}{dx}\log\Gamma(x)$, e pode ser aproximada por

$$\Psi(n) \approx \log n - c, \tag{2.59}$$

onde c é um valor constante (MUQATTASH; YAHDI, 2006).

A distribuição γ_j pode ser relacionada com o vetor A_j associado a cada documento d_j. Da mesma forma, a distribuição β pode ser relacionada a matrix B. Assim, considerando a versão do LDA com hiper-parâmetros α simétricos, é possível reescrever o ELBO usando as

correspondentes aproximações para as funções Gamma, Equação 2.58, e Digamma, Equação 2.59:

$$
\begin{aligned}
\mathscr{L} \approx \prod_{j=1}^{m} \Bigg\{ & \left[\sum_{k=1}^{k} (\alpha_k - 1) \left(\log \frac{A_{j,k}}{\sum_{r=1}^{K} A_{j,r}} \right) \right] \\
& + \left[\sum_{i=1}^{n} \sum_{k=1}^{K} f_{j,i} \varphi_{j,i,k} \left(\log \frac{A_{j,k}}{\sum_{r=1}^{K} A_{j,r}} \right) \right] \\
& + \left[\sum_{i=1}^{n} \sum_{k=1}^{K} F_{j,i} \varphi_{j,i,k} \left(\log \frac{B_{i,k}}{\sum_{l=1}^{n} B_{l,k}} \right) \right] \\
& + \left[\sum_{k=1}^{K} \left(A_{j,k} \left(\log A_{j,k} - 1 \right) - (A_{j,k} - 1) \left(\log \frac{A_{j,k}}{\sum_{r=1}^{K} A_{j,r}} \right) \right) \right] \\
& + \left[\sum_{i=1}^{n} \sum_{k=1}^{K} -F_{j,i} \varphi_{j,i,k} \log \varphi_{j,i,k} \right] \Bigg\} \\
= \sum_{j=1}^{m} \sum_{i=1}^{n} \sum_{k=1}^{K} & \left(F_{j,i} \varphi_{j,i,k} \log \frac{\frac{A_{j,k}}{\sum_{r=1}^{K} A_{j,r}} \frac{B_{i,k}}{\sum_{l=1}^{n} B_{l,k}}}{\varphi_{j,i,k}} \right. \\
& \left. + (\alpha_k - A_{j,k}) \left(\log \frac{A_{j,k}}{\sum_{r=1}^{K} A_{j,r}} \right) - A_{j,k} (\log A_{j,k} - 1) \right)
\end{aligned}
\tag{2.60}
$$

Considerando que os vetores A_j e B_i são normalizados de forma que $\sum_{k=1}^{K} A_{j,k} = 1$ e $\sum_{l=1}^{n} B_{i,l} = 1$, e definindo $\mathscr{R}(A_{j,k}, \alpha_k) = (\alpha_k - A_{j,k})(\log A_{j,k}) - A_{j,k}(\log A_{j,k} - 1)$, pode-se reescrever a Equação 2.60 para alcançar o seguinte problema de otimização

$$
\begin{aligned}
\max \mathscr{L} \approx & \max \sum_{j=1}^{m} \sum_{i=1}^{n} \sum_{k=1}^{K} \left(F_{j,i} \varphi_{j,i,k} \log \frac{A_{j,k} B_{i,k}}{\varphi_{j,i,k}} + \mathscr{R}(A_{j,k}, \alpha_k) \right) \\
\approx & \min \sum_{j=1}^{m} \sum_{i=1}^{n} \sum_{k=1}^{K} \left(F_{j,i} \varphi_{j,i,k} \log \frac{\varphi_{j,i,k}}{A_{j,k} B_{i,k}} - \mathscr{R}(A_{j,k}, \alpha_k) \right).
\end{aligned}
\tag{2.61}
$$

Sabendo que $\sum_{i=1}^{n} a_i \log \frac{a_i}{b_i} \leq \sum_{i=1}^{n} a_i \log \frac{\sum_{i=1}^{n} a_i}{\sum_{i=1}^{n} b_i}$ para qualquer a_i e b_i não negativo, e em seguida adicionando a constante $\sum_{j,i} F_{j,i} \log F_{j,i}$, tem-se

$$
\begin{aligned}
\leq & \min \sum_{j=1}^{m} \sum_{i=1}^{n} \left(F_{j,i} \sum_{k=1}^{K} \varphi_{j,i,k} \log \frac{\sum_{k=1}^{\mathscr{K}} \varphi_{j,i,k}}{\sum_{k=1}^{K} A_{j,k} B_{i,k}} - \sum_{k=1}^{K} \mathscr{R}(A_{j,k}, \alpha_k) \right) \\
\approx & \min \sum_{j}^{m} \sum_{i=1}^{n} \left(F_{j,i} \log \frac{F_{j,i}}{\sum_{k=1}^{K} A_{j,k} B_{i,k}} - \sum_{k=1}^{K} \mathscr{R}(A_{j,k}, \alpha_k) \right).
\end{aligned}
\tag{2.62}
$$

O último termo na Equação 2.62 é equivalente ao NMF (Equação 2.54) menos o termo $\mathscr{R}(A_{j,k}, \alpha_k)$. O termo $\mathscr{R}(A_{j,k}, \alpha_k)$ possui um papel importante no desempenho do LDA, ele corresponde a influência da *priori* e também inclui esparsidade na distribuição de documentos por tópicos. Quando isso é adicionado ao NMF, obtêm-se um termo regularizador que restringe os valores dos vetores A_j. Então, pode-se concluir que maximizar o ELBO do LDA com

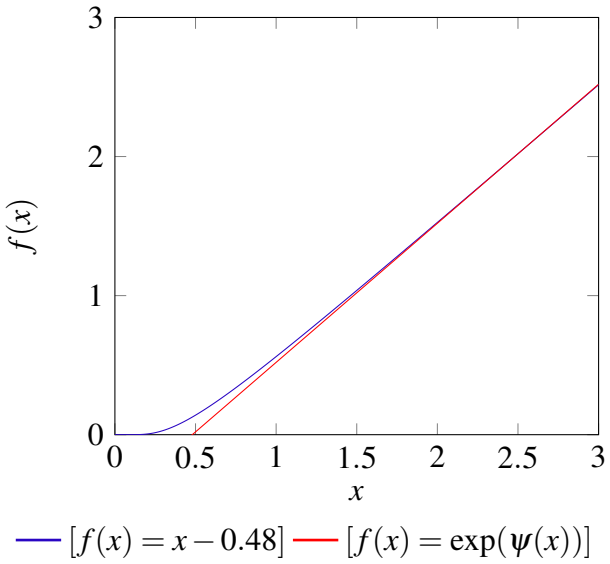
priori simétrica é proporcional a minimizar a função objetiva do NMF com divergência KL desconsiderando o termo regularizador.

■ □

Na teoria, os métodos iterativos aplicados no LDA e NMF são distintos e com diferentes fundamentações. Na prática, existem similaridades nas operações realizadas pelos seus algoritmos. Assim, será indicado essas equivalências e comparado as operações de atualizações do NMF, equações 2.55 e 2.56, e do LDA com inferência variacional, equações 2.48, 2.49 e 2.47.

Na regra de atualização do LDA, a operação de exponenciação sobre a função digama $\Psi(x)$ aproxima uma função linear quando $x > 0.48$ (MUQATTASH; YAHDI, 2006). Para perceber essa aproximação, veja a Figura 3.

Figura 3 – Plote da função linear $f(x) = x - 0.48$ e da função $f(x) = exp(\psi(x))$. Isso indica que a operação exponencial sobre a função digama aproxima uma função linear quando $x > 0.48$, *i.e.* $\exp(\psi(x)) \approx x - 0.48$ se $x > 0.48$



Fonte: Elaborada pelo autor.

Aproveitando a aproximação da função $\exp(\Psi(x))$, é possível aproximar o valor de φ utilizando apenas operações lineares

$$\varphi_{j,i,k} \approx \beta_{k,i} \times \frac{\gamma_{j,k}}{\sum_{k^*=1}^{\mathscr{K}} \gamma_{j,k^*}}. \tag{2.63}$$

Assim, o valor de $\varphi_{j,i}$ aproxima o produto de Hadamard entre o vetor normalizado γ_j e β_k. A matriz resultante, A, é relacionada a distribuição documento-tópicos γ. Da mesma forma, a matriz B é relacionada com a distribuição tópico-palavras β. Considerando essas relações, é possível aproximar a equação de atualização obtidas pelo método de inferência variacional para

o parâmetro φ,

$$\varphi_{j,i,k} \propto \left(\frac{A_{j,k} B_{k,i}}{\sum_{k^*=1}^{K} A_{j,k^*} B_{k^*,i}} \right) \tag{2.64}$$

Sem perda de generalidade, será considerada a normalização nas linhas da matriz B, *i.e.* $\sum_i B_{k,i} = 1$. Então, usando a Equação 2.64, é possível reescrever as atualizações de cada posição do fator $A_{j,k}$ na Equação 2.55 como

$$A_{j,k} = \sum_{i=1}^{\#w} F_{j,k} \varphi_{j,i,k}. \tag{2.65}$$

Note que a equação de atualização para o fator A_j, Equação 2.65, é similar a atualização da equação dos parâmetros γ_j sem o parâmetros α, Equação 2.48.

A equação de atualização do fator $B_{k,i}$ pode ser reescrita considerando a aproximação φ, Equação 2.64, e o último valor de $A_{j,k}$ obtido na Equação 2.65

$$\begin{aligned} B_{k,i} &= \frac{1}{\sum_j A_{j,k}} \frac{\sum_j F_{j,k} A_{j,k} B_{k,i}}{(AB)_{j,k}} \\ &= \frac{\sum_j F_{j,k} \varphi_{j,i,k}}{\sum_j \sum_i F_{j,k} \varphi_{j,k,i}}. \end{aligned} \tag{2.66}$$

Pela Equação 2.66, pode-se notar que o valor de $B_{k,i}$ é obtido com valor de φ para uma palavra específica w_i e tópico k para cada documento d_j, e normalizado para cada palavra w_i do vocabulário. Isso corresponde a distribuição de tópicos por palavras para um tópico k, representado pela distribuição β_k no LDA.

A indicação da relação entre o NMF com a divergência KL e o LDA com o algoritmo de inferência variacional é importante para entender os procedimentos realizados pelos modelos de tópicos. E com esse objetivo, foi mostrado que o NMF (com divergência KL) é de fato um caso especial do LDA onde é assumido uma distribuição de Dirichlet uniforme, e que o algoritmo de atualizações multiplicativas para resolução do NMF pode ser aproximado para as atualizações estabelecidas pelo algoritmo de inferência variacional do LDA.

2.4 Conclusão

Neste capítulo foi descrito os modelos probabilísticos de tópicos, especificando o LDA, que é o modelo base. O LDA foi formalmente descrito e também apresentado os dois principais algoritmos de inferência, o método de amostragem de Gibbs e o método de inferência variacional. Para explorar de forma apropriada essa área, foi realizado um estudo detalhado dos algoritmos de inferência. Por isso detalhes das derivações do modelo foram descritos. O grande objetivo do estudo apresentado neste capítulo foi registrar detalhadamente o processo de derivação do modelo para a obtenção do algoritmo de inferência. Uma outra contribuição resultante dos

estudos descritos neste capítulo está na análise comparativa do modelo LDA com o método de fatoração de matrizes NMF. Uma vez bem definida essa relação, será possível explorar o melhor desses dois métodos, possibilitando o desenvolvimento de novos algoritmos otimizados.

CAPÍTULO

3

PROPAGAÇÃO EM GRAFOS BIPARTIDOS

Atualmente, uma grande quantidade de dados estão disponíveis em formato não estruturado. A maioria desses dados são textos, como *emails*, relatórios, notícias, artigos e páginas *web*. De fato, armazenar informações como texto é a forma mais usual. Assim, técnicas automáticas que auxiliam na organização, gerenciamento e extração é um tópico de pesquisa pertinente para a comunidade de mineração de dados, possuindo vários desafios.

Outra questão está na forma de representação desses dados. A forma tradicional de representar documentos textuais é no formato vetorial (SALTON; WONG; YANG, 1975). Embora possa ser aplicada uma forma mais expressiva utilizando grafos homogêneos ou heterogêneos. Na forma de grafos homogêneo, são permitidas arestas entre os objetos do mesmo tipo, contendo apenas relações entre documentos ou relações entre termos. Em um grafo heterogêneo, são permitidas arestas ligando pares de objetos de diferentes tipos. Uma forma intuitiva de representar uma coleção de documentos é criando um grafo bipartido onde vértices correspondem a documentos e palavras, e as arestas representam a ocorrência de uma palavra no documento. Eventualmente, um peso pode ser associado a aresta de acordo com a frequência da palavra no documentos.

Descrever algoritmos usando uma representação dos dados na forma de grafos provê várias vantagens. A representação na forma de grafos (1) possui esparsidade e garante baixo consumo de memória, (2) permite uma fácil descrição das operações para a inclusão e exclusão de novos vértices, arestas e subgrafos, operações fundamentais para tratar fluxo de dados, (3) permite, a partir da topologia do grafo, distinguir entre estrutura local e estrutura global; e por fim (4) permite manter informações estrutural dos textos como ordem de palavras, localização ou estrutura sintática (SONAWANE; KULKARNI, 2014).

Algoritmos baseados em grafos são principalmente usados em esquemas de propagação de rótulos, nos quais os rótulos associados aos objetos são propagados através da estrutura do grafo para realizar a classificação ou o agrupamento dos vértices. O uso apropriado de infor-

mações enriquecidas transportadas para esses grafos podem levar a algoritmos de propagação que usam poucos rótulos, e mesmo assim são capazes de sobrepor o desempenho de algoritmos de indução que usam grande quantidade de documentos rotulados (KONG; NG; ZHOU, 2013; CHAPELLE; SCHLKOPF; ZIEN, 2010; JOACHIMS, 1999). E mesmo quando nenhuma informação de classe é fornecida, é possível obter grupos de vértices relacionados. Além disso, a estrutura do grafo pode ser utilizada para descrever o próprio modelo de classificação (BERTINI *et al.*, 2011; ROSSI *et al.*, 2014).

Dessa forma, neste capítulo é apresentado a abordagem de aprendizado de máquinas utilizando representação em grafos heterogêneos bipartidos. A representação via grafos heterogêneos bipartidos tem a vantagem de permitir que informações ou visões complementares dos dados "colaborem" entre si na detecção de padrões desses dados, como foi explorado, por exemplo, pelo método de *Cotraining* (BLUM; MITCHELL, 1998). Por exemplo, em um grafo documento-termo, alguns documentos classificados podem caracterizar os termos mais comuns que ocorrem em documentos dessa classe, e esses termos, por sua vez, podem sugerir a classe de outros documentos, não classificados, se conectados a eles.

A forma tradicional de aplicar algoritmos de agrupamento baseados em grafos em documentos textuais requer um passo crítico que é converter dados no formato vetorial para uma estrutura de grafos. Nos algoritmos tradicionais, os grafos representam apenas um único tipo de objetos, normalmente os vértices representam os documentos, e as arestas a relação entre os documentos. As arestas devem ser criadas por uma medida de similaridade entre cada par de documentos. Esse processo pode ser computacionalmente caro. Além disso, estudos prévios mostraram que a qualidade dos resultados de diferentes algoritmos de aprendizado, como agrupamento e classificação, podem se alterar dependendo do método de construção de grafos (BERTON; LOPES, 2014). Porém, na representação de um grafo bipartido documento-termo, a construção é imediata, bastando apenas conectar os vértices do tipo documento aos vértices do tipo termo e não exigindo qualquer parâmetro. Logo, grafos bipartidos é uma forma de representação simples e eficiente de documentos textuais.

Na pesquisa de doutorado focou-se, principalmente, nas abordagens não supervisionadas e semissupervisionadas, nas quais se encontram duas das principais contribuições desta tese. No entanto, para este capítulo ficar "completo" e demonstrar mais claramente a capacidade do aprendizado baseado em propagação em grafos bipartidos, é tratado também o aprendizado supervisionado. Para quem quiser se aprofundar neste último, a tese de Rossi (2015) dedica-se ao aprendizado supervisionado no contexto de classificação de documentos usando redes bipartidas. Houve uma cooperação com o trabalho de Rossi, que deu origem ao artigo (ROSSI *et al.*, 2014). Esse artigo descreve uma técnica de indução de modelos de classificação baseados em grafo bipartido, e tal técnica é sumarizada aqui na seção de aprendizado supervisionado. Assim, este capítulo abarca o aprendizado supervisionado, semissupervisionado e não supervisionado no contexto de dados estáticos, sendo as principais contribuições desta tese as abordagens descritas

nas seções 3.3 e 3.4. No próximo capítulo, Capítulo 4, uma terceira contribuição, a extensão do aprendizado baseado em propagação em grafo bipartido para o contexto de dados em fluxo é apresentada completando as propostas desenvolvidas durante esta investigação.

O restante do capítulo está organizado da seguinte forma. Na Seção 3.1 é apresentada a notação utilizada. Na Seção 3.2 é apresentado o modelo de aprendizado supervisionado utilizando grafos bipartidos. Na Seção 3.3 é apresentado a versão do algoritmo de propagação em grafos bipartidos no contexto não-supervisionado, em específico, é tratado o problema de extração de tópicos. E por fim, na Seção 3.4 é apresentado a versão semissupervisionada do algoritmo proposto, sendo essa versão uma extensão da versão não-supervisionado e com aplicação na tarefa de classificação transdutiva.

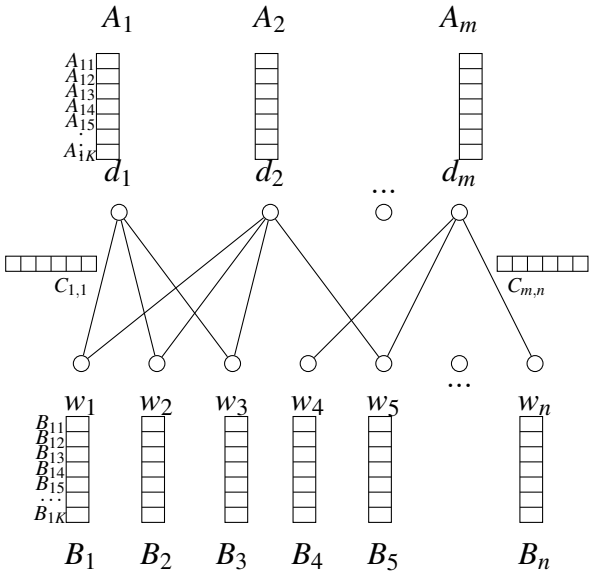
3.1 Notação

Nesta seção é introduzida a terminologia e a notação utilizada neste trabalho para a descrição dos grafos. Um grafo bipartido não direcionado é uma tripla $G = ((\mathscr{D} \cup \mathscr{W}), \mathscr{E}, f)$ onde $\mathscr{D} = \{d_1, \ldots, d_m\}$ e $\mathscr{W} = \{w_1, \ldots, w_n\}$ são os conjuntos de vértices, e $\mathscr{E}$ é o conjunto de arestas do tipo $e_{j,i} = (d_j, w_i)$. E f é uma função que atribui a cada aresta um peso, *i.e.* $\mathscr{E} \to \mathbb{R}$. Para simplificar a notação, o valor de $f(e_{j,i})$ é denotado como $f_{j,i}$. No contexto de representação de coleções de documentos, cada elemento $d_j \in \mathscr{D}$ do grafo representa um documento, e cada elemento $w_i \in \mathscr{W}$ representa uma palavra. Uma aresta $e_{j,i}$ liga a ocorrência da palavra w_i no documento d_j, e o valor de $f_{j,i}$ é a frequência do termo w_i no documento d_j.

O algoritmo proposto neste trabalho é baseado na propagação de rótulos entre os vértices. Os rótulos são identificadores de classes, e no caso não-supervisionado eles são os identificadores dos grupos. Aqui, é elaborado um esquema de rótulos multidimensionais. Em vez de cada rótulo indicar apenas um grupo (ou classe), os rótulos podem indicar diferentes graus de pertinência a vários grupos. Para manter as informações dos rótulos multidimensionais, a cada vértice é associado um vetor K-dimensional, onde K é o número de grupos. Um vetor A_j é associado a um vértice $d_j \in \mathscr{D}$ e um vetor B_i ao vértice $w_i \in \mathscr{W}$. Da mesma forma, é associado um vetor K-dimensional a cada aresta, assim, para uma aresta $e_{j,i} \in \mathscr{E}$ é associado um vetor $C_{e_{j,i}}$. Cada posição k ($1 \leq k \leq K$) dos vetores A_j, B_i e $C_{e_{j,i}}$, denotada como $A_{j,k}$, $B_{i,k}$ e $C_{e_{j,i},k}$, corresponde a pertinência do vértice ou aresta no grupo k. Veja na Figura 4 a representação do grafo bipartido, com os índices dos vértices e das arestas e os rótulos multidimensionais associados.

No caso do contexto supervisionado e semissupervisionado, define-se $\mathscr{C} = \{c_i, \ldots, c_l\}$ como o conjunto de l rótulos de classes. O conjunto $\mathscr{D}^l$ representa o conjunto de documentos rotulados e $\mathscr{D}^u$ representa o conjunto de documentos não rotulados, assim $\mathscr{D} = \mathscr{D}^l \cup \mathscr{D}^u$. Finalmente, seja $\mathscr{Y} = \{Y_1, \ldots, Y_m\}$ o conjunto de vetores l dimensionais, nos quais cada documento $d_j \in \mathscr{D}^l$ rotulado por uma classe c_k tem um vetor Y_j associado, no qual o valor da k-ésima dimensão do vetor Y_j é igual a 1, $Y_{j,k} = 1$, e 0 para todas as outras dimensões, $Y_{j,r} = 0$ para todo

Figura 4 – Grafo Bipartido G



Fonte: Elaborada pelo autor.

$r \neq k$. Esse vetor contém a real informação de classe previamente fornecida (dados rotulados). O valor de Y_j para um documento $d_j \in \mathscr{D}^u$ não rotulado é determinado depois do processo de propagação na versão semissupervisionada do algoritmo, e determinado após o processo de indução na versão supervisionada.

3.2 Aprendizado supervisionado

Uma das tarefas básicas na área de aprendizado de máquinas é a indução de um modelo para classificação de dados. Em específico, na aplicação em dados textuais, a tarefa de classificação consiste em atribuir categorias (ou rótulos) para os documentos de uma coleção. A maioria dos métodos de classificação utilizam a representação da coleção de documentos na forma matricial, formando uma matriz do tipo documento-termo. No contexto de aprendizado de máquinas, os documentos são os exemplos e os termos são as características. Em muitos casos, são utilizados algoritmos de classificação de propósito geral para a aplicação em dados textuais. Entretanto, em alguns cenários esses algoritmos não são eficientes devido a especificidades dos dados no formato textual, como alta dimensionalidade e esparsidade. Uma alternativa é a representação em grafo, na qual a coleção é representada pelas relações entre documentos e termos. Em especial, neste trabalho, é destacada a representação em grafos bipartidos, na qual o grafo consiste de documentos e palavras como vértices e a coocorrência de uma palavra no documento como as arestas. A vantagem da construção do grafo bipartido é o não requerimento da onerosa fase de criação do grafo, ou seja, não é necessário calcular as relações de similaridades entre os documentos ou entre os termos para a criação das arestas do grafo.

Alguns trabalhos tem empregado grafos bipartidos para a classificação (XUE *et al.*, 2004; YIN *et al.*, 2009; JI *et al.*, 2010), sendo que a maioria desses trabalhos utilizam a técnica transdutiva de classificação. A classificação transdutiva não cria um modelo para a classificação de novos documentos. Em vez disso, ela considera o conjunto de documentos rotulados e não rotulados para realizar a classificação. A classificação transdutiva, e os trabalhos relacionados são descritos na Seção 3.4.

Nesta seção é apresentado um algoritmo de classificação de documentos textuais que utiliza a estrutura do grafo bipartido para a indução do modelo de classificação. A indução do modelo consiste em atribuir vetores a cada vértice do tipo documento e palavra. Esses vetores mantêm as informações de classes obtidas por uma coleção de documentos já rotuladas (documentos de treino), sendo que os vetores associados aos documentos são diretamente os rótulos pré-determinados, e os valores dos vetores associados as palavras são obtidas no processo de indução do modelo. Na fase de classificação, os valores dos vetores associados as palavras são consideradas para atribuir a categoria dos novos documentos (documentos de teste).

Como já comentado, o método proposto nesta seção, o algoritmo IMBHN (*Inductive Model Based on Bipartite Heterogeneous Network*), e toda a formulação do algoritmo aqui descrita, foi desenvolvido em colaboração com outros pesquisadores do grupo de pesquisa (ROSSI *et al.*, 2012; ROSSI *et al.*, 2014). A principal contribuição desta tese é o algoritmo PBG, que também utiliza-se de uma representação em grafo bipartido, entretanto, é importante notar que os procedimentos e os passos desses dois algoritmos são diferentes. Como será descrito nas próximas seções, o enquanto que o IMBHN baseia-se na diminuição do erro quadrático, o PBG baseia-se na otimização da divergência entre as informações de classes (ou tópicos) associados aos vértices.

3.2.1 Modelo de indução baseado em grafo bipartido

O algoritmo, chamado de *Inductive Model Based on Bipartite Heterogeneous Network* (IMBHN), induz um classificador por meio da modelagem da coleção de documentos textuais como um grafo bipartido.

Para a descrição do algoritmo, é utilizada a notação descrita na Seção 3.1. Considere o grafo bipartido $G = ((\mathscr{D}^l \cup \mathscr{W}), \mathscr{E}, f)$ e os vetores Y_j e B_i associados respectivamente aos vértices $d_j \in \mathscr{D}^l$ e $w_i \in \mathscr{W}$. Note que nesse algoritmo são considerados os vetores do conjunto $\mathscr{Y}$, que contêm informações de classes do conjunto de documentos já rotulados $\mathscr{D}^l$.

O objetivo do IMBHN é computar a influência das palavras presentes na coleção considerando a relação com os documentos rotulados, *i. e.*, induzir os valores dos vetores associados às palavras da coleção em relação as classes (ROSSI *et al.*, 2014). O processo de indução é

guiado pela minimização do erro quadrático, descrito na equação a seguir,

$$Q(G) = \frac{1}{2}\left(\sum_{j=1}^{m}\sum_{k=1}^{l}(class(\sum_{i=1}^{n_{d_j}} f_{j,i}B_{i,k}) - Y_{j,k})^2\right)$$
$$= \frac{1}{2}\left(\sum_{j=1}^{m}\sum_{k=1}^{l}[erro]_{j,k}^2\right), \tag{3.1}$$

onde

$$class(\sum_{i=1}^{n_{d_j}} f_{j,i}B_{i,k}) = \begin{cases} 1 & \text{se } c_k = \arg\max_{c_r \in \mathscr{C}}(\sum_{i=1}^{n_{d_j}} f_{j,i}B_{i,r}) \\ 0 & \text{caso contrário.} \end{cases} \tag{3.2}$$

O algoritmo IMBHN induz os valores associados aos vetores B_i que minimizam o erro quadrático ($[erro]^2$), *i.e.*, basicamente a soma dos quadrados da diferença entre os vetores preditos e os vetores com as classes reais dos documentos de treino. Com isso, o problema principal dessa modelagem é encontrar os vetores B_i's que minimizam a função $Q(G)$. Para resolver esse problema, pode-se utilizar o método de gradiente descendente. Com esse método, constrói-se um algoritmo iterativo onde, em cada iteração no momento $t+1$, são feitas atualizações nos valores do vetor B_i^t.

A equação de atualização do vetor B_i é a seguinte,

$$B_i^{(t+1)} = B_i^{(t)} + \eta\left[-\frac{\partial(Q(G))}{\partial B_i^{(t)}}\right]. \tag{3.3}$$

A direção do gradiente pode ser estimada pela seguinte derivação

$$\frac{\partial(Q(G))}{\partial B_i} = \sum_{k=1}^{K}\sum_{j=1}^{m} class(\sum_{i=1}^{n_{d_j}} f_{j,i}B_{i,k} - Y_{j,k}) \times \sum_{k=1}^{K}\sum_{j=1}^{m}\sum_{i=1}^{n_{d_j}} f_{j,i}$$
$$= \sum_{k=1}^{K}\sum_{j=1}^{m}[erro]_{j,k}^2 \times \sum_{k=1}^{K}\sum_{j=1}^{m}\sum_{i=1}^{n_{d_j}} f_{j,i}. \tag{3.4}$$

Substituindo a Equação 3.4 na Equação 3.5, o valor $B_{i,k}^{(t+1)}$ da palavra w_i para a classe $c_k \in \mathscr{B}$ é o seguinte:

$$B_{i,k}^{(t+1)} = B_{i,k}^{(t)} + (\eta \sum_{j=1}^{m} f_{j,i}[error]_{j,k}^{(t)}), \tag{3.5}$$

onde η é a taxa de correção, *i.e.*, a taxa com que o erro é considerado nas atualizações.

3.2.2 Descrição do Algoritmo IMBHN

Considerando as equações de atualização dos vetores B_i, o algoritmo IMBHN possui três procedimentos principais: (1) iniciação dos vetores, (2) cálculo do erro e (3) o ajuste dos valores nos vetores.

Na iniciação dos vetores B_i associado a uma palavra w_i, os valores atribuídos podem ser gerados aleatoriamente. Porém, é mais eficiente utilizar as informações de classes atribuída aos documentos para iniciar os vetores B_i's. Dessa forma, a iniciação é realizada da seguinte maneira:

$$B_{i,k}^{(0)} = \frac{\sum_{j=1}^{m} f_{j,i} Y_{j,k}}{\sum_{j=1}^{m} f_{j,i}}, \tag{3.6}$$

note que o valor de $Y_{j,k}$ é igual a 0 caso o documento d_j não seja atribuído a classe c_k.

No passo (2) é computado o erro. O erro na classificação de um documento d_j é obtido pela soma dos pesos das arestas conectadas a todo vértice d_j multiplicado pelo valor na posição k do vetor B_i menos a classe real Y_j, *i.e.*, $erro = \left(\sum_{i=1}^{n_{d_j}} f_{j,i} B_{i,k} - Y_{j,k}\right)$.

No passo (3), o ajuste dos valores dos vetores B_i. O ajuste é feito pelo cálculo da Equação 3.5. Note que o erro influencia a atualização dos vetores associados a cada palavra conectada ao documento.

Os passos (2) e (3) são repetidos para cada documento até o critério de parada ser alcançado. O critério de parada adotado foi um número máximo de iterações ou um limiar mínimo ε para o erro.

Na fase de classificação, os vetores associados as palavras são empregados na categorização de um novo documento. Isso é alcançado por meio do argumento máximo obtido da soma dos pesos das arestas que ligam o documento aos termos multiplicado pelo valor na posição k do vetor B_i (Equação 3.2).

Para validar o algoritmo IMBHN, no trabalho de Rossi *et al.* (2014) foram realizados experimentos com 35 coleções de documentos com diferentes características. Os resultados mostraram que o algoritmo é superior, com diferença estatísticas, aos algoritmos tradicionais e estado-da-arte.

3.3 Aprendizado não-supervisionado

No contexto não supervisionado, uma tarefa comum para a extração de conhecimento é o agrupamento. O objetivo dessa tarefa é agrupar objetos que representam documentos ou palavras de modo que objetos em um mesmo grupo sejam similares (com o mesmo tópico ou assunto), e objetos não similares pertençam a diferentes grupos. Normalmente, os algoritmos de agrupamento associam a cada objeto um único grupo. Porém, existem outros tipos de algoritmos de agrupamento capazes de associar a cada documento diferentes grupos com diferentes proporções. Essa associação provê uma descrição mais realista dos conceitos temáticos presentes em documentos textuais, isso porque um mesmo documento pode se referir a vários conceitos.

A tarefa de agrupamento se assemelha com o processo de redução de dimensionalidade. Na redução de dimensionalidade, o objetivo é transformar dados que estão representados por

vetores em um espaço com alta dimensionalidade para um espaço com menor dimensionalidade. Para isso, muitas das técnicas de redução de dimensionalidade agrupam dados próximos e, com base nesses grupos, definem novos vetores com menor dimensionalidade. Isso caracteriza um problema de otimização, onde os novos valores dos vetores são ajustados de forma que ainda possa existir a distinção dos grupos de dados. Esse é um princípio no qual se baseiam técnicas bem conhecidas como LSI (*Latent Dirichlet Indexing*) e NMF (*Nonnegative Matrix Factorization* – Definição 3).

O problema de extração de tópicos pode ser visto como um caso especial do problema de agrupamento. A observação das palavras que compõe um tópico deve revelar os assuntos tratados nos documentos relacionados com esse tópico. Já a tarefa de extração de tópicos, tem como objetivo analisar um grande conjunto de documentos não rotulados, realizar o agrupamento de forma que documentos semelhantes, ou sobre um mesmo assunto, estejam em um mesmo grupo e, por fim, extrair um conjunto de palavras que sumariza cada grupo de documentos (STEYVERS; GRIFFITHS, 2007).

Técnicas de redução de dimensionalidade como o LSI e o NMF foram aplicadas em diferentes domínios, e apresentaram bons resultados práticos no problema de extração de tópicos. Entretanto, por serem baseadas em decomposição de matrizes, possuem desvantagens como o alto consumo de memória e tempo computacional. Ainda mais no contexto de mineração de texto, onde as matrizes que descrevem coleções de documentos são esparsas. Para resolver esses problemas, Hofmann (1999) propôs o pLSI e que em seguida foi evoluído para os modernos modelos probabilísticos de tópicos.

Como foi discutido no capítulo anterior, modelos probabilísticos de tópicos tem um tratamento matemático rigoroso para descrição das operações efetuadas no processo de descoberta de tópicos. Na perspectiva de um desenvolvedor de aplicações práticas, criar um modelo generativo e derivá-lo a fim de obter um algoritmo de inferência implementável é uma tarefa difícil. Além disso, o rigor matemático desafia uma rápida exploração de novas suposições, heurísticas, ou adaptações que podem ser úteis em vários cenários reais. Para superar esse problema, neste trabalho é proposto uma abordagem simples e descritiva que utiliza documentos representados em grafo bipartido.

O algoritmo proposto, chamado *Propagation in Bipartite Graph* (PBG), trabalha de forma semelhante ao bem conhecido algoritmo de propagação de rótulos (ZHU; GHAHRAMANI, 2002), a diferença é que os rótulos são propagados dos documentos para as palavras e das palavras para os documentos, obedecendo a estrutura do grafo bipartido. Outra diferença é que o objeto não tem apenas um único rótulo, mas sim um vetor K-dimensional (K é o número de grupos) onde cada dimensão é a pertinência do objeto ao grupo. A ideia é associar esses vetores a cada vértice, e iterativamente propagar a informação de grupos contida nesses vetores para os vértices vizinhos. No contexto não supervisionado, esses vetores são iniciados aleatoriamente.

Neste trabalho foi conduzido uma análise comparativa experimental do algoritmo PBG

com os métodos LDA e NMF. Os resultados encontrados pelo algoritmo PBG foram semelhantes aos métodos comparados. Com a facilidade de descrição do algoritmo PBG e a representação em grafos, foi descrito uma versão paralela do PBG com um rápido tempo de convergência. Além disso, foi utilizado heurísticas de iniciação dos rótulos para melhorar a qualidade dos resultados.

3.3.1 Algoritmo de Propagação em Grafos Bipartidos

Nesta seção é apresentada a fundamentação computacional e a descrição da proposta do algoritmo não-supervisionado para a propagação de rótulos em grafos bipartidos. Em contraste ao tradicional algoritmo de propagação de rótulos, o algoritmo PBG não atribui um simples rótulo para cada vértice do grafo. O que são propagados no PBG são rótulos multidimensionais, que correspondem aos vetores A_j, B_i e $C_{e_{j,i}}$. E a propagação obedece a estrutura da rede bipartida.

O algoritmo PBG propaga os rótulos multidimensionais (vetores) para os vértices vizinhos. As iterações de propagações operam em dois procedimentos distintos: (1) propagação local, que correspondem as propagações realizadas para um vértice a partir de sua vizinhança, e (2) propagação global, que corresponde as propagações dos rótulos percorrendo toda a estrutura do grafo. A propagação global pode ser interpretada como a disseminação das informações obtidas na fase local para todos os vértices. O algoritmo PBG é sumarizado no Algoritmo 5. A propagação local é descrita no Algoritmo 6 e a propagação global é descrita no Algoritmo 7.

Algoritmo 5: Algoritmo PBG

Entrada:
Grafo bipartido G,
K // dimensões de A_j e B_i
α // parâmetro de concentração

Saída :
rótulos multidimensionais A_j para cada documento d_j,
rótulos multidimensionais B_i para cada termo w_i

```
1 início
2     Inicie os vetores A_j para cada documento d_j ∈ G;
3     Inicie os vetores B_i para cada palavra w_i ∈ G;
4     enquanto convergência dos vetores A_j e B_i faça
5         para cada d_j ∈ 𝒟 faça
6             repita
7                 A_j ← localPropag(G,d_j, A_j, B);
8             até A_j não converge;
9         B ← globalPropag(G,A,B);
```

As entradas para o procedimento de propagação (Algoritmo 5) são o número de tópicos K (o número de dimensões dos rótulos) e a representação da coleção de documentos como um grafo bipartido G. Primeiro, são iniciados aleatoriamente os vetores A_j e B_i associados respectivamente

aos vértices $d_j \in \mathscr{D}$ e $w_i \in \mathscr{W}$. Esses vetores devem ser iniciados de forma que $\sum_{k=1}^{K} A_{j,k} = 1$ para todos os documentos $d_j \in \mathscr{D}$, e $\sum_{w_i \in \mathscr{W}} B_{i,k} = 1$ para todos os tópicos k ($1 \leq k \leq K$).

Em seguida, a propagação local é realizada para cada aresta $e_{j,i}$ incidente a um vértice d_j (ilustrada, para o vértice d_1, na Figura 5). Nesse processo é criado em um vetor K-dimensional $C_{e_{j,i}}$ resultado do produto Hadamard entre os vetores A_j e B_i,

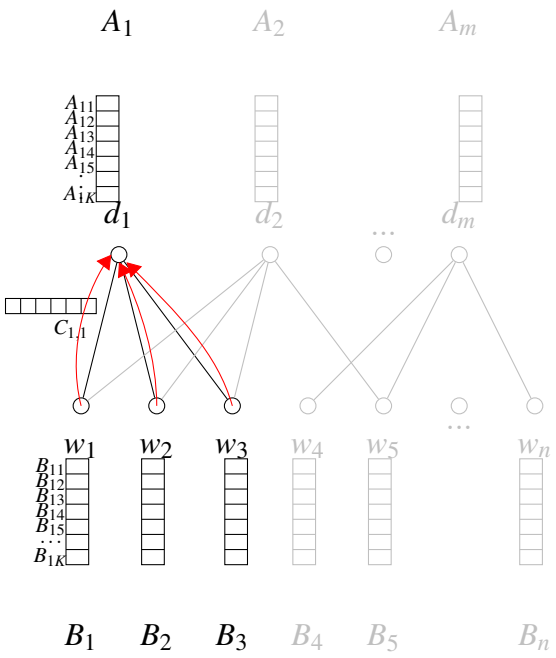
$$C_{e_{j,i}} = A_j \odot B_i. \tag{3.7}$$

O valor de $C_{e_{j,i}}$ é normalizado de forma que $\sum_{k=1}^{K} C_{e_{j,i},k} = 1$. Em seguida, os vetores $C_{e_{j,i}}$ associados as arestas $e_{j,i}$ incidentes ao vértice d_j são propagadas de volta para o vetor A_j

$$A_j = \alpha + \sum_i f_{j,i} C_{e_{j,i}}. \tag{3.8}$$

Esse processo de propagação deve ser repetido para cada vértice d_j enquanto os valores do vetor A_j se alterarem. O parâmetro α é usado para controlar o grau de concentração dos valores em A_j. O vetor A_j mantém as informações que relacionam documentos e tópicos, e caso o valor de α seja alto, esses valores estarão concentrados em poucos tópicos.

Figura 5 – Propagação local para o vértice d_1



Fonte: Elaborada pelo autor.

O processo de propagação global é realizado para todo vértice $w_i \in \mathscr{W}$, e para cada aresta $e_{j,i}$ incidente em w_i. Esse procedimento também cria um vetor K-dimensional $C_{e_{j,i}}$ dado pelo produto Hadamard de A_j e B_i (Equação 3.7). O array $C_{e_{j,i}}$ é normalizado de forma que $\sum_{k=1}^{K} C_{e_{j,i},k} = 1$ e os valores propagados de volta para B_i,

$$B_i = \sum_{d_j \in \mathscr{D}} f_{j,i} C_{e_{j,i}}, \tag{3.9}$$

Algoritmo 6: Propagação Local para PBG

```
1 function localPropag(G, d_j, A_j, B)
2     início
3         para cada aresta e_{j,i} incidente em d_j faça
4             C_{e_{j,i}} ← (A_j ⊙ B_i) / Σ_k (A_j ⊙ B_i)_k ;
5         A_j ← α + Σ_{w_i ∈ 𝒲_{d_j}} f_{j,i} C_{e_{j,i}} ;
6         retorne A_j;
```

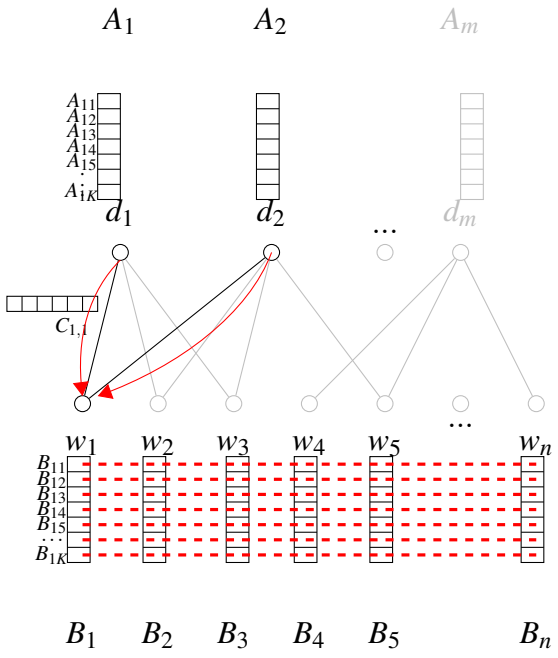
O vetor B_i é normalizado sobre todos os vértices $w_l \in \mathscr{W}$, *i.e.*

$$B_{i,k} = \frac{B_{i,k}}{\sum\limits_{w_l \in \mathscr{W}} B_{l,k}}. \tag{3.10}$$

Mesmo se o grafo bipartido for desconexo, as estatísticas nos vetores associados aos vértices do tipo palavras serão espalhadas para os vértices nos diversos componentes. Isso se deve a normalização realizada na Equação 3.10.

A Figura 6 ilustra o procedimento de propagação global para o vértice w_1 do grafo bipartido G.

Figura 6 – Propagação global para o vértice w_1



Fonte: Elaborada pelo autor.

A ideia por trás do algoritmo é localmente concentrar a influência de cada palavra w_i que ocorre em um documento d_j no array A_j. A influência ou relevância de uma palavra w_i para

Algoritmo 7: Propagação Global para PBG

função `globalPropag(`G, A, B`)`
  início
    para cada *vértice* $w_i \in \mathscr{W}$ **faça**
      para cada *aresta* $e_{j,i}$ *incident in* w_i **faça**
        $C_{e_{j,i}} \leftarrow \frac{(A_j \odot B_i)}{\sum_k (A_j \odot B_i)_k}$;
      $B_i \leftarrow \sum_{d_j \in \mathscr{D}} f_{j,i} C_{e_{j,i}}$;
    para cada *vértice* $w_i \in \mathscr{W}$ **faça**
      para $k = 1$ *to* K **faça**
        $B_{i,k} = \frac{B_{i,k}}{\sum_{w_{i^*} \in \mathscr{W}} B_{i^*,k}}$;
    retorne B;

o tópico k é dada pela k-ésima posição no array B_i. Então o algoritmo globalmente concentra a influência de todos os documentos que contêm a palavra w_i no vetor B_i. Quando o vetor B_i é atualizado, cada entrada k de 1 até K de B_i são normalizadas. Essa normalização dá a probabilidade da palavra w_i assumir o tópico k. Por exemplo, suponha que existam dois tópicos k_1 e k_2, e que o valor de B_{i,k_1} seja maior do que B_{i,k_2}. Assim, os documentos ligados a palavra w_i receberão alto valor para o índice k_1 e baixo valor para o índice k_2. Posteriormente esses valores serão propagados de volta para o vetor B_i. Quando o valor de B_i for normalizado de forma que $\sum_{w_i \in \mathscr{W}} B_{i,k} = 1$, o valor no índice k_1, que é maior, tende a aumentar, e o valor na posição k_2, que é menor, tende a diminuir. E isso é realizado até se alcançar a convergência.

3.3.2 Formulação do PBG

Os rótulos multidimensionais (ou vetores) associados a cada vértice mantêm a informação de tópicos (ou grupos). Assim, é importante que grupos de vértices altamente conexos tenham essas informações o mais similares possível, e por outro lado, grupos de vértices disjuntos tenham essas informações dissimilares. Partindo dessa ideia é possível formular um problema de otimização no qual o objetivo é maximizar a similaridade, ou minimizar a divergência, entre os rótulos multidimensionais relacionados.

No caso específico da modelagem da coleção de documentos em grafos bipartidos com vetores associados às arestas e aos vértices, o que se procura otimizar é a divergência entre os vetores resultante do produto Hadamard $A_j \odot B_i$ e o vetor $C_{e_{j,i}}$. A ideia é que quanto maior a frequência $f_{j,i}$ de uma palavra w_i em um documento d_j, maior deve ser a concordância entre as informações de tópicos contidos nos vetores $(A_j \odot B_i)$ e $C_{e_{j,i}}$. E como consequência, grupos de vetores associados a vértices altamente conexos terão informação de tópicos similares.

Para medir a similaridade entre os vetores é utilizada a divergência KL (veja a Definição 1) entre os valores de $f_{j,i}C_{e_{j,i}}$ e $A_j \odot B_i$,

$$KL((f_{j,i}C_{e_{j,i}})||(A_j \odot B_i)) = \sum_{k=1}^{K} KL(f_{j,i}C_{e_{j,i}})_k \log \frac{(f_{j,i}C_{e_{j,i}})_k}{(A_j \odot B_i)_k}. \tag{3.11}$$

Para facilitar as derivações, o problema de minimização da divergência KL é reescrito como um problema de maximização

$$\begin{aligned} \min KL((f_{j,i}C_{e_{j,i}})||(A_j \odot B_i)) &= -\max KL((f_{j,i}C_{e_{j,i}})||(A_j \odot B_i)) \\ &= \max \sum_l (f_{j,i}C_{e_{j,i}})_l \log \frac{(A_j \odot B_i)_l}{(f_{j,i}C_{e_{j,i}})_l}. \end{aligned} \tag{3.12}$$

Assim, pode-se definir a divergência entre $(A_j \odot B_i)$ e $C_{e_{j,i}}$ como:

$$Q_G(A,B,C) = \sum_{e_{j,i} \in \mathscr{E}} \left(f_{j,i}C_{e_{j,i}} \log \frac{A_j \odot B_i}{C_{e_{j,i}}} + \sum_{d_j \in \mathscr{D}} \mathscr{R}(A_j, \alpha) \right), \tag{3.13}$$

onde $\mathscr{R}(A_j, a)$ são termos regularizadores aplicados a cada documento d_j, e α é um valor constante que controla o valor de concentração do vetor A_j. Mas especificadamente, o termo regularizador é definido como

$$R(A_j, \alpha) = (\alpha - A_{j,k}) \log A_{j,k} + A_j(\log A_{j,k} - 1), \quad k = 1, \ldots, K \tag{3.14}$$

Os rótulos multidimensionais podem ser obtidos otimizando a Equação 3.13 para cada par de vértice ligados por uma aresta, isso dá origem a seguinte função de custo aplicada sobre o grafo G:

$$Q(G) = \underset{A^*,B^*,C^*}{\arg\max} \sum_{k=1}^{K} [Q_G(A,B,C)]_k. \tag{3.15}$$

O valor máximo de $Q(G)$ em relação a A_j, B_i e $C_{e_{j,i}}$ são determinados ajustando o gradiente para zero. Primeiramente, é maximizado a Equação 3.15 em relação a $C_{e_{j,i}}$. Observe que essa é uma otimização com restrição, desde que se define $\sum_{k=1}^{K} C_{j,i,k} = 1$, por isso é adicionado o apropriado multiplicador de Lagrange nas equações.

$$Q_{[C_{j,i}]} = \left(f_{j,i}C_{j,i} \log \left(\frac{A_j \odot B_i}{C_{e_{j,i}}} \right) + \lambda \left(\sum_{l=1}^{K} C_{e_{j,i},l} - 1 \right) \right), \tag{3.16}$$

onde os argumentos de Q foram removidos para simplificar a notação, e o subscrito $[C_{e_{j,i}}]$ denota que a equação apenas retêm os termos em Q que estão em função de $C_{e_{j,i}}$. Tomando as derivadas em relação a $C_{e_{j,i}}$, obtêm-se

$$\frac{\partial Q}{\partial C_{e_{j,i}}} = f_{j,i} \log (A_j \odot B_i) - f_{j,i} \log (C_{e_{j,i}}) - f_{j,i} + \lambda$$

$$= f_{j,i}\left(\log\left(A_j \odot B_i\right) - \log\left(C_{e_{j,i}}\right) - 1 + \frac{\lambda}{f_{j,i}}\right) \tag{3.17}$$

Definindo a derivada igual a zero alcança-se os valores de $C_{e_{j,i}}$ que maximizam Q,

$$C_{e_{j,i}} \propto A_j \odot B_i. \tag{3.18}$$

Como os valores dos vetores $C_{e_{j,i}}$ devem ser restringidos por definição de forma que $\sum_{e_{j,i}\in\mathscr{E}} C_{e_{j,i}=1}$, calcula-se

$$C_{e_{j,i}} = \frac{A_j \odot B_i}{\sum_{k=1}^{K}\left(A_j \odot B_i\right)_k}. \tag{3.19}$$

Em seguida, é maximizado a Equação 3.13 em relação a A_j, vetor associado a um documento $d_j \in \mathscr{D}$. Não é necessário usar o operador de Lagrange para restringir esse vetor por que ele já é restringido pelo termo regularizador. Os termos da equação que contêm A_j são:

$$\begin{aligned} Q_{[A_j]} &= \sum_{w_i\in\mathscr{W}_{d_j}} \left(f_{j,i}C_{e_{j,i}} \log A_j\right) + \mathscr{R}(A_j,\alpha) \\ &= \sum_{w_i\in\mathscr{W}_{d_j}} f_{j,i}C_{e_{j,i}} \log A_j + (\alpha - A_j)\log A_j - A_j(\log A_j - 1) \\ &= \log A_j \left(\sum_{w_i\in\mathscr{W}_{d_j}} f_{j,i}C_{e_{j,i}} - A_j + \alpha\right) - A_j(\log A_j - 1), \end{aligned} \tag{3.20}$$

onde o subconjunto $\mathscr{W}_{d_j}$ indica o conjunto de palavras que ocorrem no documento d_j.

Tomando a derivada em relação a A_j, tem-se

$$\frac{\partial Q_{[A_j]}}{\partial A_j} = \frac{1}{A_j}\left(\sum_{w_i\in\mathscr{W}_{d_j}} f_{j,i}C_{e_{j,i}} - A_{j,i} + \alpha\right). \tag{3.21}$$

Definindo essa equação igual a zero, obtêm-se o valor máximo em:

$$A_j = \alpha + \sum_{w_i\in\mathscr{W}_{d_j}} f_{j,i}C_{j,i}. \tag{3.22}$$

É importante notar que o procedimento de atualização do vetor A_j itera apenas sobre o subgrafo $G_{d_j} = (\mathscr{V} = \{\{d_j\} \cup \mathscr{W}_{d_j}\}, \mathscr{E})$, formado por arestas incidentes no vértice d_j. Essas operações são chamadas de propagação local.

Finalmente, é maximizado a Equação 3.13 em relação a B_i, vetor associado a palavra $w_i \in \mathscr{D}$. Para isso, é isolado os termos em função de B_i e adicionado o correspondente multiplicador de Lagrange.

$$Q_{[B_i]} = \sum_{k=1}^{K}\left(\sum_{j=1}^{\mathscr{D}} f_{j,i}C_{e_{j,i}} \log B_i + \lambda_k\left(\sum_{p=1}^{\mathscr{V}} B_{v,k} - 1\right)\right) \tag{3.23}$$

Tomando a derivada de $Q_{[B_i]}$ e ignorando a constante λ_k para se obter uma estimativa do gradiente

$$\frac{\partial Q_{[B_i]}}{\partial B_{i,k}} = \sum_{j=1}^{\mathscr{D}} \sum_{k=1}^{K} \frac{f_{j,i} C_{e_{j,i}}}{B_{j,i}} + \lambda_k \tag{3.24}$$

Definindo essa equação igual a zero, e resolvendo λ_k, tal que $\lambda_k = \sum_{j=1}^{D} \sum_{k^*=1}^{K} f_{j,i} C_{e_{j,i}}$. Desde que $\sum_{i=1}^{\mathscr{V}} B_{i,k} = 0$, pode-se ignorar λ_k para estimar o valor não normalizado de $B_{i,k}$

$$\hat{B}_{i,k} \propto \sum_{j=1}^{\mathscr{D}} \sum_{k=1}^{K} f_{j,i} C_{e_{j,i}}. \tag{3.25}$$

Normalizando o valor de $\hat{B}_{i,k}$ sobre todas as palavras w_p pertencentes no vocabulário, tem-se

$$B_i = \frac{\hat{B}_i}{\sum_{p \in \mathscr{W}} \hat{B}_p}. \tag{3.26}$$

Note que para obter B_i é necessário agregar todos os vetores A_j que contêm a palavra w_i, e então normalizar sobre todas as palavras do vocabulário. Essa atualização requer a iteração sobre toda a estrutura do grafo, visitando todos os vértices do tipo documento e palavra, por isso esse procedimento é denotado como propagação global.

As atualizações descritas nas equações 3.22 e 4.14, e que maximizam a Equação 3.13 são a base para os procedimentos realizados no algoritmo PBG. As propagações locais e globais são repetidas até alcançar a convergência, a qual é computacionalmente averiguada quando o valor obtido pela Equação 3.15 é menor que um dado limiar.

3.3.3 Análise comparativa entre PBG, NMF e LDA

Os procedimentos realizados pelo algoritmo PBG é diferente dos procedimentos realizados pelo algoritmo que resolve o NMF, e também diferente do algoritmo de inferência do modelo LDA. Entretanto, existem similaridades na estrutura da função objetivo quando comparado com o problema definido pelo NMF com divergência KL, e do problema de otimização definido pelo método de inferência variacional do LDA. Nesta seção são discutidos aspectos comparativos dessas diferentes abordagens.

Inicialmente, é explicitada pelo Teorema 2 a relação entre a função objetivo do PBG, Equação 3.13, com a função objetivo do problema estabelecido pelo NMF com divergência KL, Equação 2.54. Em decorrência dos teoremas 1 e 2, resulta-se no Corolário 1, que afirma que a função objetivo do algoritmo PBG é uma aproximação do ELBO (*Evidence Lower Bound)* do algoritmo de inferência variacional do LDA com hiper-parâmetros simétricos.

Teorema 2. Ao maximizar a função objetivo do algoritmo PBG, minimiza-se também a função objetivo do NMF com divergência KL mais o termo regularizador.

Demonstração. Inicialmente, é definida duas propriedades úteis nessa demonstração.

1. O algoritmo PBG assume por definição que $\sum_{k=1}^{K} C_{e_{j,i},k} = 1$, então $\sum_{k=1}^{K} f_{j,i} C_{e_{j,i},k} = f_{j,i}$ e $\sum_{w_i \in \mathscr{W}} B_{i,k} = 1$.

2. Sejam a_i e b_i números não negativos. Com a inequação da soma do logaritmo, tem-se que $\sum_{i=1}^{n} a_i \log \frac{a_i}{b_i} \leq a \log \frac{a}{b}$, onde $a = \sum a_i$ e $b = \sum b_i$.

O problema definido pelo NMF com divergência KL, correspondente a Equação 2.54, pode-se reescrito, de forma a manter a relação de proporcionalidade, da seguinte forma:

$$Q_{NMF-KL} = \sum_{j,i} f_{j,i} \log \frac{f_{j,i}}{(AB^T)_{j,i}} - \sum_{j,i} \left(f_{j,i} \log f_{j,i} + \sum_{k=1}^{K} A_{j,k} B_{i,k} \log \left(A_{j,k} B_{i,k} \right) \right) \tag{3.27}$$

Assim, pelas propriedades 1 e 2, pode-se delimitar um limite superior para a divergência KL como

$$\begin{aligned}
Q_{NMF-KL} &= \sum_{j,i} f_{j,i} \log \frac{f_{j,i}}{(AB^T)_{j,i}} - \sum_{j,i} \left(f_{j,i} \log f_{j,i} + \sum_{k=1}^{K} A_{j,k} B_{i,k} \log \left(A_{j,k} B_{i,k} \right) \right) \\
&= \sum_{j,i} \left(\sum_{k=1}^{K} f_{j,i} C_{e_{j,i},k} \right) \log \left(\frac{\sum_{k=1}^{K} f_{j,i} C_{e_{j,i},k}}{\sum_{k=1}^{K} A_{j,k} B_{i,k}} \right) \\
&\quad - \sum_{j,i} \left(f_{j,i} \log f_{j,i} + \sum_{k=1}^{K} A_{j,k} B_{i,k} \log \left(A_{j,k} B_{i,k} \right) \right) \\
&\leq \sum_{j,i} \sum_{k=1}^{K} \left(f_{j,i} C_{e_{j,i},k} \log \frac{f_{j,i} C_{e_{j,i},k}}{A_{j,k} B_{i,k}} \right) \\
&\quad - \sum_{j,i} \left(f_{j,i} \log f_{j,i} + \sum_{k=1}^{K} \mathscr{R}(A_j, \alpha) \right) \\
&= \sum_{e_{j,i} \in \mathscr{E}} \left(\sum_{k=1}^{K} f_{j,i} C_{e_{j,i},k} \left(\log \frac{C_{e_{j,i},k}}{A_{j,k} B_{i,k}} \right) + \sum_{k=1}^{K} \mathscr{R}(A_j, \alpha) \right).
\end{aligned} \tag{3.28}$$

Desde que $\alpha > 0$ e $\sum_i B_{i,k} = 1$, pode-se regularizar sobre apenas a variável A_j – o termo regularizador é descrito na Equação 3.14. Como a função $Q(G)$, definida na Equação 3.15, é limite superior de Q_{KL-NMF}, então

$$\begin{aligned}
&\arg\min_{A,B} \left(Q_{KL-NMF} \right) \\
&\quad \leq \arg\min_{A,B,C} \sum_{j,i} \left(\sum_{k=1}^{K} f_{j,i} \left(C_{e_{j,i},k} \left(\log \frac{C_{e_{j,i},k}}{A_{j,k} B_{i,k}} \right) + \mathscr{R}(A_j, \alpha) \right) \right) \\
&\quad = \arg\max_{A,B,C} \sum_{j,i} \left(\sum_{k=1}^{K} f_{j,i} C_{e_{j,i},k} \left(\log \frac{A_{j,k} B_{i,k}}{C_{e_{j,i},k}} \right) - \mathscr{R}(A_j, \alpha) \right) \\
&\quad = Q(G) \quad \text{(Equação 3.15)}
\end{aligned} \tag{3.29}$$

■ □

Corolário 1. A função objetivo do algoritmo PBG é uma aproximação do ELBO (*Evidence Lower Bound)* do algoritmo de inferência variacional do LDA com valores *a priori* simétricos.

Demonstração. Essa afirmação pode ser deduzida em decorrência imediada dos teoremas 1 e 2. ■ □

3.3.4 Melhorando o algoritmo PBG

O algoritmo PBG é uma solução aproximada para o problema de otimização estabelecida na Equação 3.13. Mas ao mesmo tempo, é um método heurístico baseado no procedimento de propagação de rótulos. Assim, aproveitando a interpretação do algoritmo como uma heurística de propagação de rótulos e a facilidade para manipulação dos dados representados como grafo bipartido, são apresentadas duas alterações no algoritmo com o objetivo de melhorá-lo tanto em desempenho quanto em qualidade dos resultados. A primeira melhoria está na geração da solução inicial. Buscando vantagem sobre a heurística de agrupamento dos vértices, é gerado rótulos iniciais já agrupados utilizando métodos rápidos de agrupamento. E a segunda melhoria aproveita a estrutura local do grafo para realizar processamento paralelizado.

3.3.4.1 Iniciação dos rótulos multidimensionais

Aqui é descrito o procedimento de iniciação dos vetores A_j associados a cada vértice do tipo documento $d_j \in \mathscr{D}$. Apenas os rótulos associados aos documentos foram iniciados porque é mais fácil criar agrupamento de documentos do que de palavras, e também por existir na literatura vários métodos eficientes para o agrupamento de documentos (STEINBACH; KARYPIS; KUMAR, 2000a; MOURA; REZENDE, 2010; FUNG; WANG; ESTER, 2003; ZHANG *et al.*, 2010; HAMMOUDA; KAMEL, 2003).

A iniciação baseada em agrupamento aplicado no algoritmo PBG pode ser descrita como: (1) computar o agrupamento π dos documentos em K grupos, tal que $\pi(d_j)$ é a associação de um documento d_j a um grupo indexado por k, *i.e.* $\pi(d_j) = k$ para $1 \leq k \leq K$; (2) atribua a k-ésima dimensão do vetor A_j o valor 1 se $\pi(d_j) = k$, 0 caso contrário; (3) propague os rótulos de A_j para os vetores B_i de cada palavra w_i conectada a um documento d_j no grafo bipartido; (4) use os rótulos A_j e B_i para iniciar o processo de propagação descrito pelo algoritmo PBG.

O agrupamento utilizado na iniciação pode ser alcançado por vários algoritmos de agrupamento, sendo preferível a utilização de métodos eficientes para não sobrecarregar o processo total. Note que é possível atribuir mais de um grupo no vetor A_j, desde que o algoritmo de agrupamento seja capaz de encontrar agrupamentos com sobreposição.

Nos experimentos realizados foram escolhidos os algoritmos de agrupamento K-means (STEINBACH; KARYPIS; KUMAR, 2000b) e *Hierarchical Link Clustering* (HLC) (BODLAJ; BATAGELJ, 2015), algoritmos conhecidos e que utilizam a representação de documentos como *bag-of-words* e grafos, respectivamente. Para usar do K-means, a coleção de documentos foi

representada na forma vetorial, com cada dimensão correspondente a uma palavra do vocabulário. A similaridade dos documentos foi calculada pela distância cosseno. Para usar o algoritmo HLC foi criado um grafo homogêneo com os vértices representando cada documento, e as arestas ligam os R documentos mais próximos. Os R vizinhos mais próximos de um documento é obtido por meio do cálculo da distância cosseno entre os pares de documentos. O HLC também foi escolhido por ser um algoritmo baseado em grafos e utilizar uma abordagem gulosa, o que provê um método computacionalmente eficiente.

3.3.4.2 Paralelização do algoritmo PBG

Outra melhoria no algoritmo PBG é o uso de técnicas de computação paralela. Ao usar a representação em grafos, é fácil dividir o conjunto de dados em subgrafos e aplicar o mesmo algoritmo nesses subgrafos. Assim, pode-se dividir o conjunto de documentos em t subconjuntos tal que $\mathscr{D} = \{\mathscr{D}_1 \cup \ldots \cup \mathscr{D}_t\}$. Cada subconjunto $\mathscr{D}_r$ induz um subgrafo G_r. Como as propagações locais utilizam apenas uma subestrutura do grafo, é possível aplicar as propagações locais para cada subgrafo G_r. Cada *thread* executa a propagação local para obter o melhor conjunto de vetores associados ao subconjunto de documentos. A propagação global não pode ser dividida pois requer a iteração sobre toda a estrutura do grafo, então ela é realizada após a finalização de cada *thread*. Os resultados de cada *thread* são unidos para obter os resultados da estrutura global do grafo. Apesar de onerosa, a propagação global percorre apenas uma vez toda a estrutura do grafo. A versão paralela do algoritmo PBG é descrito no Algoritmo 8

Algoritmo 8: Algoritmo PBG paralelo

Entrada:
grafo bipartido G,
K // dimensão dos vetores A_j e B_i
α // parâmetro de concentração
t // número de *threads*

Saída :
rótulos multidimensionais A_j para cada documento d_j,
rótulos multidimensionais B_i para cada palavra w_i

```
1  início
2      Inicia os vetores A_j para cada documento d_j ∈ G;
3      Inicia os vetores B_i para cada palavra w_i ∈ G;
4      enquanto convergence faça
5          // divide o conjunto de documentos 𝒟 em t subconjuntos
6          𝒟 ← {𝒟_1, ..., 𝒟_t} ;
7          foreach 𝒟_p ∈ 𝒟 do
8              seja 𝒜_p o conjunto de vetores associados aos documentos em 𝒟_p.
9              grafo induzido G_p dos documentos em 𝒜_p ;
10             executa a thread localPropag(G_p, 𝒟_p, 𝒜_p, B);
11         espera as threads ;
12         B ← globalPropag(G,A,B);
```

3.3.5 Resultados Experimentais

Os vetores A_j e B_i podem ser pós-processados de acordo com diferentes sub-tarefas relacionadas ao contexto não supervisionado. Nesta seção são apresentados os resultados experimentados do algoritmo PBG na tarefa de redução de dimensionalidade e extração de tópicos. Os experimentos foram conduzidos usando o conjunto de dados descritos na Tabela 2 (ROSSI; MARCACINI; REZENDE, 2013). Foram selecionados da coleção descrita em (ROSSI; MARCACINI; REZENDE, 2013) os três maiores conjuntos de dados e com melhor aplicabilidade na tarefa de extração de tópicos. Esses documentos foram pré-processados, *stop-words* foram removidas e termos foram *stemizados* por meio do algoritmo de Porter (PORTER, 1997). A frequência das palavras foram usadas como peso das arestas do grafo bipartido.

Tabela 2 – Conjuntos de dados usados na avaliação experimental. A primeira coluna é o número de documentos, a segunda coluna é o número de palavras únicas, e a última coluna é o número total de termos.

nome	m	n	$\bar{n}$	# classes
20ng	18808	45434	76.47	20
Dmoz-Business	18500	8303	11.93	37
classic4	7095	7749	35.28	4

Fonte: Dados da pesquisa.

O grafo bipartido foi criado para cada conjunto de dados e o algoritmo PBG foi executado para demonstrar sua habilidade em extrair padrões temáticos no conjunto de documentos. Os resultados foram as atribuições dos vetores A_j e B_i que otimizam a Equação 3.13.

Apesar dos documentos utilizados nessa avaliação experimental possuírem rótulos, eles foram ocultados na construção do grafo e no processo de propagação. Os rótulos foram usados depois apenas para auxiliar na avaliação dos resultados na representatividade dos documentos.

Foram avaliadas três versões do algoritmo PBG: inicialização do PBG com o algoritmo K-means (*kmeans+pbg*), inicialização do PBG com o algoritmo HLC (hlc+pbg), e a versão *multithreading* do algoritmo PBG (multi-pbg). Os resultados do PBG foram comparados com o NMF inicializado com o método *Singular Value Decomposition* (SVD) e o LDA nas tarefas de extração de tópicos e representatividade dos documentos. Para isso, duas métricas foram usadas, *Normalized Pointwise Mutual Information* (NPMI – veja a Definição 2) e acurácia de classificação dos documentos.

Na tarefa de extração de tópicos, foram definidos os número de tópicos K com o conjunto $K = \{50, 100, 150, 200\}$ para todos os algoritmos. Foi utilizada uma implementação em *python* do LDA [1] com método de inferência variacional e com processo de otimização dos hiperparâmetros. Os hiperparâmetros do LDA foram inicialmente $\alpha = \frac{1}{K}$ e $\beta = \frac{1}{n}$ com o LDA com *priori* simétricas.

[1] A implementação em *python* do método de inferência variacional do LDA é disponível em <https://github.com/kzhai/PyLDA>

O problema estabelecido pelo NMF foi resolvido utilizando o método de projeção de gradiente[2] (LIN, 2007). Para o algoritmo PBG, foram realizados experimentos iniciais com apenas 10 iterações para definir o melhor parâmetro sobre o conjunto $\alpha \in \{0.5, 0.05, 0.005\}$. O único parâmetro ($\alpha$) exigido pelo algoritmo PBG pode ser definido quando existe o conhecimento sobre a coleção, isto é, se os documentos são mais focados em um ou em vários tópicos, e portanto, pode-se utilizar um valor padrão indicado. O melhor parâmetro para todos os conjunto de dados foi $\alpha = 0.05$. O critério de parada foi definido para 100 iterações ou após 5 horas de execução para todos os algoritmos. Para a versão com iniciação dos rótulos, os documentos foram inicialmente agrupados utilizando os algoritmos *K*-means e HLC. Já na versão paralela, cada execução foi dividida em 8 *threads*.

A criação de um processo estritamente justo de comparação entre o algoritmo proposto neste trabalho e os algoritmos baseados em modelos probabilísticos é difícil, uma vez que os modelos probabilísticos tradicionalmente utilizam métricas de avaliação baseada na verossimilhança do modelo ou na perplexidade. Além de que o valor retornado pela verossimilhança (ou a perplexidade) não necessariamente está correlacionado com o julgamento humano sobre a qualidade de um conjunto de tópicos. Por isso, é utilizada a medida NPMI (*Normalized Pointwise Mutual Information*), que se aproxima da avaliação humana dos tópicos. E para avaliar a representatividade dos documentos é utilizado a medida de acurácia de classificação, sendo que os documentos são representado pelos vetores associados aos vértices do tipo documento.

3.3.5.1 Convergência

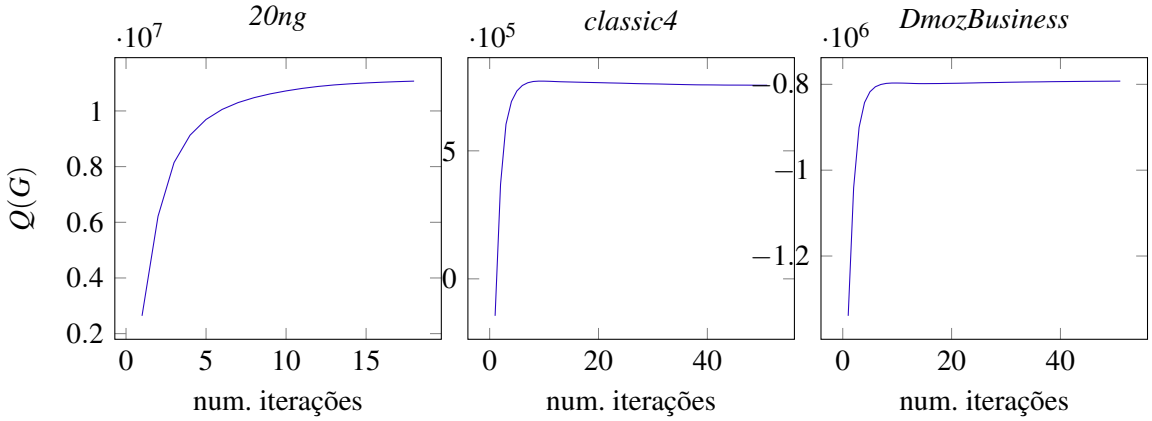
Na falta de qualquer prova geral que garanta a convergência do algoritmo PBG, foi necessário conduzir uma análise experimental para indicar a convergência do algoritmo. Na Figura 7 é mostrada a convergência do algoritmo PBG em relação ao valor da função objetivo definida na Equação 3.13. Para criar a Figura 7 foi calculado $Q(G)$, onde G é o grafo bipartido com seus respectivos vetores associados e obtidos após 50 iterações de propagações locais e globais ou no máximo 2 horas de execuções. Como pode ser notado na Figura 7, o algoritmo PBG não completou as 50 iterações para o conjunto de dados *20ng*, porém ele alcançou um estado de convergência em apenas 15 iterações. O mesmo indicativo de convergência foi percebido nos outros conjuntos de dados em aproximadamente 15 iterações.

3.3.5.2 Avaliação da representatividade dos documentos

Nesta seção é avaliada a representatividade dos vetores A_j associado aos vértices do tipo documento para representação das características dos documentos. Isso é feito da mesma forma que a distribuição de documentos por tópicos do LDA e a matriz A do NMF são usados como vetor de característica em uma tarefa de redução de dimensionalidade. O algoritmo de

[2] A implementação em *python* do NMF está disponível em <https://www.csie.ntu.edu.tw/~cjlin/nmf>

Figura 7 – Valor da função objetivo (Equação 3.13) por iterações do algoritmo PBG para os conjuntos de documentos *20ng* (esquerda), *classic4* (centro) e *Dmoz-Business* (direita).



Fonte: Dados da pesquisa.

classificação *Support Vector Machine* (SVM) [3] com parâmetros usuais foi usado em um esquema de validação cruzada para mensurar a acurácia desses vetores de característica na representação da coleção de documentos.

A acurácia de classificação é a porcentagem de acerto de um modelo de classificação aprendido do conjunto de documentos de treino aplicado em um conjunto de teste. Um alto valor de acurácia significa que o vetor de característica captura melhor o conhecimento necessário para representação do seu conteúdo.

As tabelas 3, 4 e 5 mostram respectivamente a melhor acurácia para os conjuntos de dados *20ng*, *classic4* e *Dmoz-Business*. O algoritmo PBG captura características dos documentos melhor do que o LDA em 100 iterações. Os resultados indicam que o algoritmo proposto é um método promissor na exploração da tarefa de redução de dimensionalidade. Principalmente quando o problema possuir fácil representação em grafo e ser possível a inclusão de conhecimento heurístico que possa enriquecer o processo e para obter melhores resultados.

Tabela 3 – Conjunto de documentos *20ng*. Melhores valores de acurácia obtidos pelos algoritmos utilizados nos experimentos.

	$K = 50$	$K = 100$	$K = 150$	$K = 200$
Algoritmo	**Acurácia** (%)	**Acurácia** (%)	**Acurácia** (%)	**Acurácia** (%)
pbg	70.7776	72.4559	71.5424	72.8224
kmeans+pbg	**77.5016**	**77.6184**	**77.6609**	77.0130
hcl+pbg	72.8808	76.7102	76.8271	77.2042
pbg-threads	72.5409	74.6123	75.0106	75.4461
lda	69.7100	70.1137	70.3845	71.2609
svd+nmf	73.5129	76.4075	77.1829	**77.5760**

Fonte: Dados da pesquisa.

[3] Weka 3: Software de Mineração de Dados em Java <http://www.cs.waikato.ac.nz/ml/weka/>

Tabela 4 – Conjunto de documentos *classic4*. Melhores valores de acurácia obtidos pelos algoritmos utilizados nos experimentos.

	$K = 50$	$K = 100$	$K = 150$	$K = 200$
Algoritmo	**Acurácia** (%)	**Acurácia** (%)	**Acurácia** (%)	**Acurácia** (%)
pbg	94.7005	95.0810	95.2784	95.6025
kmeans+pbg	**95.7858**	**95.9267**	**96.0395**	**96.2086**
hcl+pbg	95.3347	95.7153	95.4052	95.4898
pbg-threads	94.7710	95.0106	95.2502	95.5039
lda	94.1508	94.5736	94.1931	94.4045
svd+nmf	94.4891	95.1656	95.3206	95.7153

Fonte: Dados da pesquisa.

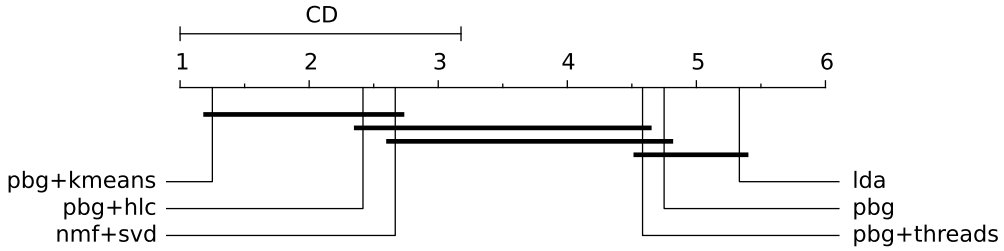
Tabela 5 – Conjunto de documentos *Dmoz-Business Dataset*. Melhores valores de acurácia obtidos pelos algoritmos utilizados nos experimentos

	$K = 50$	$K = 100$	$K = 150$	$K = 200$
Algoritmo	**Acurácia** (%)	**Acurácia** (%)	**Acurácia** (%)	**Acurácia** (%)
pbg	35.8919	43.5946	45.3351	48.7135
kmeans+pbg	**45.7189**	**54.3243**	**57.1838**	**57.8541**
hcl+pbg	46.8000	50.6649	53.4108	54.7081
pbg-threads	35.6649	41.6649	45.8649	48.8973
lda	38.4919	44.8432	48.0703	49.5459
svd+nmf	41.7459	48.6432	52.7892	55.2865

Fonte: Dados da pesquisa.

Na Figura 8 está o diagrama de diferença crítica para ilustrar os resultados do teste de significância estatística. Nesse diagrama, os algoritmos conectados por uma linha não apresentam significância estatística entre si. De acordo com a Figura 8, algoritmos com inicialização heurística são superiores e apresentam diferença estatística em relação ao LDA. Esse resultado valida a hipótese de que a inclusão de heurísticas no processo de propagação pode levar a melhores resultados.

Figura 8 – Diagrama de diferença crítica considerando as melhores acurácias para cada algoritmo.



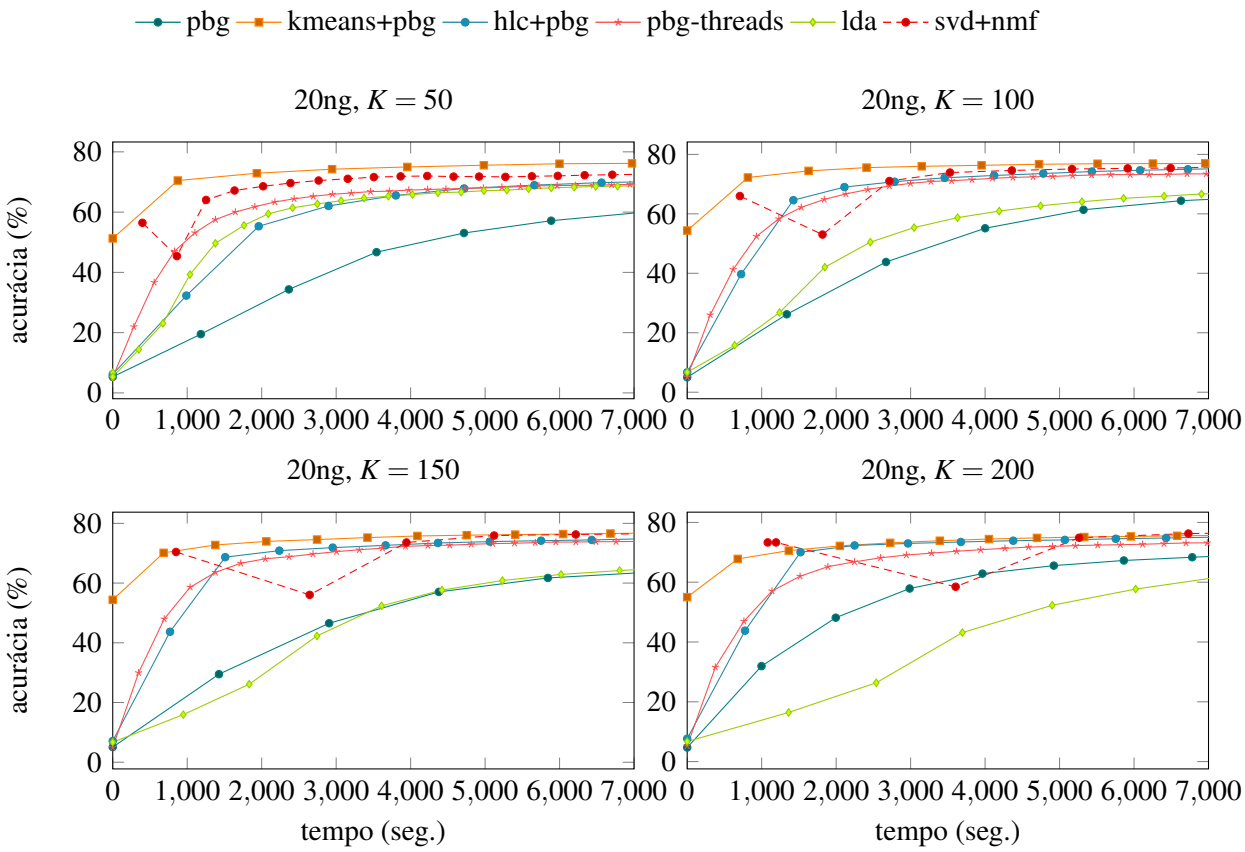
Fonte: Dados da pesquisa.

As figuras 9, 10 e 11 ilustram a acurácia por tempo na execução de todos os algoritmos dessa avaliação experimental nos conjuntos de dados *20ng*, *classic4* e *Dmoz-Business*. Cada

série ilustrada nesses gráficos são associados a um algoritmo específico. Para cada algoritmo é mantido o tempo (em segundos) e a acurácia obtida até o fim de cada iteração. Assim, o intervalo entre as marcações de tempo indicam o tempo gasto para convergência de uma iteração.

Analisando os gráficos das figuras 9, 10 e 11 observa-se que as marcações iniciais de tempo do algoritmo *svd+nmf* são melhores do que qualquer outro algoritmo. Isso porque a operação de SVD consome tempo significante e muito processamento. Por outro lado, a acurácia inicial obtida pelas simples aplicação do SVD é melhor do que a do LDA e do PBG. Da mesma forma, a acurácia obtida pelo K-means são melhores do que LDA e do PBG. Entretanto, o K-means é mais rápido do que o SVD. O vetor inicial obtido pelo algoritmo HCL não é bom suficiente para representar os documentos, por isso foi obtido baixos valores de acurácia nas iterações iniciais. Porém, mesmo com valores iniciais ruins, o algoritmo *hlc+pbg* foi capaz de obter bons valores de acurácia no final das execuções. Note que o PBG utilizou os indicativos de grupos inicialmente obtidos pelo HLC para melhorar os resultados, obtendo resultados significativamente melhores do que o LDA em todos os conjuntos de dados.

Figura 9 – Acurácias de classificação obtidas ao longo das execuções dos algoritmos utilizados nos experimentos. Os vetores de características extraídos pelos algoritmos foram usados para a representação da coleção *20ng*.

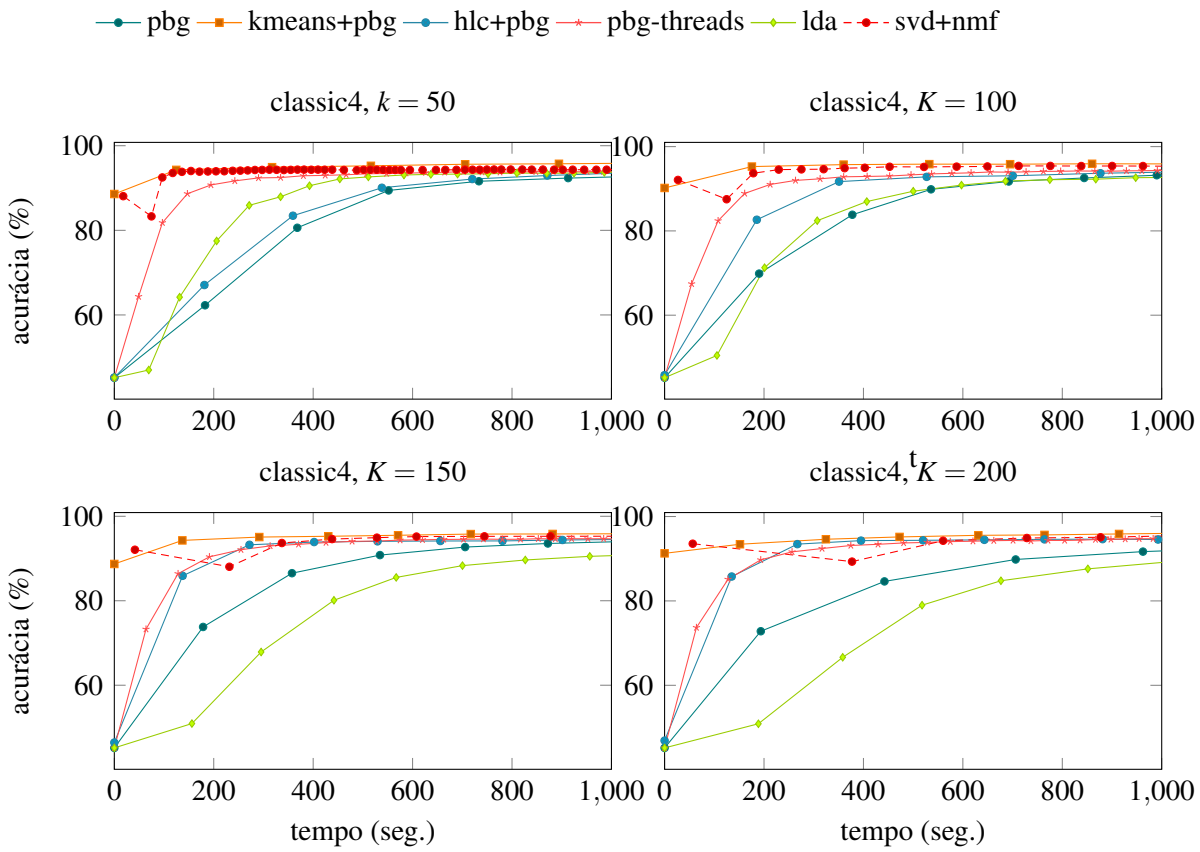


Fonte: Dados da pesquisa.

Analisando as figuras 9, 10 e 11 nota-se que o LDA converge mais rápido para $K = 50$, mas para valores maiores de K o LDA possui o tempo de convergência similar ou menor. Já a

versão *multithreading* do algoritmo PBG, como esperado, teve rápida convergência. Em alguns conjuntos de dados, como no *classic4* e *Dmoz-Business*, o algoritmo *thread-pbg* foi capaz de completar todas as 100 iterações, sendo que outros algoritmos completaram no máximo apenas 20 iterações no mesmo intervalo de tempo.

Figura 10 – Acurácias de classificação obtidas ao longo das execuções dos algoritmos utilizados nos experimentos. Os vetores de características extraídos pelos algoritmos foram usados para a representação da coleção *classic4*.



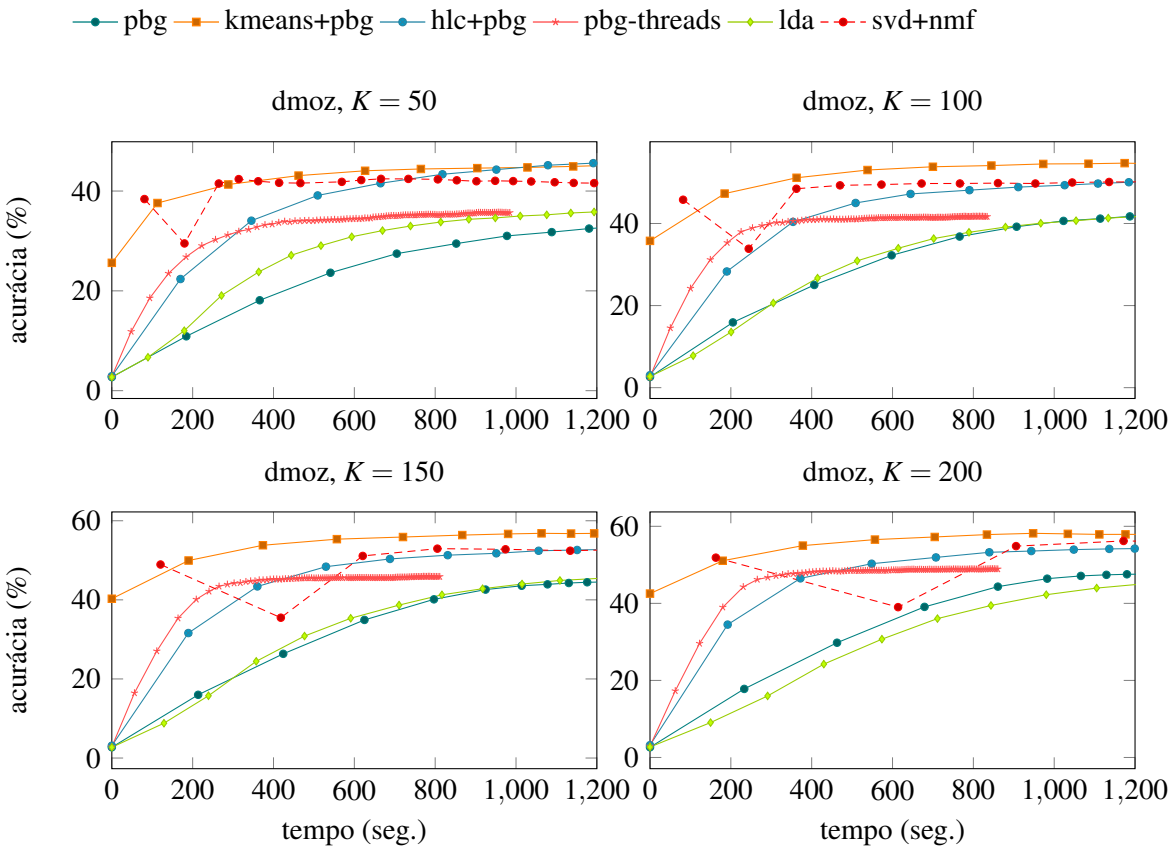
Fonte: Dados da pesquisa.

Por fim, pode-se concluir que esses experimentos indicam que mesmo a inclusão de heurísticas simples, como a iniciação baseada em agrupamento, pode ser útil para melhorar o algoritmo PBG na tarefa de extração de características. Além disso, percebe-se que o algoritmo de propagação, em cada nova iteração, obtêm resultados melhores até alcançar convergência.

3.3.5.3 *Avaliação dos tópicos utilizando NPMI (Normalized Pointwise Mutual Information)*

Como já comentado, um tópico é um conjunto de palavras que ocorrem frequentemente em documentos semanticamente relacionados e que podem ser usadas para descrever o assunto ou tema que os documentos tratam. Aqui, um tópico k é o conjunto de palavras top_k^L formada pelas L palavras melhores ranqueadas na distribuição de tópicos por palavras. No caso do PBG,

Figura 11 – Acurácias de classificação obtidas ao longo das execuções dos algoritmos utilizados nos experimentos. Os vetores de características extraídos pelos algoritmos foram usados para a representação da coleção *Dmoz-Business*.



Fonte: Dados da pesquisa.

esse conjunto de palavras é extraída da seguinte forma

$$top_k^L = \underset{top_k^{L*} \subset \mathcal{W}}{\arg\max} \sum_{w_i \in top_k^{L*}} B_{i,k} \tag{3.30}$$

tal que $|top_k^L| = L$.

Para o LDA, um tópico é a lista das $L = 10$ palavras mais prováveis extraída da distribuição ϕ. Da mesma forma, nos métodos NMF e PBG, um tópico k é o conjunto das $L = 10$ palavras do conjunto top_k^L.

Vários estudos foram feitos com o objetivo de encontrar medidas para quantificar a coerência dos tópicos e sua interpretação semântica (CHANG *et al.*, 2009; NEWMAN *et al.*, 2010; LAU; NEWMAN; BALDWIN, 2014). Nesses trabalhos podem ser encontradas avaliações experimentais mais detalhadas dos tópicos encontrados pelos métodos NMF e LDA. Aqui, o objetivo é comparar a qualidade dos tópicos obtidos pelo PBG com aqueles obtidos pelo LDA e NMF.

Para avaliar os tópicos, é usada a medida NPMI. Essa medida é baseada na associação entre pares de palavras usando dados externos (NEWMAN *et al.*, 2010; LAU; NEWMAN;

BALDWIN, 2014). Isso é correlacionado com a percepção humana sobre a coerência de um tópico e indica o quão bem as palavras se correlacionam no conjunto para descrever um tema. Seguindo o método de Newman *et al.* (2010), foram extraídas as correlações entre palavras de um conjunto de documentos da Wikipedia[4], e contada a frequência dos pares de palavras. Quanto maior a frequência das combinações dos pares de palavras em top_k^l, mais coerente será o tópico k.

Tabela 6 – Valores da media NPMI obtidos pelos algoritmos utilizados nesses experimentos. Cada conjunto de dados é seguido pelo número de tópicos em parênteses.

conj. dados (n. tópicos)	hlc+pbg	kmeans+pbg	lda	svd+nmf	pbg	pbg-thread
20ng (50)	0.1820	0.1680	0.1830	**0.2090**	0.1870	0.1820
20ng (100)	0.1870	0.1705	0.1685	**0.1975**	0.1725	0.1800
20ng (150)	0.1620	0.1713	0.1567	**0.1897**	0.1590	0.1810
20ng (200)	0.1533	0.1728	0.1512	0.1623	0.1613	**0.1733**
Dmoz-Business (50)	0.1250	0.1430	**0.1500**	0.1440	0.1100	0.1200
Dmoz-Business (100)	0.1245	**0.1535**	0.1195	0.1495	0.1010	0.1095
Dmoz-Business (150)	0.1247	**0.1467**	0.1210	0.1460	0.1090	0.1120
Dmoz-Business (200)	0.1212	0.1473	0.1182	**0.1520**	0.0950	0.1030
classic4 (50)	0.1690	0.1740	0.1700	**0.1860**	0.1790	0.1720
classic4 (100)	0.1760	0.1850	0.1885	0.1890	**0.1900**	0.1805
classic4 (150)	0.1633	0.1763	0.1773	**0.1797**	0.1713	0.1747
classic4 (200)	0.1730	**0.1818**	0.1765	0.1650	0.1642	0.1640

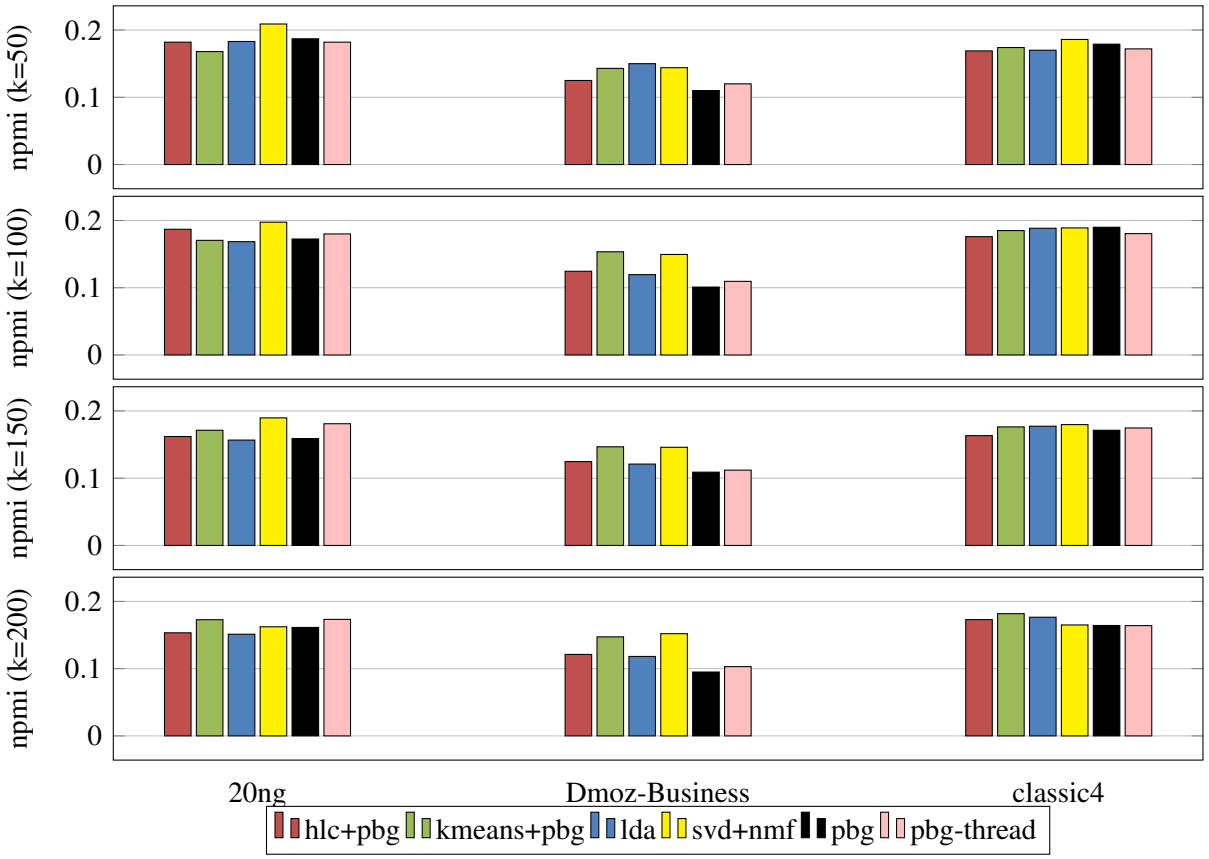
Fonte: Dados da pesquisa.

Na Tabela 6 são apresentados os resultados do NPMI. Os resultados indicam que o NMF obtém tópicos mais coerentes do que LDA. Esses resultados são ilustrados na Figura 12, onde é apresentado um grafo de barras dos valores de NPMI para todos os algoritmos e agrupando os resultados por conjunto de dados e número de tópicos. Note que o NMF, para todos os conjuntos de documentos e números de tópicos, apresentou os maiores valores de NPMI. Apesar disso, o PBG (e suas variações) obtiveram bons resultados, alcançando melhores valores de NMPI em alguns conjuntos de dados. Esses resultados indicam a viabilidade do PBG como um método competitivo, e uma nova técnica promissora na tarefa de extração de tópicos.

No Quadro 1 estão as listas com os 20 melhores tópicos entre os 50 tópicos ($K = 50$) encontrados no conjunto de dados 20*ng* para cada algoritmo utilizado nesta análise experimental. Esse conjunto de dados é formado por notícias categorizadas em 20 classes distintas. Os assuntos abordados nesse conjunto de notícias referem-se a computação, discursos políticos, religião, ciência, medicina, esporte e vários outros. Uma análise mais precisa sobre a qualidade desses tópicos deveria ser realizada por um especialista humano, apesar disso, é possível observar a relação entre as palavras de cada tópico e perceber o assunto no qual ele se trata. A lista com os tópicos encontrados nos conjuntos de dados *Dmoz-Business* e *classic4* é descrita no Apêndice A.

[4] Foram utilizados documentos da Wikipedia do ano de 2008. Esses documentos estão disponíveis livremente no site <https://dumps.wikimedia.org/>.

Figura 12 – Gráfico em barras dos valores de NPMI para todos os algoritmos com diferentes número de tópicos e conjuntos de dados.



Fonte: Dados da pesquisa.

3.3.6 Considerações finais

Nesta seção foi apresentado o algoritmo PBG, um algoritmo não supervisionado baseado na propagação de rótulos e que utiliza a representação da coleção de documentos em grafos bipartidos. A representação em grafos bipartidos oferece várias vantagens para a descrição de procedimentos não supervisionados para a extração de conhecimento de coleções textuais. A principal vantagem enfatizada neste trabalho está na facilidade das descrições das propagações do algoritmo, e como representar a informação textual usando a estrutura em grafos para a inclusão de heurísticas. Incluindo heurística de iniciação dos dados e de paralelização foi possível melhorar a qualidade dos resultados e o desempenho do algoritmo. Os experimentos usando coleções de documentos com diferentes características demonstraram que o algoritmo PBG ultrapassa o LDA no processo de extração de tópicos nas coleções utilizadas nos experimentos.

3.4 Aprendizado semissupervisionado

A tarefa de classificação de texto é normalmente realizada por um algoritmo que induz um modelo de classificação e que classifica novos documentos. Um número considerável de

Quadro 1 – Lista dos 20 melhores tópicos encontrados no conjunto de dados 20*ng* para cada algoritmo utilizado nesta análise experimental.

lda	**svd+nmf**
space orbit launch earth satellit moon system planet probe solar	muslim bosnian serb war bosnia croat europ yugoslavia serbian nazi
christian jesu god church christ bibl faith time sin dai	armenian turkish armenia turk soviet serdar argic turkei genocid extermin
team game hockei plai player year subject season nhl playoff	team hockei fan plai playoff nhl leaf cup buffalo bruin
medic diseas food patient doctor effect studi health cancer treatment	christian jesu church christ religion bibl faith homosexu paul belief
god bibl christian homosexu jesu love lord sin hell christ	mhz simm speed chip motherboard cpu ram board clock memori
armenian turkish armenia peopl turk turkei kill soviet genocid russian	imag bit gif format jpeg data displai tiff process viewer
jew arab muslim israel jewish war isra state countri nazi	israel isra arab jew palestinian jewish peac kill adam attack
drive do window disk scsi driver system problem card control	disk do copi floppi program drive instal hard mac softwar
file imag jpeg color format gif bit program version convert	player team year hit pitch win run basebal plai pitcher
game year hit player run basebal win team time pitch	gun firearm weapon crime law crimin control polic handgun arm
insur monei pai tax cost privat health columbia care system	drive scsi id hard control seagat floppi bu devic meg
graphic ftp imag pub data softwar program anonym packag mail	god sin exist jesu bibl atheist faith love hell heaven
homosexu write brian gai cramer sex articl men subject sexual	file format gif bmp convert directori ftp program postscript zip
card mac monitor appl subject video mhz board modem bit	font printer print truetyp laser charact atm deskjet problem postscript
kei encrypt chip clipper secur govern system escrow subject public	card driver video diamond bu ati vlb vga isa mode
window run program applic server subject displai widget manag sun	fbi fire batf koresh waco atf compound burn children stratu
gun fire fbi koresh write batf articl waco subject children	space orbit pat shuttl mission digex launch access hst moon
peopl don exist reason point thing evid question claim argument	system comput unix problem gatewai tape softwar user control work
presid clinton stephanopoulo vote work myer job senat don question	graphic program code softwar widget motif packag librari server sun
pbg	**pbg-thread**
space nasa gov orbit launch earth mission moon satellit shuttl	god jesu christ sin lord bibl christian church word heaven
team plai player year season game leagu hockei defens score	israel jew isra arab jewish muslim palestinian peac war kill
medic diseas patient doctor health treatment cancer studi infect effect	medic diseas patient health doctor effect treatment cancer infect studi
drive scsi disk control hard id floppi pin system bu	space gov nasa launch shuttl orbit satellit mission station pat
armenian turkish armenia turk peopl soviet russian genocid azerbaijan villag	earth henri moon toronto planet space orbit mar solar atmospher
imag data comput printer print graphic fax softwar packag program	max armenian turkish turk armenia turkei greek muslim genocid soviet
god jesu christ sin love lord man life heaven hell	god christian exist religion belief life atheist peopl faith truth
window do program softwar microsoft subject copi run disk system	imag color bit file jpeg displai gif format program convert
file email system server user font inform directori anonym run	window do run driver file microsoft printer print program font
jew war jewish muslim arab countri nazi israel nation state	drive disk hard control system tape sale floppi subject scsi
fire fbi koresh batf waco write burn atf ga compound	fire fbi koresh batf waco burn children atf compound ga
gun law weapon crime firearm peopl arm control polic crimin	card scsi video bit mhz bu board chip memori speed
card monitor subject mhz video chip ram speed appl cpu	file ftp graphic pub softwar packag data system site directori
file imag format jpeg ftp list gif graphic pub site	game plai goal score win shot boston espn subject team
christian church paul law homosexu god bibl peopl word jesu	car speed front road tire drive radar brake detector shift
monei cost year tax pai job work spend program fund	law govern right peopl state legal court case power polit
kei encrypt chip clipper secur escrow phone subject algorithm system	game year hit player basebal run pitch team win subject
game fan subject hockei playoff team cup win leaf wing	technolog secur govern commun encrypt develop system privaci agenc public
power batteri subject circuit heat signal electr light suppli radio	gun weapon arm firearm control peopl crime amend carri law
kmeans+pbg	**hlc+pbg**
god faith christian jesu bibl christ sin lord church hell	space earth orbit planet nasa mar mission moon satellit sky
drive scsi control hard disk id system problem bu floppi	game team hockei plai goal nhl playoff player win period
team plai player hockei goal nhl period game season pit	drive do disk scsi hard control system id problem floppi
imag color file bit format jpeg gif displai data program	god jesu christian christ church bibl sin lord peopl word
armenian turkish peopl armenia turk greek turkei soviet kill russian	armenian turkish muslim turk armenia peopl turkei greek genocid soviet
mac appl mhz speed chip bit cpu memori ram subject	israel isra arab jew jewish write articl palestinian subject peopl
space nasa orbit launch earth henri satellit moon gov mission	doctor effect pain patient medic studi treatment subject problem vitamin
mail post list inform address send newsgroup group email internet	card mhz bit mac bu appl chip ram speed memori
israel isra peac arab palestinian subject write state attack polici	post group newsgroup discuss messag net read new alt usenet
window do run microsoft file manag program system subject applic	file ftp program version format graphic softwar pub system gif
win team game run pitch won lost subject year score	game win team year basebal run subject pitch won lost
window widget server subject motif applic sun displai set file	gun law crime weapon firearm peopl state control crimin legal
jew jewish arab nazi write israel american adam anti articl	mail list univers inform comput address send email fax subject
kei encrypt chip clipper secur govern system escrow phone algorithm	window applic run file program widget manag server motif set
card monitor driver video subject mode vga screen color problem	kei chip encrypt bit secur de algorithm public messag block
gun law peopl weapon crime firearm state arm control amend	homosexu christian paul peopl gai sex cramer men sexual write
food msg effect eat studi peopl diet vitamin subject candida	health insur medic bank diseas columbia aid research gordon care
govern monei insur pai cost peopl privat compani isc market	monitor driver video mous subject mode vga card window problem
research inform health april medic cancer center year nation diseas	peopl exist moral god don religion reason claim question thing

Fonte: Dados da pesquisa.

documentos rotulados é necessário para criar um modelo de classificação acurado. Entretanto, um conjunto consistente de documentos rotulados para induzir um classificador não está disponível na maioria das aplicações reais. Além disso, produzir documentos rotulados é uma tarefa que consome bastante tempo e esforço de especialistas. Assim, um modo mais prático de tratar a classificação de texto em aplicações reais é aplicar métodos que fazem uso de pequenos conjuntos de documentos rotulados com documentos não rotulados (normalmente é disponibilizado um grande conjunto de documentos não rotulados) para realizar a tarefa de classificação.

Técnicas transdutivas são amplamente usadas quando a quantidade de documentos rotulados são insuficientes para gerar um modelo de classificador com bom desempenho, e também quando o objetivo é classificar um conjunto de documentos conhecidos. Nesses casos, é feito o uso de dados não rotulados para melhorar o desempenho do classificador (KONG; NG; ZHOU, 2013; CHAPELLE; SCHLKOPF; ZIEN, 2010; JOACHIMS, 1999). Classificação transdutiva estima diretamente os rótulos de documentos não rotulados sem criar um modelo de classificação.

Nesta seção é descrita a proposta de um algoritmo baseado na representação de grafos

bipartidos para a classificação transdutiva. A ideia por trás do algoritmo semissupervisionado aqui proposto é associar informações de classes para cada vértice e aresta. As informações de classes são armazenadas em vetores l-dimensionais, onde o número de dimensões é o número de classes. Esses vetores são os rótulos multidimensionais associados a cada vértice e aresta do grafo bipartido e são propagados em um esquema de propagação de rótulos. Usando a estrutura do grafo bipartido, as informações de classes são propagadas para todos os vértices e arestas, obedecendo as relações de vizinha estabelecida entre os vértices. O algoritmo tradicional de propagação de rótulos otimiza as informações de classes considerando cada dimensão independentemente, aqui, é proposto um algoritmo que otimiza a divergência entre vértices relacionados considerando todas as dimensões dos vetores com informações de classes. A divergência entre as informações de classes de vértices e arestas são otimizadas pela minimização da divergência KL (veja a Definição 1). O algoritmo proposto, chamado *Transductive Propagation in Bipartite Graph* (TPBG), alcança melhor desempenho na classificação do que os algoritmos estado da arte baseados na representação espaço vetorial e grafos, principalmente quando apenas uma pequena quantidade de dados rotulados é disponibilizada.

São três as principais contribuições deste trabalho: 1) permitir o uso vantajoso da representação em grafos bipartidos para o aprendizado semissupervisionado transdutivo; 2) propor um algoritmo que ultrapasse a acurácia de classificação obtida pelos algoritmos estado da arte baseados na representação em espaço vetorial e grafos quando considerado pequenas quantidades de documentos rotulados; 3) conduzir amplos experimentos comparativos para avaliação da proposta.

Na avaliação experimental é apresentada a acurácia do algoritmo para diferentes quantidades de documentos rotulados. Os resultados mostram que o TPBG encontra resultados consistentes, o que faz desse método uma alternativa competitiva e uma nova possibilidade de exploração no desenvolvimento de algoritmos semissupervisionados.

3.4.1 Trabalhos relacionados

Diferente da classificação supervisionada indutiva, que objetiva o aprendizado a partir de exemplos, o objetivo da classificação transdutiva é predizer os rótulos de classes dado exemplos não rotulados e rotulados. No contexto de classificação de documentos, o aprendizado transdutivo atribui pesos, ou informações de classes, para os documentos e esses documentos são classificados considerando esses pesos.

Em geral, existem dois tipos de representações de coleções de documentos para realizar o aprendizado transdutivo: modelo espaço vetorial e representação em grafos. No modelo espaço vetorial, documentos são representados como vetores e cada dimensão corresponde a palavra contida no documento. Os valores nos vetores são baseados na frequência dos termos, tais como pesos binários, tf (*term frequency*) ou tf-idf (*term frequency-inversed document frequency*). Na representação em grafos, os objetos correspondentes a documentos ou palavras

são representados como vértices, e a relação entre os pares de objetos são representados como arestas. Diferentes tipos de objetos e diferentes tipos de relações podem ser usadas para gerar o grafo. Documentos podem ser conectados de acordo com "relações explícitas", como hiperlinks e citações (OH; MYAENG; LEE, 2000; SUN *et al.*, 2009), ou considerando similaridade (ANGELOVA; WEIKUM, 2006). Termos podem ser conectados por precedência no texto (AGGARWAL; ZHAO, 2013), se eles apresentam relação semântica ou sintática (STEYVERS; TENENBAUM, 2005), ou se eles coocorrem na coleção de texto ou em sentenças (PALSHIKAR, 2007). A combinação de diferentes tipos de objetos é também usada. Nesse caso, documentos e termos são gerados por um grafo bipartido onde os termos são conectados aos documentos nos quais eles ocorrem (ROSSI *et al.*, 2014; ROSSI *et al.*, 2012).

3.4.1.1 Aprendizado transdutivo via modelo espaço vetorial

Os primeiros trabalhos de aprendizado transdutivo para a classificação de texto consideraram coleções de textos representadas no modelo espaço vetorial (YAROWSKY, 1995; BLUM; MITCHELL, 1998; JOACHIMS, 1999; NIGAM *et al.*, 2000). Talvez, a forma mais natural de realizar o aprendizado transdutivo é pela técnica de *Self-Training*. *Self-Training* assume que a classificação mais confiante estão corretas e re-induz o modelo adicionando esses novos exemplos rotulados as instâncias de teste.

Support Vector Machines (SVM) é uma das técnicas mais populares para a classificação. Sua versão transdutiva, *Transductive Support Vector Machines* (TSVM), tem sido usada para classificação de texto (JOACHIMS, 1999). TSVM considera documentos rotulados e não rotulados para obter o hiperplano com margem máxima.

Aprendizado transdutivo pode também ser realizado por modelos probabilísticos. No trabalho de Nigam *et al.* (2000) é apresentado um *framework* probabilístico no qual os dados não rotulados são usados para melhorar a classificação de texto. Um algoritmo *Expectation Maximization* (EM) baseado no *Multinomial Naive Bayes* (MNB) é usado para estimar a posterior com máxima probabilidade. O algoritmo EM é executado em dois passos. No passo E, o parâmetro θ é usado para estimar a filiação do componente de cada documento não rotulado. E no passo M, o parâmetro θ é re-estimado usando todos os documentos e isso estabelece a informação de classes dos termos. O classificador baseado em EM não é acurado se as suposições do modelo gerador forem violadas.

3.4.1.2 Aprendizado Transdutivo em Grafos

Utilizando a mesma notação descrita anteriormente na Seção 3.1, é definido um grafo como uma tripla $G = (\mathscr{V}, \mathscr{E}, f)$, onde $\mathscr{V}$ é o conjunto de vértices, $\mathscr{E}$ é o conjunto de arestas, e f é um mapeamento que associa uma aresta a um número real, *i.e.* $f : \mathscr{E} \rightarrow \mathbb{R}$. Quando $\mathscr{V}$ é composto por um um único tipo de vértice, o grafo é chamado homogêneo. Quando $\mathscr{V}$ é composto por h tipos diferentes de objetos, *i.e.*, $\mathscr{V} = \{V_1 \cup \ldots \cup V_h\}$, o grafo é chamado heterogêneo (JI

et al., 2010). Para criar um grafo no contexto textual, os vértices podem ser associados a documentos, palavras, pedaços de textos, sentenças ou parágrafos, e todos esses objetos podem ser combinados em pares para descrever uma aresta. Normalmente, redes homogêneas podem ser criadas considerando relações explícitas entre pares de documentos (OH; MYAENG; LEE, 2000; SUN *et al.*, 2009), ou considerando métricas de similaridade entre documentos (ANGELOVA; WEIKUM, 2006). Em relação a grafos heterogêneos no contexto textual, os termos podem ser conectados a documentos (ROSSI *et al.*, 2014; DHILLON, 2001) ou sentenças (WAN; YANG; XIAO, 2007) no qual eles ocorrem.

Os principais algoritmos para aprendizado transdutivo baseado em grafos objetivam maximizar uma função geral de regularização. Para explicar tal função, considere l o número de classes e Y_i um vetor l-dimensional associado a um objeto rotulado $v_i \in \mathcal{V}$. Denota-se $Y_{i,k}$ como a k-ésima dimensão do vetor Y_i. Suponha que c_k $(0 \leq k \leq l)$ é o rótulo da classe associado a v_i, então $Y_{i,k} = 1$ e $Y_{i,r} = 0$ para todo $r \neq k$. Considerando que seja fornecido um pequeno conjunto de documentos rotulados $\mathcal{V}^l$, será descrito a função geral de regularização como

$$Q(G) = \frac{1}{2} \sum_{e_{j,i} \in \mathcal{E}} f_{j,i} \Omega(R_j, R_i) + \mu \sum_{v_i \in \mathcal{V}^l} \Omega'(R_i, Y_i), \tag{3.31}$$

onde R_i é um vetor l dimensional associado a cada vértices $v_i \in \mathcal{V}$, esse vetor é o rótulo multidimensional que contem as informações de classes – cada dimensão de R_i corresponde ao grau de filiação do vértice v_i para uma classe. As funções Ω e Ω' são métricas que retornam a similaridade entre objetos representado pelo grafo G.

A função objetivo da Equação 3.31 é baseada em duas premissas. A primeira premissa afirma que os valores das informações de classes entre vértices vizinhos e altamente relacionados devem ser próximos. A segunda premissa requer que as informações atribuídas durante o processo de classificação devem ser o mais próximo possível da real informação de classe já fornecida. O parâmetro μ controla a influência da segunda premissa, *i.e.* o quanto os objetos já rotulados devem manter suas informações de classes.

Baseado nas premissas consideradas no problema de otimização descrito na Equação 3.31, existem dois tipos de algoritmos que realizam a classificação transdutiva em grafos homogêneos: (i) *Gaussian fields and Harmonic Functions* (GFHF) (ZHU; GHAHRAMANI; LAFFERTY, 2003) e (ii) *Learning with Local and Global Consistency* (LLGC) (ZHOU *et al.*, 2004). Esses dois algoritmos trabalham como o esquema de propagação de rótulos no grafo homogêneo, onde iterativamente os rótulos atribuídos aos vértices do grafo são propagados de forma a minimizar a Equação 3.31.

Em uma rede heterogênea bipartida, a propagação dos rótulos multidimensionais é feita pela propagação dos vetores de informação de classes dos documentos para os termos e vice versa. Considerando essa técnica, os três principais algoritmos que realizam classificação transdutiva em grafos heterogêneos são: (i) *Tag-based Model* (TB), (ii) *GNetMine* (GM) e (iii) *Label Propagation using Bipartite Heterogeneous Networks* (LPBHN). TB (YIN *et al.*, 2009)

estende as premissas assumidas pelo problema de otimização descrito na Equação 3.31, além disso, usa um conhecimento prévio obtido por um classificador de domínio. GM (JI *et al.*, 2010) é baseado no algoritmo LLGC e considera diferentes tipos de relações entre diferentes tipos de objetos. O algoritmo LPBHN (ROSSI *et al.*, 2014) é baseado no algoritmo GFHF e tem a vantagem de ser um algoritmo livre de parâmetros que usa a representação em redes bipartidas para a classificação transdutiva.

3.4.2 Propagação em Grafo bipartido para Classificação Transdutiva

Nesta seção é apresentada a formulação, além da fundamentação computacional e matemática, do algoritmo semissupervisionado proposto baseado em grafo bipartido para classificação transdutiva. O algoritmo *Transductive Propagation in Bipartite Graph* (TPBG) é um algoritmo de propagação de rótulos baseado na versão não-supervisionada do algoritmo de propagação em grafos bipartidos (PBG), além disso, a modelagem do problema apresenta características que se encaixam no *framework* de regularização descrito na Equação 3.31. Diferente de outros algoritmos de aprendizado transdutivo baseados em grafos, é considerado a medida de similaridade baseada na divergência KL para mensurar a divergência entre as informações de classes associadas aos termos, documentos e suas relações.

A notação é detalhadamente descrita na Seção 3.1. Aqui, a coleção de documentos é representada por um grafo bipartido do tipo $G = ((\mathscr{D} \cup \mathscr{W}), \mathscr{E}, f)$. Os l rótulos das classes estão contidos no conjunto $\mathscr{C}$, sendo que no grafo existem documentos rotulados e não rotulados, *i.e.* $\mathscr{D} = \mathscr{D}^l \cup \mathscr{D}^u$. Além disso, os vértices $d_j \in \mathscr{D}^u$ e $w_i \in \mathscr{W}$, e a aresta $e_{j,i} \in \mathscr{E}$ são associados respectivamente aos vetores com informações de classes (ou rótulos multidimensionais) A_j, B_i e $C_{j,i}$. Um documento rotulado $d_j \in \mathscr{D}^l$ é associado a um vetor do tipo Y_j. Assim, tem-se como objetivo do aprendizado transdutivo encontrar uma função $A : \{\mathscr{D}^l \cup \mathscr{D}^u\} \rightarrow \mathscr{Y}$, no qual os documentos não rotulados são usados para melhorar a acurácia de classificação.

3.4.2.1 Otimizando a divergência entre os rótulos multidimensionais

O *framework* de regularização definido na Equação 3.31 é usado em vários algoritmos baseados em propagação de rótulos (ZHOU *et al.*, 2004; ZHU; GHAHRAMANI; LAFFERTY, 2003). Tipicamente, a distância Euclidiana é utilizada nesses algoritmos para medir a similaridade entre as informações de classes dos objetos representados no grafo. O *framework* de regularização da Equação 3.31 também é usado para descrever o TPBG, entretanto é utilizada a divergência KL como função de similaridade.

Existe uma estreita relação entre os vetores que mantêm as informações de classes e a distribuição de probabilidade das classes. Nesse caso, documentos e palavras com a mesma classe terão distribuição de probabilidade com pequeno valor de divergência. Devido a essa semelhança, a distância Euclidiana pode não ser uma boa medida para mensurar a similaridade entre objetos vizinhos. Por exemplo, vetores gerados pelas distribuições normais com sobreposição do tipo

$\mathcal{N}(0,1000)$ e $\mathcal{N}(10,10000)$ tem a distância Euclidiana esperada igual a 10. Em contrapartida, vetores gerados pelas distribuições (que mal se sobrepõe) $\mathcal{N}(0,0.001)$ e $\mathcal{N}(0.1,0.01)$ terão a distância Euclidiana igual a 0.01, o que não reflete a distinção dos vetores. Assim é esperado que uma função de similaridade como a divergência KL possa ser usada para melhor distinguir as informações de classes.

O algoritmo baseado em propagação de rótulo para redes bipartidas usa informações de classes apenas dos termos e documentos, e assume que a informação de classe de uma palavra w_i no documento d_j pode ser usada para distinguir a classe do documento não rotulado $d_j \in \mathscr{D}^u$ (ROSSI *et al.*, 2012; ROSSI *et al.*, 2014). Entretanto, essa premissa não está totalmente correta devido a ocorrência de uma mesma palavra com diferentes significados em diferentes documentos, e possivelmente essa palavra possa remeter a diferentes classes. Então, a fim de resolver esse problema, é associado informações de classes para cada par documento-palavra, correspondente ao vetor $C_{e_{j,i}}$. Com isso, procura-se garantir que uma mesma palavra w_i conectada a diferentes documentos possa propagar diferentes informações de classes para seus vizinhos.

A premissa do TPBG é que a divergência entre as informações de classes dos documentos em $\mathscr{D}^l \cup \mathscr{D}^u$, das palavras em $\mathscr{W}$ e das arestas em $\mathscr{E}$ são úteis para melhorar a descoberta dos padrões de classes dos documentos em $\mathscr{D}^u$. O TPBG propaga as informações de classes das palavras e documentos para as arestas, e usa as informações de classes das arestas para inferir a informação de classe dos documentos não rotulados. Assim, um vetor l-dimensional $C_{j,i}$ é usado para armazenar a informação de classe da aresta $e_{j,i} \in \mathscr{E}$.

A ideia por trás do algoritmo TPBG é assumir que quanto maior o valor de $f_{j,i}$, maior deve ser a concordância das informações de classes dos vetores $(A_j \odot B_i)$ e $C_{e_{j,i}}$. Então, baseando-se na divergência KL, é definido a seguinte função objetivo:

$$Q_G(A,B,C) = \sum_{e_{j,i} \in \mathscr{E}} \left(f_{j,i} C_{e_{j,i}} \log \frac{A_j \odot B_i}{C_{e_{j,i}}} \right) + \sum_{d_j \in \mathscr{D}} \mathscr{R}(A_j, \alpha) + \sum_{d_j \in \mathscr{D}^l} Y_j \log \frac{A_j}{Y_j}, \tag{3.32}$$

onde $\mathscr{R}(A_j, \alpha)$ é o termo regularizador aplicado a cada documento d_j, e a constante α controla a concentração das informações de classes no vetor A_j,

$$\mathscr{R}(A_j, \alpha) = (\alpha - A_j) \log A_j + A_j (\log A_j - 1). \tag{3.33}$$

Um alto valor de α significa que os documentos provavelmente conterão uma mistura de várias classes. Por outro lado, um baixo valor de α indicará que um documento poderá conter mistura de poucas classes.

O valor de $\sum_{d_j \in \mathscr{D}^l} Y_j \log \frac{A_j}{Y_j}$ garante que as informações de classes atribuídas durante a classificação é próxima da informação de classe real fornecida pelos documentos rotulados. Isso está relacionado com as premissas estabelecidas na função de regularização geral, Equação 3.31. E também diferencia da abordagem não supervisionada descrita na Seção 3.3.

Os valores dos vetores com informações de classes são obtidos resolvendo o seguinte problema de otimização

$$Q(G) = \arg\max_{A^*,B^*,C^*} \sum_{c_k \in \mathbb{C}} [Q_G(A,B,C)]_k, \tag{3.34}$$

onde A^*, B^* e C^* correspondem aos vetores que otimizam a função Q.

As informações de classes são obtidas pelo método de gradiente descendente na otimização da função $Q(G)$. O valor máximo de $Q(G)$ em relação a A_j, B_i e $C_{e_{j,i}}$, para todos os documentos d_j, palavras w_i e arestas $e_{j,i}$ no grafo G, são determinados pela direção do gradiente. O procedimento de maximização da função objetivo (Equação 3.34) segue o mesmo procedimento da versão não supervisionada (Seção 3.3.2).

Para obter a transdução, inicialmente é maximizada a Equação 3.32 em relação ao vetor $C_{e_{j,i}}$ tal que $\sum_{c_k \in \mathbb{C}} C_{e_{j,i},k} = 1$. Assim, isolando os termos que estão em função de $C_{e_{j,i}}$ e adicionando o apropriado multiplicador de Lagrange,

$$\frac{\partial Q}{\partial C_{e_{j,i}}} = f_{j,i} \left(\log (A_j \odot B_i) - \log C_{e_{j,i}} - 1 + \frac{\lambda}{f_{j,i}} \right). \tag{3.35}$$

Fixando essa derivada igual a zero chega-se no valor máximo dos vetores $C_{e_{j,i}}$ associados as arestas do grafo G,

$$C_{e_{j,i}} \propto A_j \odot B_i. \tag{3.36}$$

Como a soma dos valores de $C_{e_{j,i}}$ tem que ser igual a 1, pode-se normalizá-lo de forma que

$$C_{e_{j,i}} = \frac{A_j \odot B_i}{\sum_{c_k \in \mathbb{C}} (A_j \odot B_i)_k}. \tag{3.37}$$

Agora, é maximizada a Equação 3.32 em relação a A_j, vetor associado a um documento $d_j \in \mathscr{D}$. Não é necessário usar Lagrange para restringir esse vetor pois ele já é restringido pelo termo regularizador (Equação 3.33). Os termos que contêm A_j são:

$$Q_{[A_j]} = \sum_{w_i \in \mathscr{W}_{d_j}} (f_{j,i} C_{e_{j,i}} \log A_j) + \mathscr{R}(A_j, \alpha) + \sum_{d_j \in \mathscr{D}^l} Y_j \log \frac{A_j}{Y_j}, \tag{3.38}$$

onde o subconjunto W_{d_j} é formado pelas palavras que conectam um documento d_j no grafo bipartido G.

O valor de A_j é fixo em Y_j para todo documento rotulado, *i.e.* $A_j = Y_j$ para todo $d_j \in \mathscr{D}^l$. Por outro lado, para obter o valor de A_j para um documento não rotulado d_j é necessário derivar em relação A_j,

$$\frac{\partial Q_{[A_j]}}{\partial A_j} = \frac{1}{A_j} \left(\sum_{w_i \in \mathscr{W}_{d_j}} f_{j,i} C_{e_{j,i}} - A_{j,i} + \alpha \right). \tag{3.39}$$

Definindo essa equação igual a zero encontra-se o máximo em:

$$A_j = \alpha + \sum_{w_i \in \mathscr{W}_{d_j}} f_{j,i} C_{e_{j,i}}. \tag{3.40}$$

Finalmente, é maximizada a Equação 3.32 em relação a B_i, vetor associado a uma palavra $w_i \in \mathscr{D}$. Para maximizar em relação a B_i, é isolado o termos que estão em função de B_i e adicionado o multiplicador de Lagrange.

$$Q_{[B_i]} = \sum_{c_k \in \mathbb{C}} \left(\sum_{d_j \in \mathscr{D}} f_{j,i} C_{e_{j,i}} \log B_i + \lambda_k \left(\sum_{w_p \in \mathscr{W}} B_{v,k} - 1 \right) \right). \tag{3.41}$$

Derivando $Q_{[B_i]}$ e ajustando o resultado dessa derivada igual a zero, obtêm-se a seguinte equação de atualização,

$$\hat{B}_{i,k} \propto \sum_{d_j \in \mathscr{D}} \sum_{c_k \in \mathbb{C}} f_{j,i} C_{e_{j,i}}. \tag{3.42}$$

Normalizando o valor de $\hat{B}_{i,k}$ sobre todas as palavras w_p no vocabulário,

$$B_{i,k} = \frac{\hat{B}_{i,k}}{\sum_{p \in \mathscr{W}} \hat{B}_{p,k}}. \tag{3.43}$$

As equações de atualização descritas nas equações 3.37, 3.22 e 4.14 que maximizam a Equação 3.15 são a base para o algoritmo TPBG, descrito na próxima seção.

3.4.2.2 O Algoritmo TPBG

A ideia do algoritmo TPBG é propagar a informação de classe por intermédio dos vértices vizinhos. Assumindo que é dado um conjunto de documentos rotulados $\mathscr{D}^l$, e a informação de classe dos vetores associados a termos e a documentos não rotulados são aleatoriamente inicializados, as iterações de atualizações são realizadas de duas maneiras: (1) atualizações locais, que respondem pela propagação por meio da vizinhança de cada vértice, e (2) propagação global, a qual propagam-se as informações de classes por meio da completa estrutura do grafo bipartido. O algoritmo TPBG é sumarizado no Algoritmo 9. As propagações locais são descritas no Algoritmo 10 e a propagação global é descrita no Algoritmo 11.

O procedimento de propagação do algoritmo TPBG (Algoritmo 9) tem como entrada o conjunto de documentos rotulados, um grafo bipartido G e o parâmetro de concentração α. Os vetores associados aos vértices $d_j \in \mathscr{D}^u$ e $w_i \in \mathscr{W}$ são iniciados aleatoriamente, tal que $\sum_{c_k \in \mathbb{C}} A_{j,k} = 1$ para todo documento não rotulado $d_j \in \mathscr{D}^u$ e $\sum_{w_i \in \mathscr{W}} B_{i,k} = 1$ para toda a classe $c_k \in \mathbb{C}$. Os vetores com informações de classes associados aos documentos já rotulados são inciados tal que $A_j = Y_j$. A propagação local é executada para todas as arestas $e_{j,i}$ incidente ao vértice d_j. Esse procedimento cria um vetor l-dimensional $C_{e_{j,i}}$ resultante do produto Hadamard

Algoritmo 9: Algoritmo TPBG

Entrada:
grafo bipartido G,
$\mathscr{D}^l$ // conjunto de documentos rotulados
α // parâmetro de concentração
Saída :
Y // rótulos atribuídos a cada documento em $\mathscr{D}^u$

início
  Inicia vetor A_j para cada documento $d_j \in \mathscr{D}$;
  Inicia vetor B_i para cada palavra $w_i \in \mathscr{W}$;
  enquanto *não alcança convergência* **faça**
    foreach $d_j \in \mathscr{D}$ **do**
      repita
        $A_j \leftarrow$ `localPropag(`G,d_j, A_j, B, $\mathscr{D}^l$`)`;
      até A_j *não converge*;
    $B \leftarrow$ `globalPropag(`G,A,B`)`;
  para todo $d_j \in \mathscr{D}^u$: $\{ Y_{j,k} = 1 \quad | \quad k = \arg\max_{\hat{k}=1}^{l} A_{j,\hat{k}} \}$;

de A_j e B_i, *i.e.* $C_{e_{j,i}} = A_j \odot B_i$. Esse vetor é normalizado tal que $\sum_{c_k \in \mathbb{C}} C_{e_{j,i},k} = 1$. Se o documento d_j está no conjunto de documentos rotulados $\mathscr{D}^l$, então $A_j = Y_j$, caso contrário, A_j recebe as informações de classes de todas as arestas incidentes no vértice d_j (Equação 3.22). A propagação local é repetida para cada vértice d_j enquanto os valores em A_j se alteram. O parâmetro α é usado para controlar o grau de concentração do vetor A_j.

Algoritmo 10: Propagação Local para TPBG

função `localPropag(`G, d_j, A_j, B,$\mathscr{D}^L$`)`
  início
    para cada *edge* $e_{j,i}$ *incident in* d_j **faça**
      $C_{e_{j,i}} \leftarrow \dfrac{(A_j \odot B_i)}{\sum_{c_k \in \mathbb{C}} (A_j \odot B_i)_k}$;
    se $d_j \in \mathscr{D}^l$ **então**
      $A_j \leftarrow Y_j$;
    senão
      $A_j \leftarrow \alpha + \sum_{w_i \in \mathscr{W}_{d_j}} f_{j,i} C_{e_{j,i}}$;
    retorne A_j;

A propagação global é realizada para todos os vértices $w_i \in \mathscr{W}$ e para toda aresta $e_{j,i}$ incidente no vértice w_i. Esse procedimento também cria um vetor l-dimensional $C_{e_{j,i}}$ resultante do produto Hadamard de A_j e B_i. O vetor $C_{e_{j,i}}$ é normalizado tal que $\sum_{c_k \in \mathbb{C}} C_{e_{j,i},k} = 1$ e o valor é propagado de volta para o vetor B_i, como descrito na Equação 4.14. Por fim, o vetor com informação de classe B_i é normalizado para todas as palavras $w_p \in \mathscr{W}$.

Algoritmo 11: Propagação Global para TPBG

função `globalPropag(`G, A, B`)`
  início
    para cada *vértice* $w_i \in \mathscr{W}$ **faça**
      para cada *aresta* $e_{j,i}$ *incidente em* w_i **faça**
        $C_{e_{j,i}} \leftarrow \frac{(A_j \odot B_i)}{\sum_k (A_j \odot B_i)_k}$;
      $B_i \leftarrow \sum_{d_j \in \mathscr{D}} f_{j,i} C_{e_{j,i}}$;
    para cada *vértice* $w_i \in \mathscr{W}$ **faça**
      para $c_k \in \mathbb{C}$ **faça**
        $B_{i,k} = \frac{B_{i,k}}{\sum_{w_p \in \mathscr{W}} B_{p,k}}$;
    retorne B;

Como na versão não supervisionada, a ideia por trás do algoritmo TPBG é localmente concentrar as informações de classes de cada palavra de um documento d_j no vetor A_j. Então concentrar toda informação global do grafo nos vetores de informação de classes associadas as palavras. Porém, diferente das versão não supervisionada, é utilizado informação de classe já conhecida advinda dos documentos rotulados.

Os procedimentos de propagação global e local são aplicados até um número máximo de iterações serem alcançadas, ou até a informação de classe contida nos documentos não rotulados não se alterarem em sucessivas iterações.

3.4.3 Avaliação Experimental

Nesta avaliação experimental, o algoritmo TPBG é comparado com os algoritmos apresentados na seção de trabalhos relacionados (Seção 3.4.1), que consideram coleções de documentos representadas no modelo espaço vetorial e em grafos. Também é considerado o algoritmo *Multinomial Naive Bayes* (MNB), que é um algoritmo indutivo supervisionado. O objetivo dessa seção é demonstrar que o algoritmo TPBG pode alcançar resultados superiores em acurácia em comparação com os algoritmos estado da arte para classificação transdutiva de textos. Nesta seção são apresentadas as coleções de textos usadas na avaliação experimental, as configurações dos experimentos, os critérios de avaliação, os resultados e as discussões.

Foram utilizadas 22 coleções de documentos textuais de diferentes domínios: *e-mails* (ES), documentos médicos (DM), artigos de notícias (AN), documentos científicos (DC), análise de sentimentos (SA) e páginas *web* (PW). Todos os documentos foram pre-processados, *stopwords* foram removidas, foram aplicados *stemm* nos termos usando o algoritmo de Porter (PORTER, 1997), marcadores HTML foram removidos, e apenas termos com frequência maior

que 2 em cada documento foram considerados. Os pesos das arestas são simplesmente a frequência do termo no documento. Maiores detalhes sobre as coleções utilizadas são encontradas em (ROSSI; MARCACINI; REZENDE, 2013).

A Tabela 7 apresenta as coleções de documentos e as suas características. Em cada coluna estão descritos os números de documentos (m), o número de termos (n), e o número médio de termos por documentos ($\overline{n_{d_j}}$), o número de classes (l), o desvio padrão considerando a porcentagem de classes em cada coleção ($\sigma(\mathscr{C})$), e a porcentagem da classe majoritária ($\max(\mathscr{C})$).

Tabela 7 – Características da coleção de documentos textuais.

Coleções	m	n	$\overline{n_{d_j}}$	l	$\sigma(\mathscr{C})$	$\max(\mathscr{C})$
Classic4 (SD)	7095	7749	35.28	4	1.94	45.16
CSTR (SD)	299	1726	54.27	4	18.89	42.81
Dmoz-Health (WP)	6500	4217	12.40	13	0.00	7.69
Dmoz-Science (WP)	6000	4821	11.52	12	0.00	9.63
Dmoz-Sports (WP)	13500	5682	11.87	27	0.00	3.70
FBIS (NA)	2463	2001	159.24	17	5.66	26.54
Hitech (NA)	2301	12942	141.93	6	8.25	26.21
La1s (NA)	3204	13196	144.64	6	8.22	29.43
La2s (NA)	3075	12433	144.83	6	8.59	29.43
NFS (CD)	10524	3888	6.65	16	3.82	13.39
Oh0 (MD)	1003	3183	52.50	10	5.33	19.34
Oh10 (MD)	1050	3239	55.64	10	4.25	15.71
Oh15 (MD)	913	3101	59.30	10	4.27	17.20
Oh5 (MD)	918	3013	54.43	10	3.72	16.23
Ohscal (MD)	11162	11466	60.39	10	2.66	14.52
opinosis (AS)	6457	2693	7.56	51	1.42	8.18
Re0 (NA)	1504	2887	51.73	13	11.56	40.43
Re1 (NA)	1657	3759	52.70	25	5.54	22.39
Re8 (NA)	7674	8901	35.31	8	18.24	51.12
Reviews (NA)	4069	22927	183.10	5	12.80	34.11
Syskillwebert (WP)	334	4340	93.16	4	10.75	41.02
WAP (WP)	1560	8461	141.33	20	5.20	21.86

Fonte: Dados da pesquisa.

3.4.3.1 Configuração dos Experimentos e Critério de Avaliação

O algoritmo TPBG foi comparado com os algoritmos tradicionais e estado da arte em aprendizado transdutivo, tanto aqueles que usam a representação em grafos quanto o modelo espaço vetorial. Além disso, foi executado um algoritmo indutivo para averiguar os resultados obtidos.

Aqui, são descritos os valores dos parâmetros dos algoritmos usados nos experimentos. Os algoritmos para o aprendizado transdutivo baseados no modelo espaço vetorial são: *Multinomial Nave Bayes with Self-Training* (ST), onde o número X de documentos com o melhor ranking de classificação foi definido pelo conjunto $X = \{5, 10, 15, 20\}$; *Expectation Maximization* (EM), no qual foi considerado a sua instanciação para classificação de texto (NIGAM *et al.*, 2000),

e foram usados os parâmetros $\Lambda = \{0.1, 0.3, 0.5, 0.7, 0.9\}$ e 1, 2, 5, 10 componentes para cada classe; *Transductive Support Vector Machines* (TSVM), no qual será considerada a proposta apresentada em (JOACHIMS, 1999) e será usado $C = 1$ para induzir o hiperplano com margem máxima e o parâmetro C' variando por um fator de dez, indo de 10^{-5} para 10^{1}. Também foi executado o TSVM com e sem a função proposta no trabalho de Joachims (1999) para manter a mesma proporção das classes dos documentos rotulas na classificação dos documentos não rotulados.

Os algoritmos para o aprendizado transdutivo baseados em grafos utilizados nessa avaliação experimental são divididos em algoritmos baseados em grafos de documentos (apenas documentos como vértices) e algoritmos baseados em grafos bipartidos (documentos e palavras como vértices). Todos esses algoritmos são iterativos, e para todos menos o TPBG foram definidos o máximo de 1000 iterações. Para o TPBG foram definidas apenas 100 iterações globais e 100 iterações locais. Os grafos de documentos foram gerados utilizando a técnica de K vizinhos mútuos, com $K \in \{7, 17, 37, 57\}$, e também foram gerados grafos utilizando a técnica *Exp* com o valor de $\sigma = \{0.05, 0.2, 0.35, 0.5\}$. Os algoritmos baseados em grafos de documentos são: *Label Propagation with Gaussian Fields and Harmonic Functions* (LP), algoritmo não paramétrico; *Learning with Local and Global Consistency* (LLGC), no qual foi utilizado o conjunto de valores para o parâmetro $\alpha \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$.

Os algoritmos baseados em grafos bipartidos são: *Label Propagation based on Bipartite Heterogeneous Network* (LPBHN), algoritmo não paramétrico; *Tab-based Model* (TB), no qual foi utilizado $\alpha = 0$, desde que existam nenhum objeto de domínio diferente, $\beta = \{0.1, 1, 10, 100, 1000\}$, e $\lambda = \{0.1, 1, 10, 100, 1000\}$; *GNetMine* (GM), no qual foi utilizado $\alpha \in \{0.1, 0.3, 0.5, 0.7, 0.9\}$.

Foi também executado o Multinomial Naive Bayes (MNB), que é um algoritmo de aprendizado supervisionado e indutivo. Ele foi utilizado para verificar os benefícios sobre o uso de documentos não rotulados no auxílio da classificação de textos. Não existem parâmetros para o MNB.

Para o algoritmo TPBG, o parâmetro de concentração foi definido com os valores do conjunto, *i.e.*, $\alpha \in \{0.5, 0.05, 0.005\}$. Um vetor com maior concentração indica mais informação para um pequeno número de classes, isso pode ser útil para diferenciar os documentos. Assim, foi decidido analisar esse parâmetro para verificar o comportamento do valor de acurácia do algoritmos. Os resultados dessa análise estão descritos no Apêndice B.

A acurácia dos classificadores foi obtida da média de 10 execuções. Em cada execução foi escolhido aleatoriamente X documentos de cada classe como documentos rotulados. Os experimentos foram conduzidos usando $X = \{1, 10, 20, 30, 40, 50\}$. As execuções começaram com o número mínimo de documentos rotulados por classes e foram acrescidos em 10 até 50. Essa variação no número de documentos rotulados permite a análise do comportamento do algoritmo e a relação inversa entre o número de documentos não rotulados e o desempenho do

classificador. Os restante dos $m-(X*l)$ documentos foram utilizados para avaliar a classificação.

3.4.3.2 Resultados

Nesta seção são apresentados os melhores valores de acurácia obtidos na avaliação experimental. Os resultados aqui descritos visam facilitar a análise e a comparação justa entre os algoritmos.

Nas figuras 13 e 14 estão presentes os valores de acurácia obtidos por diferentes algoritmos e o número de documentos rotulados por classes. O valor de acurácia nessas figuras aumentam a medida que o número de documentos rotulados crescem.

O algoritmo TPBG obteve o maior valor de acurácia para o conjunto de dados *classic4*, *Dmoz-Health*, *Dmoz-Science*, *Dmoz-Sports*, *NFS*, *oh5*, e *oh15* para todos os números de documentos rotulados. TPBG também obteve altos valores de acurácia quando o número de documentos rotulados são menores ou iguais a 20. Isso indica que a técnica proposta faz um melhor uso de poucos documentos rotulados para melhorar o desempenho do classificador.

Os algoritmos MNB e LLGC alcançam os melhores resultados em alguns conjuntos de dados do que o TPBG quando considerado um maior número de documentos rotulados. Em geral, o TPBG foi melhor do que todos os classificadores baseados em grafos bipartidos usados nesses experimentos. Da mesma forma, algoritmos baseados em grafos obteve alto desempenho na classificação do que os algoritmos baseados em modelos espaço vetorial. Algoritmos de propagação de rótulos baseados em grafos de documentos, como LLGC e GFHF, foram melhores do que algoritmos baseados em grafos bipartidos.

Os dados apresentados nas figuras 13 e 14 foram submetidos ao teste de Friedman e teste post-hoc de Li com 95% de fator de confiança para assegurar diferença estatística significante *Statistically Significant Differences* (SSD). O teste de diferença estatística aqui aplicado é aconselhável para quando existe um algoritmo de controle (que é aquele algoritmo com melhor ranque médio) e vários conjuntos de dados (GARCÍA *et al.*, 2010; TRAWINSKI *et al.*, 2012). A hipótese nula afirma que todos os algoritmos possuem desempenho de classificação equivalente, e então seus valores de ranque devem ser iguais. Por outro lado, o que se pretende determinar é se o algoritmo TPBG pode obter resultados melhores e com diferença estatística significativa em comparação com os outros algoritmos.

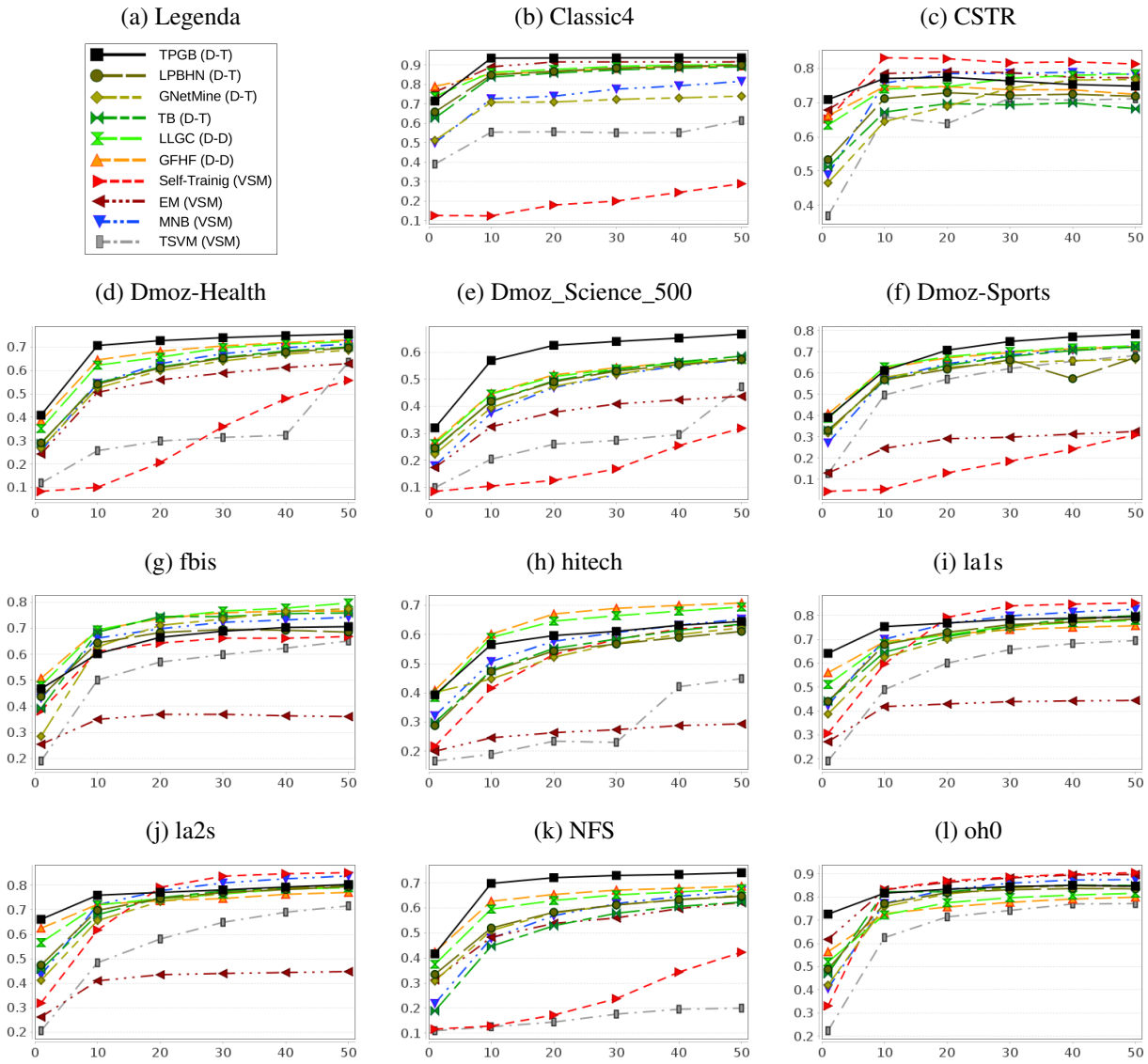
Na Tabela 8 são apresentados os resultados do teste estatístico para os diferentes algoritmos e o ranque é separado pelo número de documentos rotulados. Na Tabela 9 são apresentados os resultados do teste estatístico para todos os algoritmos e é considerado o ranque total, unindo as posições do ranque para todo o número de documento rotulado. Nessas tabelas as colunas representam respectivamente o ranque médio (AR), o ranque geral (GR), *i.e.*, o ranque dos algoritmos considerando o ranque médio, o valor de p, e o valor de p que produz diferença estatística significante (SSD). Os resultados com maior SSD são destacados em itálico.

Tabela 8 – Ranque médio (AR), ranque geral (GR) e o valor de p considerando os valores de acurácia da classificação.

	1 doc. rotulado			**10 doc. rotulados**			**20 doc. rotulados**			**30 doc. rotulados**			**40 doc. rotulados**			**50 doc. rotulados**		
Alg.	**AR**	**GR**	**valor de p**	**AR**	**GR**	**valor de p**	**AR**	**GR**	**valor de p**	**AR**	**GR**	**valor de p**	**AR**	**GR**	**valor de p**	**AR**	**GR**	**valor de p**
TPBG	2.00	1^{st}	–	2.93	1^{st}	–	3.36	1^{st}	–	3.81	3^{th}	0.550163	4.06	3^{th}	0.262568	4.22	3^{th}	0.195452
LPBHN	4.77	3^{th}	0.002387	5.56	5^{th}	*0.003877*	5.88	7^{th}	*0.005718*	6.13	7^{th}	*0.001707*	6.52	8^{th}	*0.000139*	6.63	8^{th}	*0.000084*
GNetMine	7.04	7^{th}	*0*	7.31	9^{th}	*0.000002*	7.09	9^{th}	*0.000044*	6.77	8^{th}	*0.000126*	6.36	7^{th}	*0.000278*	5.88	6^{th}	*0.001858*
TB	5.45	4^{th}	0.000154	5.72	6^{th}	*0.002197*	5.77	6^{th}	*0.008314*	5.77	6^{th}	*0.006170*	5.86	6^{th}	*0.002021*	5.97	7^{th}	*0.00132*
LLGC	3.04	2^{nd}	0.252111	3.27	2^{nd}	0.708816	3.63	3^{th}	0.765125	3.54	2^{nd}	0.765125	3.61	2^{nd}	0.533671	3.79	2^{nd}	*0.411314*
GFHF	2.00	1^{st}	–	3.43	3^{th}	0.583882	4.52	4^{th}	*0.204184*	4.70	4^{th}	0.116770	4.88	4^{th}	0.043739	5.11	4^{th}	*0.023477*
ST	8.45	8^{th}	*0*	6.50	7^{th}	*0.000093*	5.54	5^{th}	*0.16846*	5.29	5^{th}	0.026706	5.04	5^{th}	0.028460	5.18	5^{th}	*0.01927*
EM	6.45	6^{th}	0.000001	6.65	8^{th}	*0.000044*	6.63	8^{th}	*0.000337*	6.90	9^{th}	*0.000068*	7.00	9^{th}	*0.000015*	6.81	9^{th}	*0.000036*
MNB	6.09	5^{th}	0.000007	4.22	4^{th}	0.15587	3.54	2^{nd}	*0.842128*	3.27	1^{st}	–	3.04	1^{st}	–	3.04	1^{st}	–
TSVM	9.68	9^{th}	*0*	9.36	10^{th}	*0*	9.00	10^{th}	*0*	8.77	10^{th}	*0*	8.59	10^{th}	*0*	8.31	10^{th}	*0*
	SSD $p \leq 0$			SSD $p \leq 0.0.015325$			SSD $p \leq 0.05$			SSD $p \leq 0.012362$			SSD $p \leq 0.024544$			SSD $p \leq 0.030983$		

Fonte: Dados da pesquisa.

Figura 13 – Acurácia na Classificação: o eixo *x* representa o número de documentos rotulados por classe e o eixo *y* representa a acurácia obtida.

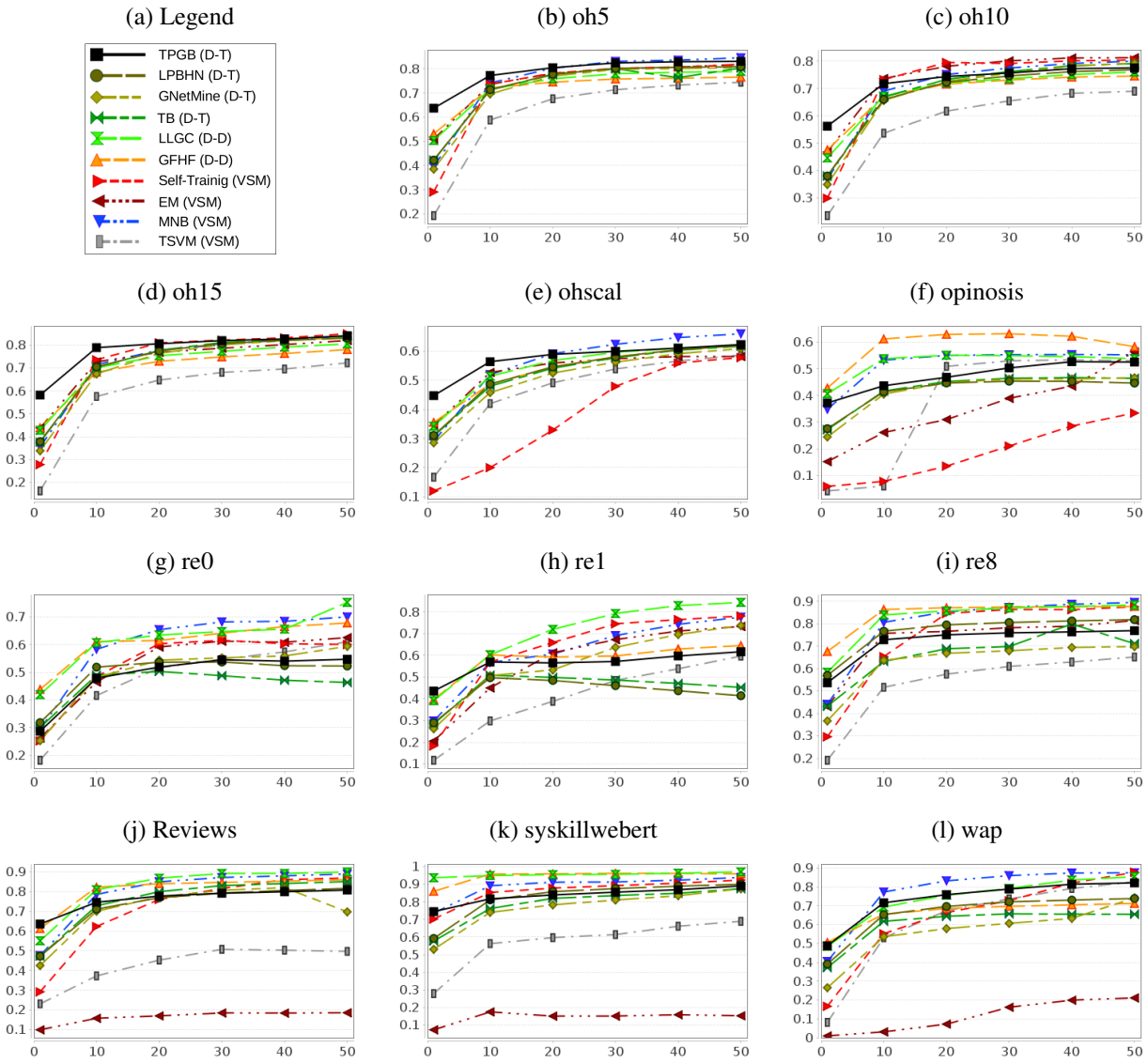


Fonte: Dados da pesquisa.

O algoritmo de controle é por padrão aquele com melhor ranque médio. Quando é considerado todos os números de documentos rotulados por classe (Tabela 9), o algoritmo TPBG apresenta o valor de ranque médio com SSD para quase todos os algoritmos, menos os algoritmos LLGC, MNB e GFHF. Quando o número de documentos rotulados são menores do que 30, o TPBG apresenta o melhor ranque médio. Usando 30, 40 e 50 documentos rotulados por classe, o TPBG fica em terceiro mas não apresenta diferença estatística quando comparado com o primeiro (MNB) e o segundo (LLGC) algoritmos do ranque.

O ranque médio e os resultados do teste de significância estatística permitem concluir que o TPBG tem melhor desempenho em casos onde exitam poucos documentos rotulados, e se tornando uma alternativa viável para os algoritmos estado da arte, tendo como vantagem o

Figura 14 – Acurácia na Classificação: o eixo *x* representa o número de documentos rotulados por classe e o eixo *y* representa a acurácia obtida.



Fonte: Dados da pesquisa.

pequeno consumo de memória pelo uso da representação em grafo da coleção de documentos, e não precisar de uma fase onerosa para criação do grafo.

3.4.4 Considerações finais

Nesta seção foi apresentado o algoritmo TPBG, que usa a representação em grafos bipartidos para realizar a classificação transdutiva de textos. O algoritmo proposto baseia-se na propagação de rótulos multidimensionais, que são vetores contendo informações de classes e estão associados a vértices e arestas do grafo bipartido. O algoritmo é semissupervisionado pois seu procedimento considera documentos rotulados e não rotulados. Os resultados experimentais obtidos com a execução do TPBG em várias bases de dados foram satisfatórios, nos quais

Tabela 9 – Ranque médio (AR), ranque geral (GR) e o valor de p considerando os valores de acurácia da classificação.

Alg.	**AR**	**GR**	**valor de p**
TPBG	3.40	1^{st}	–
LPBHN	5.92	6^{th}	*0*
GNetMine	6.74	8^{th}	*0*
TB	5.76	5^{th}	*0*
LLGC	3.48	2^{nd}	0.823063
GFHF	4.10	4^{th}	0.057347
ST	6.00	7^{th}	*0*
EM	6.74	9^{th}	*0*
MNB	3.87	3^{th}	0.207551
TSVM	8.95	10^{th}	*0*
	SSD $p \leq 0.009312$		

Fonte: Dados da pesquisa.

indicaram que o algoritmo TPBG utilizando poucos dados rotulados alcança melhor resultado na classificação do que os métodos baseados na representação no modelo espaço vetorial e em grafos.

CAPÍTULO

4

EXTRAÇÃO DE TÓPICOS *ONLINE* UTILIZANDO REDES BIPARTIDAS

Com o aumento da quantidade de novos dados no formato textual constantemente publicados, a qualidade dos métodos de mineração de texto podem ser prejudicadas. Quando se trata de notícias, ou documentos textuais disponíveis na Internet, as aplicações devem considerar documentos sobre o fluxo de textos que, em geral, são formados por grandes coleções de documentos, no sentido prático, possivelmente infinitas.

Em muitas aplicações práticas, é essencial transformar de forma rápida esse grande volume de dados textuais em informações e conhecimento útil. Tomando como exemplo o interesse em identificar os tópicos em notícias, deve-se considerar um método que não exija a especificação do número de documentos, ou seja, considera-se infinita a quantidade de documentos que chegam no fluxo. Essa exigência inviabiliza a aplicação de vários métodos encontrados na literatura que percorrem toda a coleção iterativamente e que consomem grande quantidade de memória afim de encontrar as estruturas latentes.

Em um contexto não supervisionado, esses problemas caracterizam o agrupamento em fluxo de dados. Esse problema tem como objetivo agrupar uma sequência X objetos em K grupos distintos. Cada objeto deve ser atribuído a um dos K grupos na ordem que eles chegam no fluxo (CHARIKAR *et al.*, 1997). Existem várias técnicas desenvolvidas para tratar o problema de agrupamento em fluxo, entre elas, destacam-se os algoritmos incrementais. Entre as estratégias incrementais, destacam-se os algoritmos de *microclusters* (GAMA, 2010). Esses algoritmos dividem o processo de agrupamento em duas fases, onde a primeira fase é incremental e agrupa o fluxo de dados localmente, formando os *microclusters*, e a segunda fase gera o agrupamento global.

O problema de extração de tópicos em fluxo de documentos textuais pode ser visto como um caso especial do problema de agrupamento em fluxo de dados. Porém, o agrupamento deve

ser realizado tanto para os documentos que chegam no fluxo quanto para as palavras nesses documentos. Com base nisso, pesquisadores desenvolveram alternativas para os modelos probabilísticos de tópicos em fluxo de dados (BANERJEE; BASU, 2007), e também a versão *online* do LDA (HOFFMAN; BLEI; BACH, 2010). Contudo, essas técnicas exigem o conhecimento prévio do vocabulário, uma vez que a dimensão da distribuição de tópicos deve ser conhecida. Essas técnicas não inserem mecanismo de economia de memória, como dados sumarizados, ou a capacidade de desfazer de dados que já chegaram no fluxo – técnica de "esquecimento" de dados. É difícil modelar corretamente o fluxo, com os mecanismos de sumarização e esquecimento, e descrever um algoritmo de inferência eficiente.

Neste capítulo é apresentado o algoritmo de extração de tópicos em fluxo de documentos que utiliza a representação dos dados em grafos bipartidos. A motivação para a utilização de grafos no problema de extração de tópicos em fluxo deve-se principalmente pela eficácia de sumarizar o fluxo de dados nos grafos. Devido a forma de representação com vértices e arestas, formando vários componentes conexos, é possível identificar subestruturas do grafo e realizar o processamento apenas nessas subestruturas. Outra vantagem da representação em grafos é a facilidade em remover e adicionar novos vértices e arestas. Adicionar novos vértices e arestas é basicamente incluir novos documentos que chegaram no fluxo, e excluir vértices e arestas apropriados é o mecanismo de "esquecimento" de dados. Um exemplo de esquecimento de dados é apresentado no trabalho de Bertini, Lopes e Zhao (2012). Nesse trabalho é apresentado um modelo baseada em redes complexas onde se cria um grafo, chamado grafo K-Associados, para a tarefa de classificação semissupervisionada em fluxo de dados. Bertini, Lopes e Zhao (2012) utilizaram técnica de "esquecimento" de dados e mostraram que, além de minimizar o consumo de memória, o processo de "esquecimento" é eficaz para a atualização do modelo na ocorrência de alterações de conceitos.

O restante do capítulo está organizado da seguinte forma. Na Seção 4.1 são apresentados os trabalhos relacionados com o problema de extração de tópicos considerando documentos em fluxo. Na Seção 4.2.2 é apresentada a versão online do LDA, a formulação e o algoritmo de inferência baseado no método de inferência variacional. Na Seção 4.3 é descrito a versão *online* do algoritmo PBG desenvolvido no contexto testa tese, a sua fundamentação e a descrição das operações. Na Seção 4.4 são apresentados os experimentos comparativos entre as versões *online* do LDA e do PBG. E por fim, na Seção 4.5 são apresentadas as conclusões.

4.1 Trabalhos relacionados

No levantamento bibliográfico, encontrou-se vários trabalhos recentes que estendem ou modificam o modelo LDA. Isso revela que este modelo representa o estado da arte em extração de tópicos. Uma vantagem considerável no modelo LDA é sua capacidade de extensão para outros domínios, como processamento de imagem (MACKEY, 2007; FRIEDLANDER

et al., 2012) e detecção de comunidades (LI *et al.*, 2012; YAN *et al.*, 2012). Por exemplo, no trabalho de Li *et al.* (2012), é criado um modelo para detecção de comunidades baseado no LDA. Neste modelo, é considerado a relação entre os autores, documentos e seus tópicos – são utilizadas quatro variáveis latentes, para os documentos, palavras, autores e tópicos. Além disso, com a marcação de tempo de cada documento, é identificada a evolução das comunidades ao longo do tempo. Outro trabalho que examina como os tópicos podem ser misturados com comunidades é descrito por Yan *et al.* (2012). Esse trabalho relaciona tópicos e comunidades de pesquisadores (rede de autores). É abordado um método híbrido para a extração de tópicos e a detecção de comunidades. No trabalho de Daud (2012), é apresentado um modelo de tópico chamado *Temporal-Author-Topic* (TAT) em que, simultaneamente, modela textos, autores e a marcações de tempo dos documentos. O objetivo é descobrir os autores relacionados em um domínio específico em diferentes anos. No trabalho de Blei e Lafferty (2006) é apresentado o *Dynamic Topic Model* (DTM). Esse modelo captura a evolução dos tópicos ao longo do tempo. Para isso, os dados são divididos em janelas do tempo. Por exemplo, os documentos podem ser divididos pelo ano de publicação. Em cada janela do tempo, formam-se os lotes de documentos e é aplicado o algoritmo de inferência. Em uma janela no tempo t, cria-se o modelo para k tópicos, onde os tópicos associados ao tempo t evoluíram dos tópicos encontrados na janela $t-1$. Esse modelo apresentou bons resultados para descrever um corpos de documentos e a evolução de seus K tópicos. Todos esses trabalhos citados anteriormente utilizam modelos probabilísticos de tópicos e consideram a sequência temporal dos documentos, entretanto, são métodos iterativos e que não processam os dados eficientemente em fluxo.

Já no trabalho de AlSumait, Barbará e Domeniconi (2008), é apresentado um modelo de tópicos baseado no LDA capaz de identificar tópicos emergentes e suas alterações ao longo do tempo. Inicialmente, com uma coleção de documentos, é aplicado o algoritmo de inferência sobre o modelo LDA. Em seguida, com o decorrer do fluxo de dados, novos documentos são agrupados em lotes sobre o mesmo intervalo de tempo. Esses documentos são adicionados ao modelo, reaproveitando os parâmetros já inferidos anteriormente. O método utiliza-se da representação matricial da distribuição dos termos por tópicos para calcular a alteração dos tópicos sobre o tempo. Cada matriz é obtida como resultado da inferência do amostrador de Gibbs em uma partição do tempo. Alterações nas distribuições de duas matrizes de tempos consecutivos são calculadas e, a porcentagem de alterações, podem indicar alterações de conceitos. Um problema recorrente nos modelos baseados no LDA, e consequentemente encontrado nesse modelo, é a obrigatoriedade de vocabulário e número de tópicos fixos.

Além de encontrar trabalhos que alteram o modelo LDA para o contexto dinâmico, foram encontrados trabalhos que propõe métodos de inferência otimizados para o caso de dados em fluxo. No trabalho de Yao, Mimno e McCallum (2009), são avaliados vários métodos de inferência de modelos de tópicos. Todos os algoritmos avaliados requerem uma fase de treinamento. No caso dos métodos de amostragem, inicialmente, um conjunto de dados é utilizado no treino de um amostrador de Gibbs. Em seguida, com os parâmetros inferidos, e com a distribuição

de palavras por tópicos, foram aplicados métodos de classificação para descobrir a distribuição dos tópicos para os documentos que chegam no fluxo. A maior contribuição do trabalho de Yao, Mimno e McCallum (2009) é a descrição de uma versão eficiente, tanto em tempo quanto em memória, do amostrador de Gibbs. Além disso, foram realizados vários experimentos empíricos com o objetivo de explorar a sensibilidade de cada método de inferência na variação do número de tópicos, na proporção entre documentos de treino por novos documentos e no efeito das alterações na distribuição dos tópicos no fluxo.

Outro trabalho que trata do método de inferência foi desenvolvido por Canini, Shi e Griths (2009). Nesse trabalho, são avaliados os modelos online de inferência baseado no LDA. Além disso, é proposto um novo método de inferência baseado em filtros de partículas (DOUCET; FREITAS; GORDON, 2001). O que há de interessante nesse trabalho é a sugestão de um processo de "rejuvenescimento" para atualizar o modelo. O rejuvenescimento corresponde a reamostragem das distribuições de tópicos diante dos novos dados. Apesar dessa vantagem, ainda existe o problema da grande quantidade de parâmetros requerida pelo modelo.

O algoritmo de inferência variacional transforma o problema de inferência em um problema de otimização. Devido a essa transformação, a ordem dos documentos nos lotes podem influenciar a qualidade do resultados. No trabalho de Wahabzada *et al.* (2011), é discutido o processo de escalonamento de documentos para inferência no método variacional. Nesse trabalho, o LDA estático é transformado em um método "quase *online*" que forma mini lotes de documentos mais influentes processados antes dos menos influentes, chamado isLDA. Nesse mesmo trabalho, o isLDA foi estendido para um novo modelo *online*, chamado isoLDA. Esse modelo aplica o escalonamento de documentos mais influentes em lotes formados aleatoriamente.

O modelo LDA é bastante aplicado na identificação de tópicos em notícias eletrônicas. Por isso, um problema recorrente é a dessincronização das diversas fontes de notícias, ou seja, um conjunto de documentos advindo de um canal *online* de notícias pode apresentar diferenças no conteúdo conceitual dos seus tópicos – um canal pode dar mais importância para um tipo de notícia do que outros. Não é trivial explorar a correlação existente em múltiplos fluxo assíncronos; os documentos sobre um mesmo tópico podem chegar de diferentes canais e ter diferentes marcações de tempo. No trabalho de Wang *et al.* (2009), esse problema é formalmente tratado, sendo apresentado um novo algoritmo baseado em modelos de tópicos probabilísticos. O modelo desenvolvido é baseado no modelo pLSA. O algoritmo proposto, em uma primeira fase, extrai os tópicos dos documentos desprezando a marcação de tempo. Em uma segunda fase, a marcação de tempo de todos os documentos são sincronizados e faz-se a combinação daqueles tópicos mais correlatos. Com isso, o algoritmo prossegue iterando sobre essas duas fases até alcançar um ponto de convergência. Relacionado a esse mesmo problema, tem-se o trabalho de Hong *et al.* (2011). O modelo proposto é uma extensão do LDA. Nesse modelo são considerados múltiplos fluxos de dados. Associado a cada fluxo está um conjunto de tópicos locais e, associado a todos os fluxos está um conjunto de tópicos globais. Para identificar a dinâmica temporal, cada

tópico encontrado é associado a uma função que retorna sua popularidade sobre o tempo. Quanto a inferência, ela não é aplicada de forma incremental, mas sim de forma estática com pequenos lotes de documentos.

Quando se aplica o modelo *online* em lotes de documentos, cada lote pode ser considerado uma janela do tempo. Essa janela é uma discretização do tempo, sendo a granularidade definida para alguma época – por exemplo dia, mês, ano, *etc*. Essa forma de discretizar o tempo de chegada dos documentos no fluxo foi estudada no trabalho de Iwata *et al.* (2012). Nele, foi definido um modelo baseado no LDA que analisa a evolução dos tópicos em múltiplas granularidades de tempo. O modelo proposto utiliza os tópicos anteriores e atuais para definir *a priori* para extração em épocas futuras. Já no trabalho de Iwata *et al.* (2010), foi proposto um modelo de tópicos que permite uma análise sequencial da dinâmica dos tópicos com múltiplas granularidades. A granularidade dinâmica de um tópico é proporcional a frequência com que as palavras aparecem em uma janela do tempo. Por exemplo, um tópico relacionado com "política" pode conter palavras como "senador" ou "presidente" frequentes por vários anos, por outro lado, o nome de um assessor parlamentar pode ser frequente por um curto período de tempo. Para isso, as palavras dos tópicos são distribuídas sobre a escala de tempo. Aquelas palavras populares por longo período ficarão com alta probabilidade quando se escolhe várias janelas de tempo.

Outro trabalho que estende o LDA *online* é o trabalho de Wang e Lang (2011). O modelo proposto é capaz de descobrir tópicos com unigramas, assim como bigramas, e é atualizado por um algoritmo de inferência *online*. O algoritmo de inferência é uma versão do amostrador de Gibbs que aproveita parâmetros obtidos na inferência feita em uma janela anterior do tempo.

Uma nova tendência para os modelos probabilístico de tópicos é o uso do processo de Dirichlet (DP). Essa técnica é não paramétrica e é usada para agrupamento onde não se conhece o número K de grupos. Assim como o LDA é o modelo base nos modelos probabilísticos de tópicos, o *Hierarchical Dirichlet Processes* (HDP) (TEH *et al.*, 2006) é o modelo base de modelos probabilísticos não paramétricos de tópicos. No trabalho de Ahmed e Xing (2012), foi desenvolvimento um modelo baseado no HDP capaz de detectar a "morte" e o "nascimento" de tópicos em fluxo de lote de documentos. Apesar dessas vantagens, os algoritmo de inferência em modelos HDP possuem desafios quanto a tempo de execução e consumo de memória, isso por requererem múltiplos passos por todo o conjunto de dados (WANG; PAISLEY; BLEI, 2011). Para resolver esse problema, assim como no LDA, onde foram propostos métodos de inferência *online*, foi proposto um algoritmo *online* de inferência variacional para o HDP.

Modelos probabilísticos de tópicos baseados no LDA assumem que o vocabulário de palavras é conhecido. Entretanto, quando se considera que os documentos chegam no fluxo, pouco se sabe sobre o vocabulário. O único trabalho encontrado na literatura que trata dessa lacuna é o trabalho de Zhai e Boyd-graber (2013). Nesse trabalho é estendido o método de inferência variacional *online* do LDA para considerar vocabulário infinito. Para considerar novas palavras, a distribuição de tópicos por palavras é gerada pelo processo de Dirichlet. Esse

processo é o mesmo em que se baseia o modelo HDP, entretanto, nesse caso, a distribuição não-paramétrica é aplicada na distribuição de tópicos por palavras. Com isso, o algoritmo de inferência é complexo devido a incorporação de procedimentos para dinamicamente ordenar, expandir, contrair o conjunto de palavras do vocabulário. Os experimentos realizados averiguaram a representatividade dos documentos na tarefa de redução de dimensionalidade e calcularam os valores do PMI dos tópicos. Os resultados indicaram que o modelo obtêm melhores valores de acurácia e PMI em comparação com a versão *online* do LDA com vocabulário finito.

Nesta revisão não foram encontrados apenas trabalhos que utilizam modelos probabilísticos. No trabalho de Saha e Sindhwani (2012) foi proposto um método para extração de tópicos baseado no método NMF. A matriz documento-termo é construída com documentos obtidos em janelas de tempo, onde se aproveita a matriz de janelas anteriores, removendo documentos antigos e adicionando novos. Esse método possibilita a detecção da evolução e de tópicos emergente pela análise das matrizes decompostas. Foi encontrado também métodos que utilizam a tradicional contagem da frequência de termos. No trabalho de Ma (2011), é apresentado um novo método de extração de tópicos emergentes. Os tópicos são os termos com frequência superior a um limiar pré-definido. É proposto um novo método para contagem de termos frequentes chamado TF-PDF (*Term Frequency, Proportional Document Frequency*). Já no trabalho de Bun e Ishizuka (2002), é apresentado um *framework* para auxiliar na extração dos tópicos em notícias semanais. O método considera múltiplos fluxos de dados (vários canais de notícias). O *framework* funciona da seguinte forma: o fluxo de notícias textuais chega e, com isso, é calculado o TF-PDF dos termos, em seguida, são extraídas as sentenças, ou frases das notícias, que possuem os termos de maior relevância (ou maior peso). Sobre essas sentenças é aplicado um algoritmo de agrupamento, sendo os tópicos a sumarização de cada grupo.

4.2 Aprendizado online

Os algoritmos *online* modernos de aprendizado de máquinas baseiam-se na teoria da aproximação estocástica. Nesta seção é descrito o arcabouço geral dos algoritmos *online* de aprendizado. Inicialmente é descrita a tarefa de aprendizado, com a descrição de uma função geral baseada na minimização do erro e a aplicação do método de gradiente descendente estocástico nessa função geral. Em seguida, é descrita a versão *online* do LDA (oLDA), que utiliza aproximação estocástica para otimizar o problema de otimização estabelecido pelo método de inferência variacional.

4.2.1 Otimização Estocástica

Na tarefa tradicional de aprendizado de máquina, cada exemplo é um par (x, y) composto por um conjunto de atributos x e a informação de classe y. Considere uma função $erro(\hat{y}, y)$ que mede a perda ao se estimar uma classe $\hat{y}$ conhecendo a classe real y. O objetivo é encontrar uma

função f_v, parametrizada por um vetor de pesos v, que minimiza $Q((x,y),v) = erro(f_v(x),y)$. Assim, dado um conjunto de treino $\{(x_1,y_1),\ldots,(x_n,y_n)\}$, pode-se calcular o risco empírico $E(f_v)$ da seguinte forma

$$E(f_v) = \frac{1}{n}\sum_{i=1}^{n} erro(f_v(x_i),y_i). \tag{4.1}$$

O risco empírico $E(f_v)$ pode ser minimizado utilizando o método de gradiente descendente. Nesse método, em cada iteração t, atualiza-se o vetor de pesos v em direção do gradiente de $E(f_v)$,

$$v^{(t+1)} = v^{(t)} - \rho_t \frac{1}{n}\sum_{i=1}^{n} \nabla_v Q((x_i,y_i),v^{(t)}), \tag{4.2}$$

onde ρ é a taxa com que o erro será considerada na atualização dos vetores de pesos.

Algoritmos de otimização estocástica seguem um processo não determinístico para estimar o gradiente de $E(f_v)$. Em vez de calcular exatamente o gradiente de $E(f_v)$, em cada iteração é estimado esse gradiente com base em um simples exemplo (x_t,y_t) escolhido aleatoriamente (BOTTOU, 1998):

$$v^{(t+1)} = v^{(t)} - \rho_t \nabla_v Q((x_t,y_t),v^{(t)}). \tag{4.3}$$

Com isso, espera-se que a Atualização 4.3 se comporte como a Atualização 4.2, apesar do ruído de gradiente inserido. Algoritmos baseados no gradiente estocástico utilizam estimativas instantâneas do gradiente, o que implica que o vetor com a direção de atualização está sujeito a flutuações aleatórias denominadas ruído de gradiente (BOTTOU, 2010). Por outro lado, esse processo depende apenas do exemplo escolhido aleatoriamente no momento t, não sendo necessário percorrer os exemplos das iterações anteriores. Por esse motivo, o gradiente estimado é mais fácil de computar.

A convergência do gradiente estocástico foi bastante estudada, principalmente nos trabalhos de Bottou (1998), Bottou (2004), Bottou (2010). Bottou demonstrou que o gradiente estocástico tende a uma solução ótima global v^* de $E(f_{v^*})$, caso a função Q seja uma função convexa, caso contrário, tende a uma solução ótimo local. Essa convergência é garantida desde que a taxa de erro ρ diminua ao longo das iterações, satisfazendo as seguintes condições: $\sum_t \rho_t^2 < \infty$ e $\sum_t \rho_t = \infty$.

A desvantagem do processo estocástico está na velocidade de convergência, que pode se tornar lenta devido ao ruído aplicado pelo cálculo do gradiente aproximado. Além disso, a velocidade com que o valor da taxa de erro ρ diminui influencia na velocidade de convergência.

Uma técnica comum em aprendizado estocástico é utilizar *mini-batches* para atualizar o modelo. Com isso, em vez de utilizar apenas um exemplo por vez, utiliza-se vários exemplos com o objetivo de diminuir o ruído. Utilizando um *minibatch* com b exemplos, a aproximação

do gradiente é a seguinte

$$v^{(t+1)} = w^{(t)} - \rho_t \frac{1}{b} \sum_{i=1}^{b} \nabla_v Q((x_i, y_i), v^{(t)}). \tag{4.4}$$

A média do gradiente estocástico dos b exemplos possuem o mesmo valor esperado, logo a aproximação ainda é válida.

4.2.2 Aprendizado online com o LDA

As técnicas tradicionais de inferência do LDA, como o algoritmo de amostragem de Gibbs e o algoritmo de inferência variacional, requerem uma passagem completa por toda a coleção de documentos em cada iteração. Tanto para o algoritmo de Gibbs quanto para o algoritmo variacional é necessário percorrer todos os termos de cada documento da coleção. Isso claramente pode retardar o processamento em grandes coleções de documentos, e também não é adequado aplicar tais técnicas onde novos documentos estão constantemente chegando. Assim, para resolver esses problemas, Hoffman, Blei e Bach (2010) propôs uma versão *online* do LDA. O algoritmo para inferência *online* do LDA proposto por Hoffman baseia-se na utilização da técnica de otimização estocástica para resolver o problema de otimização estabelecido no método de inferência variacional. Algoritmos de otimização estocástica seguem um processo não determinístico para estimar o gradiente de uma função objetivo. Inferência variacional estocástica provê uma abordagem escalável e muito mais eficiente para aproximar a distribuição *a posteriori* do LDA.

O método de inferência variacional tradicional aplicado no LDA e a notação utilizada nesta seção são descritos com detalhes no Capítulo 2. Na versão *online* em vez de otimizar todo o conjunto de dados, o método de inferência variacional estocástico utiliza apenas uma amostra dos dados escolhidos aleatoriamente (BOTTOU, 2004). Uma amostra pode ser um documento único ou um subconjunto de documentos da coleção. Com apenas as estatísticas obtidas por uma amostra, o algoritmo ajusta as variáveis variacionais. É importante distinguir as variáveis variacionais entre variáveis locais e globais (Veja na figura 2 a representação das variáveis variacionais do modelo LDA). As variáveis locais mantém estatísticas de cada documento especificadamente e correspondem a distribuição variacional γ e φ. As variáveis globais são as proporções que relacionam tópicos e palavras, correspondente a distribuição λ.

Note que as atualizações das variáveis locais (equações 2.48 e 2.47) utilizam apenas estatísticas do documento que está sendo processados. Assim, na versão online, é amostrado um documento d_j (ou um sub-conjunto de documentos) da coleção e computado os parâmetros γ e φ utilizando a mesma sub-rotina da versão em lote. A variável $\hat{\lambda}$ é criada para manter as estatísticas dos tópicos obtidos pelas variáveis locais,

$$\hat{\lambda} = \eta + n \sum_{i=1}^{N} \varphi_{j,i,k} w_{i,j}, \tag{4.5}$$

onde n é o número de documentos, considerando que n é um número grande para justificar o processamento *online*. Essa equação vem da Equação 2.49 aplicado n vezes para um documento amostrado.

Agora, para atualizar as variáveis globais é necessário aplicar a interpolação da variável $\hat{\lambda}$ com a variável global λ,

$$\lambda_k^{(t+1)} = (1-\rho_t)\lambda_k^{(t)} + \rho_t \hat{\lambda}_k, \tag{4.6}$$

onde ρ_t é o fator de aprendizado.

Note que a atualização da variável global λ em um momento $(t+1)$ utiliza estatística obtidas em um momento anterior (t), e que essa atualização não requer a passagem por toda a coleção de documentos. Em cada momento, novas amostras de documentos são retiradas da coleção (ou do fluxo de documentos) e são atualizadas as estatísticas locais e globais.

No Algoritmo 12 estão descritos os passos para a inferência *online* do LDA. A convergência desse algoritmo é garantida pelas propriedades de otimização estocástica (veja Seção 4.2.1).

Algoritmo 12: *online* LDA

Entrada:
Um fluxo de documentos textuais S
Número de tópicos K
hiper-parâmetros α e β

Saída :
estimativa das distribuições documento-tópicos θ e tópico-palavra ϕ

início
  Inicializa $\lambda^{(0)}$ aleatoriamente ;
  repita
    Amostre um documento d_j do fluxo de documents ;
    Inicialize $\gamma_{j,k} = 1$, para $k \in \{1,\ldots,K\}$;
    repita
      para *cada termo* w_i *do documento* d_j **faça**
        para $k = 1$ *até* K **faça**
          $\varphi_{j,i,k} \propto \exp\left(\Psi(\gamma_{j,k}) - \Psi(\sum_{l=1}^{K}\gamma_{j,l}) + \Psi(\lambda_{k,i})\right)$;
        $\gamma_j = \alpha + \sum_{i=1}^{N_{d_j}} \varphi_{j,i}$;
    até *convergência dos parâmetros locais*;
    para $k = 1$ *até* K **faça**
      $\hat{\lambda}_k = \beta + D\sum_{i=1}^{N_{d_j}} \varphi_{j,i,k} w_{j,i}$;
    $\lambda^{(t)} = (1-\rho_t)\lambda^{(t-1)} + \rho_t \hat{\lambda}$;
  até *existir documento no fluxo*;

A fundamentação da versão online do LDA é baseada na aplicação da otimização estocástica na otimização estabelecida pelo método de inferência variacional do LDA. No caso

do método de inferência variacional, objetiva-se otimizar o logaritmo da verossimilhança do modelo. Essa otimização é obtida pela maximização do ELBO, definido pela Equação 2.46. Aqui, é reescrito o ELBO $\mathscr{L}$ em função dos parâmetros variacionais locais, γ e φ, e do parâmetro global λ. Então, para o LDA, pode-se escrever o ELBO como:

$$\mathscr{L}(\gamma, \varphi, \lambda) \triangleq \sum_{d_j \in D} l(\gamma_j, \varphi_j, \lambda), \tag{4.7}$$

onde $l(\gamma_j, \varphi_j, \lambda)$ é a contribuição do documento d_j para o ELBO. Como a ocorrência das palavras para um documento d_j é observado, é possível aplicar o $E-step$ semelhante ao processo em lote para encontrar os parâmetros locais γ e φ, e mantendo o parâmetro global λ fixo. A variável intermediária $\hat{\lambda}$ mantém uma estimativa para o parâmetro global λ utilizando estatísticas da amostra e das variável locais calculadas. Com isso, é possível atualizar $\lambda^{(t+1)}$ usando uma média ponderada entre seu valor anterior, $\lambda^{(t)}$, e $\hat{\lambda}$,

$$\lambda^{(t+1)} = \lambda^{(t)} + \rho_t D \nabla_\lambda l(\gamma_j, \varphi_j, \lambda), \tag{4.8}$$

onde $\rho \triangleq (\tau_0 + t)^{-\kappa}$, $\kappa \in (0.5, 1]$ controla a taxa de esquecimento dos valores antigos de $\hat{\lambda}$ e $\tau_0 \leq 0$ diminui a importância das iterações iniciais. A condição de que $\kappa \in (0.5, 1]$ é necessária para garantir a convergência. O valor de ρ decresce ao longo do tempo. Note que ρ pode ser considerado como o coeficiente de aprendizagem do parâmetro λ.

O Algoritmo 12 também pode ser justificado pelo trabalho de Neal e Hinton (1998), nos quais foram apresentadas provas que demonstram o porquê de versões *online* dos algoritmos baseados em EM (*Expectation Maximization*) funcionarem. Considerando o contexto no qual um algoritmo tradicional de EM é aplicado, tem-se uma variável aleatória observada Z e uma outra variável aleatória não observada Y. Assume-se que a probabilidade conjunta de Y e Z é parametrizada usando uma distribuição θ, como $p(Z,Y|\theta)$. A probabilidade marginal de Z é então $P(Z|\theta) = \sum_y p(y,z|\theta)$. Com os dados observados, Z, deseja-se encontrar o valor de θ que maximiza o logaritmo da verossimilhança do modelo, $L(\theta) = \log P(z|\theta)$. Para as versões *online* do algoritmo EM, deseja-se encontrar θ, dado um número independente de dados decompostos como um fluxo do tipo $(Z_1, \ldots)$, assim como as variáveis não observadas podem ser decompostas como $(Y_1, \ldots)$. Utilizando as variáveis decompostas, tem-se uma estimativa $\hat{P}(Y) = \prod_i \hat{P}_i(Y_i)$, na qual uma função F retornará o valor dessa estimativa parametrizada por θ, $F(\hat{P}, \theta) = \sum_i F_i(\hat{P}_i, \theta)$. Então, usando o Teorema 2 apresentado no trabalho de Neal e Hinton (1998), é mostrado que: se $F(\hat{P}, \theta)$ tem máximo local em $\hat{P}^*$ e θ^*, então $L(\theta)$ também terá o máximo em θ^*. Assim, um algoritmo EM pode ser usado para otimizar a decomposições dos dados observados, e também otimizar a verossimilhança do modelo.

4.3 Algoritmo de Propagação em Grafos Bipartidos para extração de tópicos em fluxo

Para tratar dados em fluxo, o algoritmo PBG é ineficiente devido a necessidade de realizar as propagações locais para todos os conjuntos de documentos antes de realizar as propagações globais. Para resolver esse problema, nesta seção é apresentado o algoritmo *online Propagation on Bipartite Graph* (oPBG), que é uma extensão da versão não-supervisionada do algoritmo PBG (Seção 3.3) para o processamento em fluxo de documentos. A notação geral utilizada nesta seção é a mesma descrita na Seção 3.1.

4.3.1 Formulação do oPBG

Para a construção do algoritmo oPBG, é utilizado o método de gradiente estocástico para otimizar a função objetivo descrita na Equação 3.15. Em vez de computar o gradiente $\nabla Q(G)$ para todo o grafo bipartido G, cada iteração estima seu gradiente em apenas uma amostra do grafo. O método de otimização estocástica segue a estimativa do gradiente em direção do ótimo. Essa estimativa é ruidosa pois depende dos dados escolhidos aleatoriamente em cada iteração. Porém, essa estimativa é mais barata de computar do que o verdadeiro gradiente, e seguindo essa estimativa é possível escapar de ótimos locais (HOFFMAN *et al.*, 2013).

A amostra escolhida aleatoriamente é um subgrafo induzido de G, definido como $G_{d_j} = (\mathscr{V} = \{\{dj\} \cup \mathscr{W}_{d_j}\}, \mathscr{E}, f)$, onde $\{d_j\}$ é o conjunto unitário formado pelo vértice d_j e $\mathscr{W}_{d_j}$ é o conjunto de palavras que ocorrem no documento d_j. Com isso, o grafo G é formado por vários subgrafos G_{d_j}, $G = \cup_{d_j \in \mathscr{D}} G_{d_j}$, e também é possível reescrever a função objetivo Q (Equação 3.15) como a soma de vários subgrafos de G,

$$Q(G) = \sum_{d_j \in \mathscr{D}} Q(G_{d_j}). \tag{4.9}$$

Aqui, o objetivo é encontrar as equações de atualização para os vetores A_j, B_i e $C_{j,i}$ que otimizam a função Q. Isso é alcançado quando

$$\nabla Q(G) = 0. \tag{4.10}$$

Para calcular o gradiente ruidoso (ou gradiente estocástico) é necessário amostrar um subgrafo G_{d_j}, e a partir desse grafo obter uma estimativa do gradiente $\hat{\nabla} Q(G)$. Essa estimativa é igual a esperança do gradiente aplicado no subgrafo G_{d_j}, $\hat{\nabla} Q(G) = E\left[\nabla G_{d_j}\right]$.

Para otimizar $Q(G)$ em relação ao vetor A_j utilizando o gradiente estimado, basta calcular a derivada de $Q(G_{d_j})$ em relação a A_j. Tomando a derivada em relação a A_j, tem-se

$$\frac{\partial Q_{[A_j]}}{\partial A_j} = \frac{1}{A_j}\left(\sum_{w_i \in \mathscr{W}_{d_j}} f_{j,i} C_{e_{j,i}} - A_{j,i} + \alpha\right). \tag{4.11}$$

Definindo essa equação igual a zero, obtêm-se o valor máximo em:

$$A_j = \alpha + \sum_{w_i \in \mathscr{W}_{d_j}} f_{j,i} C_{e_{j,i}}. \tag{4.12}$$

Finalmente, é maximizada a Equação 3.13 em relação a B_i, vetor associado à palavra $w_i \in \mathscr{D}$. Para isso, é definido o vetor intermediário $\hat{B}_i$, que é a estimativa de B_i obtida pela amostra do subgrafo G_{d_j}. Tomando a derivada de $Q_{[\hat{B}_i]}$ e ignorando a constante λ_k para se obter uma estimativa do gradiente

$$\frac{\partial Q_{[\hat{B}_i]}}{\partial \hat{B}_{i,k}} = \sum_{k=1}^{K} \frac{f_{j,i} C_{e_{j,i}}}{B_{j,i}} + \lambda_k \tag{4.13}$$

Definindo essa equação igual a zero, e resolvendo λ_k, tal que $\lambda_k = \sum_{k^*=1}^{K} f_{j,i} C_{e_{j,i}}$. Desde que $\sum_{i=1}^{n_{d_j}} B_{i,k} = 0$, pode-se ignorar λ_k para estimar o valor de $B_{i,k}$

$$\hat{B}_{i,k} \propto \sum_{k=1}^{K} f_{j,i} C_{e_{j,i}}. \tag{4.14}$$

Para evitar a iteração sobre toda a estrutura do grafo, o vetor B_i é atualizado usando a média ponderada de seu valor anterior e o valor estimado $\hat{B}_i$. Essa é uma forma incremental de atualizar o vetor B_i aproveitando estatísticas obtidas em iterações anteriores. Assim, em uma iteração $t+1$, o gradiente ruidoso é utilizado para atualizar o vetor B_i da seguinte forma,

$$\begin{aligned} B_i^{(t+1)} &= B_i^{(t)} + \rho_t \left(\hat{B}_i - B_i^{(t)} \right) \\ &= (1-\rho_t) B_i^{(t)} + \rho_t \hat{B}_i. \end{aligned} \tag{4.15}$$

onde $\rho \triangleq (\tau_0 + t)^{-\kappa}$ define o passo em direção ao gradiente. A taxa de aprendizagem κ controla o quão rápido novos vetores substituem os antigos. A inércia de aprendizagem $\tau_0 \geq 0$ diminui a propagação antecipada e previne convergências prematuras (BOTTOU, 1998; HOFFMAN; BLEI; BACH, 2010).

A convergência é garantida quando a taxa de aprendizado é decrescente satisfazendo as condições $\sum_t \rho_t^2 < \infty$ e $\sum_t \rho_t = \infty$; isso é obtido quando $\kappa \in (0.5, 1]$ (BOTTOU, 2004).

4.3.2 Descrição do Algoritmo

Nesta seção é descrito o algoritmo oPBG, Algoritmo 13. O oPBG considera que o fluxo é formado por pequenas quantidade de documentos, chamados de *mini-batchs*. Considerar múltiplos dados observados para atualização é uma técnica comum em aprendizado estocástico pois reduz o ruído (BOUSQUET; BOTTOU, 2008; LIANG; KLEIN, 2009). Suponha que seja escolhido aleatoriamente um *mini-batch* $\mathscr{T}$ de tamanho b do fluxo de documentos S. Para esse *mini-batch* é criado um grafo $G_{\mathscr{T}} = ((\mathscr{T} \cup \mathscr{W}_{\mathscr{T}}), \mathscr{E}, f)$. Aqui assume-se que vocabulário $\mathscr{W}$ de

palavras do fluxo é conhecido, e que também a cada palavra $w_i \in \mathscr{W}$ existe um vetor B_i associado. Considerando os vetores B_i's e o grafo $G_{\mathscr{T}}$, é possível obter o vetor A_j associado ao vértice d_j sem alterar os procedimentos de propagação local que otimiza a Equação 3.15. A maximização da Equação 3.15 em relação a A_j mantém-se a mesma, assim como os procedimentos de propagação local são os mesmos descritos na Seção 3.3. Esses procedimentos mantêm-se pois as iterações locais operam apenas sobre as palavras do conjunto $\mathscr{W}_{\mathscr{T}}$. Da mesma forma, se mantêm a atualização dos vetores $C_{e_{j,i}}$ na propagação local.

A propagação global é realizada com o grafo $G_{\mathscr{T}}$ para obter a estimativa $\hat{B}_i$ dos vetores associados as palavras no conjunto $W_{\mathscr{T}}$. Essa estimativa é usada em seguida para atualizar os vetores B_i's associados a todas palavras w_i do vocabulário.

Algoritmo 13: Algoritmo oPBG

Entrada:
fluxo de documentos S,
K // dimensões de A_j e B_i
α // parâmetro de concentração

Saída :
rótulos multidimensionais A_j para cada documento d_j no fluxo,
rótulos multidimensionais B_i para cada termo w_i

início
  Inicie os vetores $B_i^{(0)}$ para cada palavra $w_i \in \mathscr{W}$;
  $t \leftarrow 0$;
  repita
    Amostre um *mini-batch* $\mathscr{T}$ de documentos do fluxo S ;
    Cria o grafo $G_{\mathscr{T}}$ com documentos e palavra de $\mathscr{T}$;
    Inicia vetores A_j para cada documento $d_j \in \mathscr{T}$;
    para cada $d_j \in \mathscr{T}$ **faça**
      repita
        $A_j \leftarrow$ `localPropag(`$G_{\mathscr{T}}$`,`d_j`,` A_j`,` B`)`;
      até A_j *não converge*;
    $\hat{B} \leftarrow$ `globalPropag(`$G_{\mathscr{T}}$`,`A`,`B`)`;
    $t \leftarrow t+1$;
    para cada $w_i \in \mathscr{W}_{\mathscr{T}}$ **faça**
      $B_i^{(t)} \leftarrow (1-\rho_t)B_i^{(t-1)} + \rho_t \hat{B}_i$;
    normaliza os vetores $B_i^{(t)}$ para todos tópicos k tal que $\sum_{w_i \in \mathscr{W}} B_{i,k}^{(t)} = 1$;
  até *enquanto existir documento no fluxo*;

4.4 Resultados Experimentais

Nesta seção são apresentados os resultados experimentais da versão online do algoritmo PBG na tarefa de redução de dimensionalidade e extração de tópicos. Os experimentos foram conduzidos usando o conjunto de dados descritos na Tabela 2. Esses dados foram pré-processados,

stop-words foram removidas e termos foram "*stemizados*" por meio do algoritmo de Porter (PORTER, 1997). As frequências das palavras em cada documento foram usadas como peso das arestas do grafo bipartido.

O objetivo dos experimentos foi comparar os resultados do oPBG com a versão online do LDA (oLDA). Um fator crítico para as comparações é o parâmetro $\rho = (\tau_0 + t)^{-\kappa}$, que define o "tamanho do passo" na direção do gradiente. Dependendo do valor de ρ, a velocidade de convergência é menor, e consequentemente os resultados finais podem ser piores. Por isso, foram definidos os seguintes conjunto de valores para esses parâmetros: $\kappa \in \{0.5, 0.6, 0.7, 0.8, 0.9\}$ e $\tau \in \{1, 64, 256, 1024\}$. Esses valores são os mesmos utilizados para o oLDA no trabalho de (HOFFMAN; BLEI; BACH, 2010).

Para simular o fluxo, os documentos foram alinhados em ordem estabelecida aleatoriamente, essa mesma ordem foi utilizada em ambos algoritmos. Uma técnica comum em aprendizado estocástico é considerar múltiplas observações por atualizações para reduzir o ruído. Essas múltiplas observações são subconjuntos de documentos que chegam no fluxo, chamados *mini-batches*. A quantidade b de documentos em cada *mini-batch* foi pré-estabelecida de forma que $b \in \{32, 64, 128, 256, 1024\}$. Um subgrafo bipartido é criado para os b documentos que chegaram no fluxo, e sobre esse subgrafo é executado a operação de propagação local. Em seguida, aproveitando a estimativa dos vetores associados as palavras, é realizado a propagação global. Os resultados foram as atribuições dos vetores A_j e B_i que otimizam a Equação 3.15.

Os rótulos dos documentos foram ocultados e foram posteriormente utilizados apenas para auxiliar na avaliação dos resultados na representatividade dos documentos. Veja Seção 3.3.5.2 sobre a avaliação sobre representatividade dos documentos. Para avaliar os tópicos foi utilizada a medida NPMI, essa medida aproxima a avaliação humana de tópicos, como já comentado na Seção 3.3.5.3.

Na tarefa de extração de tópicos, foram definidos os número de tópicos K dado pelo conjunto de valores $\{50, 100, 150, 200\}$. Foi utilizada a implementação em *python* do LDA *online* com otimização de hiperparâmetros fornecida pela ferramenta Gensim [1]. Essa implementação é baseada no método de inferência variacional *online* proposto por Hoffman, Blei e Bach (2010). Os hiperparâmetros do LDA foram inicialmente $\alpha = \beta = \frac{1}{K}$. Na versão *online* do LDA, a coleção de documentos é percorrido apenas uma vez para estimar a distribuição ϕ (distribuição de tópicos por palavras), porém, a estimativa do parâmetro θ (distribuição de documentos por tópicos) é iterativa. Assim, foi definido o máximo de 100 iterações ou a convergência. Para o algoritmo oPBG, foi utilizado o valor de $\alpha = 0.005$. Esse valor foi obtido pela análise dos resultados da versão não-supervisionada. O critério de parada foi definido para 100 iterações locais. Essas iterações locais são realizadas para os S documentos que chegaram no fluxo.

[1] Gensim: <https://radimrehurek.com/gensim/index.html">

4.4.1 Avaliação da representatividade dos documentos

Para se ter a percepção dos resultados quanto as variações dos parâmetros, foram realizados experimentos combinando todos os valores dos parâmetros τ_0, κ, b e K. Para cada conjunto de dados, foram executadas todas as 400 combinações desses parâmetros para os métodos oLDA e oPBG.

Os resultados obtidos estão ilustrados nas figuras 15, 16 e 17. Nessas figuras estão os valores de acurácia obtidos para todas as combinações de parâmetros κ, τ, b e K. Os resultados computacionais encontrados mostraram que o oPBG alcançou melhores resultados de acurácia na representatividade dos documentos, e também, na maioria dos casos, sofreu menos alterações quanto as variações dos parâmetros.

As figuras 15, 16 e 17 ilustram os valores de acurácia para todas as combinações de parâmetros definidos nestes experimentos. Analisando esses resultados, nota-se que o parâmetro $\tau \in \{1, 64, 256, 1024\}$ é o que mais influencia a diferença de acurácia nos resultados do oLDA. O parâmetro τ é inversamente proporcional ao fator de aprendizado ρ, ou seja, quanto maior o valor de τ menor será a influência dos novos documentos que chegam no fluxo. Um valor de τ alto pode ser vantajoso para retardar a convergência e evitar a propagação antecipada. Mas nos resultados obtidos na avaliação experimental, nota-se que $\tau = 1$ foi o suficiente para se obter melhores valores de acurácia.

O número de documentos no *mini-batch* influenciou os resultados, nota-se que quanto maior o número de documentos menor a acurácia. Isso ocorre devido a menor quantidade de atualizações globais realizadas. Por isso, os melhores resultados obtidos para todos os conjuntos de documentos foram para $b = 32$.

Já o parâmetro κ, que está relacionado com a velocidade com que o fator de aprendizado ρ decai em cada iteração, não influenciou significativamente no valor de acurácia.

4.4.2 Avaliação dos tópicos utilizando NPMI

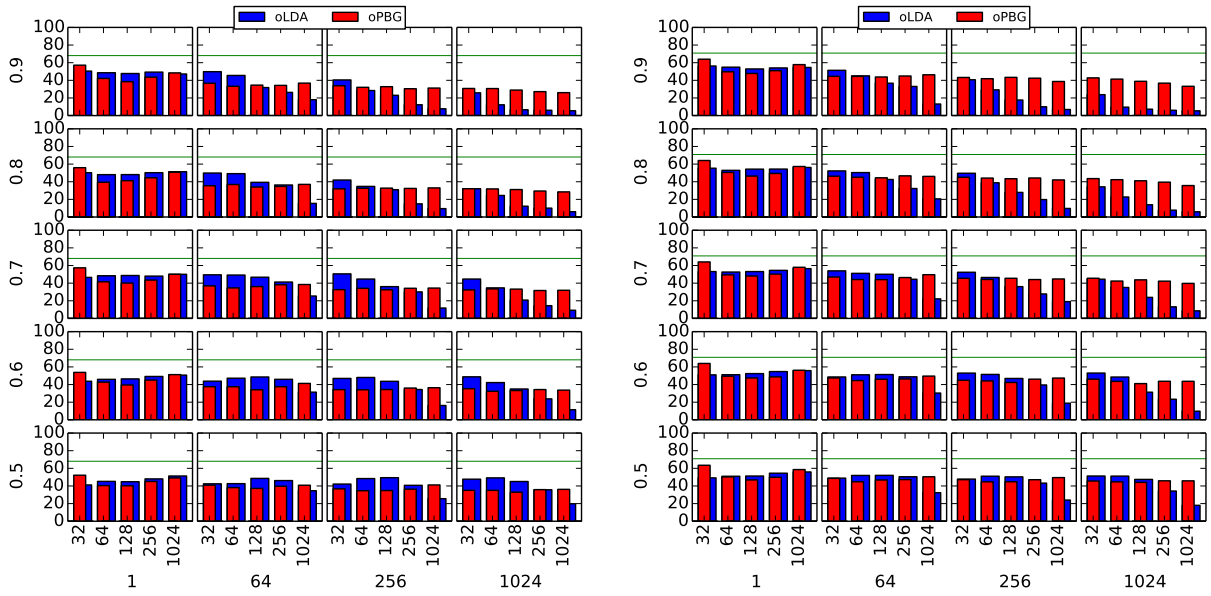
Para aplicar a medida NPMI é necessário um conjunto externo de documentos como referência. Neste trabalho foi utilizada a Wikipedia[2] em inglês. Devido ao custo de calcular a frequência dos pares de tópicos em todas as combinações de resultados obtidos, o cálculo do NPMI foi realizado apenas nos tópicos com melhores valores de acurácia obtidos na representatividade dos documentos. Os valores de NPMI estão descritos na Tabela 10.

Os melhores valores de acurácia foram obtidos para os parâmetros $\tau = 1$ para todos os conjuntos de dados, $b = 32$ para os conjuntos de dados *20ng* e *Dmoz-Business*, $b = 1024$ para os conjuntos de dados *classic4*, e $\kappa = 0.7$ para todos os conjuntos de dados.

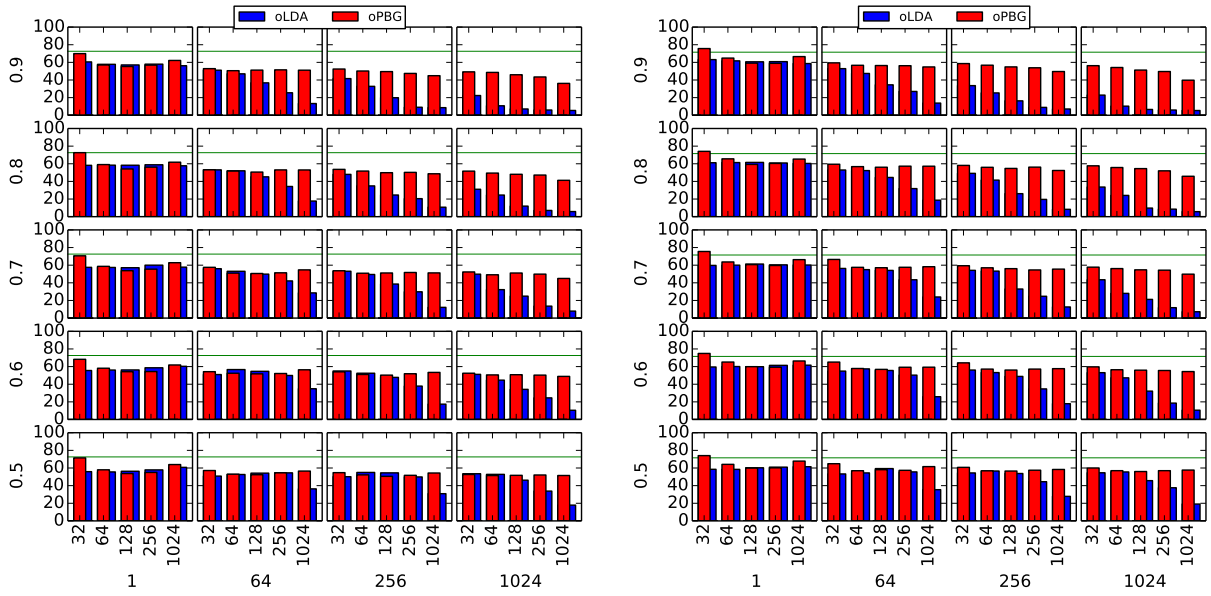
[2] Foram utilizados documentos da Wikipedia do ano de 2008. Esses documentos estão disponíveis livremente no site <https://dumps.wikimedia.org/>.

Figura 15 – Matriz de gráficos de acurácia comparando as versões online dos algoritmos PBG e LDA para o conjunto de dados *20ng*. Cada linha da matriz de gráficos corresponde a um valor de $\kappa \in \{0.5, 0.6, 0.7, 0.8, 0.9\}$, e cada coluna corresponde a um valor de $\tau_0 \in \{1, 64, 256, 1024\}$. Cada gráfico mostra a acurácia obtida considerando o número de documentos que chegaram no fluxo (eixo x) e o valor de acurácia obtidos pelos dois algoritmos (eixo y). A linha (na cor verde) cruzando horizontalmente cada gráfico corresponde a melhor acurácia obtida pelo LDA estático.

(a) $K = 50$ (b) $K = 100$



(c) $K = 150$ (d) $K = 200$

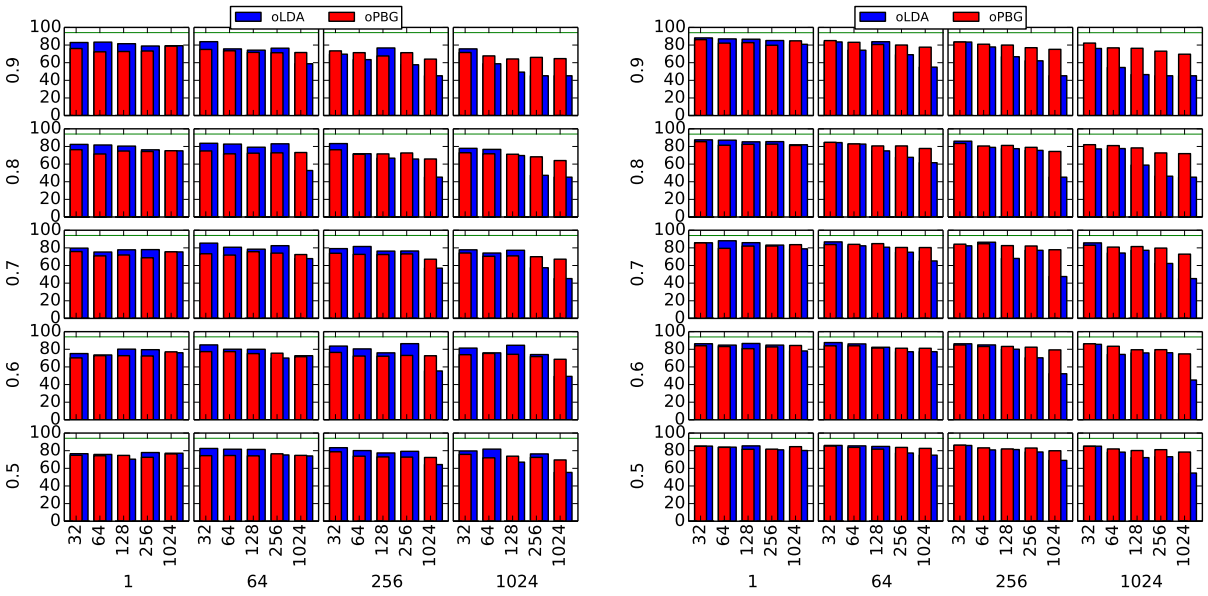


Fonte: Dados da pesquisa.

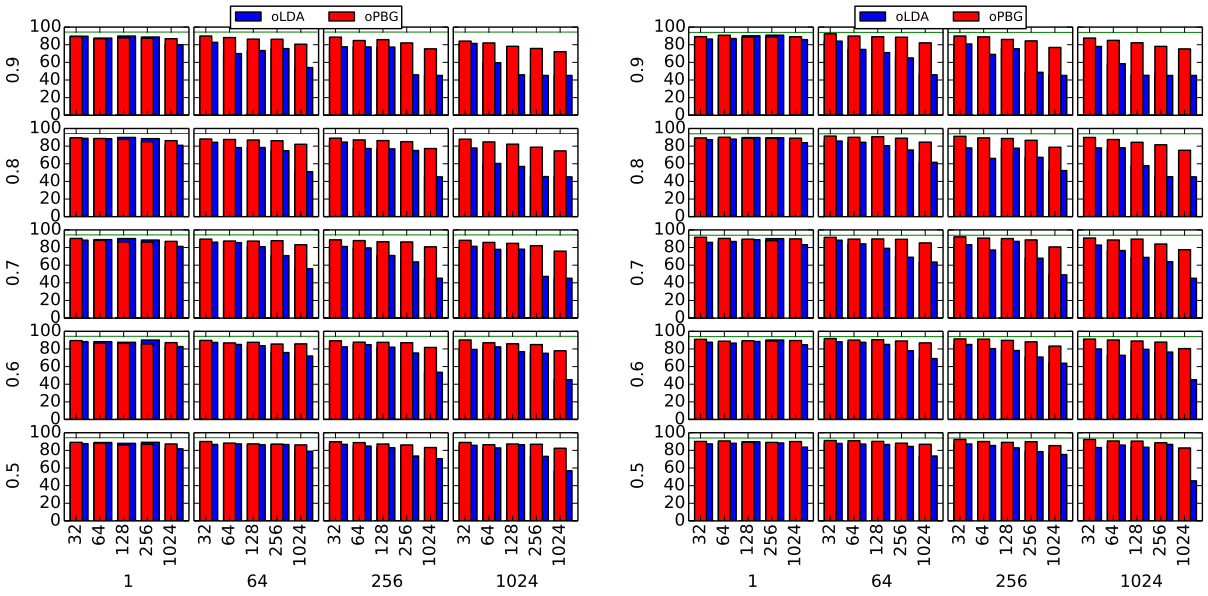
Os resultados do oPBG foram melhores nas bases *20ng* e *Dmoz-Business*, entretanto, foram piores na base *classic4*. Esses resultados podem indicar que o oPBG encontrou melhores tópicos do que o oLDA, porém, uma ressalva deve ser feita sobre esses resultados. Ao analisar o conjunto de palavras que formam os tópicos obtidos pelo oPBG, foram encontrados alguns

Figura 16 – Matriz de gráficos de acurácia comparando as versões online dos algoritmos PBG e LDA para o conjunto de dados *Dmoz-Business*. Cada linha da matriz de gráficos corresponde a um valor de $\kappa \in \{0.5, 0.6, 0.7, 0.8, 0.9\}$, e cada coluna corresponde a um valor de $\tau_0 \in \{1, 64, 256, 1024\}$. Cada gráfico mostra a acurácia obtida considerando o número de documentos que chegaram no fluxo (eixo x) e o valor de acurácia obtidos pelos dois algoritmos (eixo y). A linha (na cor verde) cruzando horizontalmente cada gráfico corresponde a melhor acurácia obtida pelo LDA estático.

(a) $K = 50$ (b) $K = 100$



(c) $K = 150$ (d) $K = 200$

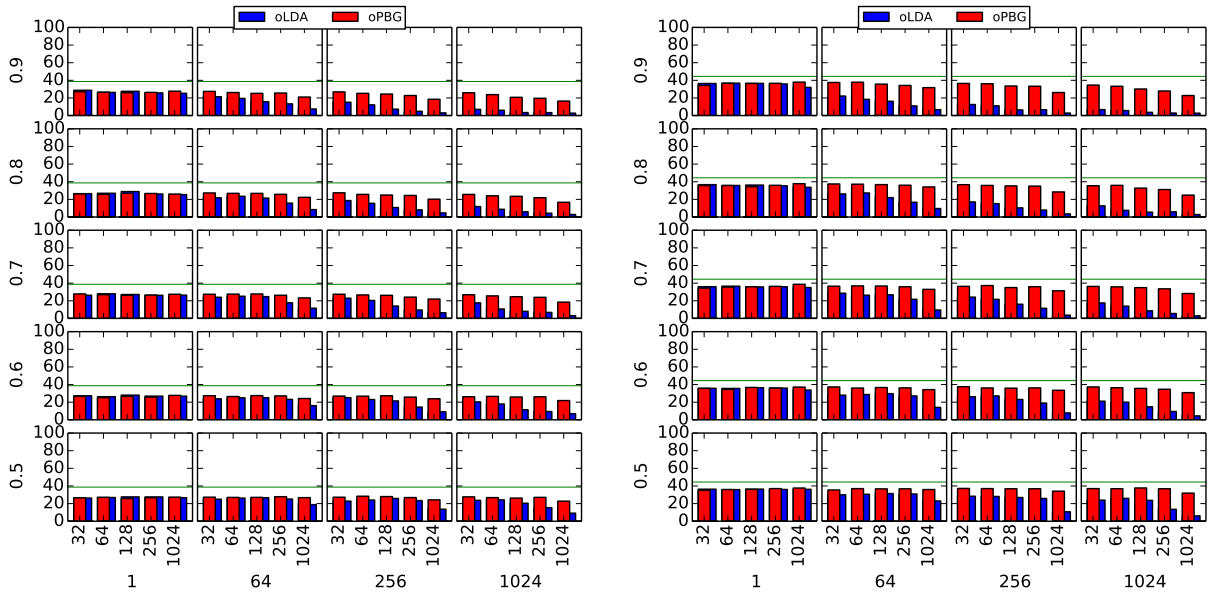


Fonte: Dados da pesquisa.

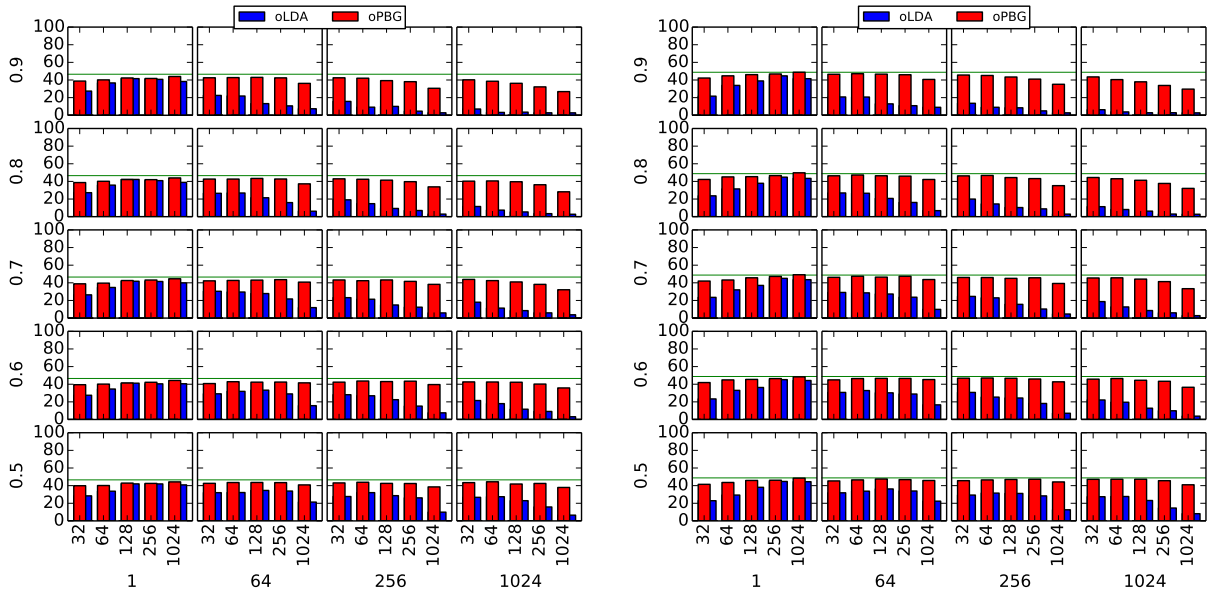
tópicos com palavras em comum. Um exemplo disso são os seguintes tópicos encontrados pelo oPBG na base *20ng*, ilustrados no Quadro 2. Note que os tópicos nas linhas 1, 2 e 3 do oPBG são semanticamente relacionados e possuem intersecções de palavras. Os valores de NPMI encontrados para esses dois tópicos são altos, pois as palavras estão semanticamente relacionadas

Figura 17 – Matriz de gráficos de acurácia comparando as versões online dos algoritmos PBG e LDA para o conjunto de dados *classic4*. Cada linha da matriz de gráficos corresponde a um valor de $\kappa \in \{0.5, 0.6, 0.7, 0.8, 0.9\}$, e cada coluna corresponde a um valor de $\tau_0 \in \{1, 64, 256, 1024\}$. Cada gráfico mostra a acurácia obtida considerando o número de documentos que chegaram no fluxo (eixo x) e o valor de acurácia obtidos pelos dois algoritmos (eixo y). A linha (na cor verde) cruzando horizontalmente cada gráfico corresponde a melhor acurácia obtida pelo LDA estático.

(a) $K = 50$ (b) $K = 100$



(c) $K = 150$ (d) $K = 200$



Fonte: Dados da pesquisa.

entre si. Consequentemente, a média geral do NPMI será alta. Porém, esse resultado não é bom quando se objetiva encontrar tópicos disjuntos, *i.e.* tópicos com palavras semanticamente relacionadas entre si e com pouca relação semântica entre os outros tópicos. No caso da coleção *classic4*, os tópicos que se sobrepuseram tinha baixo valor de NPMI, o que acarretou em um

Tabela 10 – Tabela com os valores de NPMI considerando os melhores valores de acurácia na representatividade dos documentos

conj. dados	algoritmo	$K = 50$	$K = 100$	$K = 150$	$K = 200$
20ng	oLDA	0.053	0.0436	0.0404	0.03895
	oPBG	**0.059**	**0.0533**	**0.0542**	**0.0578**
Dmoz-Business	oLDA	0.0532	0.0402	0.0352	0.0368
	oPBG	**0.0922**	**0.0943**	**0.0916**	**0.094**
classic4	oLDA	**0.121**	**0.098**	**0.0855**	**0.0804**
	oPBG	0.0774	0.0691	0.069	0.07235

Fonte: Dados da pesquisa.

Quadro 2 – Lista de tópicos para $K = 50$ obtido pelos algoritmos *online* oPBG e oLDA no conjunto de dados *20ng*.

	lista de tópicos do algoritmo oPBG ($K = 50$)	**lista de tópicos do algoritmo oLDA** ($K = 50$)
1	file entri program system output subject inform includ line write	time peopl thing point long side idea clear final don
2	file imag jpeg system program graphic ftp format data pub	don didn reason call dai doesn read care question true
3	jpeg imag file gif format color graphic version program bit	driver win love orbit understand andi cup interpret todai stephen
4	juda hang act matthew repli passag decenso contradict death fall	write gif practic view wasn produc definit rutger brown extrem
5	ripem kei public rsa pgp pem secur patent distribut standard	imag happen color object mode exist graphic write bitnet question
6	openwindow sun window file xview usr font program lib run	write articl apr subject don win gui window place rememb
7	graphic pub mail imag rai file send ftp object server	nasa gov paul space steve delet short fax easi sun
8	jehovah god elohim lord father christ mcconki mormon son jesu	end blue death red home natur jason shuttl subject frame
9	output file entri program onam printf eof line char stream	law state nation jesu weapon score arm christ subject radio
10	widget convert resourc applic visual set type file data default	good talk book discuss refer give subject articl purpos find
11	file bmp gmu dlss mml jame apr mx nmc mec	kei great left class comput line input kevin recommend cap
12	jesu matthew propheci psalm dai prophet peopl christian messiah david	softwar mac support subject version port window modem data packag
13	cesarean vaccin health malaria deliveri hiv rate infect diseas hospit	univers public scienc research center gener program depart comput mission
14	file gun firearm congress control state bill amend rkba unit	card scsi system price monitor id power appl comput subject
15	jehovah god elohim lord father christ mcconki mormon son jesu	model church write freenet cleveland articl subject visual greg toni
16	mac bit system ibm scsi file color program softwar disk	religion turkish faith islam jew armenian palestinian door bibl espn
17	widget includ version motif tar export server contrib ftp base	wrong type bit secur info basic return attempt mean code
18	planet earth spacecraft moon solar system orbit surfac atmospher sun	mail post group phone inform list subject interest includ internet
19	privaci pub eff internet comput anonym electron network email inform	file program ftp site directori pub defin anonym run includ
20	window motif inform manag offer program server list mail includ	dave sin white bob select drug wast iastat kent import
21	max bhj giz qax wwiz nrhj ghj biz nui bhjn	argument light tape medic moon berkelei adam spacecraft troubl vax
22	stephanopoulo presid don group tax discuss made packag press question	kill american player car live hit import deal sale todai
23	chz uww jpwu lhz fij rlk rmc ahf kjz rck	lost buffalo dec subject ball alaska gopher space insid nsmca
24	mac bit ibm scsi color program hardwar built cpu standard	feel mike washington school print subject printer font satellit student
25	mac bit ibm scsi color built program hardwar standard cpu	plai format fbi involv project control back boston byte instal
26	uww chz rlk rck rmc scx lhz uwt hzv rchz	problem work bad close friend subject run fine letter open
27	file gun firearm congress control state bill amend rkba unit	free david right polit peopl amend compound unit nazi church
28	client resourc server xterm program window file font suggest faq	peopl govern don bui part matter speak guess monei reason
29	jpeg imag file gif color format version program bit qualiti	ibm graphic robert includ sgi subject top std jai coach
30	output file entri program onam printf eof line char stream	rule algorithm exist theori patient atheism judg michael entri decis
31	turkish jew turkei nazi book jewish war ottoman eastern german	god christian fire bill muslim jew atheist servic wai father
32	hockei team leagu game nhl season mail list graphic draft	life compani jim encrypt technolog access system clipper sport digex
33	armenian peopl didn don karina apart call start azerbaijani hand	netcom size bhj giz valu chz prefer brain bxn rlk
34	armenian azerbaijani start didn neighbor peopl shout apart diana children	file pit run error joe averag cub rob seri app
35	hockei team leagu nhl game season draft player list divis	gun arab firearm batf koresh thoma murder smith atf ron
36	entri file program rule build info sourc remark obfusc compil	game team person board hockei fan contact land belief polici
37	bo nyr buf mon har van edm cal ott det	institut season nhl park prism target extern nick flyer randi
38	scx gcx chz rlk syx rck mcx ec rmc sq	carri unit attack roger armi upgrad offer wide night west
39	widget convert resourc applic visual set type file arg data	mark jpeg uiuc bu rai observ frank servic die virginia
40	hockei team leagu game nhl season draft divis player list	toronto stanford fall dan pitch sci villag pitcher dept reader
41	appear art wolverin ghost rider hobgoblin man punish annual sabretooth	isra convert keith titl udel clark bmp steven robi lebanes
42	remot file pwd end echo shell displai command argv shar	drive subject engin back write bike david tom dod lab
43	anonym internet system file privaci user inform address email mail	max children war pass video crime rememb physic road flame
44	wire ground circuit outlet connect gfci neutral subject electr cabl	argic serdar montreal azerbaijan azerbaijani azeri istanbul prior tartar karabakh
45	do window microsoft tcp mbyte support mous amiga graphic softwar	do window chip space disk relat quot bit direct half
46	de cipher kei bit block encrypt product data standard oper	israel moral homosexu leagu san men jewish head live longer
47	drive disk control system hard bio rom support featur card	year andrew place cmu logic total come handl clinton pittsburgh
48	jpeg imag file gif color format version program bit qualiti	point human armenian claim word evid hand world studi build
49	wire ground circuit outlet connect gfci neutral subject cabl electr	control john effect write back protect ago present report job
50	kinsei sex sexual boi orgasm year children homosexu male book	period problem sound richard write subject fix bug think articl

Fonte: Dados da pesquisa.

baixo valor geral. As listas de tópicos para as coleções *classic4* e *Dmoz-Business* estão descritas no Apêndice A.

No caso dos tópicos encontrados no oLDA, poucas sobreposição de palavras são encontrados, logo tem-se uma melhor distinção entre os tópicos.

4.4.3 Discussão dos resultados

Apesar do algoritmo oPBG encontrar valores de NPMI maiores para alguns conjunto de documentos em comparação com o oLDA, os tópicos encontrados pelo oPBG apresentaram muita sobreposição. Em contraste, no oLDA não ocorre esse problema. Nesta seção é discutida

essa diferença e, tendo como base a análise dos dados obtidos pela execução dos dois algoritmos, são dados indícios que possam explicar a sobreposição dos tópicos.

É possível fazer uma relação entre o oPBG e o oLDA. Enquanto o algoritmo de inferência variacional estocástico do oLDA baseia-se na otimização da divergência entre a distribuição real e a distribuição variacional (Veja a Seção 2.1.2), o oPBG otimiza a divergência entre os vetores associados aos vértices do grafo. Um dos fatores que existe de comuns nesses dois métodos é a distinção entre operações locais e globais. As operações locais são realizadas em cada documento, e são responsáveis pela descoberta da relação documentos-tópicos. Já as operações globais são realizadas em toda a coleção, e são responsáveis pela descoberta da relação tópicos-palavras. Nas versões *online* dos dois algoritmos as operações locais não são alteradas. Já para as operações globais, é necessário realizar a atualização das estatísticas incrementalmente. Isso é fundamentado com base no processo de otimização estocástica.

Na operação global do oPBG, as estatísticas globais são normalizadas de forma que $\sum_{i=1}^{n} B_{i,k} = 1$, para todo tópico k ($0 \leq k \leq K$) (Veja a Linha 16 do Algoritmo 13). Esse tipo de normalização caracteriza uma distribuição multinomial nos dados. Já na implementação do oLDA, para se obter a distribuição de tópicos por palavras ϕ, calcula-se a esperança do logaritmo de ϕ dada a distribuição variacional λ. Esse cálculo é feito da seguinte forma:

$$E[\log \phi_k] = \exp\left(\Psi\left(\lambda_{k,i}\right) - \Psi(\sum_{l=n}^{K} \lambda_{k,l})\right) \tag{4.16}$$

Isso é basicamente normalizar os dados de forma que eles obedeçam a distribuição de Dirichlet.

Para perceber a diferença na distribuição dos dados contido nos vetores com as estatísticas globais, tem-se o seguinte exemplo. Suponha o vetor com as estatísticas globais $X = [0.3, 0.2, 0.15, 0.01]$, fazendo as normalizações segundo as distribuições multinomial e de Dirichlet (Equação 4.16), obtêm-se as distribuições do vetor X ilustradas na Figura 18.

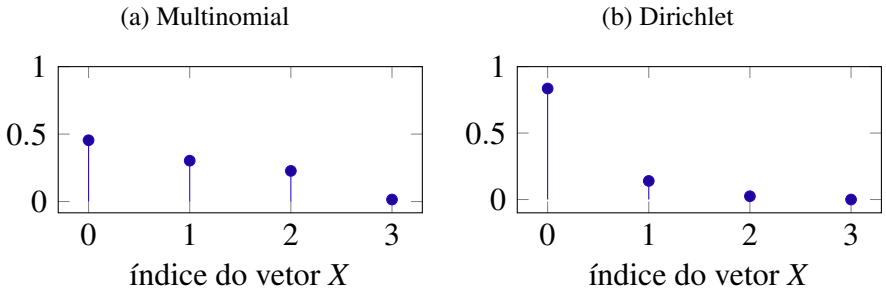


Figura 18 – Normalizações do vetor $X = [0.3, 0.2, 0.15, 0.01]$

Note que após a normalização segundo a distribuição de Dirichlet, o maior valor do vetor X aumenta exponencialmente. Suponha que cada posição i de X (índice do vetor X) é a correlação entre um tópico k com uma determinada palavra w_i. Após a normalização de Dirichlet, a relação de correlação de uma palavra para o tópico com alta probabilidade cresce

exponencialmente. Isso faz com que dificilmente mesmas palavras pertençam a mais de um tópico com alta probabilidade.

Por outro lado, a normalização do oPBG não evidencia a correlação de maior valor dos tópicos com uma palavra específica. Além disso, no caso *online*, onde as estatísticas globais são atualizadas incrementalmente, uma mesma palavra pode acumular valores com alta correlação para diferentes tópicos. Consequentemente, uma mesma palavra pode ter alta correlação para diversos tópicos, o que causa o efeito de vários tópicos relacionados.

Apesar desse efeito prejudicar a qualidade dos tópicos, não influenciou na qualidade dos resultados obtidos pela tarefa de redução de dimensionalidade. Isso ocorreu porque as dimensões latentes ainda conseguem distinguir os grupos de documentos. No caso do conjunto de dados $20ng$, no mínimo 20 dimensões latentes seriam necessárias para distinguir todos as classes. Nos experimentos foram definidas 50, 100, 150 e 200 dimensões latentes, e mesmo várias se agrupando, ainda foram suficientes para distinguir as classes dos documentos.

4.4.4 Complexidade do oPBG

O algoritmo oPBG (Algoritmo 13) processa um documento d_j (ou um subconjunto de documentos) que chega no fluxo para obter o vetor local A_j. Essa operação tem complexidade $O(Tn_{d_j}K)$, onde T é o número máximo de iterações para a convergência no processo de propagação local, n_{d_j} é o número de termos no documento d_j e K é o número de tópicos. Depois, são calculados os vetores $\hat{B}_i$ para cada palavra $w_i \in \mathcal{W}_{d_j}$. Em seguida, são estimados os vetores B_i associado a todas as palavras do vocabulário. E por fim, esses vetores são normalizados para todos os tópicos. Essas operações têm complexidade na ordem do número de palavras do vocabulário vezes o número de tópicos, $O(nK)$. Assim, a complexidade total do oPBG para as propagações de um documento d_j que chega no fluxo é $O(Tn_{d_j}K + nK)$.

O oLDA, como foi inicialmente descrito no trabalho de Dabid Blei (BLEI; NG; JORDAN, 2003), tem complexidade $O(Tn_{d_j}K + n_{d_j}K)$. Logo o oLDA é mais rápido do que o oPBG. Essa diferença está na fase de propagação global realizado pelo oPBG, que normaliza cada tópico em relação a todo o vocabulário de palavras. Entretanto, nas implementações do LDA esse processo de normalização também é realizado. A diferença da descrição do LDA e das implementações encontradas do algoritmo de inferência variacional está nas simplificações matemáticas utilizada na descrição. Logo o oPBG e o oLDA tem a mesma complexidade de tempo.

4.5 Considerações finais

Neste capítulo foi apresentado o oPBG, um algoritmo não supervisionado baseado na propagação de rótulos e que utiliza a representação em grafos bipartidos para a extração de tópicos em documentos em fluxo. A formulação do algoritmo é baseada na otimização da divergência entre vetores associados a vértices vizinhos. Para ajustar-se ao contexto dinâmico,

onde pequenos conjuntos de documentos chegam no fluxo, é utilizado o método de otimização estocástica para estimar a direção do gradiente que otimiza a função de divergência.

Os resultados obtidos indicam que o algoritmo oPBG encontra resultados melhores do que o oLDA na tarefa de redução de dimensionalidade. Já na tarefa de extração de tópicos, os valores de NPMI foram melhores do que o LDA em algumas bases. Porém, os tópicos encontrados apresentaram sobreposição, o que não pode ser desejável em algumas aplicações. Pode ser interessante incorporar no algoritmo oPBG um procedimento que evite os tópicos sobrepostos, uma indício de como isso pode ser realizado é simular o efeito da Dirichlet nos vetores associados aos termos, como discutido na Seção 4.4.3.

CAPÍTULO

5

CONCLUSÃO

Neste capítulo são apresentadas as conclusões decorrentes desta investigação de doutorado. São também comentados os objetivos alcançados durante o trabalho, bem como suas contribuições, limitações e trabalhos futuros.

O objetivo geral do trabalho consistiu no projeto e implementação de algoritmos de aprendizado de máquinas em documentos textuais representados na forma de grafos bipartidos. Os algoritmos propostos basearam-se na simples e intuitiva ideia de propagação de rótulos, o que contribuiu com a verificação da hipótese de que a representação via grafos permite a geração de modelos de extração de tópicos eficazes, eficientes e fácil de se adaptarem para trabalhar em outros domínios, como no caso de dados em fluxo.

As seguintes contribuições estão diretamente relacionadas ao objetivo e foram apresentadas em capítulos específicos desta tese:

- uma descrição detalhada dos modelos probabilísticos de tópicos, apresentada no **Capítulo 2**;
- um arcabouço para aprendizado baseado na propagação em grafos bipartidos, apresentado no **Capítulo 3**;
- um algoritmo para extração de tópicos *online*, apresentado no **Capítulo 4**.

5.1 Modelos Probabilísticos de Tópicos

A forma como pesquisadores de aprendizado de máquinas modelam textos e outros objetos mudaram após o surgimento dos modelos probabilísticos de tópicos. O arcabouço fornecido pelo LDA serviu como ferramenta para o desenvolvimento de vários outros modelos. Assim, o modelo base LDA foi investigado e detalhado a fim de entender o funcionamento prático dos algoritmos de inferência. O modelo foi descrito, e principalmente, as derivações

foram feitas durante os estudos sobre os algoritmos de inferência. Esses estudos servirão como referência para estudos futuros, principalmente para novos alunos que por ventura queiram explorar modelos probabilísticos de tópicos.

Também foi possível fundamentar o arcabouço baseado em grafos bipartidos utilizando os procedimentos de inferência aplicado no modelo LDA, em especial, o método de inferência variacional. Além disso, foi realizado um estudo comparativo do método de inferência variacional com o método de fatoração de matrizes não negativas (FALEIROS; LOPES, 2016).

Uma limitação foi na forma com que a descrição do modelo LDA foi apresentada, na qual se levou pouco em consideração a interpretação Bayesiana do modelo. Em uma descrição que enfatizam o LDA como um modelo Bayesiano completo, as distribuições *priori* deveriam ser melhores descritas. A importância da *priori* no modelo LDA é bem discutida no trabalho (WALLACH; MIMNO; MCCALLUM, 2009). Também foi pouco explorado o LDA como uma Rede Bayesiana, e como estender esse modelo a fim de se obter outros modelos.

5.2 Propagação em grafos bipartidos

Nesta tese foi apresentado um método, chamado PBG, que explora a utilização de grafos bipartidos para a representação de documentos textuais. O algoritmo PBG é um método simples, e o procedimento realizado por esse algoritmo pode ser facilmente descrito como um processo de propagação de rótulos na estrutura do grafo bipartido. Considerando esse algoritmo como um arcabouço, foi possível incluir heurísticas para melhorar os resultados.

No caso não supervisionado, considerou-se os rótulos gerados aleatoriamente e, após várias propagações, encontram-se as K dimensões latentes, que neste arcabouço são vetores K-dimensionais atribuídos a cada vértice do grafo bipartido. Algumas simples modificações foram feitas no PBG não supervisionado a fim de se obter melhores resultados, em destaque está o método de iniciação baseado em agrupamento, que é basicamente um pré-processamento para se obter valores iniciais melhores do que aqueles gerados aleatoriamente. Outra modificação tira vantagens da estrutura do grafo bipartido, que é a facilidade em dividi-lo em subgrafos. Com isso foi possível elaborar uma versão paralelizada do PBG. Os experimentos realizados foram nas tarefas de extração de tópicos e redução de dimensionalidade. Os resultados obtidos pelo PBG foram comparados com os resultados obtidos pelo NMF e LDA. Os experimentos indicaram que o algoritmo PBG possui resultados semelhantes, e em algumas bases foi capaz de ultrapassar o LDA no processo de extração de tópicos e redução de dimensionalidade, tendo resultados melhores quando se utilizou heurística de agrupamento para iniciar os valores dos vetores associados aos vértices.

No caso semissupervisionado, o algoritmo PBG foi modificado para considerar documentos com informações de classes. Essa alteração foi direta, uma vez que basta considerar os rótulos reais dos documentos rotulados e os rótulos gerado aleatoriamente para os documentos

não rotulados. Os experimentos foram específicos para a tarefa de classificação transdutiva. Os resultados indicaram que o PBG é competitivo, em comparação aos algoritmos estado da arte em classificação transdutiva, e, quando poucos documentos rotulados são considerados, ele é capaz de encontrar melhores resultados (FALEIROS; ROSSI; LOPES, 2016).

Quanto as limitações do PBG, a principal é a de que não foi demonstrada a correta convergência do algoritmo – apesar dos experimentos indicarem que o algoritmo converge. Outra limitação está na metodologia de avaliação dos tópicos no caso não supervisionado. Apesar da métrica NPMI ser usualmente aplicada para averiguar a qualidade dos tópicos, ela depende da base externa, que, no caso, foram os documentos da Wikipedia. Algumas bases podem conter vocabulários bem específicos, o que ocasionam em baixo valor de NPMI pois os pares de palavras não serão frequentes na Wikipedia. Outro problema encontrado na medida NPMI é sua incapacidade de medir a distinção entre os tópicos. Com isso, pode-se encontrar alto valor de NPMI para tópicos com palavras iguais ou redundantes (que cobrem um mesmo tema).

Como trabalhos futuros, um dos grandes desejos é incluir mais relações no grafo. Nesse caso, deseja-se não se restringir a apenas um grafo bipartido representando dois tipos de objetos e suas relações, mas sim um grafo heterogêneo com mais de dois tipos de objetos. Para isso, seria necessário alterar as operações de propagação para considerar os vetores associados aos diversos tipos de objetos. No contexto de mineração de texto, um grafo heterogêneo poderia considerar não só vértices do tipo documento e termos, mas também, por exemplo, objetos representando autores ou informações de tempo. Portanto, seria possível encontrar as dimensões latentes para as várias relações entre diferentes objetos representados no grafo heterogêneo em um procedimento que consideraria todas as relações entre esses objetos.

Outro trabalho futuro, e que tiraria vantagem da modelagem em grafos utilizada pelo algoritmo PBG, é considerar relações documentos-documentos ou termos-termos. É fácil modelar essas relações no grafo. Para criar um algoritmo que tira vantagens dessas relações, o PBG poderia ser utilizado como base para esse novo algoritmo. As operações de propagação poderiam ser adaptadas de forma a considerar os vetores de pesos associados aos vértices vizinhos e que, considerando essas novas relações, são do mesmo tipo. Isso é considerado uma forma de "enriquecimento" do método, e que, baseando na intuição do processo de propagação, poderia ser facilmente incorporando no algoritmo PBG.

5.3 Extração de tópicos *online* utilizando redes bipartidas

O arcabouço desenvolvido nesta tese, o algoritmo PBG, foi estendido para considerar o problema de extração de tópicos em fluxo de documentos. A fundamentação da versão *online* do PBG, o algoritmo oPBG, baseia-se na otimização, via método de otimização estocástica, da divergência entre vetores associados aos vértices do grafo bipartido. Na versão estática do PBG são necessárias várias iterações sobre todo o conjunto de documentos, já na versão *online*, o

documento é percorrido apenas uma vez, e as estatísticas associadas aos vetores de palavras são atualizadas de forma incremental. Com isso, tem-se um método eficiente que se adapta ao contexto dinâmico ou quando existem grandes quantidades de documentos.

Os resultados experimentais indicaram que o algoritmo oPBG encontra resultados satisfatórios, principalmente para a representabilidade dos documentos em uma tarefa de redução de dimensionalidade. Já os tópicos encontrados pelo oPBG obtiveram bons valores de NPMI em comparação com os tópicos obtidos pela versão *online* do LDA. Porém, o oLDA, em comparação com o oPBG, manteve maior variabilidade dos temas entre os diversos tópicos extraídos de uma mesma base.

Quanto aos trabalhos futuros, eles estão intimamente ligados aos grandes desafios da área de mineração em fluxo de dados. Entre esses desafios, destaca-se a capacidade de adaptar ao surgimento de novos conceitos. O ideal para o contexto dinâmico (em fluxo de dados) seria um modelo em tempo real capaz de incorporar novas informações a medida que elas chegam no fluxo, detectar alterações da distribuição dos dados para adaptar o modelo e "esquecer" os dados já processados para economizar memória. Isso está fortemente relacionado a incorporação de técnicas capazes de detectar alterações de conceitos (GAMA *et al.*, 2014; GAMA *et al.*, 2004). Na aplicação em extração de tópicos, por exemplo, um assunto (ou tópico) pode surgir ou desaparecer abruptamente, periodicamente ou mesmo, surgir gradualmente. Portanto, tem-se como trabalho futuro incorporar heurísticas de detecção de alterações de conceitos no procedimento de propagação do oPBG.

Outro trabalho futuro é realizar experimentos explorando a adição de novas palavras que chegam no fluxo e comparar os resultados com a versão do LDA com vocabulário infinito proposto por Zhai e Boyd-graber (2013). Para isso, se tiraria vantagem da formulação em grafos bipartidos que, de forma simples, permite a adição de novos vértices. Utilizando o oPBG, é fácil incluir uma heurística para criar o vetor associado a nova palavra que chega no fluxo. Essa heurística poderia utilizar vetores de palavras já conhecidas e próximas dessa nova palavra, ou iniciar o vetor com valores aleatórios, ou mesmo explorar heurísticas já aplicadas em fluxo de dados para contagem de itens, como *Count-Min Sketch* (CORMODE; HADJIELEFTHERIOU, 2008) e *Space Saving Algorithm* (METWALLY; AGRAWAL; ABBADI, 2006). Esses métodos poderiam ser utilizados junto ao PBG para estimar o número de palavras que chegaram no fluxo, e manter apenas as mais frequentes.

Por fim, o arcabouço fornecido pelo PBG é rico em possibilidades de extensões. Isso se deve a simples abordagem descritiva fornecida pela modelagem em grafo, além de endereçar problemas básicos nos contextos semissupervisionado, não supervisionado estático e não supervisionado dinâmico com fluxo de dados.

REFERÊNCIAS

AGGARWAL, C. C.; ZHAI, C. (Ed.). **Mining Text Data**. [S.l.]: Springer, 2012. ISBN 978-1-4419-8462-3. Citado 2 vezes nas páginas 31 e 33.

AGGARWAL, C. C.; ZHAO, P. Towards graphical models for text processing. **Knowl. Inf. Syst.**, v. 36, n. 1, p. 1–21, 2013. Disponível em: <http://dx.doi.org/10.1007/s10115-012-0552-3>. Citado na página 100.

AHMED, A.; XING, E. P. Timeline: A dynamic hierarchical dirichlet process model for recovering birth/death and evolution of topics in text stream. **CoRR**, abs/1203.3463, 2012. Citado na página 119.

ALSUMAIT, L.; BARBARá, D.; DOMENICONI, C. On-line lda: Adaptive topic models for mining text streams with applications to topic detection and tracking. In: **Proceedings of the 2008 Eighth IEEE International Conference on Data Mining**. Washington, DC, USA: IEEE Computer Society, 2008. (ICDM '08), p. 3–12. ISBN 978-0-7695-3502-9. Disponível em: <http://dx.doi.org/10.1109/ICDM.2008.140>. Citado na página 117.

ANGELOVA, R.; WEIKUM, G. Graph-based text classification: learn from your neighbors. In: **SIGIR '06: Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval**. New York, NY, USA: ACM, 2006. p. 485–492. ISBN 1-59593-369-7. Disponível em: <http://dx.doi.org/10.1145/1148170.1148254>. Citado 2 vezes nas páginas 100 e 101.

ARORA, S.; GE, R.; MOITRA, A. Learning topic models – going beyond svd. In: **Proceedings of the 2012 IEEE 53rd Annual Symposium on Foundations of Computer Science**. Washington, DC, USA: IEEE Computer Society, 2012. (FOCS '12), p. 1–10. ISBN 978-0-7695-4874-6. Disponível em: <http://dx.doi.org/10.1109/FOCS.2012.49>. Citado na página 32.

BANERJEE, A.; BASU, S. Topic models over text streams: A study of batch and online unsupervised learning. In: **SDM**. SIAM, 2007. ISBN 978-0-89871-630-6. Disponível em: <http://dblp.uni-trier.de/db/conf/sdm/sdm2007.html#BanerjeeB07>. Citado na página 116.

BERRY, M. W.; DUMAIS, S. T.; O'BRIEN, G. W. Using linear algebra for intelligent information retrieval. **SIAM Rev.**, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, v. 37, n. 4, p. 573–595, dez. 1995. ISSN 0036-1445. Disponível em: <http://dx.doi.org/10.1137/1037127>. Citado na página 40.

BERTINI, J. R.; LOPES, A. A.; ZHAO, L. Partially labeled data stream classification with the semi-supervised k-associated graph. **Journal of the Brazilian Computer Society**, Springer London, p. 1–12, 2012. ISSN 0104-6500. 10.1007/s13173-012-0072-8. Disponível em: <http://dx.doi.org/10.1007/s13173-012-0072-8>. Citado na página 116.

BERTINI, J. R.; ZHAO, L.; MOTTA, R.; LOPES, A. A. A nonparametric classification method based on k-associated graphs. **Information Sciences**, v. 181, n. 24, p. 5435 – 5456, 2011. Disponível em: <http://dx.doi.org/10.1016/j.ins.2011.07.043>. Citado na página 72.

BERTON, L.; LOPES, A. A. Graph construction based on labeled instances for semi-supervised learning. In: . Stockholm, Sweden: [s.n.], 2014. p. 2477–2482. Disponível em: <http://www.academia.edu/8283465/Graph_construction_based_on_labeled_instances_for_semi-supervised_learning>. Citado na página 72.

BISHOP, C. M. **Pattern Recognition and Machine Learning (Information Science and Statistics)**. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2006. ISBN 0387310738. Citado 2 vezes nas páginas 41 e 52.

BLEI, D. M. Introduction to probabilistic topic models. **Communications of the ACM**, 2011. Disponível em: <http://www.cs.princeton.edu/~blei/papers/Blei2011.pdf>. Citado 4 vezes nas páginas 31, 32, 39 e 45.

BLEI, D. M.; LAFFERTY, J. D. Dynamic topic models. In: **Proceedings of the 23rd International Conference on Machine Learning**. New York, NY, USA: ACM, 2006. (ICML '06), p. 113–120. ISBN 1-59593-383-2. Disponível em: <http://doi.acm.org/10.1145/1143844.1143859>. Citado na página 117.

BLEI, D. M.; NG, A. Y.; JORDAN, M. I. Latent dirichlet allocation. **J. Mach. Learn. Res.**, JMLR.org, v. 3, p. 993–1022, mar. 2003. ISSN 1532-4435. Disponível em: <http://dl.acm.org/citation.cfm?id=944919.944937>. Citado 8 vezes nas páginas 32, 39, 41, 43, 45, 54, 55 e 135.

BLUM, A.; MITCHELL, T. Combining labeled and unlabeled data with co-training. In: **Proceedings of the Eleventh Annual Conference on Computational Learning Theory**. New York, NY, USA: ACM, 1998. (COLT' 98), p. 92–100. ISBN 1-58113-057-0. Disponível em: <http://doi.acm.org/10.1145/279943.279962>. Citado 2 vezes nas páginas 72 e 100.

BODLAJ, J.; BATAGELJ, V. Hierarchical link clustering algorithm in networks. **Phys. Rev. E**, American Physical Society, v. 91, p. 062814, Jun 2015. Disponível em: <http://link.aps.org/doi/10.1103/PhysRevE.91.062814>. Citado na página 87.

BOTTOU, L. On-line learning in neural networks. In: SAAD, D. (Ed.). New York, NY, USA: Cambridge University Press, 1998. cap. On-line Learning and Stochastic Approximations, p. 9–42. ISBN 0-521-65263-4. Disponível em: <http://dl.acm.org/citation.cfm?id=304710.304720>. Citado 2 vezes nas páginas 121 e 126.

____. Advanced lectures on machine learning: Ml summer schools 2003, canberra, australia, february 2 - 14, 2003, tübingen, germany, august 4 - 16, 2003, revised lectures. In: ____. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004. cap. Stochastic Learning, p. 146–168. ISBN 978-3-540-28650-9. Disponível em: <http://dx.doi.org/10.1007/978-3-540-28650-9_7>. Citado 3 vezes nas páginas 121, 122 e 126.

____. Large-Scale Machine Learning with Stochastic Gradient Descent. In: LECHEVALLIER, Y.; SAPORTA, G. (Ed.). **Proceedings of COMPSTAT'2010**. Physica-Verlag HD, 2010. p. 177–186. Disponível em: <http://dx.doi.org/10.1007/978-3-7908-2604-3_16>. Citado na página 121.

BOUSQUET, O.; BOTTOU, L. The tradeoffs of large scale learning. In: PLATT, J. C.; KOLLER, D.; SINGER, Y.; ROWEIS, S. T. (Ed.). **Advances in Neural Information Processing Systems 20**. Curran Associates, Inc., 2008. p. 161–168. Disponível em: <http://papers.nips.cc/paper/3323-the-tradeoffs-of-large-scale-learning.pdf>. Citado na página 126.

BRONIATOWSKI, D. A.; MAGEE, C. L. Analysis of social dynamics on fda panels using social networks extracted from meeting transcripts. In: **SocCom**. [s.n.], 2010. Disponível em: <http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5591237&tag=1>. Citado na página 39.

BUN, K. K.; ISHIZUKA, M. Topic extraction from news archive using tf*pdf algorithm. In: **Web Information Systems Engineering, 2002. WISE 2002. Proceedings of the Third International Conference on**. [S.l.: s.n.], 2002. p. 73 – 82. Citado na página 120.

BUNTINE, W. Variational extensions to em and multinomial pca. In: **In ECML 2002**. [S.l.]: Springer-Verlag, 2002. p. 23–34. Citado na página 64.

CANINI, K. R.; SHI, L.; GRITHS, T. L. Online inference of topics with latent dirichlet allocation. In: **Proceedings of AI Stats**. [S.l.: s.n.], 2009. Citado na página 118.

CAO, L.; LI, F.-F. Spatially coherent latent topic model for concurrent segmentation and classification of objects and scenes. In: **ICCV**. IEEE, 2007. p. 1–8. Disponível em: <http://dblp.uni-trier.de/db/conf/iccv/iccv2007.html#CaoF07>. Citado na página 39.

CHANG, J.; BLEI, D. Relational topic models for document networks. In: **AIStats**. [S.l.: s.n.], 2009. Citado na página 39.

CHANG, J.; BOYD-GRABER, J.; WANG, C.; GERRISH, S.; BLEI, D. M. Reading tea leaves: How humans interpret topic models. In: **Neural Information Processing Systems**. [S.l.: s.n.], 2009. Citado 2 vezes nas páginas 62 e 95.

CHAPELLE, O.; SCHLKOPF, B.; ZIEN, A. **Semi-Supervised Learning**. 1st. ed. [S.l.]: The MIT Press, 2010. ISBN 0262514125, 9780262514125. Citado 2 vezes nas páginas 72 e 98.

CHARIKAR, M.; CHEKURI, C.; FEDER, T.; MOTWANI, R. Incremental clustering and dynamic information retrieval. In: **Proceedings of the Twenty-Ninth Annual ACM Symposium on Theory of Computing**. [S.l.: s.n.], 1997. p. 626–635. Citado na página 115.

CORMODE, G.; HADJIELEFTHERIOU, M. Finding frequent items in data streams. **Proc. VLDB Endow.**, VLDB Endowment, v. 1, n. 2, p. 1530–1541, ago. 2008. ISSN 2150-8097. Disponível em: <http://dx.doi.org/10.14778/1454159.1454225>. Citado na página 140.

DAUD, A. Using time topic modeling for semantics-based dynamic research interest finding. **Knowledge-Based Systems**, v. 26, p. 154–163, 2012. Cited By (since 1996) 0. Disponível em: <http://www.scopus.com/inward/record.url?eid=2-s2.0-84155189116&partnerID=40&md5=84ba11fec91dc4d3a43ae5ac9eeda5e1>. Citado na página 117.

DEERWESTER, S. C.; DUMAIS, S. T.; LANDAUER, T. K.; FURNAS, G. W.; HARSHMAN, R. A. Indexing by latent semantic analysis. **JASIS**, v. 41, n. 6, p. 391–407, 1990. Disponível em: <http://dx.doi.org/10.1002/(SICI)1097-4571(199009)41:6<391::AID-ASI1>3.0.CO;2-9>. Citado na página 40.

DHILLON, I. S. Co-clustering documents and words using bipartite spectral graph partitioning. In: **Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge Discovery and Data Mining**. New York, NY, USA: ACM, 2001. (KDD '01), p. 269–274. ISBN 1-58113-391-X. Disponível em: <http://doi.acm.org/10.1145/502512.502550>. Citado na página 101.

DING, C.; LI, T.; PENG, W. On the equivalence between non-negative matrix factorization and probabilistic latent semantic indexing. **Comput. Stat. Data Anal.**, Elsevier Science Publishers B. V., Amsterdam, The Netherlands, The Netherlands, v. 52, n. 8, p. 3913–3927, abr. 2008. ISSN 0167-9473. Citado na página 64.

DOUCET, A.; FREITAS, N. D.; GORDON, N. (Ed.). **Sequential Monte Carlo methods in practice**. [s.n.], 2001. Disponível em: <http://www.worldcatlibraries.org/wcpa/top3mset/839aaf32b6957a10a19afeb4da09e526.html>. Citado na página 118.

FALEIROS, T. de P.; BERTON, L.; LOPES, A. de A. Exploring data classification with k-associated network. In: **IV International Workshop on Web and Text Intelligence (WTI-2012)**. [S.l.: s.n.], 2012. Citado na página 37.

FALEIROS, T. de P.; LOPES, A. de A. On the equivalence between algorithms for non-negative matrix factorization and latent dirichlet allocation. In: **ESANN 2016, 24th European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges, Belgium, April 26-29, 2016, Proceedings**. [S.l.: s.n.], 2016. Citado 2 vezes nas páginas 37 e 138.

FALEIROS, T. P.; ROSSI, R. G.; LOPES, A. A. Optimizing the class information divergence for transductive classification of texts using propagation in bipartite graphs. **Pattern Recognition Letters**, 2016. Citado 2 vezes nas páginas 37 e 139.

FRIEDLANDER, A.; FREAN, M.; JOHNSTON-HOLLITT, M.; HOLLITT, C. Latent dirichlet allocation for image segmentation and source finding in radio astronomy images. In: **Proceedings of the 27th Conference on Image and Vision Computing New Zealand**. New York, NY, USA: ACM, 2012. (IVCNZ '12), p. 429–434. ISBN 978-1-4503-1473-2. Disponível em: <http://doi.acm.org/10.1145/2425836.2425918>. Citado 2 vezes nas páginas 116 e 117.

FRIGG, R.; WERNDL, C. **Entropy - A Guide for the Perplexed**. Oxford University Press, 2011. In ?Probabilities in Physics?, Oxford University Press. Disponível em: <http://philsci-archive.pitt.edu/8592/>. Citado na página 57.

FUNG, B. C.; WANG, K.; ESTER, M. Hierarchical document clustering using frequent itemsets. In: **IN PROC. SIAM INTERNATIONAL CONFERENCE ON DATA MINING 2003 (SDM 2003**. [S.l.: s.n.], 2003. Citado na página 87.

GAMA, J. Clustering from data streams. In: SAMMUT, C.; WEBB, G. I. (Ed.). **Encyclopedia of Machine Learning**. Springer, 2010. p. 180–183. ISBN 978-0-387-30768-8. Disponível em: <http://dx.doi.org/10.1007/978-0-387-30164-8>. Citado na página 115.

GAMA, J.; MEDAS, P.; CASTILLO, G.; RODRIGUES, P. Learning with drift detection. In: **In SBIA Brazilian Symposium on Artificial Intelligence**. [S.l.]: Springer Verlag, 2004. p. 286–295. Citado na página 140.

GAMA, J.; ZLIOBAITE, I.; BIFET, A.; PECHENIZKIY, M.; BOUCHACHIA, A. A survey on concept drift adaptation. **ACM Comput. Surv.**, v. 46, n. 4, p. 44:1–44:37, 2014. Disponível em: <http://dblp.uni-trier.de/db/journals/csur/csur46.html#GamaZBPB14>. Citado na página 140.

GARCÍA, S.; FERNÁNDEZ, A.; LUENGO, J.; HERRERA, F. Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power. **Information Sciences**, v. 180, n. 10, p. 2044 – 2064, 2010. ISSN 0020-0255. Citado na página 110.

GAUSSIER, E.; GOUTTE, C. Relation between plsa and nmf and implications. In: **Proceedings of the 28th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval**. New York, NY, USA: ACM, 2005. (SIGIR '05), p. 601–602. ISBN 1-59593-034-5. Citado na página 64.

GEMAN, S.; GEMAN, D. Stochastic relaxation, Gibbs distributions and the Bayesian restoration of images. **IEEE Transactions on Pattern Analysis and Machine Intelligence**, Taylor & Francis, v. 6, n. 6, p. 721–741, nov. 1984. Disponível em: <http://dx.doi.org/10.1080/02664769300000058>. Citado na página 46.

GERSHMAN, S. J.; BLEI, D. M. A tutorial on Bayesian nonparametric models. **Journal of Mathematical Psychology**, v. 56, n. 1, p. 1–12, fev. 2012. ISSN 00222496. Citado na página 64.

GIROLAMI, M.; KABáN, A. On an equivalence between plsi and lda. In: **Proceedings of the 26th Annual International ACM SIGIR Conference on Research and Development in Informaion Retrieval**. New York, NY, USA: ACM, 2003. (SIGIR '03), p. 433–434. ISBN 1-58113-646-3. Citado na página 64.

GRIFFITHS, T. L.; STEYVERS, M. Finding scientific topics. **PNAS**, v. 101, n. suppl. 1, p. 5228–5235, 2004. Citado 2 vezes nas páginas 39 e 45.

HAMMOUDA, K. M.; KAMEL, M. S. Incremental document clustering using cluster similarity histograms. In: **Proceedings of the 2003 IEEE/WIC International Conference on Web Intelligence**. Washington, DC, USA: IEEE Computer Society, 2003. (WI '03), p. 597–. ISBN 0-7695-1932-6. Disponível em: <http://dl.acm.org/citation.cfm?id=946251.947110>. Citado na página 87.

HENDERSON, K.; ELIASSI-RAD, T. Applying latent dirichlet allocation to group discovery in large graphs. In: **SAC '09: Proceedings of the 2009 ACM symposium on Applied Computing**. New York, NY, USA: ACM, 2009. p. 1456–1461. ISBN 978-1-60558-166-8. Disponível em: <http://portal.acm.org/citation.cfm?id=1529607>. Citado na página 39.

HOFFMAN, M. D.; BLEI, D. M.; BACH, F. R. Online learning for latent dirichlet allocation. In: LAFFERTY, J. D.; WILLIAMS, C. K. I.; SHAWE-TAYLOR, J.; ZEMEL, R. S.; CULOTTA, A. (Ed.). **NIPS**. Curran Associates, Inc., 2010. p. 856–864. Disponível em: <http://dblp.uni-trier.de/db/conf/nips/nips2010.html#HoffmanBB10>. Citado 4 vezes nas páginas 116, 122, 126 e 128.

HOFFMAN, M. D.; BLEI, D. M.; WANG, C.; PAISLEY, J. Stochastic variational inference. **J. Mach. Learn. Res.**, JMLR.org, v. 14, n. 1, p. 1303–1347, maio 2013. ISSN 1532-4435. Disponível em: <http://dl.acm.org/citation.cfm?id=2502581.2502622>. Citado na página 125.

HOFMANN, T. Probilistic latent semantic analysis. In: **UAI**. [S.l.: s.n.], 1999. Citado 3 vezes nas páginas 39, 40 e 78.

HONG, L.; DOM, B.; GURUMURTHY, S.; TSIOUTSIOULIKLIS, K. A time-dependent topic model for multiple text streams. In: **Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining**. New York, NY, USA: ACM, 2011. (KDD '11), p. 832–840. ISBN 978-1-4503-0813-7. Disponível em: <http://doi.acm.org/10.1145/2020408.2020551>. Citado na página 118.

IWATA, T.; YAMADA, T.; SAKURAI, Y.; UEDA, N. Online multiscale dynamic topic models. In: **Proceedings of the 16th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining**. New York, NY, USA: ACM, 2010. (KDD '10), p. 663–672. ISBN 978-1-4503-0055-1. Disponível em: <http://doi.acm.org/10.1145/1835804.1835889>. Citado na página 119.

____. Sequential modeling of topic dynamics with multiple timescales. **ACM Trans. Knowl. Discov. Data**, ACM, New York, NY, USA, v. 5, n. 4, p. 19:1–19:27, fev. 2012. ISSN 1556-4681. Disponível em: <http://doi.acm.org/10.1145/2086737.2086739>. Citado na página 119.

J., Z.; LIU, Z.; CAO, X. Memory-efficient topic modeling. **CoRR**, abs/1206.1147, 2012. Citado na página 64.

JI, M.; SUN, Y.; DANILEVSKY, M.; HAN, J.; GAO, J. Graph regularized transductive classification on heterogeneous information networks. In: **Proceedings of the 2010 European Conference on Machine Learning and Knowledge Discovery in Databases: Part I**. Berlin, Heidelberg: Springer-Verlag, 2010. (ECML PKDD'10), p. 570–586. ISBN 3-642-15879-X, 978-3-642-15879-7. Disponível em: <http://dl.acm.org/citation.cfm?id=1888258.1888302>. Citado 3 vezes nas páginas 75, 101 e 102.

JOACHIMS, T. Transductive inference for text classification using support vector machines. In: **Proceedings of the Sixteenth International Conference on Machine Learning**. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1999. (ICML '99), p. 200–209. ISBN 1-55860-612-2. Disponível em: <http://dl.acm.org/citation.cfm?id=645528.657646>. Citado 4 vezes nas páginas 72, 98, 100 e 109.

KONG, X.; NG, M. K.; ZHOU, Z.-H. Transductive multilabel learning via label set propagation. **IEEE Transactions on Knowledge and Data Engineering**, IEEE Computer Society, Los Alamitos, CA, USA, v. 25, n. 3, p. 704–719, 2013. ISSN 1041-4347. Citado 2 vezes nas páginas 72 e 98.

KULLBACK, S.; LEIBLER, R. A. On information and sufficiency. **Annals of Mathematical Statistics**, v. 22, p. 49–86, 1951. Citado na página 52.

LAU, J. H.; NEWMAN, D.; BALDWIN, T. Machine reading tea leaves: Automatically evaluating topic coherence and topic model quality. In: BOUMA, G.; PARMENTIER, Y. (Ed.). **Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2014, April 26-30, 2014, Gothenburg, Sweden**. The Association for Computer Linguistics, 2014. p. 530–539. ISBN 978-1-937284-78-7. Disponível em: <http://aclweb.org/anthology/E/E14/E14-1056.pdf>. Citado 3 vezes nas páginas 63, 95 e 96.

LEE, D. D.; SEUNG, H. S. Learning the parts of objects by non-negative matrix factorization. **Nature**, Nature Publishing Group, Bell Laboratories, Lucent Technologies, Murray Hill, New Jersey 07974, USA., v. 401, n. 6755, p. 788–791, out. 1999. ISSN 0028-0836. Disponível em: <http://dx.doi.org/10.1038/44565>. Citado 2 vezes nas páginas 32 e 63.

____. Algorithms for non-negative matrix factorization. In: LEEN, T. K.; DIETTERICH, T. G.; TRESP, V. (Ed.). **Advances in Neural Information Processing Systems 13**. MIT Press, 2001. p. 556–562. Disponível em: <http://papers.nips.cc/paper/1861-algorithms-for-non-negative-matrix-factorization.pdf>. Citado 2 vezes nas páginas 63 e 64.

LI, D.; DING, Y.; SHUAI, X.; BOLLEN, J.; TANG, J.; CHEN, S.; ZHU, J.; ROCHA, G. Adding community and dynamic to topic models. **Journal of Informetrics**, v. 6, n. 2, p. 237 – 253, 2012. ISSN 1751-1577. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1751157711001039>. Citado na página 117.

LI, F.-F.; PERONA, P. A bayesian hierarchical model for learning natural scene categories. In: **Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 2 - Volume 02**. Washington, DC, USA: IEEE Computer Society, 2005. (CVPR '05), p. 524–531. ISBN 0-7695-2372-2. Disponível em: <http://dx.doi.org/10.1109/CVPR.2005.16>. Citado na página 39.

LIANG, P.; KLEIN, D. Online em for unsupervised models. In: **Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics**. Stroudsburg, PA, USA: Association for Computational Linguistics, 2009. (NAACL '09), p. 611–619. ISBN 978-1-932432-41-1. Disponível em: <http://dl.acm.org/citation.cfm?id=1620754.1620843>. Citado na página 126.

LIN, C.-J. Projected gradient methods for nonnegative matrix factorization. **Neural Comput.**, MIT Press, Cambridge, MA, USA, v. 19, n. 10, p. 2756–2779, out. 2007. ISSN 0899-7667. Disponível em: <http://dx.doi.org/10.1162/neco.2007.19.10.2756>. Citado na página 90.

MA, H.-F. Hot topic extraction using time window. In: **Machine Learning and Cybernetics (ICMLC), 2011 International Conference on**. [S.l.: s.n.], 2011. v. 1, p. 56 –60. ISSN 2160-133X. Citado na página 120.

MACKEY, L. **Latent Dirichlet Markov Random Fields for Semi-supervised Image Segmentation and Object Recognition**. 2007. Citado 2 vezes nas páginas 116 e 117.

MEI, Q.; CAI, D.; ZHANG, D.; ZHAI, C. Topic modeling with network regularization. In: **WWW**. [s.n.], 2008. Disponível em: <http://portal.acm.org/citation.cfm?id=1367512>. Citado na página 39.

METWALLY, A.; AGRAWAL, D.; ABBADI, A. E. An integrated efficient solution for computing frequent and top-k elements in data streams. **ACM Trans. Database Syst.**, ACM, New York, NY, USA, v. 31, n. 3, p. 1095–1133, set. 2006. ISSN 0362-5915. Disponível em: <http://doi.acm.org/10.1145/1166074.1166084>. Citado na página 140.

MIMNO, D.; WALLACH, H. M.; TALLEY, E.; LEENDERS, M.; MCCALLUM, A. Optimizing semantic coherence in topic models. In: **Proceedings of the Conference on Empirical Methods in Natural Language Processing**. Stroudsburg, PA, USA: Association for Computational Linguistics, 2011. (EMNLP '11), p. 262–272. ISBN 978-1-937284-11-4. Disponível em: <http://dl.acm.org/citation.cfm?id=2145432.2145462>. Citado na página 63.

MOURA, M. F.; REZENDE, S. O. A simple method for labeling hierarchical document cluster. In: . In: Proceedings for the 10th IASTED - International Conference on Artificial Intelligence and Applications (IAI 2010), Calgary-Zurich: [s.n.], 2010. p. 363 – 371. Citado na página 87.

MUQATTASH, I.; YAHDI, M. Infinite family of approximations of the digamma function. **Mathematical and Computer Modelling**, v. 43, n. 11 - 12, p. 1329 – 1336, 2006. ISSN 0895-7177. Disponível em: <http://www.sciencedirect.com/science/article/pii/S0895717705004735>. Citado 2 vezes nas páginas 65 e 67.

MUSAT, C. C.; VELCIN, J.; TRAUSAN-MATU, S.; RIZOIU, M.-A. Improving topic evaluation using conceptual knowledge. In: **Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Volume Three**. AAAI Press, 2011. (IJCAI'11), p. 1866–1871. ISBN 978-1-57735-515-1. Disponível em: <http://dx.doi.org/10.5591/978-1-57735-516-8/IJCAI11-312>. Citado na página 63.

NEAL, R.; HINTON, G. E. A view of the em algorithm that justifies incremental, sparse, and other variants. In: **Learning in Graphical Models**. [S.l.]: Kluwer Academic Publishers, 1998. p. 355–368. Citado na página 124.

NEEDHAM, T. A visual explanation of jensen's inequality. **The American Mathematical Monthly**, Mathematical Association of America, v. 100, n. 8, p. 768–771, 1993. ISSN 00029890, 19300972. Disponível em: <http://www.jstor.org/stable/2324783>. Citado na página 52.

NEWMAN, D.; LAU, J. H.; GRIESER, K.; BALDWIN, T. Automatic evaluation of topic coherence. In: **Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics**. Stroudsburg, PA, USA: Association for Computational Linguistics, 2010. (HLT '10), p. 100–108. ISBN 1-932432-65-5. Disponível em: <http://dl.acm.org/citation.cfm?id=1857999.1858011>. Citado 3 vezes nas páginas 63, 95 e 96.

NIGAM, K.; MCCALLUM, A. K.; THRUN, S.; MITCHELL, T. Text classification from labeled and unlabeled documents using em. **Mach. Learn.**, Kluwer Academic Publishers, Hingham, MA, USA, v. 39, n. 2-3, p. 103–134, maio 2000. ISSN 0885-6125. Disponível em: <http://dx.doi.org/10.1023/A:1007692713085>. Citado 2 vezes nas páginas 100 e 108.

OH, H.-J.; MYAENG, S. H.; LEE, M.-H. A practical hypertext catergorization method using links and incrementally available class information. In: **Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval**. New York, NY, USA: ACM, 2000. (SIGIR '00), p. 264–271. ISBN 1-58113-226-3. Disponível em: <http://doi.acm.org/10.1145/345508.345594>. Citado 2 vezes nas páginas 100 e 101.

PAATERO, P.; TAPPER, U. Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. **Environmetrics**, John Wiley & Sons, Ltd., University of Helsinki, Department of Physics, Siltavuorenpenger 20 D, SF-00170 Helsinki, Finland, v. 5, n. 2, p. 111–126, jun. 1994. Disponível em: <http://dx.doi.org/10.1002/env.3170050203>. Citado 2 vezes nas páginas 32 e 63.

PALSHIKAR, G. K. Keyword extraction from a single document using centrality measures. In: GHOSH, A.; DE, R. K.; PAL, S. K. (Ed.). **Pattern Recognition and Machine Intelligence, Second International Conference, PReMI 2007, Kolkata, India, December 18-22, 2007, Proceedings**. Springer, 2007. (Lecture Notes in Computer Science, v. 4815), p. 503–510. ISBN 978-3-540-77045-9. Disponível em: <http://dx.doi.org/10.1007/978-3-540-77046-6_62>. Citado na página 100.

PORTER, M. F. Readings in information retrieval. In: JONES, K. S.; WILLETT, P. (Ed.). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1997. cap. An Algorithm for Suffix Stripping, p. 313–316. ISBN 1-55860-454-5. Disponível em: <http://dl.acm.org/citation.cfm?id=275537.275705>. Citado 3 vezes nas páginas 89, 107 e 128.

ROSSI, R.; MARCACINI, R. M.; REZENDE, S. O. **Benchmarking Text Collections for Classification and Clustering Tasks**. [S.l.], 2013. Disponível em: <http://www.icmc.usp.br/

CMS/Arquivos/arquivos_enviados/BIBLIOTECA_113_RT_395.pdf>. Citado 2 vezes nas páginas 89 e 108.

ROSSI, R. G. **Classificação automática de textos por meio de aprendizado de máquina baseado em redes**. Tese (Doutorado) — Universidade de São Paulo, Instituto de Ciências Matemáticas e de Computação, São Carlos, 2015. Citado na página 72.

ROSSI, R. G.; FALEIROS, T. de P.; LOPES, A. de A.; REZENDE, S. O. Inductive model generation for text categorization using a bipartite heterogeneous network. In: **Data Mining (ICDM), 2012 IEEE 12th International Conference on**. [S.l.: s.n.], 2012. p. 1086–1091. ISSN 1550-4786. Citado 6 vezes nas páginas 34, 37, 38, 75, 100 e 103.

ROSSI, R. G.; LOPES, A. A.; FALEIROS, T. P.; REZENDE, S. O. R. Inductive model generation for text classification using a bipartite heterogeneous network. **Journal of Computer Science and Technology**, Springer US, v. 29, n. 3, p. 361–375, 2014. ISSN 1000-9000. Disponível em: <http://dx.doi.org/10.1007/s11390-014-1436-7>. Citado 9 vezes nas páginas 34, 37, 72, 75, 77, 100, 101, 102 e 103.

RUSSELL, B. C.; FREEMAN, W. T.; EFROS, A. A.; SIVIC, J.; ZISSERMAN, A. Using multiple segmentations to discover objects and their extent in image collections. In: **Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 2**. Washington, DC, USA: IEEE Computer Society, 2006. (CVPR '06), p. 1605–1614. ISBN 0-7695-2597-0. Disponível em: <http://dx.doi.org/10.1109/CVPR.2006.326>. Citado na página 39.

RUSSELL, S. J.; NORVIG, P. **Artificial Intelligence: A Modern Approach**. Pearson Education, 2003. ISBN 0137903952. Disponível em: <http://portal.acm.org/citation.cfm?id=773294>. Citado na página 46.

SAHA, A.; SINDHWANI, V. Learning evolving and emerging topics in social media: a dynamic nmf approach with temporal regularization. In: **Proceedings of the fifth ACM international conference on Web search and data mining**. New York, NY, USA: ACM, 2012. (WSDM '12), p. 693–702. ISBN 978-1-4503-0747-5. Disponível em: <http://doi.acm.org/10.1145/2124295.2124376>. Citado na página 120.

SALTON, G.; WONG, A.; YANG, C. S. A vector space model for automatic indexing. **Commun. ACM**, ACM, New York, NY, USA, v. 18, n. 11, p. 613–620, nov. 1975. ISSN 0001-0782. Disponível em: <http://doi.acm.org/10.1145/361219.361220>. Citado na página 71.

SIVIC, J.; RUSSELL, B. C.; EFROS, A. A.; ZISSERMAN, A.; FREEMAN, W. T. Discovering objects and their location in images. In: **IEEE International Conference on Computer Vision**. [S.l.: s.n.], 2005. Citado na página 39.

SONAWANE, S. S.; KULKARNI, P. A. Graph based representation and analysis of text document: A survey of techniques. **International Journal of Computer Applications**, v. 96, n. 19, p. 1–8, June 2014. Full text available. Citado na página 71.

SONTAG, D.; ROY, D. Complexity of inference in latent dirichlet allocation. In: SHAWE-TAYLOR, J.; ZEMEL, R.; BARTLETT, P.; PEREIRA, F.; WEINBERGER, K. (Ed.). **Advances in Neural Information Processing Systems 24**. [S.l.: s.n.], 2011. p. 1008–1016. Citado na página 32.

STEINBACH, M.; KARYPIS, G.; KUMAR, V. A comparison of document clustering techniques. **KDD Workshop on Text Mining**, 2000. Citado na página 87.

____. ____. In: **6th ACM SIGKDD, World Text Mining Conference**. [S.l.: s.n.], 2000. Citado na página 87.

STEYVERS, M.; GRIFFITHS, T. Probabilistic Topic Models. In: ____. **Handbook of Latent Semantic Analysis**. Lawrence Erlbaum Associates, 2007. ISBN 1410615340. Disponível em: <http://www.amazon.com/exec/obidos/redirect?tag=citeulike07-20&path=ASIN/1410615340>. Citado 4 vezes nas páginas 39, 40, 60 e 78.

STEYVERS, M.; TENENBAUM, J. B. The Large-Scale structure of semantic networks: Statistical analyses and a model of semantic growth. **Cognitive Science**, v. 29, n. 1, p. 41–78, 2005. Citado na página 100.

SUMATHY, K. L.; CHIDAMBARAM, M. Text mining: Concepts, applications, tools and issues – an overview. **International Journal of Computer Applications**, v. 80, n. 4, p. 29–32, October 2013. Full text available. Citado na página 31.

SUN, Y.; HAN, J.; GAO, J.; YU, Y. itopicmodel: Information network-integrated topic modeling. In: 0010, W. W.; KARGUPTA, H.; RANKA, S.; YU, P. S.; WU, X. (Ed.). **ICDM**. IEEE Computer Society, 2009. p. 493–502. ISBN 978-0-7695-3895-2. Disponível em: <http://dblp.uni-trier.de/db/conf/icdm/icdm2009.html#SunHGY09>. Citado 2 vezes nas páginas 100 e 101.

TEH, Y. W.; JORDAN, M. I.; BEAL, M. J.; BLEI, D. M. Hierarchical dirichlet processes. **Journal of the American Statistical Association**, v. 101, p. 1566–1581, December 2006. Disponível em: <http://ideas.repec.org/a/bes/jnlasa/v101y2006p1566-1581.html>. Citado na página 119.

TRAWINSKI, B.; SMETEK, M.; TELEC, Z.; LASOTA, T. Nonparametric statistical analysis for multiple comparison of machine learning regression algorithms. **Applied Mathematics and Computer Science**, v. 22, n. 4, p. 867–881, 2012. Citado na página 110.

VALVERDE-REBAZA, J.; VALEJO, A.; BERTON, L.; FALEIROS, T. de P.; LOPES, A. de A. A naÏve bayes model based on overlapping groups for link prediction in online social networks. In: **Proceedings of the 30th Annual ACM Symposium on Applied Computing**. New York, NY, USA: ACM, 2015. (SAC '15), p. 1136–1141. ISBN 978-1-4503-3196-8. Disponível em: <http://doi.acm.org/10.1145/2695664.2695719>. Citado na página 37.

WAAL, A. de; BARNARD, E. Evaluating Topic Models with Stability. 2008. Citado na página 62.

WAHABZADA, M.; KERSTING, K.; PILZ, A.; BAUCKHAGE, C. More influence means less work: fast latent dirichlet allocation by influence scheduling. In: MACDONALD, C.; OUNIS, I.; RUTHVEN, I. (Ed.). **Proceedings of the 20th ACM Conference on Information and Knowledge Management, CIKM 2011, Glasgow, United Kingdom, October 24-28, 2011**. ACM, 2011. p. 2273–2276. ISBN 978-1-4503-0717-8. Disponível em: <http://doi.acm.org/10.1145/2063576.2063944>. Citado na página 118.

WALLACH, H. M.; MIMNO, D. M.; MCCALLUM, A. Rethinking lda: Why priors matter. In: BENGIO, Y.; SCHUURMANS, D.; LAFFERTY, J. D.; WILLIAMS, C. K. I.; CULOTTA, A. (Ed.). **Advances in Neural Information Processing Systems 22**. Curran Associates, Inc., 2009. p. 1973–1981. Disponível em: <http://papers.nips.cc/paper/3854-rethinking-lda-why-priors-matter.pdf>. Citado na página 138.

WALLACH, H. M.; MURRAY, I.; SALAKHUTDINOV, R.; MIMNO, D. Evaluation methods for topic models. In: **Proceedings of the 26th Annual International Conference on Machine Learning**. New York, NY, USA: ACM, 2009. (ICML '09), p. 1105–1112. ISBN 978-1-60558-516-1. Disponível em: <http://doi.acm.org/10.1145/1553374.1553515>. Citado na página 61.

WAN, X.; YANG, J.; XIAO, J. Towards an iterative reinforcement approach for simultaneous document summarization and keyword extraction. In: **Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics**. Prague, Czech Republic: Association for Computational Linguistics, 2007. p. 552–559. Disponível em: <http://www.aclweb.org/anthology/P07-1070>. Citado na página 101.

WANG, C.; PAISLEY, J. W.; BLEI, D. M. Online variational inference for the hierarchical dirichlet process. **Journal of Machine Learning Research - Proceedings Track**, v. 15, p. 752–760, 2011. Citado na página 119.

WANG, H.; LANG, B. Online ngram-enhanced topic model for academic retrieval. In: **Sixth IEEE International Conference on Digital Information Management, ICDIM 2011, Melbourne, Australia, September 26-28, 2011**. IEEE, 2011. p. 137–142. ISBN 978-1-4577-1538-9. Disponível em: <http://dx.doi.org/10.1109/ICDIM.2011.6093316>. Citado na página 119.

WANG, X.; ZHANG, K.; JIN, X.; SHEN, D. Mining common topics from multiple asynchronous text streams. In: **WSDM '09: Proceedings of the Second ACM International Conference on Web Search and Data Mining**. [S.l.: s.n.], 2009. p. 192–201. Citado na página 118.

WEI, X.; CROFT, W. B. Lda-based document models for ad-hoc retrieval. In: **Proceedings of the 29th annual international ACM SIGIR conference on Research and development in information retrieval**. New York, NY, USA: ACM, 2006. (SIGIR '06), p. 178–185. ISBN 1-59593-369-7. Disponível em: <http://doi.acm.org/10.1145/1148170.1148204>. Citado na página 60.

XUE, G.-R.; SHEN, D.; 0001, Q. Y.; ZENG, H.-J.; CHEN, Z.; YU, Y.; XI, W.; MA, W.-Y. Irc: An iterative reinforcement categorization algorithm for interrelated web objects. In: **ICDM**. IEEE Computer Society, 2004. p. 273–280. ISBN 0-7695-2142-8. Disponível em: <http://dblp.uni-trier.de/db/conf/icdm/icdm2004.html#XueSYZCYXM04>. Citado na página 75.

YAN, E.; DING, Y.; MILOJEVIć, S.; SUGIMOTO, C. R. Topics in dynamic research communities: An exploratory study for the field of information retrieval. **Journal of Informetrics**, v. 6, n. 1, p. 140 – 153, 2012. ISSN 1751-1577. Disponível em: <http://www.sciencedirect.com/science/article/pii/S1751157711000976>. Citado na página 117.

YAO, L.; MIMNO, D.; MCCALLUM, A. Efficient methods for topic model inference on streaming document collections. In: **Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining**. New York, NY, USA: ACM, 2009. (KDD '09), p. 937–946. ISBN 978-1-60558-495-9. Disponível em: <http://doi.acm.org/10.1145/1557019.1557121>. Citado 2 vezes nas páginas 117 e 118.

YAROWSKY, D. Unsupervised word sense disambiguation rivaling supervised methods. In: **Proceedings of the 33rd Annual Meeting on Association for Computational Linguistics**. Stroudsburg, PA, USA: Association for Computational Linguistics, 1995. (ACL '95), p. 189–196. Disponível em: <http://dx.doi.org/10.3115/981658.981684>. Citado na página 100.

YIN, Z.; LI, R.; MEI, Q.; HAN, J. Exploring Social Tagging Graph for Web Object Classification. In: **Proc. 2009 ACM SIGKDD Int. Conf. on Knowledge Discovery and Data Mining (KDD'09)**. , Paris, France: [s.n.], 2009. Citado 2 vezes nas páginas 75 e 101.

ZHAI, K.; BOYD-GRABER, J. L. Online latent dirichlet allocation with infinite vocabulary. In: DASGUPTA, S.; MCALLESTER, D. (Ed.). **Proceedings of the 30th International Conference on Machine Learning (ICML-13)**. JMLR Workshop and Conference Proceedings, 2013. v. 28, n. 1, p. 561–569. Disponível em: <http://jmlr.csail.mit.edu/proceedings/papers/v28/zhai13.pdf>. Citado 2 vezes nas páginas 119 e 140.

ZHANG, W.; YOSHIDA, T.; TANG, X.; WANG, Q. Text clustering using frequent itemsets. **Know.-Based Syst.**, Elsevier Science Publishers B. V., Amsterdam, The Netherlands, The Netherlands, v. 23, p. 379–388, July 2010. ISSN 0950-7051. Disponível em: <http://dx.doi.org/10.1016/j.knosys.2010.01.011>. Citado na página 87.

ZHOU, D.; BOUSQUET, O.; LAL, T. N.; WESTON, J.; SCHöLKOPF, B. Learning with local and global consistency. In: THRUN, S.; SAUL, L. K.; SCHöLKOPF, B. (Ed.). **Advances in Neural Information Processing Systems 16**. MIT Press, 2004. p. 321–328. Disponível em: <http://papers.nips.cc/paper/2506-learning-with-local-and-global-consistency.pdf>. Citado 2 vezes nas páginas 101 e 102.

ZHU, X.; GHAHRAMANI, Z. Learning from Labeled and Unlabeled Data with Label Propagation. In: . [S.l.: s.n.], 2002. Citado na página 78.

ZHU, X.; GHAHRAMANI, Z.; LAFFERTY, J. Semi-supervised learning using gaussian fields and harmonic functions. In: **ICML**. [S.l.: s.n.], 2003. p. 912–919. Citado 2 vezes nas páginas 101 e 102.

APÊNDICE

A

EXEMPLOS DE TÓPICOS

As listas de tópicos extraídos das coleções *Dmoz-Business* e *classic4* pelos algoritmos *online* oLDA e oPBG, estão ilustradas respectivamente nos quadros 3 e 4. Na base *classic4* existem quatro categorias distintas relacionadas a artigos de sistemas aeronáuticos, revistas médicas, títulos e resumos de artigos da ACM (*Association for Computing Machinery*) e artigos sobre recuperação de informação. Já a base *Dmoz-Business* é composta por páginas *web* com 37 categorias relacionadas a negócios e atividades comerciais.

Nos quadros 5, 6 e 7 estão as listas de tópicos dos respectivos conjuntos de documentos *20ng*, *Dmoz-Business* e *classic4*. Esses tópicos foram extraídos utilizando os algoritmos estáticos descritos no Capítulo 3. Além dos tópicos formados por conjuntos de palavras que remetem a um tema presente na coleção, estão os valores de NPMI para cada tópico. Quanto maior o valor de NPMI melhor é o tópico.

Quadro 3 – Lista de tópicos para $K = 50$ obtido pelos algoritmos *online* oPBG e oLDA no conjunto de dados *Dmoz-Business*. Conjuntos de tópicos escolhidos para as execuções com melhores valores de acurácia na representatividade dos documentos – veja Figura 16

	lista de tópicos do algoritmo oLDA ($K = 50$, $\kappa = 0.7$, $\tau_0 = 62$ **e** $b = 32$)	**lista de tópicos do algoritmo oPBG** ($K = 50$, $\kappa = 0.5$, $\tau_0 = 1$ **e** $b = 32$)
1	fabric group financi germani tool test trader project blue specialis	descript homepag harass call booksel bid facet fair radiat arm
2	paint cloth individu towel privat pvc galleri payrol embroideri blanket	chemic magnesia llp columbu fuel victoria hatcheri heater illustr asphalt
3	link relat associ treatment franc web worldwid search portal thai	bed mask jet alo louisiana confidenti pro massei caller home
4	line footwear dealer dispers sheet featur stretch fuel direct john	bait flavor acm earli versatil select expatri actor refriger sewn
5	tabl man audit wall tape health instrument return frequenc commod	group catalyst technic deutsch ascocenda lithographi ber forestri charli root
6	chemic yarn polyamid home spun glass modular short grade extrud	cyanid casualti imped particl manuscript redevelop incom longwal australasian pcp
7	equip machineri bed coat furnish lingual turkei adhes agricultur carbon	fiber work downei cola lifeguard nanomateri glendal swing nanni slag
8	leather collect activ tan garment clean raw compound laboratori japanes	vitamin activ perform project butter miss grid irish cichlid clue
9	high qualiti doubl processor emboss cleaner flax draw larg fiberglass	porcelain diesel receiv crusher devic mud pharmaci drawer commut demograph
10	account varieti charter bookkeep distribut sheep place sell taxat quickbook	applic cpe industri portrait technolog grant receptacl foodservic knive autom
11	specif plastic supplier inject medic card surfac spain enterpris publish	individu hanger box learn formalwear collect limit maid experienti clean
12	tax pharmaceut resin gmbh end texa scientif japan weld indoor	audit yarn cobalt disput vertex iceland faa windmil nonwoven color
13	comfort hous type wind cure russian fur cast commun robert	machineri treasur mannequin lectur dyestuff vita cover pdf motorhom camper
14	manufactur wool rubber curtain oil air jet thread pty mat	machin equip garment baler flexo materi specialist keyword bravo emt
15	provid rug repres consum spanish certifi make fine energi egypt	inform czech california condition street oxygen practition osha leather pattern
16	color softwar full care pvt readi import art comput contract	woven reserv yamaha pet rug ingredi suzuki laboratori kart stylist
17	manag prepar sale lubric specialist household california san republ close	made talc man soapston hotlin freelanc gfi attempt italian catalog
18	cpa base addit kitchen autom liquid bath econom ic class	fabric itali mushroom reclam windpump dy screenprint retain elliot walker
19	electr outdoor govern aerospac rope flame zealand central power hemp	standard nurs transpond host wing tehachapi pencil bespok edg cross
20	industri inform nonwoven spin food area toll petrochem needlepunch council	transmiss european imag skin relai major eucalyptu lcd tow japan
21	firm organ suppli edit healthcar white fertil core biographi pesticid	braill tabl rest recycl kite interchang crop incub phone brian
22	technic applic intermedi solut offic focus form dry safeti properti	traction nickel arc pole walk cruis mcguir freighter shade hydraul
23	product compani itali distributor profil fill seal epoxi late moor	purebr public institut taxat toggenburg helicopt informat muse nubian furnish
24	develop engin limit control new directori electron metal gener steel	multi bow list connecticut sol mit perenni emanuel trai bombardi
25	machin sew integr wholesal foam plant oper cushion spare fold	charter license shop compact lubric joe magazin promot babbitt paint
26	custom linen upholsteri fashion diversifi acryl tan calcul hide window	polyest chef includ sodium bookkeep gann newli mail speci weapon
27	own mark rigid stapl roll support jiangsu dioxid famili chlorid	edi toilet ira donut microlight forag bread mason embryo fixtur
28	includ usa sew thermoplast rich hardwar function essenti swiss cam	franc popcorn xerox osmosi wheel sourc fulfil metropolitan york facilit
29	carpet vertic water natur pdf lamin small tuft extens llc	ground tungsten licenc graphit distribut chipper career natur dresser forklift
30	design print technolog filament protect decor serv studio precis marin	deriv australia apparel fenc hamilton resal relat car fundrais let
31	ring australia open piec extrus label advisori reinforc incorpor temperatur	ccna kitchen personalis parachut cattlemen adventur ductil bantam centuri site
32	process mold file complet filtrat furnitur data acet recycl mill	sod push trainer shoe barri brass skateboard orthot tifgreen furnac
33	fiber made german corpor srl hand restaur global polish hotel	market theatric cash oxid greenvil abram tile residenti slice proofread
34	english woven blend dy chines agent jacquard quilt pattern viscos	van comput trail mcdonald copywrit pedigre packer meat web narrow
35	catalog offer world promot educ dedic free switzerland indian employ	dye driveshaft hall rotterdam supplier world alarm door pass wave
36	automot solid canada synthet composit sprai thermal shop terri proprietari	tan india bartend nonfict fold stanton australian emt program make
37	part special corp cover soft veget tent model bui sew	cotton weav trust intercultur bmw cereal aurora unifi trade literari
38	finish polyest china materi export polym wet packag cut retail	recogn parish ethylen brooklyn rehabilit trent provid reinforc bedroom hold
39	amp multi plain glossari belgium militari tube feed growth motorcycl	tax nois culinari vaccin speedwai pillow cpa molecular franchis shelter
40	cotton auxiliari spa state facil unit bag convert pakistan bond	china spong coastal deer papho flag barrett locat alma germani
41	market produc commerci turkish residenti polyurethan ceram tile refriger assembl	sew ladder part turkei cfp carol cigar extract steel auckland
42	detail rang wide trade work event nation hospit sampl spread	manufactur harvest women develop prepress valencia plow mortar solder buffet
43	site list descript price silver client staff cook poland depart	bike tekstil yellow cafe season powersport cut directori worker bakeri
44	servic india italian standard good bureau jersei websit outsourc investor	mold compactor total annuiti sweden aero gage product neem convers
45	research involv institut american public calendar guid train member assist	scanner histor nutrit simment chines water pigment turbin shortwav cooki
46	textil locat silk menu chef crust franchis attach wheel histor	specialti arizona coat dealership patient carbon zirconium sport aggreg taught
47	system perform acid european south taiwan black monthli seat focu	sheep hand join donkei nation type canadian bonanza headquart granul
48	film plan tekstil estat report elast video anti thailand block	lean shredder shetland spin true score piano intermediari divorc or
49	busi consult specialti paper construct contact profession stock mainten sport	compani propuls hear sticker gem frozen audiolog orthoped brangu groceri
50	dye apparel knit accessori weav pigment dyestuff artifici loom dy	film chimnei mcse espresso blimp respiratori satisfact chicken extra buffalo

Fonte: Dados da pesquisa.

Quadro 4 – Lista de tópicos para $K = 50$ obtido pelos algoritmos *online* oPBG e oLDA no conjunto de dados *classic4*. Conjuntos de tópicos escolhidos para as execuções com melhores valores de acurácia na representatividade dos documentos – veja Figura 17

	lista de tópicos do algoritmo oLDA ($K = 50$, $\kappa = 0.9$, $\tau_0 = 1$ **e** $b = 32$)	**lista de tópicos do algoritmo oPBG** ($K = 50$, $\kappa = 0.9$, $\tau_0 = 1$ **e** $b = 32$)
1	test lift search integr procedur essenti effect student answer direct	system inform comput program data librari result problem effect gener
2	small occur origin format displac serial rapidli converg head strong	increas number flow effect function result layer boundari pressur method
3	measur effect time cost frequenc transform word rate test kei	equat flow solut method boundari result pressur number theori layer
4	local pattern singl thermal relationship predict tumor law low squar	flow cylind number effect system result pressur method present bodi
5	treatment characterist correl agreement social behavior incid earli reflect forward	theori buckl result shell flow pressur effect number present system
6	studi surfac function relat develop report gener author discuss organ	flow number system effect boundari layer result pressur control method
7	number increas result investig separ larg experi show type reduc	boundari layer flow problem number plate effect solut method pressur
8	research growth interact user basi basic system prepar engin input	cell dna system line inform result studi flow present effect
9	free similar theoret edg linear comparison natur emploi parallel proport	system structur inform search data program number comput result effect
10	method shown produc literatur appear suggest result statist long form	gener flow method result number system pressur effect problem present
11	boundari theori wave commun skin diseas demonstr uniform distanc analyt	wing flow number bodi effect result pressur method ratio superson
12	case data analysi compress blood code medic alter histori criteria	flow number system inform result effect paper present layer shock
13	wall observ group special symmetr expans remain clinic thrust cardiac	pressur number jet flow method effect mach result increas bodi
14	high structur level region combin larg degre concept critic process	system inform program librari effect problem method flow result relat
15	made includ languag stress direct method program type purpos set	flow shock number layer pressur boundari heat bodi temperatur effect
16	gener mode cent multipl precis greater altitud plastic visual addit	inform system librari comput present data program problem develop method
17	induc close children survei posit disturb finit cell kidnei renal	number cell mach pressur flow effect result layer boundari increas
18	respons record tabl illustr applic exampl decis condit expect provid	flow number pressur result effect method system boundari layer present
19	flow pressur heat temperatur hyperson fluid control numer mach superson	librari inform system book comput develop problem data method result
20	system develop evalu initi differenti techniqu automat oscil involv requir	flow number bodi method result pressur structur cell boundari layer
21	normal blunt flutter extend plasma defect wide base hormon vibrat	system languag index program inform term comput document retriev data
22	librari number point work properti method shown reason machin wind	flow number effect result pressur boundari layer method system made
23	result ratio rang paper effect discuss present coeffici rate valu	layer boundari flow number effect pressur laminar studi heat method
24	obtain transfer document factor import refer public collect publish carri	system patient number result flow cancer effect program languag present
25	experiment consider differ chang mechan file titl lower relev unit	comput system inform librari develop patient program process data result
26	calcul algorithm scheme polynomi invers arrai fourier taper planform invert	data number flow base system pressur result comput effect method
27	servic center manag mix bend administr project varieti receiv world	effect heat flow number transfer layer boundari result system pressur
28	transit ga rat cool volum inject half isol left briefli	system librari inform comput program problem design flow data oper
29	veloc scientif journal recent dna protein neg exit deal hypertrophi	flow solut boundari layer method equat shock wave approxim number
30	cylind term compar inform process design perform oper standard nation	wing flow drag pressur effect lift number flap ratio result
31	shock stream book complet review treat concern patient effort ground	case flow inform result system present scienc patient number studi
32	year area period dai strain presenc progress failur adult mark	flow result number theori effect pressur boundari layer present method
33	field signific limit improv weight potenti current fraction primari accur	algorithm flow edg method number result pressur problem present theori
34	wing continu account elast side cultur simultan handl satisfactori instruct	effect number flow result speed system increas method high pressur
35	problem solut approxim stabil flat practic complex scale object present	test number result flow pressur effect method system measur present
36	subject jet retriev curv flight ag aircraft steadi bibliograph plane	panel group flutter children effect result number studi flow system
37	equat model buckl effect provid condit reynold maximum creep estim	system flow result number perform case effect present pressur method
38	bodi drag dynam exist mathemat text rise common optim hand	flow boundari number layer method result effect pressur solut obtain
39	base good entri deflect column articl matrix curvatur respect explor	flow number pressur method result free effect layer procedur boundari
40	layer angl order laminar mach forc lead cone circular attack	time system number flow effect result method present pressur comput
41	patient acid infect serum therapi lung marrow infant lesion random	ventricular aortic pressur left patient flow regurgit defect system librari
42	air turbul panel power predict reduct exhaust electr act need	flow number wing lift method effect result pressur boundari theori
43	comput scienc profil higher catalog classif reaction scientist energi technolog	bodi flow number pressur wing effect angl ratio result mach
44	dimension shape total part shear motion moment definit life fix	librari system data inform program number result comput effect develop
45	distribut aspect construct characterist abstract cross concentr memori interfer rough	method flow result system number calcul comput present effect pressur
46	index equilibrium librarian step revolut previou prove frequent disciplin adapt	program flow number system method result pressur bodi comput effect
47	line note product percent plan program understand human axi interest	flow surfac number pressur layer boundari result separ effect method
48	found plate thick thin static tube tissu stage cancer liver	system flow inform result method librari number problem effect studi
49	inform shell paramet determin chemic vari low involv compar error	result flow number effect pressur present layer boundari model system
50	speed cell load activ state due list sourc featur part	flow number system result method effect layer pressur boundari comput

Fonte: Dados da pesquisa.

Quadro 5 – Listas com os 20 tópicos com maiores valores de NPMI para $K = 50$. Esses tópicos foram obtidos pelos algoritmos estáticos aplicados no conjunto de dados *20ng*.

svd+nmf	npmi	lda	npmi
muslim bosnian serb war bosnia croat europ yugoslavia serbian nazi	0.25	space orbit launch earth satellit moon system planet probe solar	0.25
armenian turkish armenia turk soviet serdar argic turkei genocid extermin	0.24	christian jesu god church christ bibl faith time sin dai	0.2
team hockei fan plai playoff nhl leaf cup buffalo bruin	0.22	team game hockei plai player year subject season nhl playoff	0.19
christian jesu church christ religion bibl faith homosexu paul belief	0.22	medic diseas food patient doctor effect studi health cancer treatment	0.19
mhz simm speed chip motherboard cpu ram board clock memori	0.21	god bibl christian homosexu jesu love lord sin hell christ	0.19
imag bit gif format jpeg data displai tiff process viewer	0.2	armenian turkish armenia peopl turk turkei kill soviet genocid russian	0.19
israel isra arab jew palestinian jewish peac kill adam attack	0.19	jew arab muslim israel jewish war isra state countri nazi	0.17
disk do copi floppi program drive instal hard mac softwar	0.19	drive do window disk scsi driver system problem card control	0.15
player team year hit pitch win run basebal plai pitcher	0.19	file imag jpeg color format gif bit program version convert	0.15
gun firearm weapon crime law crimin control polic handgun arm	0.18	game year hit player run basebal win team time pitch	0.15
drive scsi id hard control seagat floppi bu devic meg	0.17	insur monei pai tax cost privat health columbia care system	0.12
god sin exist jesu bibl atheist faith love hell heaven	0.17	graphic ftp imag pub data softwar program anonym packag mail	0.11
file format gif bmp convert directori ftp program postscript zip	0.16	homosexu write brian gai cramer sex articl men subject sexual	0.1
font printer print truetyp laser charact atm deskjet problem postscript	0.16	card mac monitor appl subject video mhz board modem bit	0.1
card driver video diamond bu ati vlb vga isa mode	0.15	kei encrypt chip clipper secur govern system escrow subject public	0.09
fbi fire batf koresh waco atf compound burn children stratu	0.12	window run program applic server subject displai widget manag sun	0.08
space orbit pat shuttl mission digex launch access hst moon	0.12	gun fire fbi koresh write batf articl waco subject children	0.08
system comput unix problem gatewai tape softwar user control work	0.12	peopl don exist reason point thing evid question claim argument	0.08
graphic program code softwar widget motif packag librari server sun	0.12	presid clinton stephanopoulo vote work myer job senat don question	0.08
car engin ford dealer auto speed mile mustang bui oil	0.11	subject mail post list andrew newsgroup cmu send scott articl	0.06
pbg	**npmi**	**kmeans+pbg**	**npmi**
space nasa gov orbit launch earth mission moon satellit shuttl	0.24	god faith christian jesu bibl christ sin lord church hell	0.24
team plai player year season game leagu hockei defens score	0.22	drive scsi control hard disk id system problem bu floppi	0.19
medic diseas patient doctor health treatment cancer studi infect effect	0.22	team plai player hockei goal nhl period game season pit	0.18
drive scsi disk control hard id floppi pin system bu	0.2	imag color file bit format jpeg gif displai data program	0.18
armenian turkish armenia turk peopl soviet russian genocid azerbaijan villag	0.19	armenian turkish peopl armenia turk greek turkei soviet kill russian	0.17
imag data comput printer print graphic fax softwar packag program	0.18	mac appl mhz speed chip bit cpu memori ram subject	0.16
god jesu christ sin love lord man life heaven hell	0.18	space nasa orbit launch earth henri satellit moon gov mission	0.16
window do program softwar microsoft subject copi run disk system	0.15	mail post list inform address send newsgroup group email internet	0.14
file email system server user font inform directori anonym run	0.15	israel isra peac arab palestinian subject write state attack polici	0.13
jew war jewish muslim arab countri nazi israel nation state	0.14	window do run microsoft file manag program system subject applic	0.13
fire fbi koresh batf waco write burn atf ga compound	0.14	win team game run pitch won lost subject year score	0.12
gun law weapon crime firearm peopl arm control polic crimin	0.13	window widget server subject motif applic sun displai set file	0.11
card monitor subject mhz video chip ram speed appl cpu	0.13	jew jewish arab nazi write israel american adam anti articl	0.11
file imag format jpeg ftp list gif graphic pub site	0.13	kei encrypt chip clipper secur govern system escrow phone algorithm	0.11
christian church paul law homosexu god bibl peopl word jesu	0.12	card monitor driver video subject mode vga screen color problem	0.1
monei cost year tax pai job work spend program fund	0.11	gun law peopl weapon crime firearm state arm control amend	0.1
kei encrypt chip clipper secur escrow phone subject algorithm system	0.11	food msg effect eat studi peopl diet vitamin subject candida	0.09
game fan subject hockei playoff team cup win leaf wing	0.11	govern monei insur pai cost peopl privat compani isc market	0.09
power batteri subject circuit heat signal electr light suppli radio	0.1	research inform health april medic cancer center year nation diseas	0.09
homosexu men state ohio gai ac sex sexual write women	0.09	softwar disk program copi subject system machin work protect instal	0.09
hlc+pbg	**npmi**	**Pbg-thread**	**npmi**
space earth orbit planet nasa mar mission moon satellit sky	0.26	god jesu christ sin lord bibl christian church word heaven	0.23
game team hockei plai goal nhl playoff player win period	0.22	israel jew isra arab jewish muslim palestinian peac war kill	0.23
drive do disk scsi hard control system id problem floppi	0.21	medic diseas patient health doctor effect treatment cancer infect studi	0.22
god jesu christian christ church bibl sin lord peopl word	0.2	space gov nasa launch shuttl orbit satellit mission station pat	0.17
armenian turkish muslim turk armenia peopl turkei greek genocid soviet	0.2	earth henri moon toronto planet space orbit mar solar atmospher	0.17
israel isra arab jew jewish write articl palestinian subject peopl	0.17	max armenian turkish turk armenia turkei greek muslim genocid soviet	0.17
doctor effect pain patient medic studi treatment subject problem vitamin	0.15	god christian exist religion belief life atheist peopl faith truth	0.17
card mhz bit mac bu appl chip ram speed memori	0.15	imag color bit file jpeg displai gif format program convert	0.16
post group newsgroup discuss messag net read new alt usenet	0.14	window do run driver file microsoft printer print program font	0.15
file ftp program version format graphic softwar pub system gif	0.12	drive disk hard control system tape sale floppi subject scsi	0.15
game win team year basebal run subject pitch won lost	0.11	fire fbi koresh batf waco burn children atf compound ga	0.14
gun law crime weapon firearm peopl state control crimin legal	0.11	card scsi video bit mhz bu board chip memori speed	0.14
mail list univers inform comput address send email fax subject	0.11	file ftp graphic pub softwar packag data system site directori	0.13
window applic run file program widget manag server motif set	0.11	game plai goal score win shot boston espn subject team	0.13
kei chip encrypt bit secur de algorithm public messag block	0.11	car speed front road tire drive radar brake detector shift	0.12
homosexu christian paul peopl gai sex cramer men sexual write	0.09	law govern right peopl state legal court case power polit	0.12
health insur medic bank diseas columbia aid research gordon care	0.09	game year hit player basebal run pitch team win subject	0.12
monitor driver video mous subject mode vga card window problem	0.09	technolog secur govern commun encrypt develop system privaci agenc public	0.11
peopl exist moral god don religion reason claim question thing	0.09	gun weapon arm firearm control peopl crime amend carri law	0.11
modem port subject connect bb serial mac work softwar problem	0.08	kei encrypt chip bit de number messag block algorithm secur	0.11

Fonte: Dados da pesquisa.

Quadro 6 – Listas com os 20 tópicos com maiores valores de NPMI para $K = 50$. Esses tópicos foram obtido pelos algoritmos estáticos aplicados no conjunto de dados *Dmoz-Business*.

svd+nmf	npmi	lda	npmi
water electr suppli power energi solar control ga pump util	0.19	metal recycl steel wood manufactur aluminum weld alloi laser plastic	0.22
bank financi invest fund mortgag privat investor oper trust save	0.18	fabric textil manufactur fiber yarn applic english technic cotton woven	0.17
fabric english applic textil technic yarn cotton woven dy polyest	0.16	medic care health healthcar consum product devic servic market pharmaceut	0.16
plastic recycl paper metal mold wast inject materi usa scrap	0.15	softwar electron system comput manag autom inventori hardwar track solut	0.16
retail displai store fixtur merchandis point accessori slatwal wholesal rack	0.14	trade invest fund financ hous market capit manag mortgag investor	0.16
research medic clinic pharmaceut trial devic data biotechnolog report analysi	0.14	program univers technolog busi affili scienc offer center social depart	0.13
system control integr data electron instal secur autom radio supplier	0.12	system water instal manufactur pump design solar wireless telephon heat	0.13
commun relat public advertis agenc radio media wireless network satellit	0.12	industri manufactur product chemic materi supplier plastic india china process	0.13
consult firm strategi environment client strateg independ focus implement specialis	0.12	state unit canada south australia nation europ canadian western america	0.12
softwar hardwar comput support po track analysi data inventori integr	0.12	electr power manufactur gener util industri energi motor control cooper	0.12
web commerc site internet host electron databas applic link program	0.12	market servic technolog solut develop busi web commun design internet	0.12
trade export import stock lead trader directori global exchang foreign	0.11	paper packag industri board manufactur bag convert pulp adhes lamin	0.11
technolog high focus comput advanc univers telecommun nasdaq data scienc	0.11	train safeti school offer cours educ drive program class aid	0.1
print graphic paper label publish press book screen full color	0.11	rang product manufactur wide varieti industri applic includ coat technic	0.1
real estat apprais commerci serv properti residenti area counti valuat	0.11	engin manufactur design test control system air product compon instrument	0.1
manag project asset invest risk perform institut fund properti strateg	0.1	manag consult servic project develop train provid firm group offer	0.09
market advertis internet strategi direct brand stock promot agenc strateg	0.1	equip dealer manufactur farm fixtur sale light supplier accessori cattl	0.09
account tax firm cpa amp financi audit bookkeep prepar individu	0.1	new inform articl onlin free link report resourc featur directori	0.09
machin tool part machineri vend accessori precis cut sew suppli	0.1	product video film sport compani anim base pictur produc sound	0.09
develop applic leadership organiz strategi implement program econom team skill	0.1	machin construct industri contractor hospit vend ic offshor pipelin drill	0.08
pbg	npmi	kmeans+pbg	npmi
english plastic manufactur textil machineri fabric fiber technic applic yarn	0.14	metal steel sheet aluminum wire tube mill roll weld plate	0.21
firm account financi tax employe individu cpa complet benefit amp	0.08	water electr power gener energi sourc util light ga solar	0.16
market process high construct recycl qualiti wast consum corp tech	0.07	corpor invest bank privat secur fund investor financ credit central	0.16
equip supplier oil specialist ga hous laboratori land surfac fuel	0.07	control air aircraft transport pump aviat handl flight ship airport	0.16
develop commerci electron promot commerc privat sector countri incorpor aerospac	0.07	system softwar support data integr comput autom track hardwar regist	0.14
home water air pump lead glass clean heat gold hot	0.07	develop technolog solut site web applic internet commerc databas host	0.13
servic provid full build pennsylvania expertis contractor advisori hong kong	0.06	research medic pharmaceut publish book press clinic devic drug laboratori	0.12
system softwar integr articl mold hardwar scienc capabl mill inject	0.06	sale part sell truck vehicl rental dealer travel car trailer	0.12
design state unit varieti focus graphic art type central architectur	0.05	produc natur food low powder nutrit carbon dairi supplement feed	0.12
retail client packag advertis displai portfolio studi valuat fixtur store	0.05	industri process materi chemic food specialti india agricultur pharmaceut addit	0.11
profil data properti import rate sheet storag dairi refriger expert	0.05	test institut qualiti standard requir measur certif inspect describ complianc	0.11
web network internet agenc assist report coach partner help interact	0.05	amp account financi tax firm individu cpa audit prepar certifi	0.1
line canada stock publish fund investor farm invest net contain	0.05	custom print label screen color card made graphic maker printer	0.1
engin rang wide compon specialti administr purchas color magnet mechan	0.04	technic english specif fabric textil catalog applic fiber manufactur yarn	0.1
industri profession medic worldwid instal devic telecommun translat staf optic	0.04	associ organ nation american member profit repres issu govern membership	0.1
oper pharmaceut global limit advanc field clinic format nasdaq capit	0.04	commerci area serv real estat apprais properti counti residenti citi	0.1
metal sell steel larg screen bui european aluminum roll anim	0.04	group insur health care plan independ benefit employe agent administr	0.09
organ area chemic energi transport protect trailer bag tube recoveri	0.04	machin tool drill cut precis hand set user core hydraul	0.09
train safeti career employ mainten aid marin cours seminar magazin	0.04	manag consult project client human plan strateg practic risk asset	0.09
custom plan order strateg requir car mail sampl creativ item	0.04	usa environment recycl wast advanc plant type clean contain treatment	0.09
hlc+pbg	npmi	Pbg-thread	npmi
system applic data databas access file user pdf end voic	0.14	firm plan project invest bank newslett strateg partner financ capit	0.13
research test analysi focus facil field clinic pharmaceut drug laboratori	0.14	technolog solut corpor test agenc integr insur strategi benefit implement	0.11
invest stock report privat fund partner investor capit screen asset	0.14	technic plastic recycl ga wast sheet fiber optic file plate	0.1
technic fabric manufactur specif textil steel fiber applic collect yarn	0.13	retail pharmaceut health publish care american univers florida consum healthcar	0.08
high radio telecommun card mobil cabl tech page wireless phone	0.12	site fabric textil manufactur finish applic technic yarn english cotton	0.08
industri control automot instrument devic autom monitor measur automat valv	0.12	account financi link tax cpa life audit prepar radio inventori	0.05
new articl cover newslett issu forum magazin author current review	0.12	nation english assist german multi germani posit french spanish italian	0.05
process materi water metal recycl sell machineri wast pump wood	0.12	profession educ promot resourc human govern membership canadian effect affili	0.05
video photographi imag instal film wed photograph audio music record	0.11	develop search mainten aid mobil aerospac cpr hardwar structur weld	0.05
aircraft australia repair truck vehicl rental aviat inventori car trailer	0.11	busi trade internet global worldwid cover larg countri peopl start	0.05
manag consult plan oper project client strateg assess risk survei	0.11	market sale area directori report wholesal profit standard agent help	0.04
environment qualiti air specialist standard specialis document complianc environ iso	0.1	full oper north specialist south telecommun nasdaq mortgag africa east	0.04
firm amp account financi tax cpa individu audit prepar certifi	0.1	custom support center emerg phone low furnitur call email concept	0.04
print chemic paper specialti coat label color paint sheet fine	0.09	electron water commerc distribut enterpris advanc pump rental open tank	0.04
manufactur plastic compon mainten mold assembl precis militari aerospac composit	0.09	system paper power util track convert item instal dairi pulp	0.04
equip suppli supplier distributor machin accessori dealer protect cut sport	0.09	serv real estat video apprais properti counti tip natur valuat	0.04
develop custom solut softwar support packag enterpris implement india host	0.09	softwar environment store displai transport protect fixtur contain agricultur merchandis	0.03
base technolog internet integr comput long window thermal door platform	0.08	manufactur machin electr institut china corp present focu precis vend	0.03
locat food canada histori restaur menu coffe british ontario bar	0.08	commun stock analysi photographi time automot advic wed specialis photograph	0.03
product network health care pharmaceut consum life natur supplement feed	0.08	advertis sell drill media dealer plant farm bui exchang voic	0.03

Fonte: Dados da pesquisa.

Quadro 7 – Listas com os 20 tópicos com maiores valores de NPMI para $K = 50$. Esses tópicos foram obtido pelos algoritmos estáticos aplicados no conjunto de dados *classic4*.

svd+nmf	npmi	lda	npmi
matrix invers eigenvalu symmetr real eigenvector element hilbert tridiagon spars	0.27	ventricular defect case left aortic patient cardiac heart septal pressur	0.22
polynomi root evalu symmetr zero orthogon legendr recurs coeffici jacobi	0.25	patient cancer treatment therapi case excret diseas hepat breast result	0.22
cell marrow cultur tissu dna bone lymphocyt antigen lung viru	0.24	inform scienc research scientif studi commun social scientist work technolog	0.16
patient case cancer diseas treatment children group therapi ventricular clinic	0.18	hypothermia sodium diabet effect increas insipidu decreas perfus dog blood	0.16
buckl shell cylind load stress cylindr thin axial circular compress	0.18	case patient speech pneumonia syndrom hydrocephalu lesion develop diseas normal	0.16
scienc scientif research social scientist studi commun literatur work technolog	0.16	problem solut method equat numer linear point approxim solv obtain	0.16
librari book servic univers librarian public academ research school educ	0.15	protein fraction len nickel weight rna crystallin acid solubl compon	0.16
layer boundari laminar turbul plate separ veloc wall transit friction	0.15	case renal amyloidosi diseas lesion report present system amyloid kidnei	0.16
equat solut differenti linear numer approxim ordinari simultan exact obtain	0.14	boundari layer flow heat temperatur laminar veloc solut transfer wall	0.15
growth hormon rat acid increas effect dai kidnei serum plasma	0.14	function algorithm integr error differenti formula polynomi squar seri evalu	0.15
fortran subroutin charact input manipul compil format chain output cobol	0.13	flow nozzl superson symmetr compressor dimension axial part veloc design	0.15
method calcul iter compar gauss applic point appli comparison requir	0.13	inform system search data retriev base file user servic comput	0.15
symbol manipul algebra subroutin string express formula notat techniqu charact	0.13	algorithm matrix grammar pattern fit gener invers matric set permut	0.14
languag translat compil formal syntax semant grammar natur featur machin	0.12	growth hormon rat effect antigen inject dai tissu activ serum	0.14
jet pressur number mach angl nozzl cone reynold measur ratio	0.12	shock wave bodi flow ga air hyperson heat stagnat temperatur	0.14
matric test determin invers evalu tridiagon probabl tunnel set boolean	0.12	arithmet list bit number point binari time float digit decim	0.14
flutter panel plate speed aerodynam edg flat mach high rectangular	0.12	stage oxygen fluid mycoplasma pressur increas cerebr blood tension brain	0.13
arithmet precis compil float point multipl express complex error binari	0.12	cell dna cultur marrow strain kidnei infect lymphocyt studi label	0.13
flow bodi hyperson field superson past theori fluid viscou dimension	0.12	wing lift theori bodi calcul aerodynam airfoil flow effect ratio	0.13
function gamma error bessel minim complex coulomb real argument distribut	0.11	cell electron membran lung alveolar inclus surfac bodi line cytoplasm	0.12
pbg	**npmi**	**kmeans+pbg**	**npmi**
blood acid level increas concentr oxygen glucos fatti plasma metabol	0.25	patient case treatment cancer diseas therapi renal clinic drug excret	0.27
boundari layer flow laminar turbul veloc surfac separ fluid dimension	0.21	ventricular blood defect left pressur patient oxygen arteri aortic heart	0.22
flow ga air tube nozzl equilibrium pressur superson veloc static	0.19	cell tissu antigen human cultur tumor rnarrow studi bone lung	0.21
algorithm matrix polynomi graph squar gener invers arithmet root find	0.18	algorithm function complex graph find polynomi permut exponenti plot ration	0.18
cell human antigen cultur infect tumor fraction strain isol tissu	0.18	boundari layer laminar separ turbul flow transit plate surfac effect	0.15
error flutter frequenc correct panel mode vibrat nois oscil damp	0.17	flow fluid dimension veloc compress superson theori airfoil steadi past	0.15
wing jet lift ratio effect airfoil pressur forc aerodynam drag	0.16	research scienc social knowledg scientif book scientist studi work field	0.15
growth hormon dai rat kidnei increas renal effect inject diabet	0.15	problem integr formula solv formul finit differ satisfi involv treat	0.14
buckl load cylind shell plate thin circular stress pressur cylindr	0.15	method determin calcul valu approxim paramet factor squar appli accuraci	0.14
ventricular defect left case side patient curv aortic cardiac heart	0.15	growth increas rat acid hormon effect protein dai kidnei concentr	0.13
program languag compil fortran statement algol instruct symbol manipul translat	0.14	dna infect case isol hydrocephalu infant famili pneumonia phage hepat	0.13
cell stage marrow bone label lymphocyt irradi mark liver hepat	0.13	program languag compil machin grammar translat context semant formal implement	0.12
comput machin input output displai gener aid graphic termin ibm	0.13	system oper user design interact cost implement displai graphic environ	0.12
bodi flow angl blunt shape cone pressur hyperson nose theori	0.12	inform servic current technic user research center scienc technolog scientif	0.12
patient treatment case therapi period primari cancer diseas year month	0.12	gener distribut statist frequenc correl random probabl mode nois engin	0.12
method techniqu applic appli calcul estim gener compar result comparison	0.11	free jet pressur ratio stream nozzl effect mix exhaust diamet	0.11
solut equat approxim case numer linear differenti seri obtain exact	0.11	wing lift speed flutter edg effect ratio aerodynam lead forc	0.11
shock wave pressur free mach stream interact transit superson hyperson	0.11	librari univers book librarian survei servic educ school public medic	0.11
index retriev document inform search term refer automat word relev	0.1	shock wave ga air equilibrium flow tube dissoci pressur high	0.11
scienc research scientif literatur field review year product scientist publish	0.1	heat temperatur transfer rate stagnat wall inject conduct thermal surfac	0.11
hlc+pbg	**npmi**	**Pbg-thread**	**npmi**
diseas patient renal hepat liver amyloidosi clinic lesion syndrom develop	0.27	level acid plasma concentr glucos fatti increas blood liver serum	0.25
cell human cultur dna strain antigen tumor infect tissu mycoplasma	0.22	human activ cell antigen tumor detect specif reaction tissu tumour	0.2
patient primari cancer therapi tumor treatment carcinoma drug surviv agent	0.21	patient case diseas clinic therapi lesion treatment year diagnosi report	0.2
flow boundari solut layer veloc fluid equat compress field plate	0.16	inform scienc research commun scientif scientist social technolog studi technic	0.17
program comput languag machin compil input implement fortran process manipul	0.15	method solut equat numer problem approxim linear obtain differenti appli	0.16
blood marrow cell bone oxygen label lymphocyt dog increas hypothermia	0.14	boundari layer heat transfer temperatur laminar wall flow surfac plate	0.16
algorithm effici gener graph random find bound minim number permut	0.14	program languag compil machin fortran input algol grammar statement translat	0.15
inform research commun network scientif scientist process technic scienc technolog	0.14	algorithm multipl find polynomi arithmet recurs precis permut elimin divis	0.15
cell electron lung membran inclus alveolar synthesi structur studi epitheli	0.13	dna cell marrow label bone sensit synthesi nickel irradi phage	0.14
point equat integr numer variabl transform differenti seri formula comput	0.13	shock wave ga air flow equilibrium tube temperatur dissoci hyperson	0.14
shock bodi flow wave hyperson pressur blunt cone nose shape	0.13	cell lung electron studi membran alveolar inclus tissu type lymphocyt	0.14
cylind buckl shell load plate circular thin stabil pressur stress	0.13	wing lift flutter aerodynam ratio speed panel forc calcul aircraft	0.13
method problem solut appli solv linear applic present gener illustr	0.12	function order integr deriv express error formula valu calcul squar	0.13
level normal high degre higher lower low significantli patient excret	0.12	pressur jet free ratio stream nozzl mix static exhaust superson	0.13
languag set structur concept defin rule definit algol formal grammar	0.12	inform retriev search document relev question user kei answer request	0.13
rat acid increas kidnei dai anim fatti inject content liver	0.12	growth hormon rat dai increas kidnei anim renal diabet respons	0.12
action effect increas plasma activ parathyroid phosphat diabet water sodium	0.11	buckl stress load shell cylind thin compress theori circular stabil	0.11
number product matrix element multipl size probabl invers sequenc segment	0.11	flow lead dimension edg superson theori distribut pressur veloc airfoil	0.11
approxim object paramet complex valu squar polynomi curv fit root	0.11	system design oper user interact cost implement develop share capabl	0.11
index retriev document search term inform word automat relev abstract	0.11	bodi blunt angl cone pressur hyperson nose shape theori attack	0.1

Fonte: Dados da pesquisa.

APÊNDICE

B

ANÁLISE DO PARÂMETRO DO TPBG

Neste capítulo são descritos os ranques médios obtidos pela aplicação do procedimento de Friedman nos resultados do algoritmo semissupervisionado TPBG. O objetivo é analisar a influência dos resultados na variação do parâmetro $\alpha \in \{0.5, 0.05, 0.005\}$.

Nessa análise, cada resultado da execução do TPBG para um valor de α é comparado com os resultados da execução de todos os outros valores de α. Os resultados são os valores de acurácia obtidos pela execução do TPBG nos conjuntos de documentos descritos na Tabela 7. Os resultados também variaram quanto ao número de documentos rotulados, onde foram conduzido experimentos considerando 1, 10, 20, 30, 40 e 50 documentos rotulados.

Tabela 11 – Ranques médios obtidos pela aplicação do procedimento de Friedman sobre os resultados do TPBG com diferentes número de documentos rotulados.

(a) 1 doc. rotulado por classe

TPBG α	Ranque
0.5	1.875
0.05	1.75
0.005	2.375

(b) 10 docs. rotulados por classe

TPBG α	Ranque
0.5	2.1875
0.05	1.625
0.005	2.1875

(c) 20 docs. rotulados por classe

TPBG α	Ranque
0.5	1.9375
0.05	1.75
0.005	2.3125

(d) 30 docs. rotulados por classe

TPBG α	Ranque
0.5	1.625
0.05	1.75
0.005	2.625

(e) 40 docs. rotulados por classe

TPBG α	Ranque
0.5	1.5625
0.05	1.8125
0.005	2.625

(f) 50 docs. rotulados por classe

TPBG α	Ranque
0.5	1.5625
0.05	1.8125
0.005	2.625

Fonte: Dados da pesquisa.

Na Tabela 11 estão os ranques médios. Note que o parâmetro com o melhor ranque é $\alpha = 0.05$. Testes *post hoc* foram feitos para identificar quais dos pares de grupos diferem. Com isso, foi possível aceitar ou rejeitar a hipótese nula (a de que não há diferença estatística

significativa entre os grupos). O procedimento de Nemenyi's rejeita aquelas hipóteses que tem valor de $p \leq 0.016667$. Na Tabela 12 estão os valores de p alcançados pelos testes *post hoc* com nível de confiança de 95%. Na Tabela 13 estão os testes considerando todos os números de documentos rotulados. Portanto, com esses testes, percebe-se que não foram obtidos diferença significativa entre os parâmetros $\alpha = 0.5$ e $\alpha = 0.05$.

Tabela 12 – Teste *post hoc* para teste de hipótese nula do conjunto de valores do parâmetros α do algoritmo TPBG.

(a) 1 doc. rotulado por classe

i	α vs. α	p
3	0.05 vs. 0.005	0.0771
2	0.5 vs. 0.005	0.157299
1	0.5 vs. 0.05	0.723674

(b) 10 docs. rotulados por classe

i	α vs. α	p
3	0.5 vs. 0.05	0.111612
2	0.05 vs. 0.005	0.111612
1	0.5 vs. 0.005	1

(c) 20 docs. rotulados por classe

i	α vs. α	p
3	0.05 vs. 0.005	0.111612
2	0.5 vs. 0.005	0.288844
1	0.5 vs. 0.05	0.595883

(d) 30 docs. rotulados por classe

i	α vs. α	p
3	0.5 vs. 0.005	0.004678
2	0.05 vs. 0.005	0.013328
1	0.5 vs. 0.05	0.723674

(e) 40 docs. rotulados por classe

i	α vs. α	p
3	0.5 vs. 0.005	0.002654
2	0.05 vs. 0.005	0.021556
1	0.5 vs. 0.05	0.4795

(f) 50 docs. rotulados por classe

i	α vs. α	p
3	0.5 vs. 0.005	0.002654
2	0.05 vs. 0.005	0.021556
1	0.5 vs. 0.05	0.4795

Fonte: Dados da pesquisa.

Tabela 13 – Teste *post hoc* para teste de hipótese nula do conjunto de valores do parâmetro α do algoritmo TPBG. Nessa tabela é considerando todos os números de documentos rotulados.

i	α vs. α	p
3	0.05 vs. 0.005	0.000001
2	0.5 vs. 0.005	0.000004
1	0.5 vs. 0.05	0.77283

Fonte: Dados da pesquisa.