UNIVERSIDADE FEDERAL DO AMAZONAS
INSTITUTO DE COMPUTAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA
DENYS DIONÍSIO BEZERRA SILVEIRA

MODELOS DE TÓPICOS BASEADOS EM AUTOCODIFICADORES VARIACIONAIS UTILIZANDO AS DISTRIBUIÇÕES GUMBEL-SOFTMAX E MISTURA DE NORMAIS-LOGÍSTICAS

Manaus
Dezembro de 2018

UNIVERSIDADE FEDERAL DO AMAZONAS
INSTITUTO DE COMPUTAÇÃO
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA
DENYS DIONÍSIO BEZERRA SILVEIRA

MODELOS DE TÓPICOS BASEADOS EM AUTOCODIFICADORES VARIACIONAIS UTILIZANDO AS DISTRIBUIÇÕES GUMBEL-SOFTMAX E MISTURA DE NORMAIS-LOGÍSTICAS

Dissertação apresentada ao Programa de Pós-Graduação em Informática do Instituto de Computação da Universidade Federal do Amazonas como requisito parcial para a obtenção do grau de Mestre em Informática.

Orientador: Marco Antônio Pinheiro de Cristo
Coorientador: André Luiz da Costa Carvalho

Manaus
Dezembro de 2018

Ficha Catalográfica

Ficha catalográfica elaborada automaticamente de acordo com os dados fornecidos pelo(a) autor(a).

Silveira, Denys Dionísio Bezerra
S587m Modelos de Tópicos baseados em Autocodificadores Variacionais utilizando as distribuições Gumbel-Softmax e mistura de Normais-Logísticas / Denys Dionísio Bezerra Silveira. 2018
115 f.: il. color; 31 cm.

Orientador: Marco Antônio Pinheiro de Cristo
Coorientador: André Luiz da Costa Carvalho
Dissertação (Mestrado em Informática) - Universidade Federal do Amazonas.

1. Modelos de Tópicos. 2. Autocodificadores Variacionais. 3. Inferência Bayesiana. 4. Aprendizagem Profunda. I. Cristo, Marco Antônio Pinheiro de II. Universidade Federal do Amazonas III. Título

FOLHA DE APROVAÇÃO

"Modelos de Tópicos baseados em Autocodificadortes Variacionais utilizando as distribuições Gumbel-Softmax e Mistura de Normais-Logísticas"

DENYS DIONÍSIO BEZERRA SILVEIRA

Dissertação de Mestrado defendida e aprovada pela banca examinadora constituída pelos Professores:

Prof. Marco Antonio Pinheiro de Cristo - PRESIDENTE

Prof. Juan Gabriel Colonna - MEMBRO INTERNO

Prof. André Luiz da Costa Carvalho - MEMBRO EXTERNO

Profa. Gisele Lobo Pappa - MEMBRO EXTERNO

Manaus, 11 de Dezembro de 2018

Dedico este trabalho aos meus pais, que são os dois maiores incentivadores das realizações dos meus sonhos e que sempre me apoiaram, mesmo nos momentos mais difíceis. A eles expresso a minha eterna gratidão.

Agradecimentos

Primeiramente gostaria de agradeço à Deus por tudo. Sem Ele nada seria possível.

Agradeço imensamente aos meus pais pelo apoio e carinho incondicional, principalmente nos momentos mais difíceis do mestrado. Os senhores são os meus heróis e palavras seriam poucas para expressar minha gratidão. Também agradeço ao meu irmão Davy pelo companheirismo e pelas palavras de incentivo.

Gostaria de expressar os meus sinceros agradecimentos ao meu orientador, Prof. Marco Cristo, que com sua grande experiência e conhecimento orientou-me na condução deste trabalho, sempre disposto a me ajudar em qualquer situação. Agradeço não apenas as palavras de motivação mas também as críticas construtivas que me ajudaram durante o mestrado e que com toda a certeza continuarão a me ajudar ao longo da minha jornada acadêmica e profissional.

Também quero agradecer ao meu co-orientador, Prof. André Carvalho, pelas valiosas e incontáveis horas dedicadas a este projeto de pesquisa, mesmo quando estava claramente sobrecarregado com as demais tarefas. Além de me auxiliar no âmbito da pesquisa sempre com muita paciência e compreensão, deu-me grandes conselhos pessoais. Para mim é uma honra tê-lo como co-orientador de Mestrado.

Agradeço àqueles que contribuíram diretamente ou indiretamente, seja com ou palavras de motivação ou conselhos, especialmente aos meus amigos Alef e Daniel Fernandes. Também agradeço aos meus colegas de laboratório: Anderson, Daniel Xavier, Erick, Ludimila, Josiane e Ivanilse. E aos meus colegas da Flaner: Alexandre, Ana Vitória, Brandell e Diego.

Por fim, agradeço à Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES) pelo auxílio financeiro para o desenvolvimento deste trabalho.

"Se vi mais longe foi por estar de pé sobre ombros de gigantes."
(Isaac Newton)

Resumo

Modelos probabilísticos de tópicos são modelos estatísticos capazes de identificar tópicos em uma coleção de texto. Eles são amplamente aplicados em tarefas relacionadas à área de Processamento de Linguagem Natural, uma vez que capturam com sucesso relações latentes por meio da análise de dados não rotulados. Entretanto, soluções analíticas para a inferência Bayesiana desses modelos são geralmente intratáveis, dificultando a proposta de modelos probabilísticos que sejam mais expressivos. Neste cenário, os Autocodificadores Variacionais (ACVs), métodos que empregam uma rede de inferência baseada em redes neurais responsável por estimar a distribuição *a posteriori*, tornaram-se uma alternativa promissora para inferir distribuições de tópicos em coleções de texto. Estes modelos, contudo, também introduzem novos desafios, tal como a necessidade de distribuições contínuas e reparametrizáveis que podem não se ajustar às distribuições reais dos tópicos. Além disso, redes de inferência tendem a apresentar um problema conhecido como colapso de componentes, onde apenas alguns tópicos contendo poucos termos correlacionados são efetivamente extraídos. Para tentar evitar estes problemas, propõem-se dois novos métodos de tópicos. O primeiro (GSDTM) é baseado em uma distribuição contínua pseudocategórica denominada *Gumbel-Softmax*, capaz de gerar amostras aproximadamente categóricas, enquanto o segundo (LMDTM) adota uma mistura de distribuições Normais-logísticas, que pode ser adequada em cenários onde a distribuição dos dados é complexa. Apresenta-se também um estudo sobre o impacto que diferentes escolhas de modelagem têm sobre os tópicos gerados, observando um compromisso entre coerência dos tópicos e a qualidade do modelo gerador. Por meio de experimentos usando duas coleções de dados de referência, três métricas distintas de avaliação quantitativa e uma inspeção qualitativa, mostra-se que o modelo GSDTM supera de forma significativa os modelos de tópicos considerados estado da arte em grande parte dos cenários de teste, em termos de coerência média de tópicos e perplexidade.

Palavras-chave: Modelos de Tópicos, Autocodificadores Variacionais, Inferência Bayesiana, Aprendizagem Profunda.

Abstract

Probabilistic topic models are statistical models which are able to identify topics on textual data. They are widely applied in many tasks related to Natural Language Processing due to their effective use of unlabeled data to capture latent relations. Analytical solutions for Bayesian inference of such models, however, are usually intractable, hindering the proposition of highly expressive text models. In this scenario, Variational Auto-Encoders (VAEs), where an artificial neural-based inference network is used to approximate the posterior distribution, became a promising alternative for inferring latent topic distributions of text documents. These models, however, also pose new challenges such as the requirement of continuous and reparameterizable distributions which may not fit so well the true latent topic distributions. Moreover, inference networks are prone to a well-known problem called component collapsing, where a little number of topics are effectively retrieved. To overcome these problems, we propose two new text topic models. The first (GSDTM) is based on the pseudo-categorical continuous distribution called *Gumbel-Softmax* which is able to generate categorical-like samples, while the second (LMDTM) adopts a mixture of Normal-Logistic distributions which can fits well in scenarios where the data distribution is complex. We also provide a study on the impact of different modeling choices on the generated topics, observing a trade-off between topic coherence and generative model quality. Through experiments using two reference datasets, three different quantitative metrics and one qualitative inspection, we show that GSDTM largely outperforms previous state-of-the-art baselines in most of scenarios, when considering average topic coherence and perplexity.

Palavras-chave: Topic Modeling, Variational Auto-Encoders, Bayesian Inference, Deep Learning.

Lista de Figuras

Lista de Tabelas

Nomenclatura

$P(X = x)$	Probabilidade do evento X ocorrer dado um valor x;
$p(x)$	Distribuição de probabilidades sobre uma variável x;
$\mathbf{x}$	Vetor;
$\mathbf{X}$	Matriz;
$\mathcal{G}$	Grafo;
$Pa_{\mathcal{G}}(x_i)$	Os pais de x_i em $\mathcal{G}$;
$[a, b]$	Intervalo no domínio dos números reais incluindo a e b;
λ	Parâmetros variacionais ou coeficiente angular de uma função linear;
$D_{KL}(P\|\|Q)$	Divergência Kullback-Leibler (KL) entre P e Q;
$x \sim P$	Variável aleatória x tem distribuição P;
$\mathbb{E}_{x\sim P}$ ou $\mathbb{E}[f(x)]$	Esperança de f(x) em função de P(x);
$f(\mathbf{x}; \theta)$	Uma função de $\mathbf{x}$ com parâmetro θ;
$\nabla_x f(x)$	Gradiente de $f(x)$ em relação a x;
μ	Parâmetro representando a média;
σ	Parâmetro representando o desvio-padrão;
$\mathcal{N}(\mathbf{x}; \mu, \mathbf{\Sigma})$	Distribuição Normal sobre $\mathbf{x}$ com média μ e covariância Σ;
$Var(f(x))$	Variância de $f(x)$ em relação a sua distribuição $P(x)$;
Δ^T	Representação de um simplex com T dimensões;
ν	Parâmetro da distribuição Multinomial;
τ	Parâmetro de temperatura da distribuição *Gumbel-Softmax*;
π	Vetor de probabilidades;
$log\ x$	Logaritmo natural de x;
p_{dado}	Distribuição real geradora de dados;
p_{modelo}	Distribuição estimada aprendida através da base de treinamento;
$\mathcal{X}$	Conjunto de todas as amostras provenientes de uma base de treinamento;
$arg\ max$	Função que retorna o índice do elemento com valor máximo de uma lista;

Sumário

1

Introdução

Modelos probabilísticos de tópicos têm sido aplicados com sucesso em várias tarefas relacionadas ao Processamento de Linguagens Naturais (PLN), tal como recuperação de documentos [Wei & Croft, 2006], agrupamento [Xie & Xing, 2013], classificação [Rubin et al., 2012], identificação de autoria [Seroussi et al., 2014] e análise de sentimentos baseada em aspectos [Lu et al., 2011]. Pode-se definir estes modelos como algoritmos que, dada uma coleção vasta e não estruturada de documentos, identificam qual a probabilidade de cada documento ou palavra estarem relacionados a um determinado tópico.

Grande parte do sucesso destes modelos probabilísticos geradores deve-se à eficiência deles em utilizar dados não rotulados para capturar dependências entre documentos e palavras e associá-los a um conjunto de tópicos. Neste contexto, modelos probabilísticos estruturados como os Autocodificadores Variacionais (ACVs) têm se destacado como o estado da arte em modelagem de tópicos, sendo capazes de aumentar consideravelmente a qualidade da aprendizagem destas relações de dependências, provendo tópicos com maior nível de coerência quando comparado com outros tipos de modelos disponíveis na literatura de modelagem de tópicos [Miao et al., 2016; Srivastava & Sutton, 2017].

Dentre as abordagens que adotam Autocodificadores Variacionais em modelagem de tópicos em texto, dois métodos denominados respectivamente *Neural Variational Document Model* (NVDM) e o *Product Latent Dirichlet Allocation* (ProdLDA) destacam-se como o estado da arte. No entanto, eles ainda possuem algumas limitações em cenários onde os dados são complexos e categóricos, como palavras, documentos

e tópicos, pois utilizam distribuições que não são próximas de distribuições categóricas. Desta forma, informações que poderiam contribuir com uma melhoria dos níveis de qualidade dos tópicos podem ser perdidas. Neste contexto, surge a necessidade de novos modelos capazes de se ajustar melhor à natureza dos dados provenientes de coleções de texto, de forma que aprendam tópicos coerentes e relevantes a um observador humano.

1.1 Motivação

Grande parte da informação está contida em dados não estruturados, *i.e.*, dados não rotulados ou não categorizados, sendo a própria WEB um exemplo notável deste fato. Quando se observa as páginas presentes na Internet, percebe-se claramente que grande parte das informações estão organizadas em formato de texto livre, o que torna difícil a categorização deste material.

A categorização dos documentos baseada em tópicos é uma tarefa importante para a organização e sumarização de elementos textuais. Por exemplo, é interessante para o usuário ter a possibilidade de navegar através dos documentos de acordo com o assunto contido no material. Um aluno de computação, por exemplo, pode estar lendo um documento sobre o lema do bombeamento e deseja ler posteriormente algum documento relacionado com o tema (*e.g.*, linguagens regulares). A comodidade é maior na procura se os materiais que o estudante busca estiverem categorizados como linguagens formais e autômatos, junto com outros textos que são provavelmente relevantes ao usuário.

Os benefícios de se agrupar os documentos através dos tópicos vão além da navegação mais fácil entre os textos. Os modelos de tópicos são utilizadas em diversas tarefas relacionadas à área de Processamento de Linguagem Natural. Por exemplo, estes modelos podem ser empregados em métodos de sumarização, onde informações relevantes são extraídas de uma vasta coleção de dados e resumidas para facilitar o processo de interpretação dos dados analisados [Arora & Ravindran, 2008]. Outra aplicação relevante consiste na extração de aspectos provenientes de avaliações de produtos. Um aspecto é uma característica que é passível de avaliação do consumidor, como por exemplo a tela de um celular ou a qualidade sonora de um fone de ouvido. Modelos de tópicos podem ser adaptados para extraírem as palavras que possuem maior probabilidade de serem consideradas um aspecto relevante [Titov & McDonald, 2008; Lu et al., 2011]. Além dessas aplicações, existem outras que utilizam modelos de tópicos em algum grau, como identificação de autoria [Seroussi et al., 2014], recuperação de documentos [Wei & Croft, 2006], classificação [Rubin et al., 2012] e agrupamento [Xie

& Xing, 2013]. Portanto, melhorias aplicadas em modelagem de tópicos beneficiam não só a própria área, quanto em outras correlacionadas, permitindo um avanço em diversos campos da literatura de Processamento de Linguagem Natural.

1.2 Justificativa

Grande parte das abordagens tradicionais baseadas em modelos gráficos probabilísticos direcionados (cf. Seção 3.1) requerem o uso de distribuições de probabilidade complexas e específicas para cada método. Mesmo para modelos que capturam relações simples de dependência entre as variáveis, tal como dependências sequenciais (*e.g.*, dependências entre palavras presentes em uma sentença) e espaciais (*e.g*, dependências entre *pixels* em uma imagem), pode ser difícil a derivação de um algoritmo de inferência plausível. Soluções analíticas para tais modelos geralmente resultam em integrais intratáveis, necessitando de métodos que lidem com distribuições *a posteriori* aproximadas (mais simples) ou que estimam a distribuição *a posteriori* real por meio de estratégias de amostragem. Como consequência, modelos altamente expressivos que operam com dados textuais são geralmente evitados.

Técnicas recentes baseadas em Autocodificadores Variacionais amenizam grande parte destes problemas. Quando aplicados em modelos de tópicos, os ACVs aproximam distribuições *a posteriori* reais por meio de redes neurais, utilizando como dados de entrada a frequência dos termos de cada documento que compõe um determinado *corpus*. Mais especificamente, estes modelos treinam uma rede de inferência capaz de aprender os parâmetros de uma distribuição de probabilidade (*e.g.*, média e variância no caso de uma distribuição Normal), distribuição esta que se aproxima da distribuição *a posteriori* real. Para atingir este objetivo, o algoritmo de retropropagação é usado para minimizar o erro de reconstrução destes documentos e, por meio das variáveis latentes aprendidas neste processo de reconstrução (por exemplo, variáveis relativas aos tópicos), extraem-se as informações desejadas. Devido à flexibilidade das redes neurais, modelos mais complexos podem ser projetados, capazes de capturar explicitamente dependências sequenciais e espaciais entre as variáveis de interesse e aprender distribuições complexas de forma plausível.

Estes modelos, contudo, são difíceis de se utilizar na prática devido a alguns problemas. Por exemplo, eles necessitam reparametrizar a função de amostragem como uma função diferenciável, já que o algoritmo de retropropagação não é capaz de realizar a diferenciação das funções estocásticas, que são necessárias para gerar as amostras. Também é possível que as redes de inferência fiquem presas em um ótimo local proble-

mático, de forma a gerar tópicos muitos semelhantes entre si e com má qualidade, um problema conhecido como colapso de componentes. Para lidar com o colapso, várias heurísticas têm sido propostas, tal como ajustes ótimos dos parâmetros, realização de recorte (*clipping*) de certas componentes da função de custo ou adoção de técnicas de regularização tal como *Batch Normalization* e *Dropout*. Contudo, o impacto do uso dessas técnicas na qualidade dos tópicos ainda não é bem-compreendido.

1.3 Hipótese de Pesquisa

Visto que tópicos representam grupos semânticos distintos que são de natureza categórica, a adoção de distribuições que gerem amostras pseudo-categóricas, tal como a *Gumbel-Softmax*, possibilita uma aprendizagem mais adequada em relação à natureza dos dados, podendo implicar em maiores índices de coerência de tópicos. Além disso, para permitir a reparametrização, geralmente se adotam distribuições tais como as distribuições Normais ou Normais-Logísticas, que podem não se adequar a distribuições *a posteriori* complexas. Desta forma, é possível que uma combinação de distribuições probabilísticas, como por exemplo a mistura de Normais-Logísticas, ajuste-se melhor às bases de dados mais complexas.

1.4 Objetivos

Tendo em vista a hipótese de pesquisa, pode-se definir como objetivo geral:

> Investigar o impacto do uso de diferentes configurações e distribuições probabilísticas na qualidade dos modelos de tópicos que são baseados em Autocodificadores Variacionais, elucidando os cenários que contribuem para o aumento da coerência dos tópicos e para a melhoria dos níveis de qualidade apresentados pelo modelo probabilístico.

Os objetivos específicos deste trabalho de pesquisa são:

- Aplicar as distribuições *Gumbel-Softmax* e a mistura de distribuições Normais-Logísticas para modelar as relações latentes de tópicos, com o objetivo de maximizar os níveis de coerência média dos tópicos e a qualidade do modelo gerador probabilístico;
- Compreender o impacto da aplicação das técnicas *Batch Normalization* e *Dropout* no processo de modelagem de tópicos;

- Avaliar o impacto destes modelos de tópicos na tarefa de recuperação de documentos;
- Avaliar qualitativamente os tópicos e o *embedding* gerados pelos métodos propostos e pelos demais métodos baseados em Autocodificadores Variacionais;

1.5 Contribuições da dissertação

A principal contribuição deste trabalho é a proposta do *Gumbel-Softmax Document Topic Model* (GSDTM) e do *Logistic-Normal Mixture Document Topic Model* (LMDTM), os primeiros modelos baseados em Autocodificadores Variacionais que usam respectivamente *Gumbel-Softmax* e Mistura de distribuições Normais-Logísticas diretamente em modelagem de tópicos. Também é estudado como diferentes escolhas na arquitetura das redes neurais afetam a qualidade dos tópicos gerados.

Baseado em uma comparação dos modelos propostos com duas coleções de dados de referência e três métricas de avaliação (coerência média de tópicos, perplexidade e precisão em diferentes frações de documentos coletados), mostra-se que: (i) o modelo GSDTM alcançou em geral os melhores resultados em termos de coerência média de tópicos e perplexidade, ultrapassando outras abordagens avaliadas em vários cenários, que incluem dois métodos considerados o estado da arte em modelos de tópicos baseados em Autocodificadores Variacionais; (ii) o uso de técnicas de estabilização como *Dropout* e *Batch Normalization* impactam diretamente nas métricas avaliadas, tendo um impacto positivo na coerência dos tópicos mas negativo em relação à perplexidade, e (iii) que o modelo LMDTM é competitivo em cenários com vastas coleções de dados sem a adoção das técnicas *Batch Normalization* e *Dropout*. Os resultados deste trabalho de pesquisa foram publicados no *International Joint Conference on Neural Networks* (IJCNN) 2018 [Silveira et al., 2018].

1.6 Organização do trabalho

O resto deste trabalho é organizado como se segue. O capítulo 2 apresenta os principais conceitos sobre modelos de tópicos baseados em Autocodificadores Variacionais, enquanto que o capítulo 3 aborda os trabalhos relacionados aos métodos propostos neste trabalho. No capítulo 4, apresentam-se os modelos de tópicos GSDTM e LMDTM. Reportam-se os experimentos e discussões dos resultados no capítulo 5. Finalmente, no capítulo 6, apresentam-se as conclusões obtidas e direções para trabalhos futuros.

2

Fundamentos Teóricos

Neste capítulo serão apresentados os conceitos necessários para o entendimento e desenvolvimento deste trabalho, bem como uma revisão da literatura em relação à área de modelagem de tópicos e subáreas correlatas. A Seção 2.1 aborda um resumo sobre as principais definições referentes aos modelos de tópicos. Depois, a Seção 2.2 compreende uma breve explicação do funcionamento dos modelos gráficos probabilísticos, base de vários modelos de tópicos. Nas Seções 2.3 e 2.4 discutem-se respectivamente conceitos básicos de inferência estatística e o processo de inferência variacional, mostrando como este método pode ser aplicado para tornar factível o processo de inferência estatística em modelos complexos de aprendizagem. A seguir, aborda-se uma breve conceituação sobre redes neurais artificiais na Seção 2.5. Ainda neste capítulo, é apresentado na Seção 2.6 o *embedding* de palavras, que consiste em representações de palavras em um espaço vetorial segundo a semântica, onde os vetores que representam palavras semanticamente próximas tendem a estar localizados próximos entre si. Mais adiante, descreve-se na Seção 2.8 os Autocodificadores Variacionais, métodos capazes de aprender estruturas latentes por meio da reconstrução de dados. Por fim, apresentam-se na Seção 2.9 as considerações finais do capítulo.

2.1 Modelos Probabilísticos de Tópicos

Modelos de tópicos podem ser definidos como algoritmos capazes de descobrir os principais tópicos presentes em uma coleção de documentos vasta e não estruturada [Blei, 2012]. Deste modo, o objetivo desta área consiste em prover um procedimento automa-

Tópico 8	**Tópico 15**	**Tópico 16**	**Tópico 19**	**Tópico 35**
guns	lib	mission	azerbaijan	game
gun	window	orbit	armenians	playing
msg	widget	lunar	armenia	season
criminals	xlib	rocket	armenian	fans
laws	xterm	solar	town	played
crime	string	satellite	soldiers	players
constitution	root	launch	troops	playoffs
weapons	application	flight	turks	team
rights	motif	space	apartment	hit
criminal	client	shuttle	villages	baseball

Tabela 2.1: Exemplo de cinco (de um total de cinquenta) tópicos extraídos da coleção 20newsgroups.

tizado para analisar e organizar coleções textuais em um conjunto de categorias com significados semânticos próprios, denominado tópicos ou temas [Mohr & Bogdanov, 2013].

Para exemplificar o conceito de tópicos, considere uma coleção de artigos não categorizados contendo notícias sobre assuntos variados. O conteúdo desse tipo de documento muda de acordo com tema explorado na notícia, de tal forma que existem notícias focadas em temas relevantes como política, tecnologia, economia e outros. Assim, categorizá-los por tema facilita a organização e a sugestão de outras notícias semelhantes. Entretanto, uma notícia não possui apenas um tópico relevante, tendo em vista que existem temas que são correlacionados. Por exemplo, uma notícia sobre economia pode conter parágrafos citando o panorama político, implicando na existência de dois tópicos em um mesmo documento. Logo, pode-se inferir que para identificar os temas contidos nas notícias, não basta analisar o documento como um todo, pois ele pode estar relacionado a vários tópicos. Geralmente, a solução para este problema é diminuir o grau de granularidade do dado observado, analisando cada palavra contida no documento. Desta forma, cada tópico é um agrupamento de termos contendo significados semânticos em comum. Exemplificando melhor, um tema denominado "economia" está atrelado a um conjunto de palavras como "dólar", "libra", "inflação", "*superávit*" e outras, enquanto que o tema relativo à política contém palavras como "partido", "eleição", "voto" e "constituição". Embora uma notícia possa conter todos estes termos, um modelo de tópicos será capaz de categorizá-los em temas e inferir a probabilidade de cada notícia estar relacionado à um tópico especifico de acordo com a proporção das palavras relacionadas a este tópico no texto.

Com o propósito de tornar modelos de tópicos possíveis de serem aplicados de

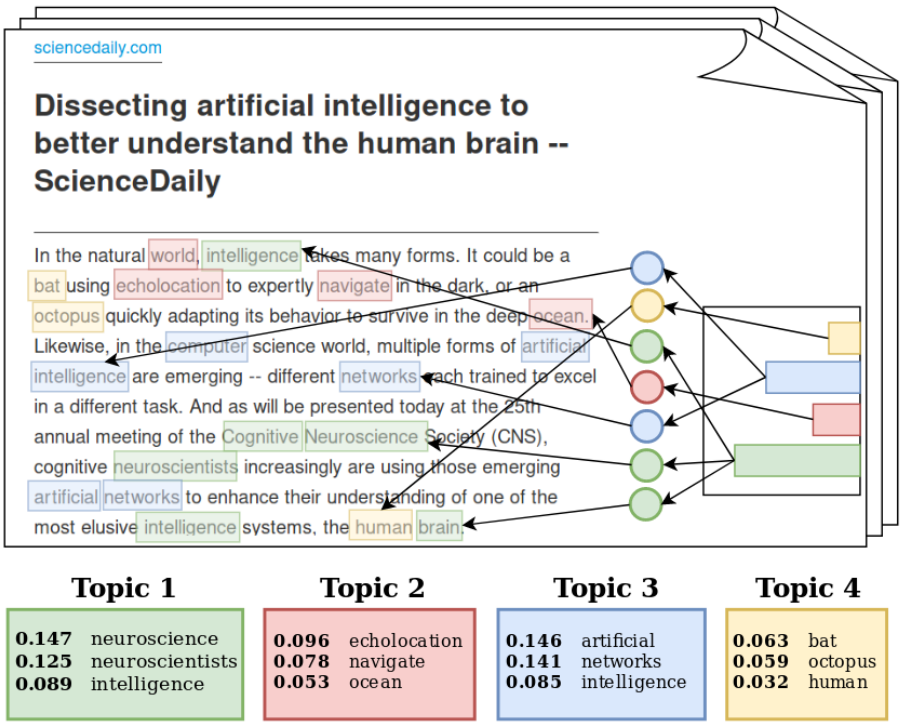


Figura 2.1: Diagrama mostrando as intuições do Latent Dirichlet Allocation (LDA), um modelo probabilístico de tópicos. Figura autoral baseada em uma representação do LDA presente no trabalho de Blei [2012].

forma eficiente no âmbito da Aprendizagem de Máquina, foram desenvolvidos os modelos probabilísticos de tópicos, que são métodos estatísticos que analisam as palavras extraídas de uma coleção de texto para descobrir tópicos de uma coleção, como estes estão relacionados entre si e como eles mudam durante um determinado período [Blei, 2012]. Neste modelo, um tópico é formalmente descrito como uma distribuição de probabilidades sobre um conjunto finito de termos denominado vocabulário. Desta forma, a distribuição de tópicos atribui uma probabilidade de cada termo contido no vocabulário ocorrer dado um tópico específico, isto é, a análise dos tópicos resulta em um ranqueamento de palavras dispostas em ordem de relevância, baseado na distribuição de probabilidades proveniente de cada tópico. Consequentemente, as palavras que estão no topo do ranqueamento representam os termos com maior probabilidade de ocorrer em um tópico específico.

A Tabela 2.1 apresenta cinco tópicos treinados com o modelo de tópicos GSDTM e escolhidos deliberadamente como forma de ilustrar o conceito de tópico. Pode-se observar que as dez palavras com maior índice de probabilidade em cada tópico possuem uma forte correlação entre si. Por exemplo, enquanto o Tópico 8 trata-se sobre armas, o Tópico 16 relaciona-se fortemente com foguetes e espaço sideral. Desta forma, o

ranqueamento da probabilidade torna-se útil para determinar o grau de correlação entre termos e tópicos.

A Figura 2.1 apresenta de forma simplificada o processo de extração de tópicos provenientes de uma coleção de texto. Embora tenha a finalidade de mostrar as intuições do método *Latent Dirichlet Allocation* (LDA) [Blei, 2012], a ilustração pode ser usada para demonstrar o funcionamento dos modelos probabilísticos de tópicos em geral. Observa-se à direita da figura uma representação da distribuição de tópicos em cada documento, responsável por determinar o grau de ocorrência do tópico em um determinado documento. À esquerda localiza-se uma representação da distribuição de palavras geradas a partir da distribuição de tópicos, enquanto que embaixo ilustra-se o ranqueamento de tópicos a partir da distribuição de palavras, contendo os termos com maior probabilidade de ocorrer na coleção dado um determinado tópico. Desta forma, todos os modelos probabilísticos de tópicos incorporam em algum grau este conceito de geração de palavras determinadas por uma distribuição de tópicos.

Normalmente, os modelos de tópicos existentes exploram padrões de ocorrência de palavras no texto e aprendem a relacionar documentos que compartilham padrões similares [Alghamdi & Alfalqi, 2015]. Mais especificamente, estes modelos são baseados na ideia de que documentos são resultantes da mistura entre tópicos, onde os tópicos são distribuições de probabilidades sobre as palavras. Assim, pode-se dizer que um modelo de tópicos é um modelo gerador de documentos, onde cada documento pode ser gerado estatisticamente [Steyvers & Griffiths, 2007]. Este processo ocorre da seguinte forma [Steyvers & Griffiths, 2007]:

- Escolhe-se uma distribuição de probabilidade relacionada aos tópicos;
- Para cada palavra presente em um documento qualquer, escolhe-se aleatoriamente um tópico t de acordo com a distribuição de probabilidade;
- Amostra-se uma nova palavra da distribuição de tópicos t.

Segundo Steyvers & Griffiths [2007], este processo de representação de documentos e palavras como tópicos probabilísticos possuem certas vantagens quando comparados com representações puramente vetoriais. Entende-se como uma representação puramente vetorial quando os termos do documento são representados como vetores em um espaço vetorial sem analisar o contexto nos quais estão empregados e nem a relação de dependência com outros termos. Em primeiro lugar, os agrupamentos de termos em tópicos são melhores interpretados do ponto de vista do entendimento humano, já que modelos de tópicos aprendem o processo de geração de documentos, diferentemente de

métodos de agrupamento em texto que analisam apenas as ocorrências das palavras. Por último, modelos de tópicos são flexíveis de tal forma que o processo de geração de documentos pode ser alterado a fim de expandir o uso para outras áreas.

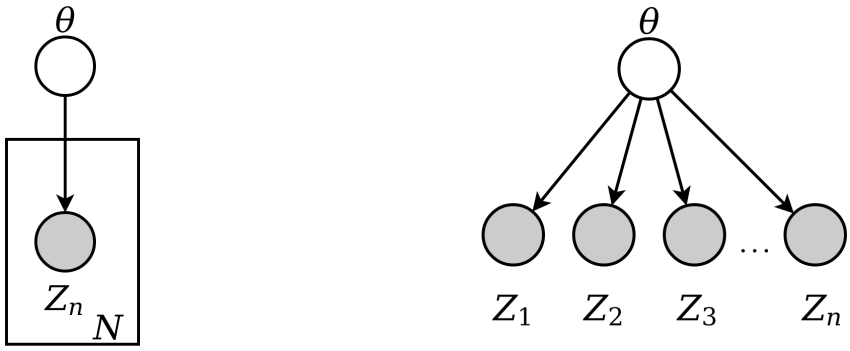
Os modelos probabilísticos de tópicos têm sido aplicados com sucesso em várias tarefas relacionadas ao Processamento de Linguagens Naturais (PLN) tais como recuperação de informação [Wei & Croft, 2006], agrupamento [Xie & Xing, 2013], classificação [Rubin et al., 2012], identificação de autoria [Seroussi et al., 2014], recomendação [Wang & Blei, 2011] e análise de sentimento baseada em aspectos [Lu et al., 2011]. Também têm sido aplicado em áreas fora do escopo de Recuperação de Informação, como análise sociológica [DiMaggio et al., 2013] e genética [Chen et al., 2010]. Qualquer tarefa que envolva categorizar documentos de forma não supervisionada baseado em conteúdo é passível de utilização de modelos de tópicos.

Embora a área tenha evoluído substancialmente ao longo dos anos, alguns desafios continuam presentes. Em particular, a noção de tópicos em coleções de texto é de natureza abstrata e subjetiva. Por exemplo, para um determinado indivíduo, um documento pode ser melhor classificado em um tópico, enquanto que para outro indivíduo o mesmo documento pode ser melhor classificado em um tópico diferente. Além disso, diferentes seções de um mesmo documento podem pertencer a tópicos distintos. Outra dificuldade encontrada decorre do fato de que linguagens naturais são inerentemente ambíguas [Resnik, 1999], permitindo que uma mesma palavra possa ser atribuída em tópicos distintos.

2.2 Modelos Gráficos Probabilísticos

Modelos de tópicos geralmente analisam entidades dependentes estatisticamente entre si, tais como palavras, documentos e tópicos. Entretanto, representar estas relações de dependência de forma elegante e formalizada constitui-se um problema significativo. Felizmente, pode-se representar formalmente estas relações estatísticas por meio de modelos gráficos probabilísticos (MGP), que são modelos formais estruturados como grafos que expressam as relações de dependência condicional entre variáveis aleatórias em um sistema estatístico [Jordan, 2004]. Pode-se definir uma variável aleatória (V.A) como uma variável quantitativa cujo valor depende de eventos aleatórios. Desta forma, entidades como palavras, documentos e tópicos podem ser representadas como variáveis aleatórias e os valores de probabilidades que não são conhecidos podem ser estimados utilizando o ferramental estatístico provido por estes modelos probabilísticos.

A principal motivação do uso desse formalismo em modelos estatísticos de tópicos



(a) Representação em pratos. (b) Representação estendida.

Figura 2.2: Diagrama de um modelo gráfico probabilístico (a) em uma estrutura condensada, denominada de estrutura em pratos e (b) em forma estendida, mostrando detalhes todas as variáveis aleatórias ocultas na estrutura em pratos.

provém da capacidade dos modelos gráficos probabilísticos de representar o processo de geração de documentos de forma objetiva e compreensível para sistemas computacionais. Além disso, sistemas definidos em termos de modelos gráficos probabilísticos se beneficiam de ferramentas estatísticas de inferência, tais como o máximo a *posteriori* (MAP) [Azzopardi et al., 2004], inferência variacional [Blei et al., 2003a], amostragem de Gibbs [Rosen-Zvi et al., 2004] e outras.

A representação formal dos modelos gráficos probabilísticos é descrita conforme mostrado na Figura 2.2. Considere $\mathbf{x} = (x_1, x_2, \cdots, x_n)$ um vetor contendo variáveis aleatórias. Um modelo gráfico probabilístico para $\mathbf{x}$ é uma representação em forma de grafo da distribuição conjunta $P(\mathbf{x})$. Por sua vez, uma distribuição conjunta consiste do produto das probabilidades individuais de cada variável aleatória. Esta representação consiste de dois componentes [Larrañaga, 2002]:

- Uma estrutura $\mathcal{G}$ para $\mathbf{x}$ composta por um grafo direcionado ou não direcionado que representa um conjunto dependências condicionais entre os elementos de $\mathbf{x}$;
- Um conjunto de distribuições de probabilidades individuais $P(x_i)$. Pode-se definir como distribuições individuais aquelas relativas a cada variável aleatória $x_i \in \mathbf{x}$.

Os modelos gráficos probabilísticos podem ser divididos em duas categorias: modelos baseados em grafos direcionados acíclicos e modelos baseados em grafos não direcionados [Goodfellow et al., 2016]. No primeiro caso, também conhecido como redes de crenças (*belief network*) ou redes Bayesianas (*Bayesian network*) [Pearl, 1986], as dependências condicionais das variáveis aleatórias possuem um sentido determinado, representado pelo grafo direcionado. No segundo, denominado também na literatura

como Campos Aleatórios Markovianos (*Markov Random Fields* ou MRF) ou rede Markovianas (*Markov Networks*) [Kindermann, 1980], a dependência condicional é bidirecional. Em outras palavras, duas variáveis aleatórias são mutualmente dependentes nesta abordagem.

Em modelos gráficos direcionados, a direção de dependência condicional é representada através de uma seta que indica que uma distribuição de probabilidade de uma variável aleatória é definida em termos da outra. Logo, uma seta que parte da variável a e incide na variável b significa que a distribuição de probabilidade de b depende condicionalmente do valor de a. Formalmente, um modelo gráfico direcionado possui como estrutura um grafo $\mathcal{G}$ direcionado e acíclico cujos vértices são variáveis aleatórias. Além disso, o modelo contém um conjunto de distribuições individuais de probabilidade $P(x_i|Pa_{\mathcal{G}}(x_i))$, onde $Pa_{\mathcal{G}}(x_i)$ fornece os nós parentes de x_i em $\mathcal{G}$. Logo, a distribuição de probabilidade em relação ao vetor de variáveis $\mathbf{x}$ é dado pela Equação 2.1 [Goodfellow et al., 2016].

$$p(\mathbf{x}) = \prod_i p(x_i|Pa_{\mathcal{G}}(x_i)) \tag{2.1}$$

Por outro lado, modelos gráficos não direcionados são modelos probabilísticos definidos em um grafo não direcionado $\mathcal{G}$. Em modelos de tópicos, destaca-se o uso de modelos probabilísticos gráficos direcionados. Deste modo, este trabalho de pesquisa foca majoritariamente neste tipo de modelo gráfico.

Os modelos gráficos probabilísticos têm como vantagem a construção de modelos estruturados com custo computacional reduzido referente à representação de distribuições de probabilidades, bem como do processo de inferência e aprendizagem. Desta forma, esses modelos permitem o projeto de métodos probabilísticos complexos de aprendizagem, reduzindo o uso de memória e o tempo de execução ao reduzir o número de interações necessárias entre as variáveis aleatórias para o processo de inferência [Goodfellow et al., 2016]. Além disso, existe uma separação explícita entre a representação do conhecimento em modelos gráficos e os métodos de inferência existentes. Em outras palavras, as técnicas clássicas de inferência estatística, tais como Monte Carlo via Cadeia de Markov (*Markov Chain Monte Carlo* ou MCMC), Máximo a Posteriori (MAP) e Inferência Variacional podem ser utilizadas de forma independente do domínio de aplicação, desde que estejam estruturadas em modelos gráficos probabilísticos.

Devido as vantagens proporcionadas, os modelos gráficos probabilísticos têm sido usados em diversas áreas da literatura científica. Em particular, a área de reconhecimento de padrões tem abrangido uma grande quantidade de estudos destinados no melhoramento de métodos de segmentação e rotulamento de objetos em imagens e ví-

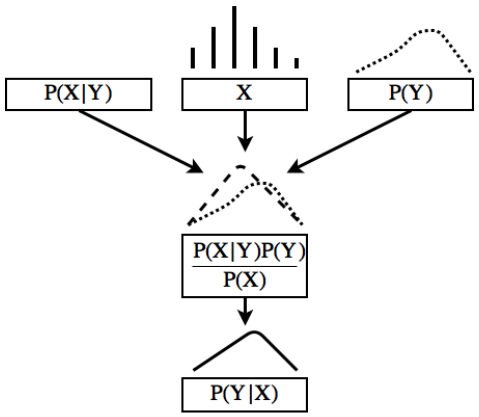


Figura 2.3: Diagrama representando o teorema de Bayes. O termo X representa a distribuição dos dados de entrada. Pode-se observar pelo exemplo que a distribuição de probabilidade *a posteriori* busca um compromisso entre a distribuição evidenciada nos dados e a distribuição fornecida *a priori*.

deos [He et al., 2004; Khatoonabadi & Bajic, 2013], e reconhecimento de gestos [Wang et al., 2006]. Embora a área de aprendizagem profunda tenha se tornado dominante em diversas aplicações, trabalhos recentes ainda utilizam métodos tradicionais baseados em modelos gráficos probabilísticos, sobretudo em aplicações que demandam eficiência e não possuem coleções vastas de dados rotuladas [Xu et al., 2018; Chen et al., 2018]. Outros métodos incorporam redes neurais profundas com métodos baseados em modelos gráficos probabilísticos, como o estudo de Arnab et al. [2018], que propõe um método de segmentação semântica unindo o modelo de Campos Aleatórios Condicionais (*Conditional Random Fields*) com redes neurais dinâmicas, redes estas que são capazes de mudar a sua própria topologia para aceitar entrada de dados que são mutáveis em um espaço de tempo [I. W. Lang et al., 2002].

2.3 Inferência Estatística

O principal propósito dos modelos gráficos probabilísticos é a realização de inferências consideradas adequadas e por meio da representação do conhecimento em modelos gráficos. Entende-se inferência como uma decisão baseada em alguma evidência observada. Em estatística, a inferência se resume no cálculo da probabilidade *a posteriori* $P(Y|X)$, onde se calcula a probabilidade de uma variável aleatória Y obter um determinado valor dado uma evidência contida no valor da variável X. Desta forma, $P(Y|X)$ representa a incerteza em termos de probabilidades condicionais, assumindo o intervalo de valores $[0, 1]$, onde 0 indica total descrença sobre a evidência e 1 assume crença absoluta.

Em teoria das probabilidades, categoriza-se o processo de inferência em abordagem frequentista e Bayesiana. Na primeira abordagem, considera-se a inferência como uma frequência relativa resultante da razão do número da população presente em uma amostra com o tamanho do espaço amostral. Assim, apenas evidências extraídas diretamente dos dados são consideradas. Em oposição à abordagem frequentista, a Bayesiana considera que a inferência pode incluir um conhecimento *a priori* sobre os dados. Em outras palavras, pode-se incluir uma probabilidade sobre a variável Y sem verificar a evidência proporcionada por X. A inferência Bayesiana é baseada no teorema de Bayes [Bayes & Price, 1763], indicado na Equação 2.2, onde a probabilidade *a posteriori* $P(Y|X)$ é obtida multiplicando $P(Y)$, que consiste na probabilidade *a priori* da hipótese relativa à variável Y, por $P(X|Y)$, que é a probabilidade da evidência em X ser verdadeira dado o valor de Y, também denominada de verossimilhança (*likelihood*), conforme explicitado na Figura 2.3.

$$P(Y|X) = \frac{P(X|Y)P(Y)}{P(X)}, \tag{2.2}$$

A probabilidade $P(X)$ pode ser considerada como uma constante de normalização. Desta forma, a inferência Bayesiana tem como desafio estimar os valores de $P(X|Y)$ e $P(X)$, visto que os mesmos não são conhecidos, ao contrário de $P(Y)$, cujo valor sabe-se previamente.

Assim como demais métodos probabilísticos, os modelos gráficos utilizam técnicas disponíveis na literatura de probabilidade para a realização de inferência Bayesiana. Segundo Box & Tiao [2011], o processo de inferência pode ser realizado de duas formas:

- Métodos exatos: calculam a probabilidade *a posteriori* por meio da resolução analítica via regra de Bayes;

- Métodos aproximativos: A probabilidade *a posteriori* é estimada, de forma que o valor seja o mais próximo possível daquele obtido via método exato.

Em geral, o uso da abordagem exata em modelos gráficos probabilísticos não é factível, pois a probabilidade de normalização $P(X)$ frequentemente contém integrais sem solução analítica ou computacionalmente intratáveis. Desta forma, grande parte dos modelos gráficos existentes utilizam a abordagem aproximativa. Na Subseção 2.4 será apresentado uma abordagem aproximativa popular em modelos gráficos probabilísticos, denominada de inferência variacional.

2.4 Inferência Variacional

A inferência variacional pode ser definida como o processo de inferência que utiliza uma distribuição mais simples para aproximar a distribuição real, de forma que o problema causado pelas integrais sem solução analítica ou computacionalmente intratáveis seja evitado. Formalmente, consiste em aproximar uma distribuição *a posteriori* complexa $p(\theta|x)$[1] por meio de uma distribuição mais simples $q(\theta;\lambda)$, que pertence a uma família restrita de distribuições indexadas por um parâmetro variacional λ [Jordan et al., 1999; Beal et al., 2003]. Entende-se como métodos variacionais aqueles que convertem um problema complexo em um problema simples, onde o problema simples é geralmente caracterizado como um relaxamento do problema original. Este relaxamento é obtido por meio da expansão do problema via adição de parâmetros adicionais denominados variacionais [Jordan et al., 1999].

Para aproximar duas distribuições distintas, é necessário uma métrica que indique o quão próximos são estas duas distribuições. Para tal, pode-se utilizar, por exemplo, a divergência Kullback-Leibler (KL), uma quantificação estatística proveniente da área de teoria da informação [MacKay, 2003]. A divergência KL entre duas distribuições p e q para distribuições discretas e continuas é mostrada nas equações 2.3 e 2.4, respectivamente [Wainwright et al., 2008]:

$$D_{KL}(q||p) = \sum_{\{Z\}} q(Z)\log \frac{q(Z)}{p(Z|x)}, \text{ para distribuições discretas} \tag{2.3}$$

$$D_{KL}(q||p) = \int q(Z)\log \frac{q(Z)}{p(Z|x)}dZ, \text{ para distribuições contínuas} \tag{2.4}$$

onde Z representa todos os nós do grafo $\mathcal{G}$ proveniente de um modelo gráfico probabilístico. O processo de aproximação, representado na Figura 2.4, é realizado escolhendo-se uma distribuição dentre uma família de distribuições aproximativas q, que minimiza a divergência KL $D_{KL}(q||p)$ em relação aos parâmetros variacionais λ [Jordan et al., 1999], conforme mostrado na Equação 2.5:

$$\lambda^* = \arg\min_{\lambda} D_{KL}(q(\theta|\lambda,x)||p(\theta|x)) \tag{2.5}$$

Entretanto, a divergência KL é computacionalmente difícil de ser calculada, pois a mesma necessita de um conhecimento prévio da distribuição que está sendo aproximada, representada por $p(x)$, que não é factível de ser computada analiticamente [Wain-

[1] Utilizou-se θ no lugar da representação y pois a literatura de inferência variacional adota a primeira representação com mais frequência.

wright et al., 2008]. A fim de lidar com este problema, o objetivo do processo de inferência variacional é alterado, de forma que o cálculo do valor de $p(x)$ não seja mais necessário. Isto é possível explorando a relação inversamente proporcional entre a divergência KL e o limite inferior da evidência, também denominada de energia livre negativa, conforme mostrado na Figura 2.4. Desta forma, a minimização da divergência KL pode ser remodelada como um processo de maximização do limite inferior da evidência (*Evidence of Lower Bound* ou ELBO). Em outras palavras, pode-se afirmar que a inferência variacional utiliza um artifício estatístico que permite expressar a divergência KL em termos do limite inferior, que é factível de ser calculado.

Para se obter o limite inferior da evidência a partir da divergência KL, é necessário aplicar a inequação de Jensen [1906] com o propósito de permitir a derivação do limite inferior a partir da log-verossimilhança (*log-likelihood*) $\log p(x)$. Este processo de derivação é explicitado melhor no teorema abaixo.

Teorema 1. *Considere $f(x)$ uma função estritamente convexa definida sobre o intervalo I. Se $x_1, x_2, \cdots, x_N \in I$ e $\lambda_1, \lambda_2, \cdots, \lambda_N \geq 0$, com $\sum_{i=1}^{N} \lambda_i f(x_i)$, então*

$$f\left(\sum_{i=1}^{N} \lambda_i x_i\right) \leq \sum_{i=1}^{N} \lambda_i f(x_i)$$

De forma alternativa, se $f(x)$ é uma função convexa e $X \in \{x_i : 1, \cdots, N\}$ é uma variável aleatória, então $f(\mathbb{E}[X]) \leq \mathbb{E}[f(X)]$, onde $\mathbb{E}[X]$ é a esperança de X.

Como a função logarítmica é convexa, então se pode afirmar que $\log\left(f(\mathbb{E}\{X\})\right) \leq \mathbb{E}[\log\left(f(X)\right)]$. Em termos menos formais, a inequação de Jensen permite determinar o logaritmo da esperança de X em função da esperança do logaritmo de X, por meio de uma relação de inequação. Consequentemente, é possível obter o limite inferior da

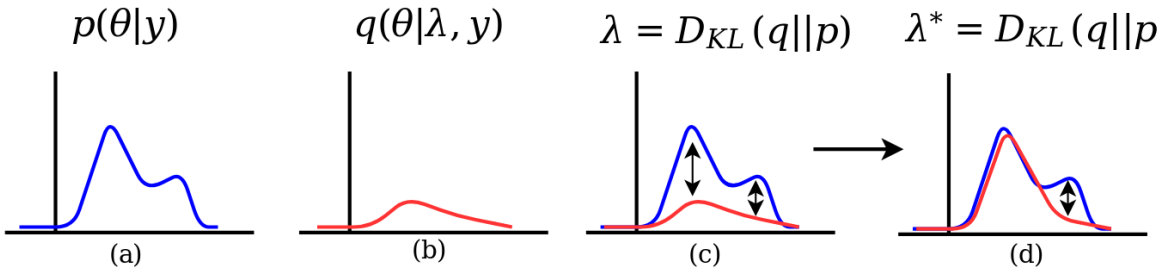


Figura 2.4: Representação ilustrativa do processo de inferência variacional, onde a distribuição de probabilidade original p (a) é aproximada pela distribuição variacional q (b). Por meio da minimização da divergência KL (c), é obtido uma distribuição variacional similar à original (d).

evidência $\mathcal{L}$ reconstruindo a marginalização relativa à variável aleatória Z e aplicando a inequação de Jensen, conforme apresentado na Equação 2.6.

$$\begin{aligned}
\log p(x) &= \log \int_Z p(x, Z) \\
&= \log \int_Z p(x, Z) \frac{q(Z)}{q(Z)} \\
&= \log \left(\mathbb{E}_q \left[\frac{p(x, Z)}{q(Z)} \right] \right) \\
&\geq \mathbb{E}_q \left[\log p(x, Z) \right] - \mathbb{E}_q \left[\log q(Z) \right] \\
&\geq \mathcal{L}
\end{aligned} \tag{2.6}$$

A divergência KL é correlacionada com o limite inferior da evidência. Uma vez que a divergência KL pode ser reescrita em função da esperança, tal que

$$D_{KL}(q||p) = \mathbb{E}_q \left[\log \left[\frac{q(Z)}{p(Z|x)} \right] \right], \tag{2.7}$$

e sabendo que $\mathbb{E}_q[\log p(x)] = \log p(x)$, pois $p(x)$ é independente da distribuição variacional q, pode-se realizar uma manipulação algébrica usando a regra de Bayes em $p(Z|x)$, de tal forma que se obtém como resultado a seguinte Equação (2.8):

$$\begin{aligned}
\log p(x) &= D_{KL}(q||p) + \mathbb{E}_q[\log [q(Z)]] - \mathbb{E}_q[\log p(x, Z)] \\
&= D_{KL}(q||p) - \mathcal{L}
\end{aligned} \tag{2.8}$$

Visto que o valor de $\log p(x)$ é imutável, então incrementar $D_{KL}(q||p)$ implica em decrementar $\mathcal{L}$ e vice-versa. Logo, para minimizar a divergência KL, basta maximizar o limite inferior.

Na prática, a família da distribuição variacional necessita ser o mais simples possível para que o processo de inferência seja eficiente em termos de tempo de execução. Para isso, utiliza-se a inferência variacional de campo médio, que consiste em uma simplificação da distribuição de probabilidade q, assumindo que a mesma pode ser fatorizada, conforme mostrado na Equação 2.9.

$$q(z_1, z_2, \cdots, z_m) = \prod_j^m q(z_j) \tag{2.9}$$

Na inferência variacional de campo médio, assume-se que cada uma das variáveis aleatórias é condicionalmente independente em relação às demais. Desta forma, a distribuição variacional q é simplificada de modo que o tempo necessário para a realização da inferência seja reduzido [Blei, 2011]. Definido o limite inferior da evidência

por meio da inferência variacional de campo médio, pode-se usar diferentes métodos de otimização de funções contínuas, como o gradiente descendente estocástico [Ranganath et al., 2014] e métodos iterativos de ponto fixo [Blei et al., 2003a], a fim de otimizá-la e consequentemente realizar a aprendizagem do modelo gráfico probabilístico.

No geral, a inferência variacional é vastamente aplicada em modelos de tópicos, sobretudo aqueles tradicionais baseados em métodos que não utilizam redes neurais e em Autocodificadores Variacionais. Na próxima seção, serão apresentados conceitos básicos de redes neurais, que são necessários para a compreensão dos Autocodificadores Variacionais.

2.5 Modelo de Redes Neurais Artificiais

Dentre diversos modelos que surgiram com o intuito de simular a aprendizagem humana via técnicas computacionais, alguns se inspiravam em como o processo de aprendizagem acontece ou pode acontecer no cérebro [Goodfellow et al., 2016]. Como resultado, foram criadas as redes neurais artificiais, modelos inspirados na organização dos neurônios, capazes de receber alguns valores, processá-los, ajustá-los automaticamente utilizando pesos e fornecer a saída do processamento. Mais especificamente, estes modelos são projetados para receber n entradas de dados, nomeados de sinais de entrada e associá-los à saída y. O processo de aprendizagem é realizado ajustando um conjunto de pesos $w_1, w_2, \cdots, w_n$ denominados pesos de sinapse e a combinação linear destes pesos com a entrada de dados, expressa pela função $f(\mathbf{x}, \mathbf{w}) = x_1 w_1 + \cdots + x_n w_n$ calcula a saída y, conforme mostrado na Figura 2.5.

Entretanto, modelos expressos utilizando combinação linear possuem pouca capacidade de representação. Por exemplo, a Figura 2.6a mostra a aplicação da função XOR [Minksy & Papert, 1969], que evidencia o problema de modelos lineares. Pode-se observar que não é possível separar corretamente as instâncias pretas das brancas usando uma superfície de decisão linear. Contudo, é possível separá-las utilizando uma superfície de decisão curva, que é uma característica de modelos não lineares, conforme mostrado na Figura 2.6b. Desta forma, as redes neurais podem aplicar uma função denominada função de ativação, que tem como objetivo transformar a função $f(\mathbf{x}, \mathbf{w})$ em uma função não linear e consequentemente aumentar a capacidade do modelo.

A capacidade das redes neurais pode ser aumentada conectando os neurônios (ou *perceptrons*) entre si, formando as redes denominadas de redes neurais de múltiplas camadas (*multilayer perceptrons* ou MLP). O objetivo de uma MLP é aproximar uma função ótima f^*. Por exemplo, considere, para um classificador, que $y = f^*(\mathbf{x})$ seja

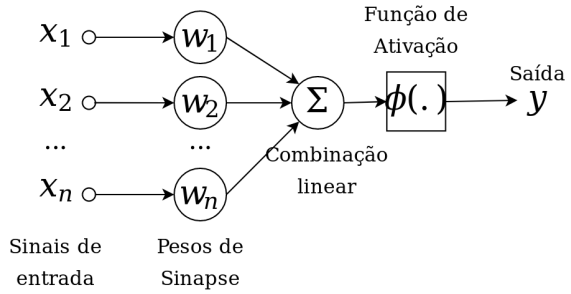
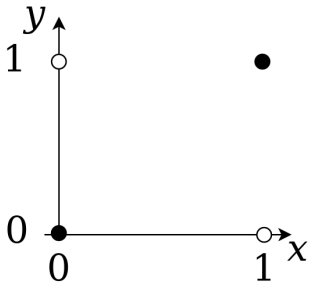


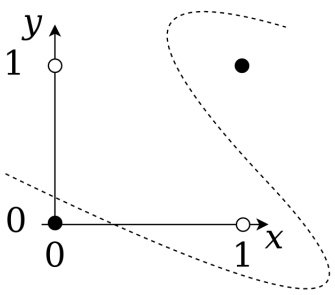
Figura 2.5: Representação de um neurônio (ou *perceptron*) artificial. Figura autoral baseada no trabalho de Boukadida et al. [2011].

um mapeamento entre a entrada $\mathbf{x}$ e uma categoria y. A MLP define um mapeamento $\hat{y} = f(\mathbf{x}; \theta)$ que aprende o valor dos parâmetros θ que resulta na melhor aproximação possível da função f^*. Estes modelos são denominados propagação (*feedforward*) devido ao fato da informação proveniente dos dados de entrada fluírem pela rede até a camada de saída, passando pelas camadas intermediárias. Nesta arquitetura, não existem conexões de *feedback* que utilizam a própria saída da rede como dados de entrada. Quando as redes do tipo propagação contém conexões de *feedback*, elas são denominadas de Redes Neurais Recorrentes (*Recurrent Neural Networks* ou RNN) [Goodfellow et al., 2016].

Estes modelos são denominados redes porque são formados pela composição de várias funções. Estas funções são organizadas em um grafo acíclico direcionado que descreve como as funções estão relacionadas entre si, conforme mostrado na Figura 2.7. Por exemplo, pode-se organizar três funções $f^{(1)}, f^{(2)}$ e $f^{(3)}$ de forma encadeada de tal maneira que a o valor retornado por uma função seja o valor de entrada da próxima. Considerando a ordem de encadeamento como $f^{(3)}(f^{(2)}(f^{(1)}(x)))$, denomina-se $f^{(1)}$ como a primeira camada, $f^{(2)}$ como a segunda camada e assim por diante [Good-



(a) Função XOR.



(b) Função não linear para a função XOR.

Figura 2.6: Ilustração do (a) problema da função XOR e (b) do poder de expressão das funções não lineares.

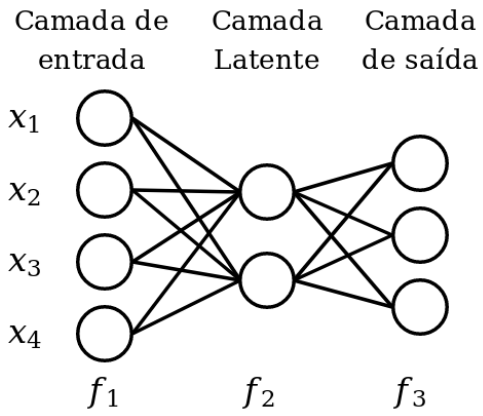


Figura 2.7: Representação de uma MLP.

fellow et al., 2016]. A camada inicial é chamada de camada de entrada, enquanto que a final é denominada de camada de saída. As demais camadas são chamadas de latentes ou ocultas, já que geralmente o valor produzido por estas camadas não são de interesse do usuário, diferentemente dos valores da camada de saída.

A fim de tornar a aprendizagem possível em MLPs, é necessário que o mapeamento aprendido pela rede neural seja o mais próximo possível do mapeamento real, ou seja, que a saída proveniente desta rede seja a mais correta possível. Para tal, pode-se definir uma função denominada de função de custo para medir o erro entre a saída $\hat{y}$ inferida pela rede e a saída esperada y. Existem diversas funções de custo que podem ser usadas em MLPs, sendo a mais comum a função de erro médio quadrático ($f(y, \hat{y}) = (y - \hat{y})^2$), que calcula o quadrado da diferença entre $\hat{y}$ e y. Neste caso, o objetivo da rede é ajustar os parâmetros θ da rede de tal forma que o valor da função f seja minimizado.

A minimização ou maximização da função de custo pode ser efetuada utilizando métodos de otimização para variáveis contínuas. Existem diversos métodos na literatura, sendo os mais utilizados aqueles baseados no cálculo da função gradiente da função de custo. Em termos simplificados, o gradiente é a generalização da derivada para múltiplas dimensões. Conforme mostrado no exemplo presente na Figura 2.8, o gradiente $\nabla_x f$ (gradiente da função f em relação à variável x) é utilizado para indicar a direção dos pontos candidatos em relação ao máximo ou mínimo local da função $f(x)$. No caso do exemplo, o objetivo é encontrar o menor valor possível de $y = f(x)$, denotado por y^*. Partindo-se de um valor aleatório y_0, é possível encontrar um valor próximo de y^* utilizando o valor do gradiente como indicador da posição do valor mínimo. Desta forma, pode-se alterar iterativamente o valor de x de modo que o novo valor x_{n+1} seja equivalente a $x_n - \alpha \nabla_x f(x_n)$, onde α é a taxa de aprendizagem, cujo

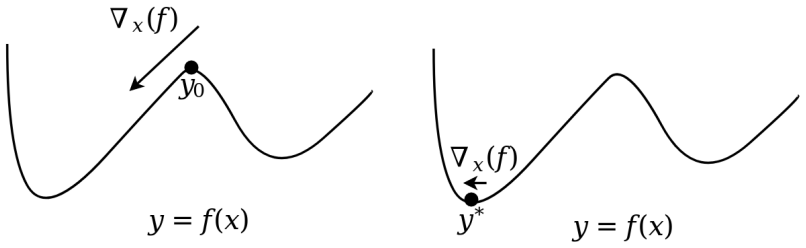


Figura 2.8: Figura mostrando o processo de minimização da função de custo.

propósito é controlar o deslocamento, reduzindo a influência do gradiente. Neste caso, o método é denominado gradiente descendente caso o problema seja de minimização, e gradiente ascendente caso contrário.

O ajuste dos pesos de toda a MLP pode ser efetuado de forma eficiente utilizando o algoritmo de retropropagação (*backpropagation*) [Rumelhart et al., 1986]. Este algoritmo ajusta todos os pesos da rede baseado na função de custo, porém de forma inteligente: as derivadas parciais em relação à última camada são computadas diretamente da função de custo, enquanto que as derivadas parciais das demais camadas são calculadas em função das derivadas parciais da camada posterior. Desta forma, evita-se recalcular os gradientes de forma desnecessária, tornando o processo mais eficiente. Para a realização do ajuste dos pesos, pode-se utilizar uma abordagem como gradiente descendente, descrita no parágrafo anterior. O nome deste método deriva do fato que o cálculo do erro computado pela função de custo é propagado da última camada até a primeira (retropropagação dos erros).

O treinamento de redes neurais com várias camadas latentes é complexo, pois a distribuição dos dados de entrada de cada camada muda durante o processo de aprendizagem, já que os parâmetros da camada anterior também são alterados. Isto retarda o tempo de aprendizagem da rede, especialmente quando taxas de aprendizagem menores são necessárias para lidar com a flutuação da distribuição dos dados. Além disso, alguns neurônios da MLP podem se especializar em algum tipo específico de dado, comprometendo a generalização da inferência, isto é, a capacidade do modelo ter alta acurácia em dados não utilizados na fase de treinamento da rede. Logo, foram propostas várias técnicas auxiliares de aprendizagem de redes neurais, das quais se destacam o *Batch Normalization* (BN) [Ioffe & Szegedy, 2015] e *Dropout* [Srivastava et al., 2014].

O *Batch Normalization* é um método que normaliza cada *batch* de dados aplicando uma transformação $\hat{\mathbf{x}} = Norm(\mathbf{x}, \mathcal{X})$, onde $\mathbf{x}$ é a entrada de dados de uma camada qualquer da rede e tratado como um vetor e $\mathcal{X}$ é o conjunto de todas essas entradas, ou seja, toda a base de treinamento. Uma vez que um dos objetivos desta transformação é a normalização dos dados de entrada, pode-se aplicar a Equação 2.10

para cada dimensão do vetor $\mathbf{x}$, onde $\mathbb{E}[\mathbf{x}]$ e $Var(\mathbf{x})$ são respectivamente a esperança e a variância de $\mathbf{x}$.

$$\hat{\mathbf{x}} = \frac{\mathbf{x} - \mathbb{E}[\mathbf{x}]}{\sqrt{Var(\mathbf{x}) + \epsilon}} \tag{2.10}$$

Tendo os dados sido normalizados, o último passo é alterar a escala e a dispersão dos dados aplicando a Equação 2.11, onde γ e β são parâmetros dados como entrada.

$$\hat{\mathbf{y}} = \gamma\hat{\mathbf{x}} + \beta \tag{2.11}$$

Como resultado, o *Batch Normalization* permite o uso de taxas de aprendizagem mais altas, acelerando o tempo de treinamento da rede. Esta técnica também atua como um regularizador, diminuindo as flutuações do treinamento resultante da escolha dos parâmetros da rede. Desta forma, o uso de *Batch Normalization* tende a melhorar a tarefa de classificação de dados usando menos amostras de treinamento quando comparado com redes neurais que não utilizam esta técnica [Ioffe & Szegedy, 2015].

A outra técnica auxiliar, denominada *Dropout*, consiste em simplesmente remover temporariamente a presença de uma porcentagem dos neurônios de uma MLP durante a fase de treinamento. Mais especificamente, cada neurônio está atrelado a uma probabilidade r amostrada de uma distribuição binomial com parâmetro p (probabilidade de se manter o neurônio na rede). Caso o valor de p seja 0, o neurônio tem suas arestas removidas temporariamente. Caso contrário, as arestas são mantidas. A motivação deste processo é que neurônios tendem a se especializar em um tipo de entrada de dados ao longo do treinamento. Desta forma, o "desligamento" temporário de alguns neurônios da rede evita a super-especialização deles. Como resultado da aplicação desta técnica, aumenta-se a capacidade de generalização da rede, regulando assim o funcionamento da MLP [Srivastava et al., 2014].

2.6 *Embeddings* de palavras

As abordagens baseadas em redes neurais se popularizaram de forma intensiva nos últimos anos, muito devido ao sucesso do campo de aprendizagem profunda [Schmidhuber, 2015]. Embora não haja uma definição precisa do conceito de aprendizagem profunda, entende-se como profunda uma rede com várias camadas latentes. Entretanto, redes que não são de aprendizagem profunda, denominadas de redes de aprendizagem rasa, também são empregadas em diversas áreas de estudo. Uma desses áreas consiste em representar as relações semânticas das palavras como um vetores em um espaço vetorial

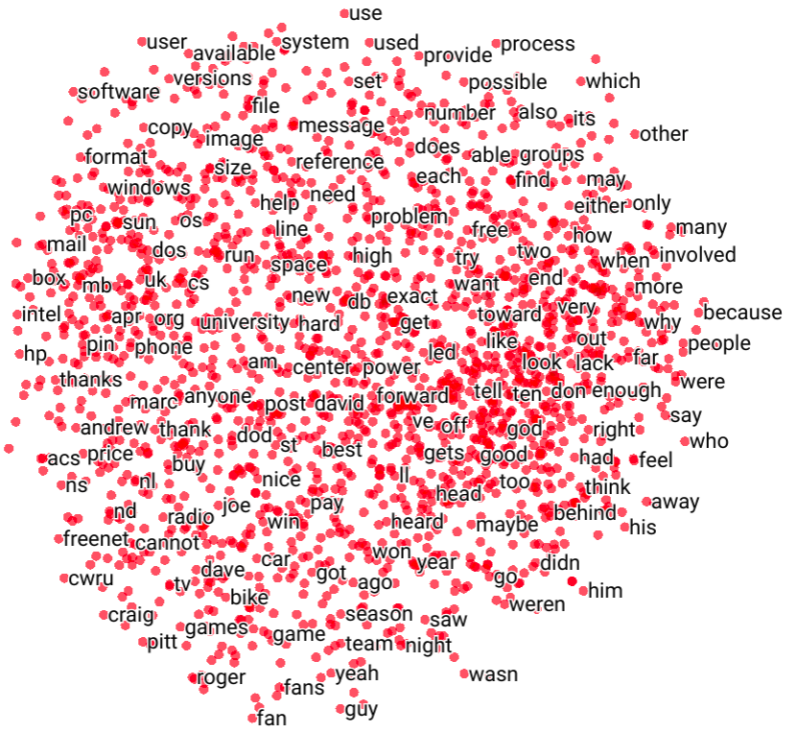


Figura 2.9: Representação de um exemplo de *embedding* de palavras em duas dimensões, treinados com o modelo GSDTM na base de dados 20newsgroups. É possível notar que palavras semanticamente similares tendem a estar próximas no espaço vetorial do *embedding*.

de baixa dimensão, isto é, vetores cuja dimensão é menor que a do vocabulário de palavras. Este mapeamento entre palavras e suas representações vetoriais é denominado *embedding* de palavras (*word embeddings*).

Um *embedding* é um termo usado para designar qualquer mapeamento entre um espaço de alta dimensionalidade para um espaço de baixa dimensão, de forma que a representação de alguma característica relativa aos dados originais sejam preservadas no outro espaço vetorial. No caso do *embedding* de palavras, almeja-se representar as relações de coocorrência em um espaço de dimensão menor com a menor perda de informação possível. Na Figura 2.9, pode-se observar um exemplo de *embedding* de palavras. É possível notar que palavras semanticamente similares tendem a estar próximas no espaço vetorial do *embedding*. Por exemplo, os termos "os", "dos" e "windows" estão próximos no espaço vetorial representado na Figura 2.9 e estão associados à área de informática, da mesma forma que as palavras "game", "season" e "team" podem ser associadas ao conceito de esporte.

Embora existam vários modelos de *embedding* de palavras, dois modelos denomi-

nados Word2Vec [Mikolov et al., 2013] e o Glove [Pennington et al., 2014] têm atraído a atenção dos pesquisadores nos últimos anos por aliar boa representatividade de linguagem com eficiência. Tal possibilidade tem sido aproveitada para melhorar diversas tarefas de processamento de linguagens naturais. Entretanto, outros modelos são capazes de gerar *embeddings* de palavras. Por exemplo, os Autocodificadores Variacionais podem gerar representações vetoriais com grande qualidade, com a vantagem de incorporar outros tipos de relações além da coocorrência das palavras. Desta forma, os modelos de tópicos baseados em Autocodificadores Variacionais podem construir *embeddings* de palavras que refletem a distribuição de tópicos aprendida pela rede de inferência, desde que a rede responsável por reconstruir os dados seja uma regressão multinomial logística.

Na próxima seção, será apresentado o Autocodificador Variacional, modelo que obteve resultados significativos na literatura de modelos de tópicos nos últimos anos.

2.7 Modelos geradores

Para entender o objetivo dos Autocodificadores Variacionais, faz-se necessário revisar o conceito de modelos geradores, já que um Autocodificador Variacional é categorizado como um tipo de modelo gerador. Na literatura, o termo "modelo gerador" é usado em diferentes sentidos. No contexto deste trabalho, utiliza-se esse termo para definir qualquer modelo que, dado um conjunto de dados distribuídos sob uma distribuição p_{dado}, aprende de alguma forma a representar uma estimativa desta distribuição por meio de uma distribuição p_{modelo}, cujos parâmetros são obtidos via aprendizagem [Doersch, 2016].

Com o objetivo de apresentar de forma mais clara a função dos modelos geradores, considere a tarefa de sintetização de imagens, mostrada na Figura 2.10. Esta tarefa tem como objetivo aprender padrões provenientes da organização dos *pixels* provenientes de um conjunto de amostras de imagens e gerar imagens semelhantes às originais utilizando os padrões aprendidos. A fim de realizar esta tarefa, pode-se utilizar modelos geradores. Para tal, considera-se que os *pixels* estão organizados segundo a distribuição de probabilidade p_{dado}. Para fins de formalização, o conjunto de dados é denotado por $\mathbf{X} = \{\mathbf{X}_1, \mathbf{X}_2, \cdots, \mathbf{X}_N\}$, onde $\mathbf{X}_i$ é uma imagem identificada por i e composta por uma matriz de *pixels*. Desta forma, a média de valores de uma distribuição de probabilidade $p_{dado}(\mathbf{X}_i)$ de uma imagem específica é alta quando os *pixels* de uma imagem estão fortemente correlacionados entre si, formando uma imagem reconhecível para um julgador humano, enquanto que um valor médio baixo tende a indicar que os *pixels* estão

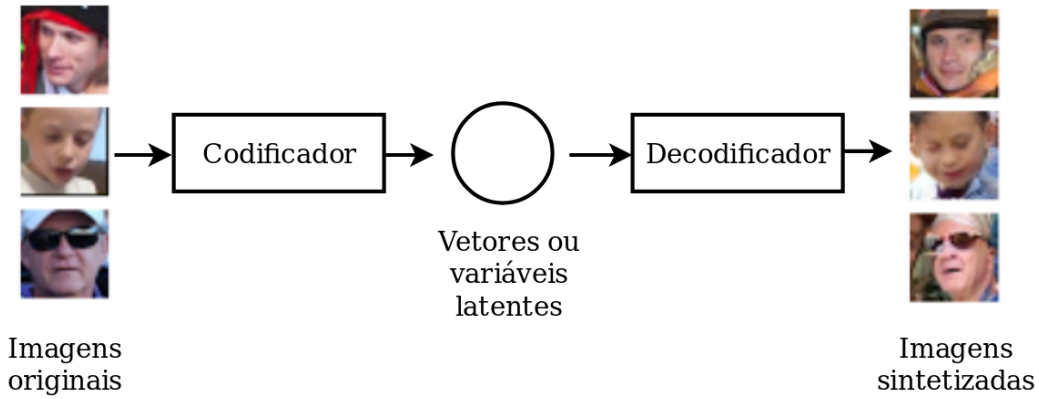


Figura 2.10: Representação de um modelo gerador de forma simplificada. As imagens reais (à esquerda) e as geradas via aprendizagem (à direita) foram extraídas do trabalho de van den Oord et al. [2016].

dispostos de forma aleatória. Logo, com um número suficiente de amostras, pode-se treinar um modelo de aprendizagem para aproximar a distribuição $p_{dado}(\mathbf{X})$ por meio de outra distribuição mais simples $p_{modelo}(\mathbf{X})$, de tal forma que é possível sintetizar novas amostras de dados $\hat{\mathbf{X}}$ por meio do processo de amostragem $\hat{\mathbf{X}} \sim p_{modelo}(\mathbf{X})$.

Embora modelos geradores e os Autocodificadores sejam topologicamente parecidos, os objetivos dos dois modelos são distintos. Enquanto que modelos geradores visam gerar novas amostras semelhantes às originais por meio de uma distribuição de probabilidades, um Autocodificador visa apenas extrair uma representação em baixa dimensão proveniente dos dados de entrada com o mínimo possível de perda, não sendo possível a sintetização de novos dados usando este modelo [Hinton & Zemel, 1994]. Este problema de terminologia é evidente na denominação dos Autocodificadores Variacionais. Embora tenham o termo "Autocodificador" no nome, estes modelos são categorizados como modelos geradores. Assim, tal termo apenas faz alusão à semelhança da arquitetura das redes neurais dos Autocodificadores Variacionais com a rede de Autocodificadores.

Embora existam diversos modelos geradores na literatura, Doersch [2016] enumera três problemas compartilhados pela maioria desses modelos:

1. Requerem um conhecimento prévio sobre a estrutura dos dados de entrada;

2. Necessitam de aproximações severas da distribuição de probabilidade $p(\mathbf{X})$, levando a uma aprendizagem subótima;

3. Frequentemente recorrem aos métodos de inferência como Monte Carlo via Cadeia de Markov (MCMC), que podem ser caros computacionalmente.

Logo, os Autocodificadores Variacionais se tornaram populares por resolverem estas três dificuldades encontradas nos outros modelos. Isto se deve ao fato que os Autocodificadores Variacionais necessitam de pouco conhecimento prévio sobre os dados e utilizam inferência variacional como método de inferência, que realiza aproximações próximas do ótimo global sem ser caro computacionalmente. Os Autocodificadores Variacionais são definidos com mais profundidade na Seção 2.8.2.

2.8 Autocodificadores Variacionais

Os Autocodificadores Variacionais (ACVs) consistem em modelos de variáveis latentes que reconstroem uma entrada de dados por meio de amostragem de uma distribuição probabilística aproximada via processo de inferência variacional [Kingma & Welling, 2013]. Diferentemente dos modelos gráficos probabilísticos clássicos, são usadas redes neurais para realizar a inferência dos parâmetros das distribuições de probabilidade. Inicialmente, este modelo foi desenvolvido especificamente para a tarefa de sintetização de imagens. Entretanto, estudos na área de modelos de tópicos têm expandido o uso dos Autocodificadores Variacionais para extrair tópicos em coleção de texto. Devido ao fato deste trabalho de pesquisa ser baseado em Autocodificadores Variacionais, será apresentado um resumo descrevendo os principais conceitos relativos a estes modelos.

2.8.1 Modelo gráfico probabilístico de ACVs

Mais formalmente, um Autocodificador Variacional é um modelo gráfico probabilístico contendo duas variáveis aleatórias $\mathbf{z}$ e $\mathbf{x}$, que indicam respectivamente as relações latentes presentes na coleção e o dado observável. Desta forma, tem-se uma distribuição de probabilidade p com conjunto de parâmetros θ sob a forma $p_\theta(\mathbf{z})p_\theta(\mathbf{x}|\mathbf{z})$ que indica que os dados observáveis em $\mathbf{x}$ podem ser gerados por meio dados latentes em $\mathbf{z}$. Entretanto, conforme visto anteriormente, a integral da verossimilhança marginalizada $p_\theta(\mathbf{x}) = \int p_\theta(\mathbf{z})p_\theta(\mathbf{x}|\mathbf{z})$ é intratável, inviabilizando o uso de técnicas clássicas de inferência como o Esperança-Maximização (*Expectation-Maximization*). Assim, usa-se uma distribuição variacional $q_\phi(\mathbf{z}|\mathbf{x})$ com parâmetros variacionais denotados por ϕ para a realização da inferência variacional. Os parâmetros ϕ são aprendidos juntamente com os parâmetros θ do modelo gerador. O modelo gráfico probabilístico dos Autocodificadores Variacionais são mostrados na Figura 2.11.

No processo de inferência variacional, os Autocodificadores Variacionais geralmente aproximam uma distribuição intratável $p_\theta(\mathbf{z}|\mathbf{x})$ por meio de uma distribuição $q_\Phi(\mathbf{z}|\mathbf{x})$, denominada de modelo de reconhecimento. Uma vez que a coleção de da-

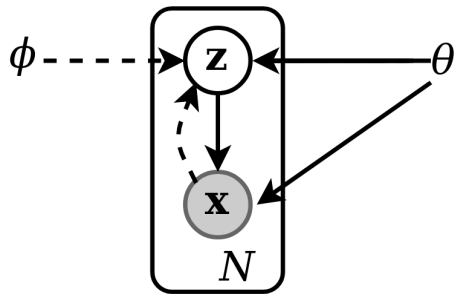


Figura 2.11: Representação do modelo gráfico probabilístico direcionado dos Autocodificadores Variacionais, presente em Kingma & Welling [2013]. Linhas sólidas denotam o modelo gerador $p_\theta(\mathbf{z})p_\theta(\mathbf{x}|\mathbf{z})$, enquanto linhas tracejadas representam a aproximação variacional $q_\phi(\mathbf{z}|\mathbf{x})$ em relação à distribuição intratável $p_\theta(\mathbf{z}|\mathbf{x})$.

dos $\mathbf{X}$ produz uma distribuição sobre todos os possíveis valores da variável latente $\mathbf{z}$, o modelo de reconhecimento é denominado codificador [Kingma & Welling, 2013]. De maneira similar, $\mathbf{z}$ produz uma distribuição sobre todos os possíveis valores de $\mathbf{x}$ e é frequentemente referido como rede de reconstrução ou decodificador. Em vez de usar *frameworks* tradicionais para aproximar os valores dos parâmetros do modelo, Autocodificadores Variacionais usam redes neurais para estimar os parâmetros das distribuições utilizadas no codificador e no decodificador. A entrada de dados da rede de reconstrução corresponde aos valores de $\mathbf{z}$, sendo estes valores amostrados da distribuição de probabilidade $q_\Phi(\mathbf{z}|\mathbf{x})$, que é provida pela rede de codificação.

Em Autocodificadores Variacionais clássicos, a distribuição variacional utilizada é proveniente da família das distribuições Normais, também denominadas de Gaussianas. Deste modo, um Autocodificador clássico deve estimar dois parâmetros para a realização da inferência variacional, que são a média e o desvio-padrão da distribuição Gaussiana, representados respectivamente pelas letras gregas μ e σ.

2.8.2 Arquitetura de redes neurais em ACVs

A Figura 2.12 mostra a arquitetura de um Autocodificador de forma detalhada. Diferentemente das abordagens estatísticas clássicas, os parâmetros são estimados pelo codificador, composto por redes neurais de múltiplas camadas que tem como entrada uma matriz de frequência termo-documento representada por $\mathbf{x}$. Tendo os parâmetros da distribuição variacional sido estimados, obtém-se os dados latentes representados pela variável aleatória $\mathbf{z}$ por meio do processo de amostragem, de tal forma que $\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})$. Em seguida, os valores provenientes de $\mathbf{z}$ são dados como entrada para o decodificador, que assim como o codificador, é formado por redes neurais artificiais de

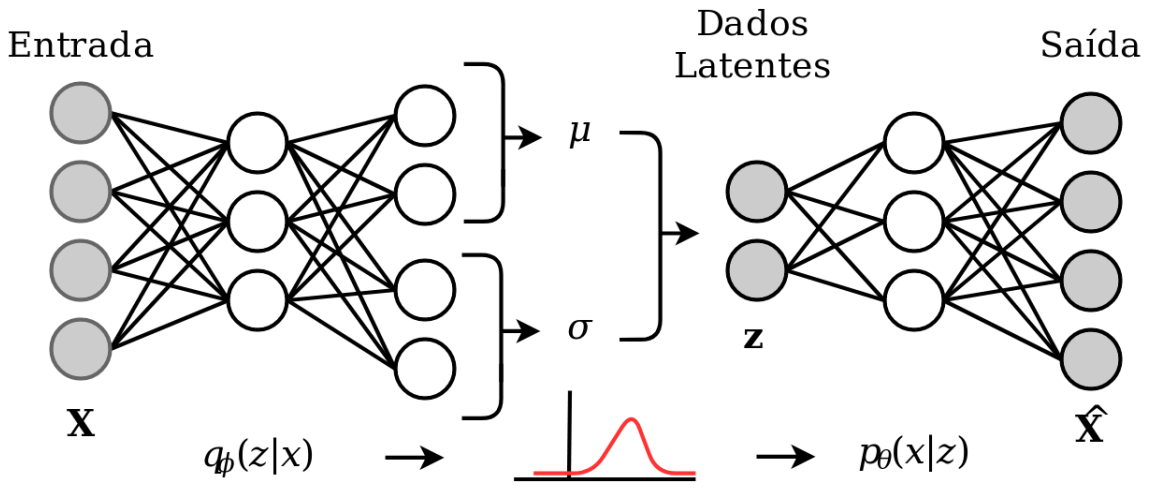


Figura 2.12: Representação de um Autocodificador Variacional padrão. A rede neural à esquerda representa o codificador, enquanto que a rede disposta à direita denota o decodificador. Repare que as redes neurais são utilizadas para a estimativa dos parâmetros das distribuições, ou seja, participam indiretamente do processo de reconstrução dos dados de entrada. A distribuição representada na parte inferior é da família das distribuições Gaussianas, cujos parâmetros μ e σ são estimados pela rede de codificação.

múltiplas camadas. O decodificador participa na estimativa da distribuição geradora $p_\theta(\mathbf{x}|\mathbf{z})$ e produz como saída os dados reconstruídos representados por $\hat{\mathbf{x}}$.

Os Autocodificadores Variacionais têm como função de perda o limite inferior da evidência. Em outras palavras, como o processo de inferência é realizado usando o *framework* variacional, a rede tem como objetivo calcular os parâmetros das distribuições de probabilidade que maximizam o limite inferior variacional. Assim, o processo de treino visa aproximar $q_\phi(\mathbf{z}|\mathbf{x})$ à distribuição $p_\theta(\mathbf{z}|\mathbf{x})$, tendo como consequência a sintetização de dados cada vez mais similares em comparação com os dados presentes na coleção de treino. Na próxima subseção, será apresentado o limite inferior variacional proveniente do processo de inferência variacional (cf. Seção 2.4) adaptado para Autocodificadores Variacionais.

2.8.3 Limite inferior variacional

A distribuição marginal de probabilidade é composta por um somatório de probabilidades marginalizadas sobre amostras individuais de dados $\log p_\theta(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \cdots, \mathbf{x}^{(N)}) = \sum_{i=1}^{N} \log p_\theta(\mathbf{x}^{(i)})$, que pode ser reescrita utilizando a teoria de inferência variacional, conforme indicado na Equação 2.12 [Kingma & Welling, 2013].

$$\log p_\theta(\mathbf{x}^{(i)}) = D_{KL}(q_\phi(\mathbf{z}|\mathbf{x}^{(i)})||p_\theta(\mathbf{z}|\mathbf{x}^{(i)})) + \mathcal{L}(\theta; \phi; \mathbf{x}^{(i)}) \tag{2.12}$$

A primeira parcela da Equação 2.12 denota a divergência KL da aproximação da distribuição variacional em relação à distribuição à posteriori original. A segunda denota o limite inferior da evidência, que pode ser reformulada conforme visto na Seção 2.4:

$$\begin{aligned} \log p_\theta(\mathbf{x}^{(i)}) &\geq \mathcal{L}(\theta;\phi;\mathbf{x}^{(i)}) \\ &= \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[-\log q_\phi(\mathbf{z}|\mathbf{x}) + \log p_\theta(\mathbf{x},\mathbf{z})] \end{aligned} \tag{2.13}$$

Por fim, o limite inferior da evidência pode ser formulado a partir da Equação 2.13:

$$\mathcal{L}(\theta;\phi;\mathbf{x}^{(i)}) = D_{KL}(q_\phi(\mathbf{z}|\mathbf{x}^{(i)})||p_\theta(\mathbf{z})) + \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[-\log p_\theta(\mathbf{x}^{(i)}|\mathbf{z})] \tag{2.14}$$

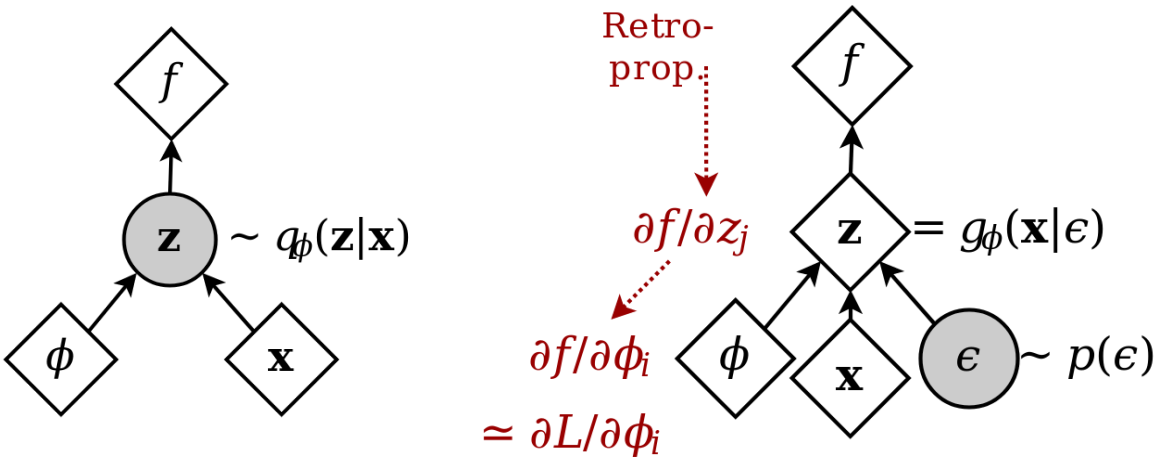
Desta forma, a Equação 2.14 pode ser maximizada em relação aos parâmetros ϕ e θ. A propagação na rede neural dos Autocodificadores Variacionais é realizada utilizando o limite inferior da evidência como função de perda. A divergência KL $D_{KL}(q_\phi(\mathbf{z}|\mathbf{x}^{(i)})||p_\theta(\mathbf{z}))$ pode ser calculada de forma analítica, caso as duas distribuições sejam da família das distribuições Gaussianas. Por outro lado, computa-se $\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[-\log p_\theta(\mathbf{x}^{(i)}|\mathbf{z})]$ por meio de processo de amostragem, conforme indicado na Equação 2.15.

$$\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[-\log p_\theta(\mathbf{x}^{(i)}|\mathbf{z})] \approx -\frac{1}{L}\sum_{l=1}^{L} \log p_\theta(\mathbf{x}^{(i)}|\mathbf{z}^{(l)}) \tag{2.15}$$

Embora a propagação em Autocodificadores seja simples, a efetuação da retropropagação do erro na rede possui uma série de obstáculos, em geral proporcionados pela variável latente $\mathbf{z}$, cujos valores não são diretamente fornecidos pela rede neural e sim obtidos via amostragem da distribuição $q_\Phi(\mathbf{z}|\mathbf{x})$. Na Subseção 2.8.4 aborda-se melhor este problema, descrevendo uma solução denominada truque de reparametrização.

2.8.4 Truque de reparametrização

Existem várias famílias de distribuição de probabilidade que podem ser usadas nos codificadores e nos decodificadores, dependendo das características dos dados e do modelo utilizado. Entretanto, para que a otimização no codificador por meio de métodos que usam diferenciação seja possível, é necessário realizar o processo de retropropagação [LeCun et al., 1988] através da variável aleatória contínua $\mathbf{z}$, amostrada da distribuição $q_\Phi(\mathbf{z}|\mathbf{x})$. Este fato é um problema, pois a retropropagação do erro não é possível em nós aleatórios. Embora a aplicação de métodos de diferenciação por meio de amostragem tal como estimador de gradiente Monte Carlo ingênuo seja possível,



(a) (b)

Figura 2.13: Fluxograma dos nós de um Autocodificador Variacional sob a forma original (2.13a) e a forma reparametrizada (2.13b). Losangos representam nós determinísticos, ao passo que círculos indicam nós aleatórios. Além disso, setas contínuas indicam o sentido da propagação, enquanto que as tracejadas denotam a direção da retropropagação. Baseado no material de Kingma [2015].

incorrem em estimadores de alta variância [Paisley et al., 2012], tornando impraticável a maximização do limite inferior variacional via técnicas clássicas.

Para lidar com esta questão, Kingma & Welling [2013] propuseram uma técnica denominada de truque de reparametrização. Este artifício teórico consiste em assumir que a distribuição variacional pode ser redefinida como uma função diferenciável $g_\phi(.)$ com um conjunto de parâmetros ϕ, que pode ser aprendido pela rede de codificação e com uma variável auxiliar não determinística ϵ. Assim, as derivadas usadas no processo de retropropagação do erro podem ser calculadas, permitindo o treino destas redes, conforme indicado na Figura 2.13.

Segundo Kingma & Welling [2013], a reparametrização é útil no contexto dos Autocodificadores Variacionais, já que a técnica pode ser usada para reformular a esperança relativa à distribuição $q_\phi(\mathbf{z}|\mathbf{x})$ tal que a estimativa da esperança via abordagem Monte Carlo seja diferenciável com respeito a ϕ. A prova é dada como se segue [Kingma & Welling, 2013]. Considere um mapeamento determinístico $\mathbf{z} = \boldsymbol{g_\phi(\epsilon, x)}$, onde se sabe que a integral $q_\phi(\mathbf{z}|\mathbf{x}) \sum_i dz_i = p(\epsilon) \sum_i d\epsilon_i$. Portanto[2], $\int q_\phi(\mathbf{z}|\mathbf{x}) f(\mathbf{z}) d\mathbf{z} = \int p(\epsilon) f(\mathbf{z}) d\epsilon$. Uma vez que as integrais correspondem respectivamente às esperanças de $q_\phi(\mathbf{z}|\mathbf{x})$ e $p(\epsilon)$, um estimador diferenciável pode ser construído por meio da Equação $q_\phi(\mathbf{z}|\mathbf{x}) f(\mathbf{z}) d\mathbf{z} \approx \frac{1}{L} \sum_{l=1}^{L} f(g_\phi(\mathbf{x}, \epsilon^{(l)}))$, onde $\epsilon^{(l)} \sim p(\epsilon)$. No ponto de vista do processo de diferenciação,

[2]Note que para representação infinitesimal usa-se a notação convencional $d\mathbf{z} = \prod_i d\mathbf{z}_i$

isto significa que o processo estocástico é deslocado para a distribuição $p(\epsilon)$, que fica fora do escopo do gradiente. Desta forma, este truque estatístico pode ser usado para estimar o valor do limite inferior variacional de forma eficiente [Rezende et al., 2014].

Em Autocodificadores Variacionais clássicos, onde $\mathbf{z} \sim q_\Phi(\mathbf{z}|\mathbf{x})$ segue uma distribuição Normal $\mathcal{N}(\mu, \sigma^2)$, a função de reparametrização é $z = \mu + \sigma\epsilon$, onde ϵ é uma variável auxiliar de ruído tal que $\epsilon \sim \mathcal{N}(0, 1)$. Logo, um estimador determinístico pode ser formulado como se segue (Equação 2.16):

$$\mathbb{E}[f(g_\phi(\mathbf{x}, \epsilon^{(l)})] \simeq \frac{1}{L}\sum_{l=1}^{L} f(g_\phi(\mathbf{x}, \epsilon^{(l)})) = \frac{1}{L}\sum_{l=1}^{L} f(\mu + \sigma\epsilon^{(l)}) \tag{2.16}$$

Obter um truque de reparametrização para uma distribuição específica pode ser um desafio considerável. Três abordagens podem ser utilizadas para se obter uma função de reparametrização [Kingma & Welling, 2013]:

- Função de distribuição acumulada (FDA) inversa: Neste caso, $\epsilon \sim \mathcal{U}(0, I)$, onde $\mathcal{U}$ é a distribuição Uniforme, e $g_\phi(\mathbf{x}, \epsilon)$ é a função FDA inversa de $q_\theta(\mathbf{z}|\mathbf{x})$. A função deve ser tratável, o que não é possível na maioria das distribuições. A amostragem por meio da FDA inversa é um processo complexo que pode ser explorado com mais detalhes no trabalho de Sillitto [1969]. Exemplos de distribuições de probabilidades passíveis de serem reparametrizadas por este método: Exponencial, Cauchy, Logística, Rayleigh, Pareto, Weibull, Reciprocal, Gompertz, Gumbel e Erlang.

- Família de distribuições do tipo "localização-escala": distribuições formadas por dois parâmetros indicando a localização (*e.g*, a média) e a escala (*e.g*, o desvio-padrão) podem ser reparametrizadas na forma padrão, quando a localização e a escala equivalem a 0 e 1, respectivamente. Neste caso, ϵ é amostrado da distribuição padrão e $g(.) = \text{localização} + \text{escala} \; . \; \epsilon$. Exemplos: distribuição Gaussiana, de Laplace, Elíptica, T de Student, Logística, Uniforme e Triangular.

- Composição: Nesta abordagem, as variáveis randômicas são expressas como transformações de variáveis auxiliares. Exemplos: Normal-Logística (exponenciação de variável normalmente distribuída), Gamma (soma de variáveis exponencialmente distribuídas), Dirichlet (soma ponderada de valores distribuídos sobre a distribuição Gamma), Beta, Qui-Quadrado e distribuições F.

Para distribuições com FDA inversa intratável, é possível gerar boas aproximações da FDA que requerem uma complexidade computacional comparável com o cálculo da função densidade de probabilidade [Devroye, 1986].

2.9 Considerações Finais

Neste capítulo apresentou-se os conceitos necessários para o entendimento deste trabalho, tais como modelos probabilísticos de tópicos, inferência estatística (com foco na inferência variacional), redes neurais de múltiplas camadas, *embedding* de palavras e Autocodificadores Variacionais. Outros conceitos mais específicos dos modelos propostos, como as distribuições Normais-Logísticas e *Gumbel-Softmax* são apresentados no Capítulo 4. Adicionalmente, as métricas de avaliação são detalhadas no Capítulo 5.

No próximo capítulo serão mostrados alguns modelos de tópicos, que utilizam alguns ou todos os conceitos apresentados neste capítulo de fundamentos. Consequentemente, estes conceitos são fundamentais para o entendimento dos modelos de tópicos propostos neste trabalho de pesquisa (GSDTM e LMDTM).

3

Trabalhos Relacionados

Este capítulo tem como objetivo apresentar os principais trabalhos disponíveis na literatura relacionados à área de modelagem de tópicos. Dividiu-se os modelos de tópicos em três categorias: modelos gráficos probabilísticos tradicionais (Seção 3.1), que não utilizam redes neurais no processo de inferência; modelos gráficos probabilísticos neurais e não direcionados (Seção 3.2), que ao contrário dos modelos tradicionais são baseados em redes neurais estruturadas em modelos probabilísticos não direcionados; e modelos baseados em Autocodificadores Variacionais (Seção 3.3), que são baseados em redes neurais estruturadas em modelos gráficos probabilísticos direcionados e que utilizam o processo de inferência variacional. Mais adiante, na Seção 3.4, discute-se sobre os métodos baseados em Autocodificadores Variacionais voltados à sintetização de imagens e geração de sentenças temáticas, métodos que não são modelos de tópicos mas que estão relacionados em algum grau com este trabalho de pesquisa. Por fim, aborda-se na Seção 3.5 um comparativo entre os métodos propostos e os trabalhos relacionados, seguido das considerações finais (Seção 3.6).

3.1 Modelos de Tópicos baseados em Modelos Gráficos Probabilísticos Tradicionais

Estes modelos aplicam a teoria e as técnicas de inferência dos modelos gráficos probabilísticos (*e.g.*, inferência variacional ou amostragem de *Gibbs*) para realizar a tarefa de modelagem de tópicos, sem a utilização de redes neurais no processo de aprendizagem.

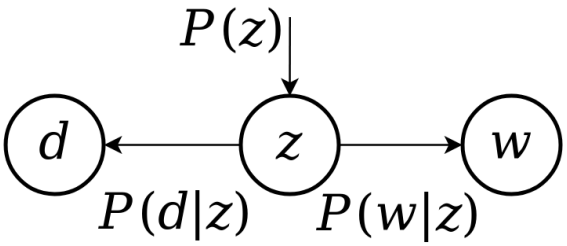


Figura 3.1: Representação do modelo gráfico do PLSI.

O primeiro modelo probabilístico de tópicos foi o *Probabilistic Latent Semantic Indexing* (PLSI) [Hofmann, 1999], também denominado na literatura como *Probabilistic Latent Semantic Analysis* (PLSA). Surgiu como uma versão probabilística do *Latent Semantic Indexing* (LSI) [Deerwester et al., 1990], que consiste de uma técnica de recuperação de informação baseada na análise da matriz de contagem entre termos e documentos que visa descobrir estruturas semânticas por meio de métodos baseados em álgebra linear [Deerwester et al., 1990]. O intuito do PLSI no contexto de modelos de tópicos é realizar a organização e sumarização de vastas coleções de documentos eletrônicos, conciliando a qualidade do LSI com as vantagens da abordagem probabilística.

Conforme mostrado na Figura 3.1, os componentes que definem a abordagem PLSI são especificamente um conjunto de documentos $D = \{d_1, \ldots, d_N\}$, sendo N o número total de documentos presentes na coleção, um conjunto $Z = \{z_1, \ldots, z_K\}$ de tópicos e um conjunto $W = \{w_1, \ldots, w_M\}$ de palavras. A principal vantagem do PLSI é que tópicos podem ser gerados *a priori*, e tanto os documentos quanto as palavras podem ser determinados a partir do tópico gerado. Deste modo, tem-se um modelo que estima a distribuição de probabilidades para os tópicos, assumindo que os tópicos são latentes, ou seja, não são conhecidos no momento da inferência. Por meio do teorema de Bayes e da incorporação da variável latente z (onde z representa um tópico) a Equação 3.1 pode ser obtida através do cálculo da probabilidade de um documento d conter a palavra w.

$$\widehat{P}_{LSI}(d, w) = \sum_{z \in Z} P(d|z)P(z)P(w|z) \tag{3.1}$$

O processo pode ser realizado através do método Esperança-Maximização, um método probabilístico robusto capaz de estimar os parâmetros utilizados em um modelo probabilístico gerador.

Embora o método PLSI tenha marcado o início do estudo de modelos probabilísticos voltados para modelagem de tópicos, a popularidade desta área de estudo deve-se

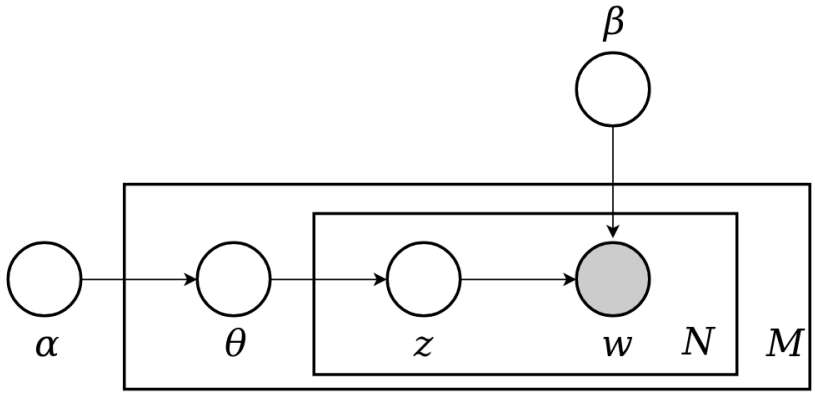


Figura 3.2: Representação do modelo gráfico do LDA, baseada na figura presente em Blei et al. [2003a]. As caixas são representações em "pratos" indicando replicação de variáveis aleatórias. O "prato" mais externo representa os M documentos, enquanto que o "prato" mais interno representa a escolha de tópicos e palavras dentro de um documento.

sobretudo ao modelo denominado *Latent Dirichlet Allocation* (LDA), proposto por Blei et al. [2003b]. O LDA é um modelo gerador probabilístico para dados discretos, tais como coleções de texto. A intuição principal do LDA é que os documentos cobrem um determinado número de tópicos, ao passo que estes são representados por um determinado número de palavras. Os documentos são analisados por meio do uso da distribuição de Dirichlet, utilizada como distribuição *a priori* e por meio da aplicação da distribuição Multinomial sobre a relação entre documentos e tópicos e entre tópicos e palavras. No LDA, cada item da coleção de dados é modelado como um modelo de mistura finita sobre os tópicos. Entende-se como um modelo de mistura finita como combinação convexa de k densidades f_i cada uma associada a um peso $p_i > 0$ [de Oliveira & Loschi, 2013]. Cada tópico é, por sua vez, modelado como uma mistura infinita (quando o número de densidades tente ao infinito) de distribuições sobre um conjunto de tópicos [Blei et al., 2003b]. O processo gerador para cada documento d em uma coleção de dados $\mathcal{D}$, onde α e β são parâmetros com k dimensões, é definido a seguir conforme representado na Figura 3.2 [Blei et al., 2003b]:

- Escolha um valor para θ proveniente de uma distribuição de Dirichlet com parâmetro α.
- Para cada palavra w_n presente no documento:
  - Escolha z_n proveniente de uma distribuição Multinomial com parâmetro θ;
  - Escolha uma palavra w_n proveniente da distribuição de probabilidades $p(w_n|z_n, \beta)$ condicionada ao tópico z_n e tendo β com parâmetro.

A fim de proporcionar uma maior clareza do funcionamento do modelo LDA, considere a analogia do modelo gerador com um possível processo de escrita de uma redação dissertativo-argumentativa. O autor não conhece previamente as palavras que ele escreverá, porém, baseado na sua própria experiência, ele sabe *a priori* a proporção de tópicos que ele deverá abordar em cada parágrafo. Por exemplo, o autor pode decidir focar em política e economia em um parágrafo, e em meio ambiente e ecologia no outro parágrafo, a fim de dar suporte às argumentações presentes na redação. Desta forma, essa proporção *a priori* de tópicos é dada pela distribuição de Dirichlet no caso do LDA. Após decidir a proporção de tópicos na redação, o autor utilizará este parâmetro para decidir quais tópicos ele irá abordar em cada parágrafo. Por fim, o autor escolherá qual a palavra mais relacionada ao tópico escolhido para redigir o parágrafo da redação. Desta forma, pode-se observar que o modelo gerador representa com certa verossimilhança o processo de escrita de textos baseados em tópicos, permitindo assim a inferência de tópicos mais coerentes quando comparado ao modelo PLSI.

Dado os parâmetros α e β, a verossimilhança marginal de w, ou seja, a distribuição conjunta de probabilidades entre w e θ (parâmetro *a priori*) onde θ sofre o processo de marginalização, possui a seguinte Equação (3.2):

$$p(\mathbf{w}|\alpha,\beta) = \frac{\Gamma(\sum_{i=1}^{k}\alpha_i)}{\prod_{i=1}^{k}\Gamma(\alpha_i)} \int \left(\prod_{i=1}^{k}\theta_i^{\alpha_i-1}\right)\left(\prod_{n=1}^{N}\sum_{i=1}^{k}\prod_{j=1}^{V}(\theta_i\beta_{ij})^{w_n^j}\right) d\theta \tag{3.2}$$

Repare que a verossimilhança marginal segue a regra de Bayes. A primeira razão pode ser considerada como uma constante de normalização, enquanto que o primeiro item entre parênteses localizado no começo da integral corresponde a probabilidade *a priori* $p(\theta|\alpha)$ do modelo. Por fim, o segundo item entre parênteses localizado no final da integral está associado a distribuição multinomial. A probabilidade conjunta pode ser computada usando o ferramental de inferência Bayesiana, tal como amostragem de Gibbs e inferência variacional. Observe também que a escolha das distribuições Multinomial e Dirichlet no modelo não é aleatória. A distribuição Multinomial representa bem dados discretos que podem ter mais de dois estados possíveis, tais como palavras e tópicos. A distribuição de Dirichlet, por ser uma distribuição conjugada da distribuição multinomial[1], é escolhida como distribuição *a priori* com o intuito de evitar integrais impróprias e viabilizar o cálculo de inferência. Devido a estas características, aliada

[1]Uma distribuição *a priori* p_1 é conjugada de outra (p_2) quando a distribuição *a posteriori* de p_2 é da mesma família de distribuições da p_1 [Gelman et al., 2013]. No contexto do LDA, a distribuição de Dirichlet é conjugada da Multinomial, pois a distribuição *a posteriori* da Multinomial pode ser definida como uma distribuição de Dirichlet.

à alta coerência dos tópicos obtida utilizando este modelo, o LDA se popularizou e tornou-se o estado da arte na literatura de modelos de tópicos por vários anos.

Vários estudos posteriores se dedicaram em melhorar o LDA para cada domínio. Por exemplo, Griffiths et al. [2004] propuseram o *hierarchical LDA* (hLDA), que utiliza uma otimização do cálculo de probabilidades no modelo LDA utilizando árvores, denominado *Nested Chinese Restaurant Process* (NCRP). Relaxando-se o modelo LDA e assumindo que palavras e tópicos podem ser representados hierarquicamente, é possível utilizar árvores para representar as distribuições de probabilidade. Desta forma, a complexidade computacional necessária para se computar as distribuições de probabilidade é reduzida para uma escala logarítmica $O(log(n))$. Logo, o modelo hLDA se propõe a tentar resolver o cenário onde existem coleções vastas de dados a serem processadas. Griffiths et al. [2004] concluem que o modelo proposto é flexível e generalista para representar hierarquias de tópicos, que naturalmente acomodam coleções de dados que crescem de forma significativa.

Blei & Lafferty [2006] propuseram uma modificação do LDA para analisar o cenário onde os tópicos se alteram ao longo do tempo, denominada de *Dynamic Topic Model* (DTM). Este modelo supõe que a coleção de dados é dividida em períodos de tempo, como por exemplo em períodos anuais. Para isso, o método DTM modela os documentos pertencentes a cada período em um modelo de tópicos com K componentes, onde os tópicos associados com o período t são derivados do período de tempo anterior $(t-1)$. Mais formalmente, este modelo emprega K representações LDA, cada uma associada a um período de tempo t. Cada parâmetro α e β relativos a um período t de tempo são dependentes dos parâmetros do período anterior. Os autores mostraram que é possível utilizar uma mistura de distribuições Gaussianas para estimar os valores dos parâmetros α e β para cada período de tempo e que é possível adotar a distribuição Normal-Logística para estimar a proporção θ de tópicos. Os resultados mostraram que o modelo DTM pode fornecer um modelo preditivo mais acurado capaz de oferecer novos meios de navegação entre tópicos em coleções de dados vastas e não estruturas.

Ainda estudando as possibilidades do uso da distribuição Normal-Logística no parâmetro de proporção de tópicos, Blei & Lafferty [2007] propuseram o *Correlated Topic Model* (CTM), que visa substituir a distribuição *a priori* do LDA (Dirichlet) pela Normal-Logística. A hipótese principal dos autores é que a distribuição Normal-Logística consegue aprender melhor a correlação entre os tópicos do que a distribuição *a priori* original do LDA. Os autores testaram o modelo proposto em coleções de artigos científicos e mostraram que para este domínio, o CTM consegue atingir valores de perplexidade melhores do que o obtido com o LDA.

Li & McCallum [2006] abordaram uma alternativa ao CTM denominada de *Pa-*

Reinforcement Learning			Human Receptive System		
LDA	n-gram (2+)	n-gram (1)	LDA	n-gram (2+)	n-gram (1)
state	reinforcement learning	action	motion	receptive field	motion
learning	optimal policy	policy	visual	spatial frequency	spatial
policy	dynamic programming	reinforcement	field	temporal frequency	visual
action	optimal control	states	position	visual motion	receptive
reinforcement	function approximator	actions	figure	motion energy	response
states	prioritized sweeping	function	direction	tuning curves	direction
time	finite-state controller	optimal	fields	horizontal cells	cells
optimal	learning system	learning	eye	motion detection	figure
actions	reinforcement learning rl	reward	location	preferred direction	stimulus
function	function approximators	control	retina	visual processing	velocity
algorithm	markov decision problems	agent	receptive	area mt	contrast
reward	markov decision processes	q-learning	velocity	visual cortex	tuning
step	local search	goal	vision	light intensity	moving
dynamic	state-action pair	space	moving	directional selectivity	model
control	markov decision process	step	system	high contrast	temporal
sutton	belief states	environment	flow	motion detectors	responses
rl	stochastic policy	system	edge	spatial phase	orientation
decision	action selection	problem	center	moving stimuli	light
algorithms	upright position	steps	light	decision strategy	stimuli
agent	reinforcement learning methods	transition	local	visual stimuli	cell

Figura 3.3: Dois tópicos extraídos de 50 tópicos aprendidos pelo modelo TNG [Wang et al., 2007]. Repare que a saída destes modelos de tópicos são n-gramas relacionados com um tópico específico.

chinko Allocation Model (PAM). A ideia principal deste modelo é permitir a correlação entre N tópicos quaisquer, diferentemente do CTM que é capaz apenas de correlacionar pares de tópicos. Segundo os autores, o PAM consegue identificar correlações possivelmente esparsas e aninhadas entre os tópicos. O modelo foi testado e comparado com os modelos LDA, CTM e hLDA segundo a métrica de perplexidade, obtendo os melhores resultados quando testado com mais de 100 tópicos. Também o modelo PAM consegue superar o LDA em termos de acurácia de classificação de documentos.

Rosen-Zvi et al. [2004] introduziram uma extensão do modelo LDA, nomeado de *Author-Topic Model*, a fim de identificar informação de autoria em coleção de documentos. Neste modelo, cada autor é associado a uma distribuição multinomial sobre os tópicos presentes no texto e cada tópico é associado a uma distribuição multinomial sobre as palavras. Um documento com múltiplos autores é modelado como um modelo de mistura associado aos autores. Rosen-Zvi et al. [2004] mostraram que este modelo é capaz de identificar a probabilidade do autor estar relacionado ao documento analisado, superando o LDA nesta aplicação em termos de perplexidade.

Alguns modelos gráficos probabilísticos de tópicos exploraram a relação entre as palavras no texto por meio de N-gramas, em oposição à representação em *bag of words*, que considera apenas as relações entre palavras e documentos. Desta forma, estes modelos tendem a apresentar resultados melhores, pois capturam informações referentes ao significado do texto, que são vantajosas em várias tarefas de mineração em texto [Wang et al., 2007]. Neste contexto, Wallach [2006] propôs o *Bigram Topic Model* (BTM), um modelo derivado do LDA que incorpora sequências de texto em

n-gramas e variáveis latentes de tópicos. Mais especificamente, o BTM opera com uma sequência de palavras $(\cdots, w_{i-1}, w_i, w_{i+1}, \cdots)$, onde cada palavra desta sequência está associada a uma variável latente de tópico z. Também cada palavra w_i depende apenas da palavra anterior w_{i-1}, formando relações em bigramas. Para treinar o modelo, o autor utilizou o método de Esperança-Maximização. Como resultado, o BTM supera o LDA em termos de *Bits* por palavra. Outro modelo desenvolvido com base em n-gramas foi o *Topical N-gram Model* (TNG) [Wang et al., 2007]. A vantagem deste modelo sobre o BTM é a capacidade de aprender um número arbitrário de gramas, diferentemente do BTM, que é capaz de aprender apenas duas gramas (bigramas). Desta forma, tem-se como resultado um modelo baseado em n-gramas mais generalista, conforme mostrado na Figura 3.3.

3.2 Modelos de Tópicos baseados em Redes Neurais e em modelos não direcionados

Na última década, modelos baseados em redes neurais surgiram como uma alternativa viável para modelagem de tópicos, em virtude do crescimento da área de aprendizagem profunda. Por exemplo, Salakhutdinov & Hinton [2009] propuseram uma rede neural estruturada em um modelo não direcionado, denominada de *Replicated Softmax* (RS), capaz de estimar a probabilidade de observar uma nova palavra em um documento dado um conjunto de palavras observadas previamente. Além de gerar documentos, o modelo RS é capaz de aprender representações interpretáveis dos documentos. Mais especificamente, o *Replicated Softmax* é uma técnica pertencente à família do *Restricted Boltzmann Machine (RBM)*, uma rede neural artificial cujo propósito é aprender a distribuição de probabilidade sobre o conjunto de entrada por meio da distribuição de Boltzmann. O termo *Restricted* deriva do fato de que a estrutura da rede necessita estar disposta conforme um grafo bipartido, onde uma partição contém as variáveis observáveis e outra contém as variáveis latentes. O resultado desta restrição é um treino mais eficiente, principalmente quando realizado concomitantemente com o algoritmo de *Contrastive Divergence* [Hinton, 2002], que consiste em um método que estima as derivadas parciais da função de custo de uma rede RBM utilizando uma versão simplificada do MCMC, já que a aprendizagem por meio da estimativa por máxima verossimilhança (*Maximum-likelihood Estimation* ou MLE) é intratável devido ao custo exponencial [Salakhutdinov & Hinton, 2009].

Larochelle & Lauly [2012] propuseram o DocNADE, uma versão do *Neural Autoregressive Distribution Estimator* (NADE) [Larochelle & Murray, 2011] para mo-

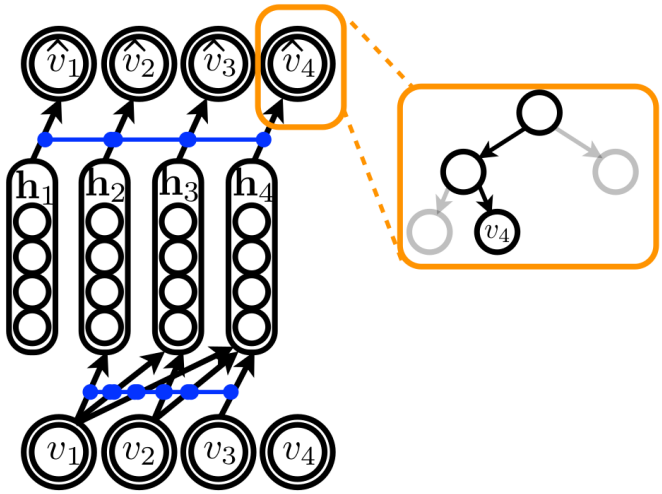


Figura 3.4: Ilustração do DocNADE extraída do trabalho de Larochelle & Lauly [2012]. Repare que as conexões de cada observação multinomial v_i são compartilhadas entre si. O compartilhamento também ocorre com cada variável latente do modelo. Por fim, a probabilidade condicional é representada por uma árvore binária de regressões logísticas.

delagem de tópicos em documentos. O NADE consiste em uma rede similar a uma rede neural do tipo Autocodificador, onde o vetor de entrada de dados observáveis e o vetor de saída possuem o mesmo tamanho. Entretanto, diferentemente de uma rede Autocodificador padrão, o modelo NADE fornece como dados de saída da rede as probabilidades condicionais de cada dado observável em função dos dados observáveis anteriores [Larochelle & Lauly, 2012]. Em outras palavras, dado um vetor de dados observáveis $\mathbf{v} = (v_1, v_2, \cdots, v_N)$, o NADE calcula a probabilidade condicional $p(v_i|\mathbf{v}_{<i})$, onde probabilidade de v_2 depende de v_1, e a probabilidade de v_3 depende de v_2 e v_1, e assim por diante. A principal contribuição do modelo DocNADE em relação do NADE é que este modelo aplica uma simplificação da probabilidade condicional $p(\mathbf{v})$ inspirada nas equações recursivas de campo-médio condicional do RS, de forma que a probabilidade de uma palavra ser observada em um documento é condicionada às palavras previamente observadas no documento [Larochelle & Lauly, 2012], conforme mostrado na Figura 3.4. Como resultado deste processo, a função *Softmax* usada como distribuição sobre as palavras pode ser substituída por uma distribuição hierárquica definida sobre os caminhos de um árvore binária de palavras. Desta forma, o DocNADE escala de forma logarítmica de acordo com o tamanho do vocabulário, ao invés de escalar linearmente como o RS. Por fim, os autores mostraram também que o DocNADE supera o RS na tarefa de recuperação de texto.

Posteriormente, Srivastava et al. [2013] introduziram uma versão profunda do RS

denominada *Over-Replicated Softmax* (ORS). Este modelo é uma generalização do RS que permite a utilização de mais de uma camada de unidades latentes com estrutura bipartida na rede neural. A utilização de várias camadas permite uma flexibilização maior da representação dos *priors* na rede. Entretanto, o custo computacional resultante deste processo é consideravelmente maior quando comparado ao RS. Para amenizar este problema, o método ORS realiza um pré-treinamento do modelo utilizando uma rede RBM simples com os pesos escaladas por um fator de $1 + \frac{M}{N}$. Segundo Srivastava et al. [2013], embora este processo de pré-treinamento não seja capaz de se aproximar do valor máximo de verossimilhança, na prática isto efetua uma boa estimativa das unidades latentes, reduzindo o número de iterações necessárias para o treinamento deste modelo.

Outros modelos neurais linguísticos interessantes foram propostos para aplicação em dados textuais, como por exemplo o GMNTM [Yang et al., 2015] e SLRTM [Tian et al., 2016]. O GMNTM representa cada tópico como um vetor multidimensional onde cada palavra é dependente não somente do tópico a qual ela está relacionada, mas também pelo vetor de *embedding* das palavras próximas. Tanto as sentenças e palavras quanto o agrupamento responsável por categorizar as palavras em tópicos são aprendidos de forma conjunta, ou seja, o GMNTM incorpora em uma mesma função de custo o objetivo de minimizar o erro de embedding e de agrupamento. Como o GMNTM, o SLRTM também enfatiza a modelagem da ordem entre as palavras com o objetivo de melhorar a identificação de tópicos usando um ferramental baseado em Redes Neurais Recorrentes.

3.3 Modelos de Tópicos baseados em Autocodificadores Variacionais

Recentemente, modelos baseados em Autocodificadores Variacionais foram adaptados com sucesso como modelos de tópicos, resultando em métodos como o *Neural Variational Document Model* (NVDM) [Miao et al., 2016] e o *Product Latent Dirichlet Allocation* (ProdLDA) [Srivastava & Sutton, 2017]. Estes métodos propõem uma nova abordagem que consiste na utilização de Autocodificadores Variacionais para a realização da inferência de tópicos. O principal objetivo na adaptação desses modelos geradores de imagens para a aplicação de modelagem de tópicos constitui-se na eficácia dessas redes em aprender uma distribuição latente por meio do processo de reconstrução de dados. Desta forma, assumindo que tópicos podem ser aprendidos por meio de distribuições latentes, estes modelos podem ser capazes de aprender tópicos mais

relevantes presentes em uma coleção de documentos quando comparados com outros métodos presente na literatura, em termos de perplexidade e coerência de tópicos e quando usados nas coleções de dados 20newsgroups e RCV1-v2 [Miao et al., 2016; Srivastava & Sutton, 2017].

O NVDM foi proposto por Miao et al. [2016] como uma extensão do Autocodificador Variacional para modelagem de tópicos. Mais detalhadamente, este modelo assume que a variável latente segue uma distribuição Normal, da mesma forma que o Autocodificador Variacional padrão, que é utilizado na aplicação de sintetização de imagens. Entretanto, o NVDM propõe duas abordagens provenientes dos modelos baseados no *Replicated Softmax*, com o intuito de aplicar estas redes na tarefa de modelagem de tópicos. A primeira é a utilização de uma matriz de frequência entre termos e documentos como entrada de dados do Autocodificador Variacional. Desta forma, o NVDM obtém as relações de tópicos por meio da análise das relações de coocorrência dos termos em cada documento presente na coleção de dados. A segunda abordagem consiste em utilizar uma rede de decodificação *softmax*. Em outras palavras, esta rede consiste de uma regressão multinomial logística definida sob a forma $softmax(\mathbf{W}\mathbf{x} + \mathbf{b})$, onde $\mathbf{W}$ constitui-se na matriz de pesos e $\mathbf{b}$ consiste de um vetor de viés da rede. Os autores mostram que a utilização de um decodificador *softmax* é capaz de reconstruir documentos em um espaço semântico de *embedding* de palavras, de forma similar ao DocNADE. Assumindo que existem N palavras observadas em todos os documentos e que $p(\mathbf{z})$ corresponde a um *prior* Gaussiano para $\mathbf{z}$, o limite inferior variacional do NVDM pode ser definido pela equação 3.3:

$$\mathcal{L} = \mathbb{E}_{q_\phi(z|x)}[\sum_{i=1}^{N} \log\, p_\theta(\mathbf{x}_i|\mathbf{z})] - D_{KL}[q_\phi(\mathbf{z}|\mathbf{x})||p(\mathbf{z})] \tag{3.3}$$

Pode-se observar dois termos na equação acima. O primeiro corresponde à função de perda do processo de reconstrução dos dados, caracterizado pela esperança do logaritmo da função *softmax* em relação à distribuição variacional. Deste modo, esta parcela garante que a rede tentará reconstruir os dados com maior verossimilhança possível. A segunda parcela representa a divergência KL entre a distribuição variacional e a distribuição do prior Gaussiano, garantindo que essas distribuições sejam próximas entre si. Logo, esta parcela atua como um regularizador do processo de inferência.

Por fim, os autores mostram que o NVDM supera outras técnicas disponíveis na literatura de modelagem de tópicos em termos de perplexidade, como o LDA, *Replicated Softmax* e DocNADE. Deste modo, o NVDM tem como mérito ser o primeiro modelo baseado em Autocodificadores Variacionais para modelagem de tópicos e mostrar que

esta abordagem é capaz de produzir os melhores resultados, quando comparado aos principais modelos de tópicos existentes na literatura.

O outro modelo de tópicos baseado em Autocodificadores Variacionais, denominado ProdLDA, foi proposto por Srivastava & Sutton [2017]. Os autores definem este método como a primeira abordagem proposta cujo funcionamento baseia-se em uma aproximação da inferência proveniente do modelo LDA por meio de Autocodificadores Variacionais, com o intuito de extrair tópicos com maior coerência. Essa aproximação é interessante do ponto de vista teórico, visto que o LDA possui uma estrutura probabilística bem definida para a efetuação do processo de inferência de tópicos.

Os autores propõem a utilização da distribuição conjunta p do modelo LDA como distribuição a ser inferida e da distribuição variacional de campo médio q como forma de aproximar a distribuição real. Seguindo o processo de inferência dos Autocodificadores Variacionais, pode-se obter a seguinte equação (3.4):

$$L(\gamma, \phi|\alpha, \beta) = -D_{KL}[q(\theta, z|\gamma, \phi)||p(\theta, z|\alpha)] + \mathbb{E}_{q(\theta,z|\gamma,\phi)}[\log\, p(w|z, \theta, \alpha, \beta)] \tag{3.4}$$

onde γ e ϕ representam os parâmetros variacionais, e θ, z, w são os parâmetros aprendidos pelo modelo, conforme explicitado na Seção 3.1.

Entretanto, existem dois problemas que impedem o uso desta abordagem. Em primeiro lugar, o truque de reparametrização referente à distribuição Dirichlet é difícil de ser derivado [Srivastava & Sutton, 2017]. Em segundo, a utilização da variável latente discreta $\mathbf{z}$ é problemática devido ao fato do truque de reparametrização não ser possível para distribuições discretas. A fim de resolver o problema referente ao truque de reparametrização, os autores propuseram computar a distribuição Dirichlet por meio de uma distribuição Normal-Logística utilizando a técnica de aproximação Laplaceana. Desta forma, é possível obter uma aproximação dos parâmetros *a priori* de uma distribuição Dirichlet usando uma distribuição Normal-Logística (DNL), cujo truque de reparametrização é conhecido na literatura de Autocodificadores Variacionais. Para o segundo problema, os autores propuseram colapsar a variável latente por meio do processo de marginalização, de forma que a distribuição $p(w|z, \theta, \alpha, \beta)$ torne-se $p(w|\alpha, \beta)$, que é possível de ser reparametrizada. Através destas simplificações do modelo, o Autocodificador pode ser usado de forma similar ao NVDM. A rede de codificação estima os parâmetros da DNL, enquanto que a rede de codificação baseada na regressão multinomial logística realiza o processo de reconstrução.

Os autores justificam que a regressão multinomial logística pode ser interpretada como uma mistura de multinomiais, onde $w_n|\beta, \theta \sim \text{Multinomial}(1, \sigma(\beta\theta))$. O parâ-

metro β corresponde a matriz de pesos W da rede de decodificação, enquanto θ é a variável latente estimada pela rede de codificação e σ corresponde à aplicação da função *softmax*. Por fim, os autores afirmam que por meio desta interpretação, o ProdLDA pode ser descrito como uma versão do LDA computada por meio do produto de *experts* definido como $p(w_n|\theta, \beta) \propto \prod_k p(w_n|z_n = k, \beta)^{\theta_k}$.

O prodLDA utiliza como *baseline* as versões *Collapsed Gibbs* e variacional do modelo LDA, e o modelo NVDM. Como métricas de comparação, os autores adotaram duas métricas principais para comparar quantitativamente os métodos: coerência média dos tópicos e perplexidade. Os resultados mostram que o ProdLDA supera significativamente os métodos comparados em termos de coerência média dos tópicos. Entretanto, em termos de perplexidade, o método é superado por quase todos os métodos comparados.

Embora o modelo ProdLDA tenha méritos em realizar uma aproximação do modelo LDA, pode-se levantar algumas ressalvas sobre este modelo. Em primeiro lugar, embora haja correlação entre o LDA e o prodLDA, as sucessivas simplificações e a aproximação da distribuição de Dirichlet por meio da Normal-Logística usando aproximação laplaceana torna essa aproximação distante do valor ideal. Desta forma, em termos práticos, a diferença entre este modelo e o NVDM é que este utiliza uma distribuição Normal-Logística ao invés da distribuição Normal. Em segundo lugar, os autores empregam os métodos *Batch Normalization* e *Dropout* como forma de evitar problemas de instabilidade durante o processo de treinamento do modelo. Entretanto, estes métodos podem influenciar significativamente as métricas de perplexidade e coerência de tópicos, sendo esta influência estudada empiricamente nesta dissertação de mestrado. Por fim, este modelo é extremamente dependente dos parâmetros da rede, de modo que é necessário utilizar um valor alto de taxa de aprendizagem para que o funcionamento da rede seja adequado.

3.4 Outros trabalhos relacionados baseados em Autocodificadores Variacionais

Além dos Autocodificadores Variacionais voltados para modelagem de tópicos, outras técnicas baseadas neste modelo estão intrinsecamente relacionadas com este trabalho de pesquisa. Por exemplo, variações do Autocodificador Variacional padrão têm sido desenvolvidos com o intuito de melhorar a tarefa de sintetização de imagens ou adequá-la para outras aplicações. Pode-se citar como exemplo o *Gaussian Mixture Variational Auto-Encoder* (GMVAE), proposto por Dilokthanakul et al. [2016], que

consiste em uma extensão do Autocodificador Variacional padrão que aproxima uma mistura Gaussiana finita e uniforme com o objetivo de melhorar o agrupamento em tarefas de reconstrução de imagens. Mais especificamente, o GMVAE assume que a variável latente segue um modelo de mistura de Gaussianas, onde o coeficiente de mistura é definido *a priori* com o valor K^{-1} (mistura uniforme), onde K consiste no número de componentes presentes no modelo de mistura. Desta maneira, o modelo é simplificado para possibilitar a adoção do mesmo em um Autocodificador Variacional. Como resultado, o GMVAE é capaz de extrair com melhor qualidade informações provenientes de agrupamentos naturais existentes nas coleções de dados, tornando a tarefa de reconstrução de imagens competitiva quando comparada com outros métodos presentes na literatura. Outra conclusão interessante deste modelo é que dados complexos podem ser agrupados de forma eficiente por meio de uma mistura de distribuições cujos parâmetros são estimados por um Autocodificador Variacional.

Recentemente, Jang et al. [2016] propuseram uma adaptação do Autocodificador Variacional padrão para a tarefa de síntese de imagens, denominada *Gumbel-Softmax Variational Auto-Encoder* (GSVAE), utilizando a distribuição Gumbel-Softmax para aproximar dados categóricos. A escolha desta distribuição decorre da necessidade de reparametrização para que o cálculo dos gradientes da amostragem $z \sim q(z|x)$ seja possível. Deste modo, os autores recorrem ao *Gumbel-Max trick* [Maddison et al., 2014; Yellott, 1977], que consiste em uma técnica de amostragem para dados categóricos utilizando a distribuição Gumbel [Gumbel, 1954]. Mais especificamente, o *Gumbel-Max trick* provê uma forma eficiente de realizar a amostragem de uma distribuição categórica com probabilidades π aplicando a Equação 3.5, onde g_i é um ruído amostrado de uma distribuição Gumbel padrão ($g_i \sim Gumbel(0, 1)$), *arg max* é a função que retorna o índice do vetor cujo valor associado é máximo, e *one hot* transforma este índice em uma notação do tipo *one hot encoding* [Jang et al., 2016].

$$z = one\ hot(arg\ max_i[g_i + \log \pi_i]) \tag{3.5}$$

Visto que a função *arg max* não é diferenciável, Jang et al. [2016] aplicaram um relaxamento desta função, utilizando a função *softmax*. Logo, aplicando este método, a utilização da distribuição *Gumbel-Softmax* em Autocodificadores Variacionais torna-se possível. Em termos de resultados obtidos, os autores mostraram que a utilização da distribuição *Gumbel-Softmax* em um Autocodificador Variacional melhora significativamente a qualidade do modelo gerador em termos do valor negativo do limite inferior variacional. Além disso, eles mostraram que a abordagem proposta é competitiva em tarefas de síntese de imagens e em predição de classes em imagens contendo dígitos

escritos manualmente, considerando a métrica de acurácia.

3.5 Diferenças entre os métodos propostos e os trabalhos relacionados

Os métodos propostos neste trabalho de pesquisa, denominados GSDTM e LMDTM (cf. Capítulo 4), são baseados em Autocodificadores Variacionais. Desta forma, assim como outros modelos de tópicos baseados neste modelo, o GSDTM e o LMDTM possuem diferenças significativas em relação às outras abordagens presentes na literatura de modelagem de tópicos.

Em primeiro lugar, pode-se observar que os modelos propostos possuem menor capacidade de representação das relações de dependência entre as variáveis aleatórias, quando comparados com abordagens baseadas em modelos gráficos probabilísticos tradicionais. Uma vez que o modelo probabilístico dos Autocodificadores Variacionais é restrito à arquitetura semelhante ao de um Autocodificador, a construção de modelos probabilísticos gráficos mais ricos é impossibilitada. Entretanto, a falta de flexibilidade dos modelos propostos é compensada pela maior capacidade dos mesmos em aprender estruturas latentes presentes nos dados de entrada. Desta forma, conforme mostrado por Miao et al. [2016], modelos de tópicos baseados em Autocodificadores Variacionais tendem a obter uma métrica de perplexidade superior quando comparado ao LDA, que é um dos modelos de tópicos mais conhecidos.

Em segundo lugar, os modelos GSDTM e o LMDTM são semelhantes às abordagens baseadas em redes neurais estruturadas em modelos gráficos não direcionados, dado que estes modelos utilizam redes neurais para efetuar o treinamento. No entanto, os modelos baseados em redes neurais de linguagens geralmente utilizam redes RBM (*Restricted Boltzmann Machines*), o que restringe o tipo de distribuição que pode ser usado por estes modelos. Por outro lado, as abordagens propostas são baseadas em Autocodificadores Variacionais, o que permite a adoção de distribuições diferentes para modelar as variáveis latentes. Por exemplo, o GSDTM adota a distribuição *Gumbel-Softmax*, enquanto o LMDTM utiliza uma mistura de distribuições Normais-Logísticas.

Por último, as abordagens propostas são comparadas com outros métodos baseados em Autocodificadores Variacionais. Como o intuito de apresentar as diferenças entre os métodos de forma clara, listam-se as principais características dos mesmos na tabela 3.1. Nesta tabela, os métodos são descritos de acordo com as seguintes características: (i) aplicação do método, que pode ser voltado para modelagem de tópicos ou para a tarefa de síntese de imagens; (ii) distribuição do codificador, que indica qual a

Método	**Aplicação**	**Distribuição do codificador**	**Distribuição do decodificador**
NVDM	Modelo de Tópicos	Normal	Regressão Multinomial Logística
ProdLDA	Modelo de Tópicos	Normal-Logística	Regressão Multinomial Logística
GSVAE	Síntese de imagens	*Gumbel-Softmax*	Normal ou Bernoulli
GMVAE	Síntese de imagens	Mistura Gaussiana	Normal
LMDTM	Modelo de Tópicos	Mistura de Normais-Logísticas	Regressão Multinomial Logística
GSDTM	Modelo de Tópicos	*Gumbel-Softmax*	Regressão Multinomial Logística

Tabela 3.1: Principais diferenças entre as abordagens propostas (LMDTM e GSDTM) e trabalhos relacionados baseados em Autocodificadores Variacionais.

distribuição usada para modelar a variável latente da distribuição variacional da qual ocorre o processo de amostragem ($\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})$); e (iii) distribuição do decodificador, que descreve a distribuição ou função que realiza o processo de decodificação dos dados no Autocodificador Variacional.

As abordagens propostas estão localizadas na penúltima e última linha da tabela. Como pode ser observado, os métodos GSVAE e GMVAE possuem uma abordagem significativamente diferente do LMDTM e GSDTM, pois eles são voltados para aplicações em base de dados constituídas por imagens. Como consequência desta diferença de aplicação, a distribuição do decodificador é constituída por uma distribuição Normal ou Bernoulli para a efetuação da decodificação dos dados. Já os métodos NVDM e ProdLDA, que possuem o mesmo domínio de aplicação do LMDTM e do GSDTM, diferenciam-se das abordagens propostas em relação à distribuição do codificador. Tanto o LMDTM quanto o GSDTM utilizam distribuições diferentes das empregadas pelos métodos NVDM e ProdLDA. Entretanto, a escolha das distribuições que compõem o codificador das abordagens propostas não foi feita ao acaso. No caso do GSDTM, a aplicação da distribuição *Gumbel-Softmax*, que é capaz de aproximar dados de natureza categórica, possibilita melhor qualidade de inferência em modelagem de tópicos, visto que os elementos relacionados com esta aplicação são inerentemente categóricos. Já o método LMDTM utiliza uma mistura de distribuições Normais-Logísticas com o intuito de realizar uma inferência com maior qualidade em cenários onde os dados de entrada são complexos.

3.6 Considerações Finais

Neste capítulo foram apresentados os principais trabalhos relacionados, os quais foram categorizados em:

1. Modelos de Tópicos baseados em Modelos Gráficos Probabilísticos Tradicionais;
2. Modelos de Tópicos baseados em Redes Neurais estruturadas em modelos não direcionados;
3. Modelos de Tópicos baseados em Autocodificadores Variacionais;
4. Outros trabalhos relacionados baseados em Autocodificadores Variacionais;

Além disso, os métodos propostos foram comparados com outras abordagens presentes na literatura de modelagem de tópicos.

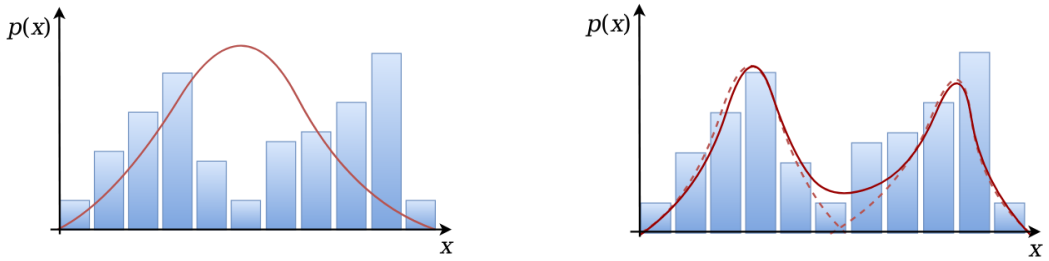
Como no NVDM e ProdLDA, neste trabalho adapta-se os Autocodificadores Variacionais para a tarefa de modelagem de tópicos. Contudo, diferentemente destes modelos, os codificadores aproximam uma distribuição *Gumbel-Softmax* ou uma mistura de distribuições Normais-Logísticas. Também se adota um número diferente de configurações de arquitetura de modo a estudar o impacto dessas escolhas na qualidade da reconstrução e na coerência dos tópicos obtidas pelos modelos. No próximo capítulo, as abordagens propostas serão abordadas com maior nível de detalhes.

4

Métodos propostos

Neste capítulo são apresentados dois modelos de tópicos baseados em Autocodificadores Variacionais. O primeiro método proposto denomina-se *Logistic-Normal Mixture Document Topic Model* (LMDTM). Este método tem como principal característica o uso de um modelo de mistura de distribuições Normais-Logísticas (MMNN) para representar a distribuição da variável latente, que é responsável por representar a estrutura de tópicos proveniente de uma coleção de dados. O objetivo em adotar o modelo de mistura consiste em representar melhor dados textuais que são significativamente complexos. O segundo método, denominado *Gumbel-Softmax Document Topic Model* (GSDTM), propõe o uso da distribuição *Gumbel-Softmax* (GS) como distribuição probabilística da variável latente. O uso desta distribuição é capaz de melhorar a qualidade da modelagem de tópicos, uma vez que ela é capaz de aproximar a distribuição de dados categóricos (e.g., palavras, documentos, tópicos e outras entidades) em uma rede de Autocodificador Variacional, com melhor qualidade quando comparado com outras distribuições de probabilidades [Jang et al., 2016].

O capítulo é organizado como se segue. A seção 4.1 descreve o método LMDTM com detalhes, mostrando como o modelo probabilístico está estruturado e como o truque de reparametrização para a distribuição de mistura de Normais-Logísticas é realizado, além de descrever a arquitetura de rede neural utilizada no método proposto. Depois, a seção 4.2 expõe as particularidades do método GSDTM, seguindo o mesmo processo utilizado para descrever o método LMDTM. Logo após, o processo de treino de ambos os métodos é relatado na seção 4.3. Por fim, na seção 4.4 fazem-se algumas considerações finais referentes às abordagens propostas.



(a) Distribuição única.

(b) Mistura de duas distribuições.

Figura 4.1: Ilustração da vantagem do modelo de mistura em relação a uma única distribuição. Pode-se observar que o modelo de mistura presente em (b) é capaz de realizar uma aproximação melhor da distribuição dos dados do que uma distribuição simples (a).

4.1 Logistic-Normal Mixture Document Topic Model (LMDTM)

O LMDTM (***L**ogistic-Normal **M**ixture **D**ocument **T**opic **M**odel*) é baseado na ideia de utilizar uma combinação linear de distribuições com o intuito de realizar uma inferência sobre os dados com maior verossimilhança quando comparado com o processo de inferência proveniente de uma única distribuição. De fato, Bishop [2006] afirma que um modelo de mistura de distribuições geralmente se ajusta melhor a dados complexos. Por exemplo, a Figura 4.1 ilustra como um modelo de mistura pode se adequar melhor quando os dados estão distribuídos de forma mais complexa. Mais especificamente, a ilustração localizada à esquerda apresenta uma única distribuição que não é capaz de aproximar a distribuição dos dados, representada pelo conjunto de retângulos azuis. Entretanto, se um modelo de mistura formada por duas distribuições for empregado, a aproximação torna-se mais factível em relação à distribuição dos dados.

Assim como o GMVAE (c.f Seção 3.4), este método aplica um modelo de mistura de distribuições em Autocodificadores Variacionais. Entretanto, ao invés de usar uma distribuição Gaussiana, foi usado uma mistura de distribuições Normais-Logísticas (NL). Segundo Aitchison & Shen [1980], a Normal-Logística é uma distribuição resultante da transformação logística aplicada a uma distribuição Normal. Mais formalmente, considerando $\mathbf{v} \sim \mathcal{N}(\mu, \Sigma)$ uma amostra de uma distribuição Normal multivariada com d dimensões, pode-se obter amostras da distribuição Normal-Logística aplicando a seguinte transformação logística (c.f Equação 4.1) [Huang & Malisiewicz, 2009]:

$$\mathbf{u} = \frac{\exp \mathbf{v}}{\sum_{j=1}^{d} \exp\, v_j} \tag{4.1}$$

A função de densidade de probabilidade para $\mathbf{u}$, representada graficamente na Figura 4.2, pode ser escrita conforme mostrado na Equação 4.2 [Aitchison & Shen, 1980]:

$$p(\mathbf{u}, \mu, \Sigma) = |2\pi\Sigma|^{-\frac{1}{2}} \left(\prod_{j=1}^{d} u_j \right)^{-1} \exp \left[-\frac{1}{2} \left\{ \log \left(\frac{\mathbf{u}}{u_d} \right) - \mu \right\}^T \Sigma^{-1} \left\{ \log \left(\frac{\mathbf{u}}{u_d} \right) - \mu \right\}^T \right] \tag{4.2}$$

onde

$$u_d = 1 - \sum_{j=1}^{d-1} u_j \tag{4.3}$$

Visto que Srivastava & Sutton [2017] mostraram que o uso de distribuições Normais-Logísticas geralmente melhora a coerência dos tópicos, adotou-se um modelo de mistura desta distribuição para compor o método LMDTM. Assim, o uso deste modelo pode ser uma melhor forma de aproximar a distribuição *a posteriori* real e, consequentemente, pode melhorar a tarefa de modelagem de tópicos.

Nas seções seguintes, apresentar-se-á o modelo probabilístico do LMDTM, bem como o truque de reparametrização e a arquitetura de rede neural empregada neste método.

4.1.1 Modelo probabilístico

O modelo probabilístico do LMDTM pode ser descrito como se segue. Considere um modelo gerador $p_\theta(\mathbf{x}, \mathbf{y}, \mathbf{z})$, cuja distribuição conjunta pode ser denotada por $p(\mathbf{y})p_\theta(\mathbf{z}|\mathbf{y})p_\theta(\mathbf{x}|\mathbf{z}, \mathbf{y})$, onde $\mathbf{x}$ é uma amostra observada gerada a partir de um conjunto de variáveis latentes $\mathbf{y}$ e $\mathbf{z}$, e θ representa o conjunto de parâmetros da rede neural usada para estimar os parâmetros das distribuições $p_\theta(\mathbf{z}|\mathbf{y})$ e $p_\theta(\mathbf{x}|\mathbf{z}, \mathbf{y})$. O processo ge-

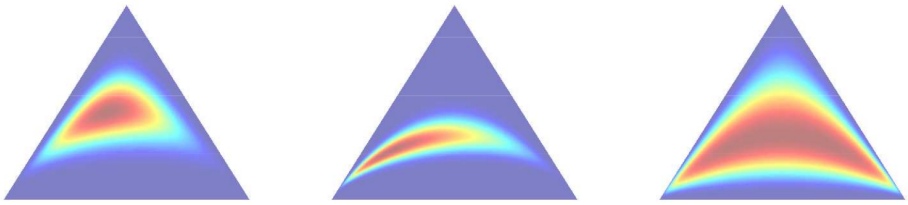


Figura 4.2: Gráfico de distribuições Normais-Logísticas para várias configurações de parâmetros. Figura extraída do trabalho de Huang & Malisiewicz [2009].

rador das variáveis latentes $\mathbf{y}$ e $\mathbf{z}$, e da variável observável $\mathbf{x}$ é definido respectivamente pelas Equações 4.4, 4.5 e 4.6, onde μ_{y_k} e $\sigma^2_{y_k}$ correspondem aos parâmetros de média e variância da k-ésima componente da mistura de distribuições Normais-Logísticas, representada pela notação $\mathcal{LN}$:

$$\mathbf{y} \sim Multinomial(\pi) \tag{4.4}$$

$$\mathbf{z} \sim \prod_k \mathcal{LN}(\mu_{y_k}, \sigma^2_{y_k})^{y_k} \tag{4.5}$$

$$\mathbf{x} \sim softmax(\mathbf{W}\mathbf{z} + \mathbf{b}) \tag{4.6}$$

A Equação 4.4 inicia o processo gerador escolhendo uma das K componentes do modelo de mistura. Esta escolha segue uma distribuição multinomial, com probabilidades de mistura $\pi = (\pi_1, \pi_2, \cdots, \pi_K)$, que determina a probabilidade de cada componente ser escolhida. Em outras palavras, π representa o peso de cada componente no modelo de mistura. Com o intuito de simplificar o modelo, considera-se $\mathbf{z}$ uniformemente distribuído, assumindo que $\pi = K^{-1}$. Ou seja, o LMDTM adota um modelo de mistura onde cada componente possui o mesmo peso. Desta forma, viabiliza-se a construção das redes neurais responsáveis por estimar os parâmetros das distribuições do modelo. A escolha da componente do modelo de mistura é representada pela variável categórica $\mathbf{y} = (y_1, y_2, \cdots, y_K)$. Esta variável é do tipo *one-hot*, ou seja, consiste de um vetor com K posições, onde uma das posições possui valor equivalente a um e as demais posições contém valor zero. Desta forma, se a primeira componente for escolhida, o valor de $\mathbf{y}$ será $(1, 0, \cdots, 0)$. Caso a segunda componente seja escolhida, o valor será $(0, 1, \cdots, 0)$, e assim por diante. Por fim, a variável latente $\mathbf{z}$ é gerada por meio da amostragem da componente Normal-Logística escolhida pela distribuição Multinomial, conforme mostrada na equação 4.5.

A distribuição $p_\theta(\mathbf{x}|\mathbf{z}, \mathbf{y})$ denota uma regressão multinomial logística com parâmetros constituídos pela matriz de pesos $\mathbf{W}$ e pelo vetor de viés $\mathbf{b}$, conforme mostrado pela Equação 4.6. Deste modo, o método LMDTM transforma os dados latentes amostrados do modelo de mistura de distribuições Normais-Logísticas em dados observáveis, representados pela variável $\mathbf{x}$. Em outras palavras, a regressão multinomial logística aprende como gerar os dados de entrada, constituídos por frequências entre termos e documentos, por meio das relações latentes de tópicos geradas pelo modelo de mistura.

Contudo, devido ao fato do cálculo da probabilidade conjunta não ser tratável, adota-se um processo de inferência baseado em inferência variacional de campo médio. Mais especificamente, o processo de inferência no modelo LMDTM é efetuado usando

uma distribuição variacional simples $q_\Phi(\mathbf{x}, \mathbf{y}, \mathbf{z}) = \prod_y q_\Phi(\mathbf{y}|\mathbf{x}) q_\Phi(\mathbf{z}|\mathbf{x}, \mathbf{y})$, onde $q_\Phi(\mathbf{y}|\mathbf{x})$ e $q_\Phi(\mathbf{z}|\mathbf{x}, \mathbf{y})$ seguem respectivamente as distribuições Multinomial e Normal-Logística. O modelo variacional do LMDTM possui K distribuições Normais-Logísticas independentes, cada uma denotada pela variável multinomial $\mathbf{y}$. Deste modo, cada uma das distribuições funcionam como componentes individuais da mistura de Normais-Logísticas. Esta modelagem pode permitir que o processo de inferência variacional aproxime melhor uma distribuição do modelo gerador complexo quando comparado com outros modelos que utilizam apenas uma componente Normal-Logística. Seguindo o processo padrão do processo de inferência (c.f. Seção 2.8.4), encontra-se o limite inferior variacional, conforme indicado na Equação 4.7, usando a divergência Kullback-Leibler (KL) entre as distribuições q_Φ e p_θ.

$$\begin{aligned} log p_\theta \geq -\mathcal{U}(\mathbf{x}) = \frac{1}{K}\mathbb{E}_{z\sim q_\theta(\mathbf{y},\mathbf{z}|\mathbf{x})}[log p_\theta(\mathbf{x}|\mathbf{y},\mathbf{z})] - \\ D_{KL}[q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y})||p_\theta(\mathbf{z}|\mathbf{y})] - D_{KL}[q_\phi(\mathbf{y}|\mathbf{x})||p(\mathbf{y})] \end{aligned} \tag{4.7}$$

Seguindo a abordagem usada no ProdLDA [Srivastava & Sutton, 2017], usa-se a aproximação da divergência KL entre distribuições Normais-Logísticas, que consiste na divergência KL definida para distribuições Normais. Deste modo, $D_{KL}[q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{y})||p_\theta(\mathbf{z}|\mathbf{y})]$ pode ser computada analiticamente sob a forma fechada. Considere que μ_k e σ_k sejam respectivamente a média e o desvio-padrão da k-ésima componente do modelo de mistura de Normais-Logísticas proveniente da distribuição variacional $q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{y})$, enquanto que $\mu_{\rho k}$ e $\sigma_{\rho k}$ sejam os parâmetros da k-ésima componente do modelo de mistura proveniente da distribuição geradora $p_\theta(\mathbf{z}|\mathbf{y})$. Logo, a divergência KL pode ser definida pela equação abaixo (4.8):

$$D_{KL}[q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{y})||p_\theta(\mathbf{z}|\mathbf{y})] = \log\frac{\sigma_k}{\sigma_{\rho k}} + \frac{(\mu_{\rho k} - \mu_k)^2 + \sigma_{\rho k}^2}{2\sigma_k^2} - \frac{1}{2} \tag{4.8}$$

Uma vez que $\mathbf{y}$ é uma variável discreta, calcula-se $D_{KL}[q_\phi(\mathbf{y}|\mathbf{x})||p(\mathbf{y})]$ usando a definição de divergência KL para distribuições discretas, de acordo com a Equação 4.9.

$$D_{KL}[q_\phi(\mathbf{y}|\mathbf{x})||p(\mathbf{y})] = -\sum q_\phi(\mathbf{y}|\mathbf{x}) \log\left[\frac{q_\phi(\mathbf{y}|\mathbf{x})}{p(\mathbf{y})}\right] \tag{4.9}$$

Entretanto, a distribuição *a priori* $p(\mathbf{y})$ é definida como uniforme, ou seja, $p(\mathbf{y}) = K^{-1}$. Desta forma, pode-se rescrever a Equação 4.9 da seguinte forma:

$$D_{KL}[q_\phi(\mathbf{y}|\mathbf{x})||p(\mathbf{y})] = -\sum q_\phi(\mathbf{y}|\mathbf{x}) \left[\log q_\phi(\mathbf{y}|\mathbf{x}) + \log(K)\right] \tag{4.10}$$

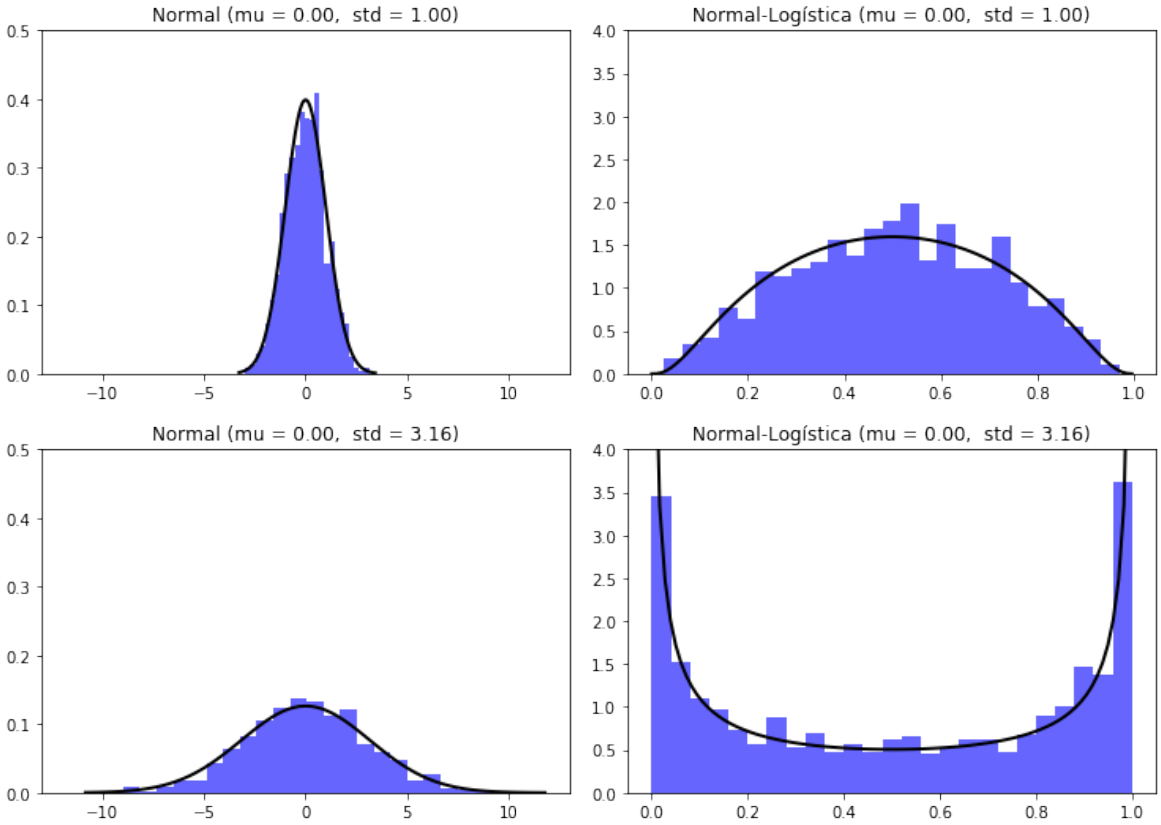


Figura 4.3: Histograma comparativo entre as amostras provenientes do truque de reparametrização de distribuições Normais e do truque de reparametrização de distribuições Normais-Logísticas. Para a elaboração destes histogramas, foram geradas 1000 amostras.

Por fim, a entropia de informação $\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} log p_\theta(\mathbf{x}|\mathbf{y},\mathbf{z})$ é estimada usando a entropia cruzada definida para a regressão multinomial logística. A fórmula da entropia cruzada é mostrada na Equação 4.11, onde V é o tamanho do vocabulário de termos e $p_\theta(\mathbf{x}_j|\mathbf{y},\mathbf{z})$ é a probabilidade do termo x_j dada pela regressão multinomial logística (c.f. Seção 4.1.3).

$$\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}[\sum_{j=1}^{V} log p_\theta(\mathbf{x}_j|\mathbf{y},\mathbf{z})] = -\sum_{j=1}^{V} \mathbf{x}_j log p_\theta(\mathbf{x}_j|\mathbf{y},\mathbf{z}) \tag{4.11}$$

O objetivo desta equação é mensurar a capacidade de reconstrução do modelo. Mais formalmente, a entropia cruzada visa estimar o quão próximo a distribuição dos dados reconstruídos pelo modelo são em comparação com os dados reais fornecidos como entrada do modelo. Desta forma, um dos objetivos do modelo é minimizar essa diferença entre as distribuições, produzindo uma matriz de frequência dos termos em cada documento próxima da fornecida como entrada de dados.

4.1.2 Truque de reparametrização

O modelo LMDTM aplica o truque de reparametrização com uma transformação por meio da função *softmax* para cada componente da mistura de Normais-logísticas, adotando uma abordagem similar ao de Srivastava & Sutton [2017]. Assim, a variável contínua $\mathbf{z}$ é dada pela equação $softmax(\mu_k + \epsilon\sigma_k)$, onde μ_k é a média e σ_k é o desvio-padrão da k-ésima componente Normal-Logística presente no modelo de mistura. Assim como nos Autocodificadores Variacionais onde a variável latente segue uma distribuição Normal, a variável não determinística ϵ representa um ruído aleatório amostrado de uma distribuição Normal padrão, ou seja, com média equivalente ao valor 0 e variância com valor 1 ($\epsilon \sim \mathcal{N}(0, 1)$). Na Figura 4.3, pode-se observar que embora a diferença entre a equação do truque de reparametrização de distribuições Normais e de Normais-Logísticas seja relativamente pequena, a distribuição das amostras difere consideravelmente. Pode-se observar que diferentemente da distribuição Normal, a Normal-Logística produz amostras restritas em um intervalo entre 0 e 1, além de apresentar um comportamento mais maleável do que a Normal, principalmente quando o desvio-padrão possui um valor consideravelmente elevado.

4.1.3 Estrutura dos dados de entrada e arquitetura do LMDTM

Diferentemente dos Autocodificadores Variacionais padrões, a entrada de dados fornecida à rede é uma representação vetorial do tipo *bag-of-words*, onde cada vetor representa um documento da coleção de dados. Embora usada em quase a totalidade dos modelos de tópicos baseados em Autocodificadores Variacionais, esta representação para redes neurais foi proposta no trabalho de Salakhutdinov & Hinton [2009], como uma forma de permitir a modelagem de documentos em modelos geradores. A rede também inclui vetores do tipo *one-hot* representando cada componente do modelo de mistura de Normais-Logísticas, como ilustrado na Figura 4.4. Mais especificamente, os documentos são denotados por um conjunto de vetores $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_N\}$, onde cada $\mathbf{x}_i$ é associado com um documento d_i. O vetor $\mathbf{x}_i \in \mathcal{R}^V$ representa o vetor de frequência de palavras onde cada palavra pertence a um vocabulário de tamanho V. O vetor do tipo *one-hot* $\mathbf{Y}$ representa o discriminante da componente Normal-Logística. Em outras palavras, este vetor indica qual componente da mistura de Normais-Logísticas está sendo estimada pela rede. A concatenação de $\mathbf{X}$ e $\mathbf{Y}$ é a entrada de dados para a rede de inferência.

A fim de maximizar o valor do limite inferior variacional no LMDTM, todos

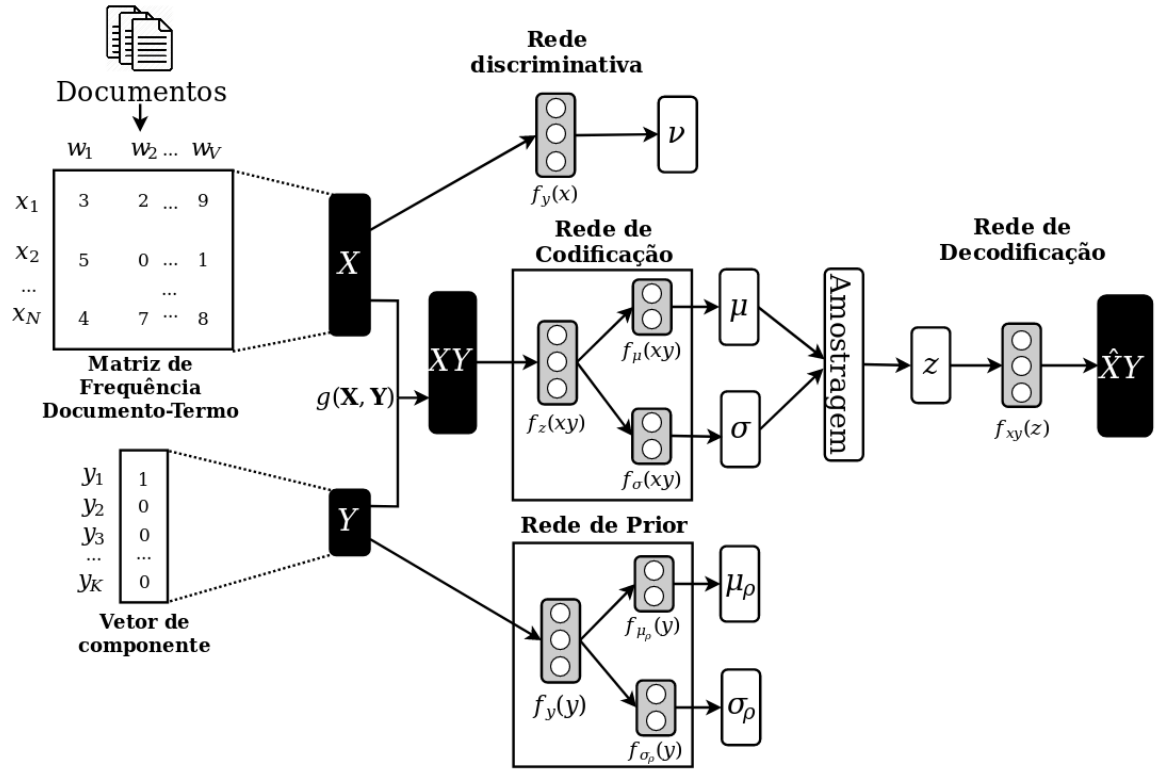


Figura 4.4: Arquitetura de uma rede LMDTM. Retângulos negros representam entrada e saída de dados, enquanto retângulos cinzas representam redes neurais e retângulos brancos indicam parâmetros e variáveis do modelo. A função $g(\mathbf{X}, \mathbf{Y})$ concatena $\mathbf{X}$ a $\mathbf{Y}$.

os parâmetros de todas as distribuições de probabilidade são treinadas usando uma rede de Autocodificador Variacional, ao invés de usar métodos tradicionais Bayesianos tal como o método de Esperança-Maximização. No modelo LMDTM, a rede neural compreende três sub-redes neurais. A primeira é uma rede discriminativa $f_y(\mathbf{x})$ que possui uma função de ativação realizada pela função *softmax* e aprende o parâmetro ν da distribuição Multinomial $q_\Phi(\mathbf{y}|\mathbf{x})$. Desta forma, esta rede determina a classe dos documentos (tópicos) dado a variável observável $\mathbf{x}$. A segunda sub-rede é a principal do método LMDTM, que compreende a rede de codificação (*encoder*) e a rede de decodificação (*decoder*). A primeira é representada por $f_z(xy)$, onde xy é uma notação abreviada para $g(\mathbf{X}, \mathbf{Y})$, função esta usada para concatenar $\mathbf{X}$ a $\mathbf{Y}$. A segunda é denotada por $f_{xy}(z)$ e é responsável por decodificar os dados latentes em $\mathbf{z}$ em uma matriz de frequência documento-termo reconstruída e denotada por $\mathbf{X\hat{Y}}$. Por fim, a terceira sub-rede é nomeada de rede de *prior*, tendo como principal função estimar os parâmetros da distribuição Normal-Logística *a priori* com base nos valores presentes no vetor de componentes $\mathbf{Y}$.

Na rede de codificação, os vetores de média (μ) e os vetores de desvio-padrão (σ) correspondente a cada componente da mistura de Normais-Logísticas denotada por $q_\theta(\mathbf{y}, \mathbf{z}|\mathbf{x})$ são computados respectivamente pelas redes neurais $f_\mu(xy) = l(f_z(xy))$ e $f_\sigma(xy) = l(f_z(xy))$, onde $l(.)$ indica que não há funções de ativação (*e.g*, função *softmax* ou $ReLU$). Deste modo, o truque de reparametrização é aplicado nestes vetores e as amostras $\mathbf{z}$ são obtidas.

Enquanto Autocodificadores Variacionais geralmente usam distribuições Normais ou Binomiais em redes de decodificação, o LMDTM segue a abordagem de Miao et al. [2016] e adota uma regressão multinomial logística, denotada pela distribuição $p_\theta(\mathbf{x}|\mathbf{y}, \mathbf{z})$, para reconstruir os dados de entrada $\mathbf{X}$, conforme mostrado na Equação 4.12, onde $\mathbf{W}$ é o parâmetro de peso da regressão multinomial logística capaz de representar um *embedding* de palavras em um espaço vetorial $\mathcal{R}^{TxV}$ compartilhado entre os documentos, onde T é o número de tópicos e $\mathbf{b}$ é o vetor de *bias*:

$$p_\theta(x|y, z) = \frac{exp(\mathbf{W}\mathbf{z}_i + \mathbf{b}_i)}{\sum_{j=1}^{k} exp(\mathbf{W}\mathbf{z}_j + \mathbf{b}_j)} \tag{4.12}$$

Por fim, a média e o desvio-padrão de cada componente da mistura de Normais-Logísticas *a priori* $p_\theta(\mathbf{z}|\mathbf{y})$ são computados respectivamente pelas redes neurais $f_{\mu_\rho}(y) = l(f_y(y))$ e $f_{\sigma_\rho}(y) = l(f_y(y))$, onde $l(f_y(y))$ é uma rede totalmente conectada onde a última camada não possui funções de ativação.

4.2 Gumbel-Softmax Document Topic Model (GSDTM)

Tópicos podem ser vistos como categorias de palavras e documentos que estão semanticamente associados a um ou mais temas. Desta forma, eles podem ser melhor modelados por meio de distribuições probabilísticas categóricas. De acordo com Bishop [2006], uma distribuição categórica, também conhecida como multinomial, é uma distribuição probabilística capaz de modelar variáveis discretas que podem tomar um dentre K estados possíveis. No contexto de modelagem de tópicos, um estado configura-se em um tópico. Logo, a utilização de distribuições categóricas permite uma representação mais verossímil da distribuição de dados, o que possibilita a geração de tópicos mais compreensíveis de acordo com as observações provenientes de um julgador humano [Chen et al., 2016; Rae et al., 2016]. De fato, pode-se observar que modelos de tópicos baseados em modelos probabilísticos gráficos, como o LDA, geralmente assumem que tópicos são amostrados de distribuições multinomiais, objetivando a melhor

representação possível da distribuição de um tópico.

Entretanto, devido ao fato de que distribuições categóricas geram amostras discretas, torna-se difícil o desenvolvimento de um truque de reparametrização para elas. Deste modo, os modelos de tópicos baseados em Autocodificadores Variacionais adotam, em sua grande maioria, uma variável latente que segue uma distribuição Normal ou Normal-Logística. Contudo, a distribuição de dados sobre elementos categóricos dificilmente consegue ser bem aproximada por meio destas distribuições contínuas.

Neste contexto, uma distribuição contínua interessante que pode ser utilizada em Autocodificadores Variacionais como forma de mitigar este problema é a *Gumbel-Softmax*(GS) [Jang et al., 2016]. De modo simplificado, esta é uma distribuição contínua baseada na distribuição de Gumbel e que pode se ajustar mais adequadamente a dados que sejam de natureza categórica. Desta forma, é possível propor um truque de reparametrização para esta distribuição. Isto é realizado por meio de uma modificação do método conhecido como truque Gumbel-Max [Luce, 1959; Maddison et al., 2014], um artifício estatístico que utiliza a função *max* (função que retorna o valor máximo presente em uma lista de itens numéricos) sobre a distribuição Gumbel para gerar amostras. Como a função *max* não é reparametrizável, Jang et al. [2016] propuseram uma modificação da distribuição de Gumbel denominada *Gumbel-Softmax*, que utiliza uma aproximação contínua da função *max* por meio da função *softmax*, para gerar amostras. Logo, o truque de reparametrização é possibilitado, uma vez que a função de geração de amostras é contínua.

Neste contexto, propõe-se o método baseado em Autocodificadores Variacionais denominado ***G**umbel-**S**oftmax **D**ocument **T**opic **M**odel* (GSDTM). Este método emprega a *Gumbel-Softmax* como distribuição da variável latente, de modo que o processo de aprendizagem de elementos categóricos como tópicos, documentos e palavras possa ser mais efetivo e, deste modo, melhores tópicos possam ser extraídos. A distribuição *Gumbel-Softmax* pode ser descrita como se segue. Considere que a variável discreta $\mathbf{y}$, responsável pela representação de tópicos, seja amostrada de uma distribuição multinomial com probabilidades $\pi = \{\pi_1, \cdots, \pi_T\}$, tal que $\sum_i \pi_i = 1$. A distribuição *Gumbel-Softmax* assume que as amostras categóricas $\mathbf{y}$ são codificadas como um vetor com T dimensões do tipo *one-hot*, representados como vértices do simplex Δ^{T-1}. Mais especificamente, este simplex consiste em um espaço vetorial com $T-1$ dimensões formado pelos possíveis valores de π [Bishop, 2006]. A densidade da distribuição *Gumbel-Softmax* [Jang et al., 2016] é denotada pela Equação 4.13, onde o parâmetro τ, referido como a temperatura da função *softmax*, é usado para controlar a discretização da distribuição de probabilidade sobre $\mathbf{y}$ e T é o número de dimensões e de tópicos da distribuição *Gumbel-Softmax*.

$$p_{\pi,\tau}(\mathbf{y}) = \Gamma(T)\tau^{T-1}(\sum_{i=1}^{T}\pi_i/\mathbf{y}_i^{\tau})^{-T}\prod_{i=1}^{T}(\pi_i/\mathbf{y}_i^{\tau+1}) \tag{4.13}$$

Para temperaturas baixas, as amostras provenientes da distribuição *Gumbel-Softmax* tornam-se próximas a uma configuração do tipo *one-hot*, enquanto que para altas temperaturas elas se tornam cada vez mais próximas de distribuição uniformemente categórica, onde a probabilidade de escolha é igual para todas as categorias. Mais especificamente, valores altos das probabilidades π_i fazem com que a massa de probabilidade da *Gumbel-Softmax* tenda a estar concentrada em direção dos vértices $\mathbf{y}_i$ do simplex.

4.2.1 Modelo probabilístico

O modelo GSDTM usa uma distribuição *Gumbel-Softmax* para modelar a variável latente de tópicos $\mathbf{z}$. Adicionalmente, o modelo probabilístico incorpora a variável $\mathbf{y}$, que segue uma distribuição multinomial, com o intuito de modelar o parâmetro π utilizado pela distribuição *Gumbel-Softmax*, enquanto que os dados observados $\mathbf{x}$ seguem um modelo de regressão multinomial logística. A fórmula da distribuição conjunta de probabilidades é dada pela Equação 4.14:

$$p_\theta(\mathbf{x}, \mathbf{y}, \mathbf{z}) = \sum_y \sum_z p(\mathbf{y})p(\mathbf{z})p_\theta(\mathbf{x}|\mathbf{z}, \mathbf{y}) \tag{4.14}$$

A grande vantagem da distribuição *Gumbel-Softmax* é a propriedade de aproximar melhor sobre valores categóricos em comparação com a maioria das distribuições contínuas existentes na literatura. Consequentemente, a codificação dos documentos no espaço de tópicos pode ser melhorada, já que o tópico é uma entidade inerentemente categórica. Além disso, a distribuição *Gumbel-Softmax* tem um processo de amostragem simples e eficiente, o que é benéfico para o treinamento em vastas coleções de dados. Outra vantagem é que a distribuição *Gumbel-Softmax* produz estimadores enviesados de gradientes com baixa variância [Maddison et al., 2016], o que aumenta a robustez do modelo estatístico.

A distribuição conjunta do GSDTM possui algumas diferenças quando comparadas com o modelo LMDTM que são importantes de serem abordadas. Devido ao fato de que a *Gumbel-Softmax* não ser uma distribuição de misturas, a variável latente $\mathbf{z}$ não é condicionada à variável discreta $\mathbf{y}$. Deste modo, as probabilidades $p(\mathbf{y})$ e $p(\mathbf{z})$ são computadas de forma independente, o que permite uma grande simplificação do modelo.

O método GSDTM adota o mesmo processo de inferência variacional do LMDTM. Assim, seguindo a abordagem descrita na Seção 4.1, o limite inferior variacional pode ser denotado por:

$$\begin{aligned} logp_\theta \geq -\mathcal{U}(\mathbf{x}) = \mathbb{E}_{\mathbf{z}\sim q_\theta(\mathbf{y},\mathbf{z}|\mathbf{x})}[logp_\theta(\mathbf{x}|\mathbf{y},\mathbf{z}) \\ + logp(\mathbf{y}) + logp(\mathbf{z}) - q_\phi(\mathbf{y},\mathbf{z}|\mathbf{x})] \end{aligned} \tag{4.15}$$

Uma vez que a variável $\mathbf{y}$ não é observável e o problema de modelagem de tópicos não é supervisionado, segue-se a abordagem proposta por Kingma & Welling [2013], onde a variável $\mathbf{y}$ passa pelo processo de marginalização. Assim, o limite inferior variacional pode ser redefinido na seguinte equação (Equação 4.16):

$$logp_\theta \geq -\mathcal{U}(\mathbf{x}) = \mathbb{E}_{\mathbf{z}\sim q_\theta(\mathbf{z}|\mathbf{x})}[logp_\theta(\mathbf{x}|\mathbf{z})] - D_{KL}[q_\phi(\mathbf{z}|\mathbf{x})||logp(\mathbf{z})] \tag{4.16}$$

Desta forma, o limite inferior variacional do GSDTM tem a mesma estrutura quando comparado ao Autocodificador Variacional padrão, conforme pode ser visto na Seção 2.8.4. Assim como no LMDTM, a distribuição do decodificador ($p_\theta(\mathbf{x}|\mathbf{z})$) segue uma regressão multinomial logística. Em relação à divergência KL, segue-se a abordagem de Jang et al. [2016] e utiliza-se a divergência KL voltada para distribuição categórica ao invés da divergência KL entre distribuições *Gumbel-Softmax*. Esta simplificação é justificada pelo fato da distribuição *Gumbel-Softmax* ser uma aproximação da distribuição Multinomial e pelo fato do cálculo da divergência KL categórica ser mais eficiente do que a divergência KL entre distribuições *Gumbel-Softmax*. Logo, a distribuição $q_\phi(\mathbf{z}|\mathbf{x})$ segue uma distribuição Multinomial cujo parâmetro ν é calculado a partir do valor presente na variável $\mathbf{x}$. Já a distribuição $p(\mathbf{z})$ segue uma distribuição Multinomial *a priori*, cujo valor é uniformemente distribuído e equivale a T^{-1}, onde T é o número de tópicos.

4.2.2 Truque de reparametrização

Diferentemente do Autocodificador Variacional padrão, onde a variável latente está condicionada a uma distribuição Normal, o modelo GSDTM utiliza uma distribuição *Gumbel-Softmax* para gerar amostras provenientes da variável latente $\mathbf{z}$. Para realizar este processo, o GSDTM adota o truque de reparametrização voltado para a distribuição *Gumbel-Softmax*, que consiste na suavização contínua do truque *Gumbel-Max*. Por sua vez, o *Gumbel-Max* é um método que gera amostras de uma distribuição categórica

com probabilidades de classes π, conforme mostrado na Equação abaixo (4.17):

$$\mathbf{z} = \text{one\_hot}(\arg\max_i[\mathbf{g}_i + log\pi_i]) \tag{4.17}$$

onde $g_1, g_2, \cdots, g_k$ são amostras geradas de $Gumbel(0, 1)$. Pode-se realizar este processo usando o método da transformação inversa, por meio da amostragem $\mathbf{U} \sim Uniform(0, 1)$ e computando $\mathbf{g} = -log(-log(\mathbf{U}))$ [Maddison et al., 2016]. Em razão da função de probabilidade não ser diferenciável, o truque de reparametrização voltado para a distribuição *Gumbel-Softmax* substitui a função *argmax* pela versão suavizada e contínua, a *softmax*, conforme mostrado na Equação 4.18:

$$\mathbf{z}_i = \frac{exp((\mathbf{g}_i + log\pi_i)/\tau)}{\sum_{j=1}^{k} exp((\mathbf{g}_j + log\pi_j)/\tau)}, i = 1, \cdots, k \tag{4.18}$$

onde τ é o parâmetro que representa a temperatura do modelo. Desta forma, tem-se um truque de reparametrização eficiente e flexível, capaz de gerar amostras com distribuição de probabilidade próxima de uma distribuição categórica.

4.2.3 Arquitetura do método GSDTM

Assim como o método LMDTM, o GSDTM adota uma matriz de frequência documento-termo como entrada de dados do Autocodificador Variacional, representada pela variável $\mathbf{X}$. A arquitetura desta rede neural é ilustrada com detalhes na Figura 4.5. Como observado na figura, a representação dos dados de entrada é codificada pela rede de inferência $f_z(\mathbf{x})$, composta por uma rede neural de duas camadas com funções de ativação do tipo *ReLU*. A saída da rede de inferência conecta-se com outra camada de rede neural, representada por π_x e denominada de rede discriminativa.

Assim como a rede discriminativa do LMDTM, esta rede possui uma camada e não contém funções de ativação, sendo responsável por estimar a proporção de tópicos presente na coleção de dados utilizando a distribuição Multinomial. Esta rede gera dois conjuntos de representações não normalizadas, nomeados de *logits*. Cada *logit* é normalizado por meio da aplicação da função *softmax*, gerando respectivamente dois vetores de probabilidades, um representado por ν e outro por π. O vetor ν é utilizado como parâmetro da distribuição Multinomial $q_\phi(\mathbf{y}|\mathbf{x})$. Por outro lado, o vetor de probabilidades π é o parâmetro da distribuição $q_\phi(\mathbf{z}|\mathbf{x})$, que segue uma distribuição *Gumbel-Softmax*. Em seguida, as amostras $\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x})$ são obtidas via truque de reparametrização da distribuição *Gumbel-Softmax* usando os parâmetros π e τ, conforme descrito na Equação 4.18. Diferentemente de π, o parâmetro de temperatura τ não é aprendida pela rede do modelo. Ao invés disso, define-se um valor inicial de temperatura e aplica-se o

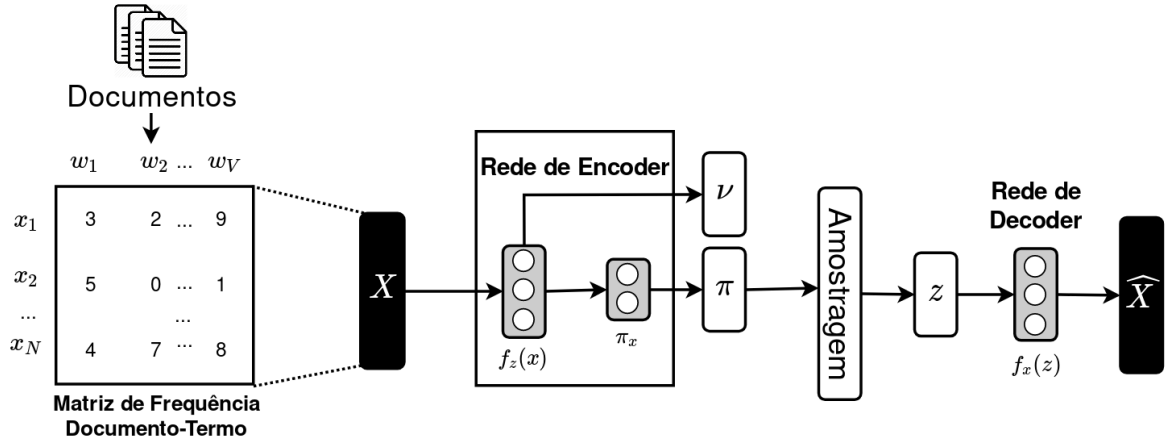


Figura 4.5: Arquitetura do método GSDTM. O parâmetro π corresponde à probabilidade de cada tópico, utilizado no processo de amostragem da distribuição *Gumbel-Softmax*.

processo de ajuste gradual conhecido como *simulated annealing* ao longo das iterações de treinamento, conforme a abordagem proposta por Jang et al. [2016]. Assim como no método LMDTM, o modelo GSDTM adota uma regressão multinomial logística para a distribuição de reconstrução $p_\theta(\mathbf{x}|\mathbf{z})$. Uma rede neural $f_x(\mathbf{z})$, constituída de uma camada sem funções de ativação (uma vez que esta rede tem como objetivo estimar os parâmetros da regressão multinomial logística) e com parâmetros $\mathbf{W}$ (matriz de pesos) e $\mathbf{b}$ (viés), decodifica as amostras geradas de $\mathbf{z}$ em amostras reconstruídas $\hat{\mathbf{X}}$. De forma equivalente ao método LMDTM, a matriz de peso $\mathbf{W}$ representa um *embedding* de palavras em um espaço vetorial $\mathcal{R}^{TxV}$, onde V é o tamanho do vocabulário e T é o número de tópicos.

4.3 Treino via SGVB

Assim como em outros modelos baseados em Autocodificadores Variacionais, o treinamento dos métodos LMDTM e GSDTM é efetuado via *Stochastic Gradient Variational Bayes* (SGVB) [Kingma & Welling, 2013]. Este método consiste em possibilitar a diferenciação do limite inferior variacional através da aplicação do truque de reparametrização (c.f Seção 2.8.4). Desta forma, é possível realizar o treinamento da rede por meio de técnicas de otimização baseadas em gradiente estocástico, de forma similar a outras arquiteturas redes neurais.

Em virtude da possibilidade do treinamento ser efetuado por meio da otimização do gradiente, pode-se estimá-lo de forma eficiente utilizando um tamanho fixo de M amostras provenientes da base de treinamento, em vez de utilizar todo o conjunto de

Algoritmo 1: Algoritmo de otimização em Autocodificadores Variacionais utilizando *minibatches*. Adaptado de Kingma & Welling [2013].

$\theta, \phi \leftarrow$ Inicialize os parâmetros
repita
  $\mathbf{X}^M \leftarrow$ *Minibatch* com M amostras aleatórias extraídas da base de treino
  $\epsilon \leftarrow$ Amostras aleatórias da distribuição de ruído $p(\epsilon)$
  $g \leftarrow \nabla_{\theta,\phi}\tilde{\mathcal{L}}^M(\theta, \phi, \mathbf{X}^M, \epsilon, \tau)$ (gradiente do limite inferior variacional)
  $(\theta, \phi) \leftarrow$ Atualize os parâmetros θ, ϕ baseado no gradiente g
até *a convergência dos parâmetros* (θ, ϕ);
retorna (θ, ϕ)

dados. Este tipo de treinamento é tradicionalmente denominado *minibatch* ou *mini-batch* estocástico [Goodfellow et al., 2016]. Tal método é realizado conforme descrito a seguir. Dado N amostras de uma base de dados $\mathbf{X}$ e o limite inferior da evidência (ELBO), pode-se construir um estimador baseado em *minibatches*, de tal forma que:

$$\mathcal{L}(\theta, \phi, \mathbf{X}) \simeq \tilde{\mathcal{L}}^M(\theta, \phi, \mathbf{X}) = \frac{N}{M}\sum_{i=1}^{M}\tilde{\mathcal{L}}^M(\theta, \phi, \mathbf{x}^{(i)}) \tag{4.19}$$

onde M denota o número de amostras em cada *minibatch* e $\tilde{\mathcal{L}}^M$ representa o valor estimado do limite inferior utilizando as amostras presentes no *minibatch*. Em cada *minibatch* $\mathbf{X}^M = \{x^{(i)}\}_{i=1}^M$, existem M amostras extraídas aleatoriamente da coleção de dados. Com um número suficiente de amostras no *minibatch*, é possível computar o valor do gradiente $\nabla_{\theta,\phi}\tilde{\mathcal{L}}^M(\theta, \phi, \mathbf{X}^M)$ e assim realizar a otimização por meio de métodos estocásticos baseados em diferenciação, tais como Gradiente Descendente Estocástico, AdaGrad [Duchi et al., 2011] e o ADAM [Kingma & Ba, 2014], conforme mostrado no Algoritmo 1. Em suma, o algoritmo de otimização treina os parâmetros (conjunto de pesos) da rede de decodificação, representado por θ, e de codificação, representado por ϕ, por meio do gradiente estimado via *minibatches*. A atualização dos parâmetros é realizada de forma iterativa, até que um critério de convergência seja alcançado.

Como visto nos métodos LMDTM e GSDTM, o limite inferior variacional pode ser condensado em dois termos. O primeiro termo é composto pelas divergências KL da aproximação da distribuição variacional em relação à distribuição *a posteriori*. O segundo refere-se à perda de reconstrução cujo valor equivale a $\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})}logp_\theta(\mathbf{x}|\mathbf{y}, \mathbf{z})$. Enquanto que o primeiro termo pode ser diferenciado normalmente, já que as divergências KL podem ser calculadas analiticamente, o mesmo não ocorre com o segundo termo, visto que o mesmo é definido sob a forma de entropia. Contudo, a perda de reconstrução pode ser estimada por meio de amostragem Monte Carlo, onde a média de L observações provenientes de $\mathbf{z}$ são utilizadas para fornecer um valor esti-

mado. Desta forma, o gradiente pode ser calculado de forma eficiente, de tal forma que $\nabla_{\theta,\phi}\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x})} logp_\theta(\mathbf{x}|\mathbf{y},\mathbf{z}) \approx \frac{1}{L}\sum_{i=1}^{L} logp_\theta(\mathbf{x}|\mathbf{y},\mathbf{z}^{(i)})$.

Baseado nas amostras $\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x},\mathbf{y})$, treinou-se os métodos LMDTM e GSDTM com otimizador ADAM [Kingma & Ba, 2014] a fim de maximizar limite inferior variacional. Seguindo a mesma abordagem do modelo NVDM e ProdLDA, adotou-se uma amostra de tamanho $L = 1$ a fim de diminuir o custo de treinamento. Em ambos os métodos, foram utilizados *minibatches* compostos por 200 amostras ($M = 200$).

4.4 Considerações finais

Neste capítulo foram apresentados dois modelos de tópicos baseados em Autocodificadores Variacionais. O primeiro método proposto, denominado LMDTM, visa obter melhores resultados em modelagem de tópicos em base de dados textuais onde a distribuição dos dados é complexa. O segundo, intitulado GSDTM, busca melhorar a inferência dos tópicos utilizando a distribuição *Gumbel-Softmax*, com o intuito de aproximar com melhor qualidade a distribuição de elementos textuais na base de dados, que são inerentemente categóricos. No próximo capítulo será abordado o protocolo experimental e serão apresentados os resultados dos experimentos, considerando três métricas quantitativas de avaliação e uma inspeção qualitativa em duas coleções de dados distintas.

5

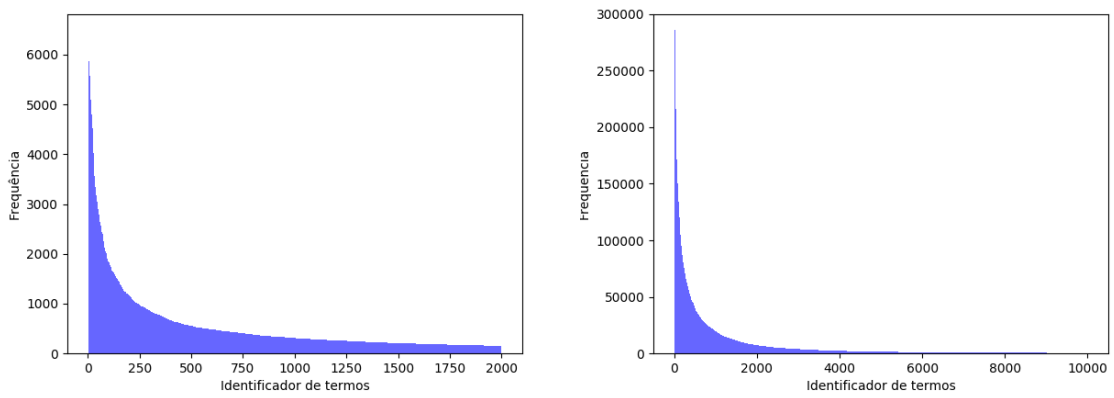
Metodologia e Experimentos

Neste capítulo, serão abordados a metodologia utilizada no processo de experimentação e os resultados obtidos provenientes desse processo, com o intuito de validar os métodos propostos (detalhados no Capítulo 4) e compará-los com outros métodos que são considerados o estado da arte na tarefa de modelagem de tópicos. A primeira Seção (5.1) apresenta as coleções de dados utilizadas no processo de experimentação, coleções estas que são recorrentes na literatura de modelagem de tópicos. Em seguida, a Seção 5.2 aborda a metodologia utilizada, definindo a configuração e o protocolo experimental do processo de experimentação, a fim de garantir que a comparação entre os métodos propostos e os métodos adotados como *baseline* seja válida. Posteriormente, descreve-se na Seção 5.3 os resultados obtidos pelos métodos por meio de três métricas quantitativas e uma análise qualitativa de modelagem de tópicos. Por fim, fazem-se as considerações finais do capítulo na Seção 5.4.

5.1 Coleções de Dados

Com o intuito de entender melhor a qualidade dos modelos propostos quando comparados aos modelos de referência, realizou-se experimentos em duas coleções de dados: 20newsgroups[1] e RCV1 Volume 2 (RCV1-v2)[2]. A base de dados 20newsgroups é uma coleção de artigos provenientes de fóruns de discussão de notícias, dividida em 11.314 documentos de treino e 7.531 documentos de teste. Já a base RCV1-v2 é uma vasta

[1]Disponível em `http://qwone.com/~jason/20Newsgroups`
[2]Disponível em `http://trec.nist.gov/data/reuters/reuters.html`



(a) Coleção 20newsgroups.

(b) Coleção RCV1-v2

Figura 5.1: Distribuição de frequência dos termos presentes na coleção 20newsgroups (Figura 5.1a) e na coleção RCV1-v2 (Figura 5.1b), ordenados de forma decrescente em relação à frequência. Os termos foram trocados por identificadores numéricos com valores dentro do intervalo $(1, \cdots, V)$, onde V consiste no tamanho do vocabulário.

coleção de dados provenientes de artigos de notícias da Reuters composta por 794.414 documentos de treino e 10.000 documentos de teste.Assim como em Larochelle & Lauly [2012], Miao et al. [2016] e Srivastava & Sutton [2017], o tamanho do vocabulário foi truncado em 2.000 termos para a base 20newsgroups e 10.000 termos para a coleção RCV1-v2. Aplicou-se o mesmo processo de pré-processamento de Miao et al. [2016] para a coleção 20newsgroups. Mais especificamente, adotou-se o seguinte procedimento:

1. Foram removidos os termos considerados *stopwords* e todas as palavras que não estão codificadas em padrão UTF-8.

2. Para cada documento d presente no conjunto de dados, contabilizou-se a frequência de cada termo w presente neste documento, gerando uma matriz de frequência documento-termo.

3. As frequências foram salvas no mesmo formato adotado por Miao et al. [2016].

Pode-se definir *stopwords* como o conjunto de termos que são omitidos ou retirados de um conjunto de dados. No contexto deste trabalho, esse conjunto é composto por palavras de sentido amplo que normalmente não contribuem com o processo de aprendizagem, como artigos, preposições e verbos de estado. Uma vez que não existe um consenso exato na literatura sobre quais palavras devem estar contidas neste conjunto, utilizou-se o mesmo conjunto de termos adotados pela biblioteca NLTK como *stopwords*.

A fim de realizar os experimentos de avaliação de recuperação de documentos, foi seguida a abordagem de Larochelle & Lauly [2012] e extraiu-se aleatoriamente das duas coleções de dados 100 documentos da base de treino para compor a base de validação. Após a etapa de pré-processamento, foram obtidos 10.870 documentos da base de treino, 7.280 para a base de teste e 100 documentos para a base de validação para a coleção 20newsgroups. Em razão da coleção RCV1-v2 já ser disponibilizada pré-processada, apenas se mudou o formato para adequá-la ao mesmo formato utilizado pela base 20newsgroups.

É importante notar que a distribuição dos termos em coleções de texto possui um comportamento específico. Conforme mostrado nos histogramas de frequência dos termos presentes na Figura 5.1, a distribuição dos termos segue uma distribuição similar à distribuição de Zipf [Sichel, 1975; Zipf, 2013]. Desta forma, poucos termos da base possuem alta probabilidade de ocorrência, enquanto grande parte deles ocorrem com menor probabilidade. Esta é uma característica de coleções textuais, que afeta significativamente a aprendizagem dos modelos baseados em Autocodificadores Variacionais.

5.2 Metodologia

Para comparar os métodos propostos com o *baseline* adotado, usou-se a mesma configuração experimental de Miao et al. [2016] e Srivastava & Sutton [2017]. Para a rede de codificação do GSDTM foi definido uma rede do tipo *feed-forward* com três camadas, sendo as duas primeiras camadas com ativações ReLU e dimensão igual a 100 e a última uma camada de saída sem funções de ativação, com tamanho igual ao número de tópicos adotado, que define o valor do parâmetro da distribuição *Gumbel-Softmax*. Por sua vez, a rede de codificação do modelo LMDTM é similar ao GSDTM, exceto no que diz respeito ao número de camadas de redes sem funções de ativação. Devido ao fato do LMDTM ter dois parâmetros no modelo probabilístico na rede de codificação (média e desvio-padrão), a implementação emprega as duas últimas camadas sem funções de ativação, cada uma responsável por estimar um parâmetro de uma das componentes da mistura de distribuições Normal-Logísticas. Em ambos os métodos, foi aplicado *mini-batches* com tamanho de 200 amostras.

A rede discriminativa do modelo LMDTM consiste em uma rede com duas camadas, tendo a primeira uma função de ativação do tipo ReLU com dimensão igual a 100, seguida de uma camada sem funções de ativação com dimensão equivalente ao número de tópicos. A rede de decodificação é uma rede com função de ativação do

tipo *Softmax* com uma camada latente em ambos os modelos, com dimensão igual ao número de termos no vocabulário. Também cada coleção de dados foram treinadas com 50 e 200 tópicos usando o otimizador ADAM [Kingma & Ba, 2014], seguindo o padrão adotado na literatura de modelo de tópicos. Adicionalmente, os códigos-fonte da implementação dos modelos LMDTM e GSDTM foram disponibilizados publicamente na plataforma GitHub[3].

Seguiu-se o procedimento adotado por Srivastava & Sutton [2017] para calcular as métricas de coerência de tópicos e perplexidade, usando o limite inferior variacional para computar a perplexidade média por documento na base de teste. Também foi usado o *Normalized Pointwise Mutual Information* (NPMI) como métrica de coerência de tópicos. Essa métrica é dada pela seguinte Equação abaixo (5.1):

$$\text{NPMI}(w_i,\, w_j) = \frac{log\frac{P(w_i,w_j)}{P(w_i)P(w_j)}}{-logP(w_i,w_j)} \tag{5.1}$$

O objetivo desta métrica é estimar o grau de coocorrência entre dois termos quaisquer (w_i e w_j) em uma coleção de documentos. Enquanto $P(w_i, w_j)$ denota a probabilidade dos dois termos ocorrerem juntos em um mesmo documento, $P(w_i)$ e $P(w_j)$ indicam respectivamente a probabilidade dos termos w_i e w_j ocorrerem na base de dados. O numerador da razão é denominado *Pointwise Mutual Information* (PMI) e corresponde ao logaritmo da razão entre $P(w_i, w_j)$ e $P(w_i)P(w_j)$. O PMI será alto se os termos w_i e w_j ocorrerem com frequência no mesmo documento e se a frequência individual de cada termo for baixa na coleção. O denominador funciona como um normalizador, permitindo que o valor do PMI seja normalizado no intervalo $[-1, 1]$, onde -1 indica que os termos nunca coocorrem na coleção, 0 denota que a coocorrência entre os termos é estatisticamente independente e 1 denota coocorrência perfeita.

Na tarefa de avaliação de recuperação de documentos foi usado o mesmo processo adotado por Larochelle & Lauly [2012], definindo 100 documentos como base de validação. Quanto ao método LMDTM, realizou-se experimentos específicos de validação para encontrar o número ideal de componentes, utilizando os valores $K = \{5, 10, 15, 20\}$ onde K é o número de componentes. Em virtude do valor $K = 5$ apresentar em média os melhores resultados no experimento de validação, adotou-se este número de componentes nos demais experimentos.

Uma vez que os modelos de tópicos baseados em Autocodificadores Variacionais tendem a apresentar problema de colapso de tópicos, isto é, quando o algoritmo falha devido à instabilidade do processo de treinamento e define apenas um ou poucos tópi-

[3] `https://github.com/denyssilveira/gsdtm_lmdtm_topic_model/`

cos distintos contendo apenas termos com alta frequência, autores tal como Srivastava & Sutton [2017] observaram empiricamente que este problema pode ser amenizado por meio do uso de técnicas auxiliares tais como *Dropout* e *Batch Normalization* (BN) seguindo uma escolha cuidadosa dos parâmetros utilizados pelo otimizador tal como a taxa de aprendizagem. Enquanto o *Dropout* melhora a generalização e retarda o processo de treinamento a fim de evitar o problema de *overfitting*, o *Batch Normalization* minimiza a amplitude da covariância, auxiliando a estabilização da aprendizagem. Contudo, devido ao grande impacto dessas técnicas no processo de aprendizagem, cada um dos métodos foram avaliados adotando-se todas as combinações possíveis da aplicação ou não-aplicação de tais técnicas. Em particular, definiu-se a probabilidade de *Dropout* para 0.6 em todos os experimentos utilizando *Dropout*, uma vez que Srivastava & Sutton [2017] adotaram este valor na implementação do método ProdLDA. Avaliou-se os modelos usando os melhores resultados obtidos na etapa de validação. Com a finalidade de determinar as configurações ótimas, foram realizados testes preliminares em cada cenário utilizando como taxa de aprendizagem todos os valores em $\alpha_t = \{0.02, 0.002, 0.0002\}$. A escolha destes valores são similares aos adotados nas implementações dos métodos avaliados e disponibilizadas publicamente. Também foi utilizado um número fixo de iterações, sendo adotadas 300 iterações em experimentos utilizando a coleção RCV1-V2 e 2.000 iterações nos cenários onde a base 20newsgroups é utilizada.

Em relação ao modelo DocNade, não foram aplicados métodos auxiliares destinadas à redes neurais, tais como *Batch Normalization* e *Dropout*, devido às limitações da arquitetura do método. Mais especificamente, diferentemente dos métodos baseados em Autocodificadores Variacionais, o DocNADE faz uso de uma estrutura hierárquica baseada em uma árvore binária de regressões logísticas, o que inviabiliza a adotação de métodos auxiliares sem alterar significativamente a arquitetura de rede neural do DocNADE. Logo, não se adotou a implementação do *Batch Normalization* e do *Dropout* para este método.

Outra limitação na metodologia decorre da falha do LMDTM em treinar a coleção 20newsgroups utilizando apenas o *Dropout*, tanto com 50 tópicos quanto com 200 tópicos. A causa da falha foi erro numérico no valor do gradiente, que ocorre devido à instabilidade do treinamento ocasionado pela falta da aplicação do *Batch Normalization* e principalmente pelo modelo de mistura, que propicia o crescimento anormal do valor do gradiente. Este comportamento será melhor investigado em trabalhos futuros. Deste modo, não foram realizados experimentos neste cenário específico.

5.3 Resultados dos Experimentos

A descrição dos resultados dos experimentos foi dividida em quatro subseções. A primeira descreve a avaliação dos tópicos extraídos dos modelos avaliados utilizando como métrica de avaliação o *Average Topic Coherence* (ATC), ou Coerência Média de Tópicos em tradução livre, cujo objetivo é avaliar o quão coerentes são as primeiras N palavras extraídas em cada tópico, isto é, qual o grau em que essas palavras coocorrem em cada tópico. A segunda subseção mostra os resultados da avaliação do modelo gerador, que avalia a capacidade de generalização do modelo probabilístico para cada método analisado. Mais especificamente, mede-se o quão bem a distribuição inferida por estes métodos prevê uma amostra não presente na base de treino. A terceira subseção analisa os resultados provenientes da tarefa de recuperação de documentos, cujos dados utilizados nesta tarefa provém da representação vetorial latente dos documentos aprendida por cada método. Por fim, a última seção descreve os resultados qualitativos, mostrando o ranqueamento de tópicos e a qualidade do *embedding* obtidos em cada modelo.

Nos experimentos utilizou-se as implementações disponíveis publicamente dos métodos de referência para fins de comparação, que são: ProdLDA[4], NVDM[5], e DocNADE[6]. Para cada implementação, foram utilizados os parâmetros definidos por padrão.

5.3.1 Avaliação de tópicos

Avaliou-se a coerência média dos tópicos (ATC) dos modelos usando a métrica NPMI. Uma vantagem desta métrica é que, comparada com outras tal como perplexidade (cf. Seção 5.3.2), ela se mostra melhor correlacionada com a percepção humana sobre qualidade dos tópicos gerados [Lau et al., 2014].

A fim de realizar a extração de tópicos dos modelos treinados, seguiu-se a abordagem de Srivastava & Sutton [2017], descrita na Figura 5.2. A partir da regressão multinomial logística que calcula a distribuição de probabilidade $p_\theta(\mathbf{x}|\mathbf{y}, \mathbf{z})$, extraem-se os pesos $\mathbf{W}$ da regressão logística. Pode-se interpretar $\mathbf{W}$ como uma matriz onde cada linha representa um tópico $t \in [1, 2, \cdots, T]$ e cada coluna como um termo w_i presente em um vocabulário V de termos, onde $i \in [1, 2, \cdots, |V|]$. Uma vez que $\mathbf{W}$ é uma representação vetorial das palavras em um espaço de tópicos, o valor $\mathbf{W}_{ti}$ indica o *score*,

[4] https://github.com/akashgit/autoencoding_vi_for_topic_models/
[5] https://github.com/ysmiao/nvdm/
[6] http://www.dmi.usherb.ca/~larocheh/code/DocNADE.zip

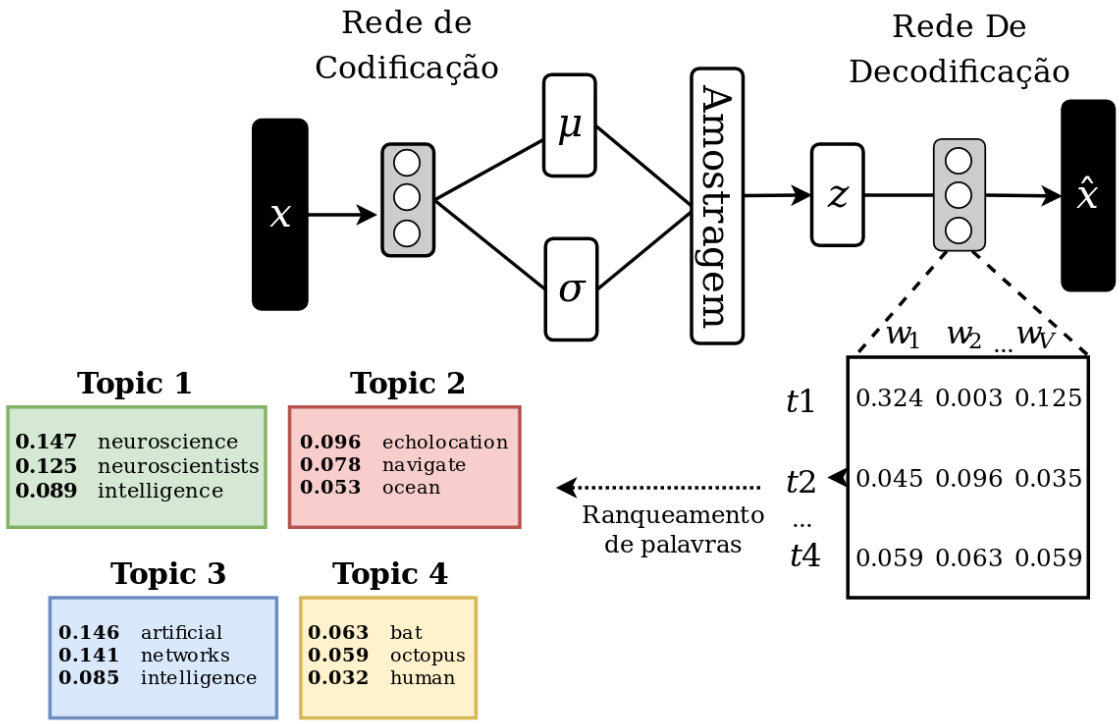


Figura 5.2: Esquematização do processo de extração de tópicos, mostrando como ranqueamentos de tópicos podem ser obtidos a partir da matriz de pesos presente na regressão multinomial logística utilizada na rede de decodificação.

também denominado como intensidade de conexão, entre um tópico t e uma palavra w_i. Logo, para cada tópico t, obtém-se N palavras com os maiores *scores* na linha t da matriz de pesos $\mathbf{W}$. O resultado desta operação é um ranqueamento de tópicos contendo as N palavras com a maior probabilidade de pertencer a um tópico específico.

Coleção 20newsgroups (50 Tópicos)				
	Com Dropout		**Sem Dropout**	
Método	**Com BN**	**Sem BN**	**Com BN**	**Sem BN**
GSDTM	**0.273**	0.236	0.180	**0.186**
LMDTM	0.218	-	0.190	0.153*
ProdLDA	0.266	**0.270***	**0.194**	0.109*
NVDM	0.150	0.177	0.069	0.081
DocNADE	-	-	-	0.141

Tabela 5.1: Resultado do ATC obtido da coleção 20newsgroups utilizando 50 tópicos. Valores mais altos indicam resultados melhores. Asteriscos indicam que houve problema de colapso de tópicos durante o treinamento.

Considere que w_{ti} seja a i-ésima palavra presente no ranqueamento de tópicos. A métrica ATC é dada pela Equação abaixo (5.2):

Coleção 20newsgroups (200 Tópicos)				
	Com Dropout		**Sem Dropout**	
Método	**Com BN**	**Sem BN**	**Com BN**	**Sem BN**
GSDTM	**0.236**	**0.211**	**0.162**	0.134
LMDTM	0.133	-	0.122	0.119*
ProdLDA	0.225	0.046*	0.159	0.047*
NVDM	0.091	0.152	0.055	0.076
DocNADE	-	-	-	**0.139**

Tabela 5.2: Resultado do ATC obtido da coleção 20newsgroups utilizando 200 tópicos. Valores mais altos indicam resultados melhores. Asteriscos indicam que houve problema de colapso de tópicos durante o treinamento.

Coleção RCV1-v2 (50 tópicos)				
	Com Dropout		**Sem Dropout**	
Método	**Com BN**	**Sem BN**	**Com BN**	**Sem BN**
GSDTM	**0.177**	**0.147**	**0.176**	0.028
LMDTM	0.140	0.140	0.141	0.102
ProdLDA	0.147	0.096	0.147	**0.110**
NVDM	0.105	0.110	0.052	0.060
DocNADE	-	-	-	0.037

Tabela 5.3: Resultado de ATC obtido da coleção RCV1-v2 utilizando 50 tópicos. Valores mais altos indicam resultados melhores. Asteriscos indicam que houve problema de colapso de tópicos durante o treinamento.

Coleção RCV1-v2 (200 tópicos)				
	Com Dropout		**Sem Dropout**	
Método	**Com BN**	**Sem BN**	**Com BN**	**Sem BN**
GSDTM	**0.183**	0.089	0.138	0.022
LMDTM	0.090	**0.108**	**0.152**	0.020
ProdLDA	0.154	0.088	0.110	**0.063**
NVDM	0.058	0.087	0.032	0.031
DocNADE	-	-	-	0.026

Tabela 5.4: Resultado de ATC obtido da coleção RCV1-v2 utilizando 200 tópicos. Valores mais altos indicam resultados melhores. Asteriscos indicam que houve problema de colapso de tópicos durante o treinamento.

$$\text{ATC} = \frac{1}{T}\sum_{t=1}^{T}\sum_{j=2}^{N}\sum_{i=1}^{j-1}\frac{log\frac{P(w_{ti},w_{tj})}{P(w_{ti})P(w_{tj})}}{-logP(w_{ti},w_{tj})} \tag{5.2}$$

onde $P(w_{ti})$ representa a probabilidade da palavra w_{ti} ocorrer em um documento e $P(w_{ti}, w_{tj})$ indica a probabilidade de w_{ti} e w_{tj} ocorrerem no mesmo documento. Com o

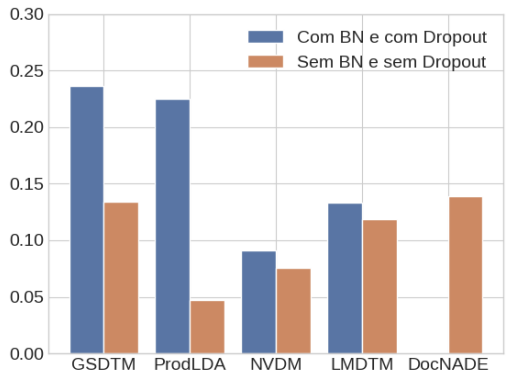
intuito de realizar este experimento, usou-se a implementação de Lau et al. [2014] para calcular a métrica ATC de cada método. Essa métrica é extraída do ranqueamento de tópicos obtido pela análise da matriz de pesos **W**. A implementação está disponível publicamente na plataforma Github[7].

As Tabelas 5.1, 5.2, 5.3 e 5.4 mostram os resultados obtidos nos experimentos em relação à coerência média de tópicos. Como pode ser visto, o método GSDTM claramente supera os demais métodos quando utilizadas todas as abordagens auxiliares (*Batch Normalization* e *Dropout*). O mesmo teve ganhos de aproximadamente 20,41% em comparação com o segundo melhor resultado, alcançado pelo método ProdLDA, no cenário onde se adota a coleção RCV1-v2 com 50 tópicos. Em outros cenários onde todas as abordagens auxiliares são adotadas, o GSDTM superou os demais métodos com menor margem de ganho: 18,83% em relação ao ProdLDA na coleção RCV1-v2 com 200 tópicos e 4,89% e 2,63% na coleção 20newsgroups em relação ao ProdLDA utilizando respectivamente 200 tópicos e 50 tópicos.

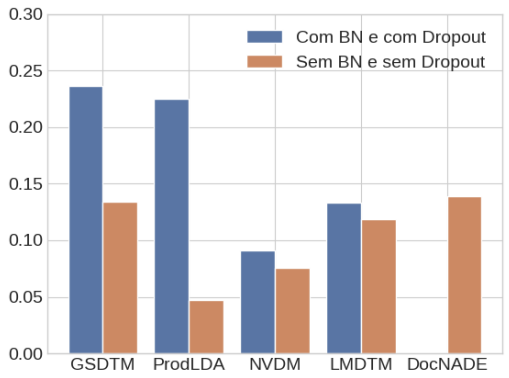
Também os resultados mostram que o método proposto GSDTM alcançou índices competitivos de coerência média de tópicos utilizando a coleção 20newsgroups em outros cenários onde pelo menos uma técnica auxiliar deixa de ser utilizada. Por exemplo, quando as técnicas *Batch Normalization* e *Dropout* não são utilizadas, o GSDTM supera os demais métodos, tendo ganho de 21,56% em relação ao segundo melhor método (LMDTM) utilizando 50 tópicos. Entretanto, à medida que a complexidade aumenta em termos de tamanho da coleção de dados e número de tópicos, o GSDTM perde a capacidade de gerar tópicos altamente coesos, sendo superado pelos outros métodos. Desta forma, é perceptível que o GSDTM depende de técnicas de estabilização para que a coerência de tópicos se mantenha elevada em cenários onde a complexidade é alta. Contudo, este fenômeno não é exclusivo dos métodos propostos, uma vez que todos os outros métodos têm algum grau de diminuição da coerência média de tópicos neste cenário.

Quando os métodos analisados são treinados sem o uso de *Batch Normalization* e *Dropout*, estas mudanças induzem em uma queda visível dos níveis de ATC em todos os métodos analisados, conforme pode ser observado na Figura 5.3. Isto indica que o emprego de métodos de estabilização no processo de treinamento de *Auto-Encoders Variacionais* em modelagem de tópicos traz melhorias. Ainda neste cenário, os métodos LMDTM e ProdLDA sofreram de um efeito negativo na modelagem de tópicos quando treinados na coleção 20newsgroups, denominado colapso de tópicos [Srivastava & Sutton, 2017], que consiste na extração de tópicos idênticos entre si, ou seja, o mo-

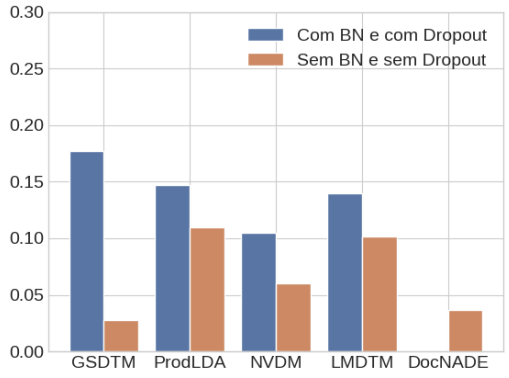
[7] `https://github.com/jhlau/topic_interpretability`



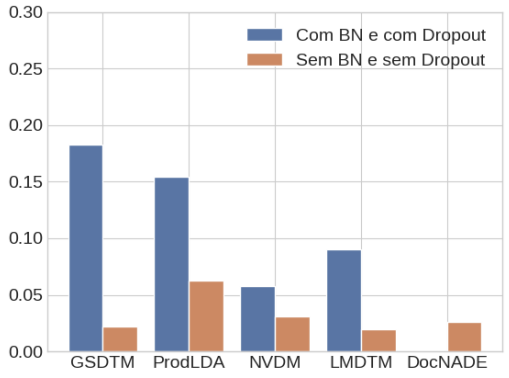
(a) Coleção 20newsgroups com 50 tópicos.



(b) Coleção 20newsgroups com 200 tópicos.



(c) Coleção RCV1-v2 com 50 tópicos.



(d) Coleção RCV1-v2 com 200 tópicos.

Figura 5.3: Comparativo gráfico entre os resultados obtidos utilizando Batch Normalization (BN) e Dropout e os resultados obtidos sem utilizar nenhuma destas técnicas.

delo não é capaz de identificar tópicos distintos, colapsando o espaço de tópicos em um ponto específico, conforme exemplificado na Tabela 5.5.

Por outro lado, o método LMDTM foi superado pelos métodos GSDTM e ProdLDA em grande parte dos cenários analisados. Entretanto, os resultados obtidos pelo LMDTM são competitivos em alguns cenários na coleção RCV1-v2. Por exemplo, os valores de coerência média de tópicos na coleção de dados RCV1-v2 utilizando *Batch Normalization* são próximos àqueles obtidos com o método ProdLDA. Estes resultados indicam que possivelmente o método LMDTM não é adequado para coleção de dados menores e, em coleções maiores, este método requer o uso de *Batch Normalization* para evitar o problema de colapso de tópicos. Este índice de coerência consideravelmente inferior, em geral, pode ser atribuído à grande complexidade do modelo, onde se utilizam várias distribuições Normais-Logísticas. Assim, o LMDTM tem uma maior quantidade de parâmetros a serem aprendidos quando comparado com outros métodos,

que não utilizam um modelo de mistura. Este problema pode ser mitigado por meio do aumento do tamanho do *corpus* presente na coleção de dados, o qual se pretende realizar em estudos futuros.

Enquanto o uso da distribuição *Gumbel-Softmax* diminuiu o problema de colapso no GSDTM, os modelos que empregam distribuição Gaussiana ou Normal-Logística como distribuições da rede de codificação são mais suscetíveis a esse problema e necessitam ser treinados com métodos como *Batch Normalization* e *Dropout* a fim de evitar o problema de colapso de tópicos e alcançar melhores resultados de coerência de tópicos.

Em relação à coleção de dados RCV1-v2, foi registrado um baixo nível de coerência de tópicos pelo método GSDTM sem o uso de nenhuma técnica auxiliar, como o *Batch Normalization* e o *Dropout*. Pode-se observar neste cenário que o GSDTM alcança baixos índices de perplexidade (quanto menor a perplexidade, melhor a generalização do modelo). Isto claramente indica que existe um compromisso entre coerência de tópicos (utilizando NPMI como métrica) e perplexidade, que será melhor explorado na subseção 5.3.2.

Ainda em relação ao uso das técnicas auxiliares, pode-se observar que o nível de influência do *Batch Normalization* e do *Dropout* difere dependendo do tamanho da coleção utilizada. Por exemplo, nos experimentos executados na coleção 20newsgroups, o *Dropout* obteve no geral um peso maior na melhoria da coerência média dos tópicos do que o *Batch Normalization*, enquanto o efeito inverso foi constatado na coleção RCV1-v2. Este comportamento se deve ao *Dropout* ter efeito mais visível na diminuição do *overfitting* em coleções de dados menores.

Tópico 1	**Tópico 2**	**Tópico 3**	**Tópico 4**	**Tópico 5**
people	thanks	thanks	thanks	thanks
don	anyone	anyone	anyone	windows
think	know	please	please	anyone
more	out	am	am	please
than	just	out	windows	am
just	like	had	know	advance
who	had	mail	hi	hi
much	get	like	mail	mail
their	am	get	like	card
government	don	who	out	help

Tabela 5.5: Exemplo de colapso de tópicos. É possível observar que quando este problema ocorre, não existe uma aprendizagem, sendo extraídos palavras com alta frequência na coleção de dados.

Como esperado, quase todos os métodos tiveram quedas nos níveis de coerência média de tópicos quando um número maior de tópicos é utilizado, uma vez que um número de tópicos mais elevado aumenta consideravelmente a complexidade da tarefa de modelagem de tópicos. As exceções são os métodos GSDTM e ProdLDA treinados na coleção de dados RCV1-V2, utilizando 200 tópicos e os métodos auxiliares *Batch Normalization* e *Dropout*, cuja taxa de ganho é respectivamente 3,39% e 4,76% quando comparados com os mesmos cenários treinados com 50 tópicos. Outro método que foi exceção é o LMDTM, quando o mesmo é treinado na coleção de dados RCV1-V2 sem uso de *Dropout* e com uso de *Batch Normalization*. Neste caso, o ganho percentual foi de 7,8%, quando comparado com o mesmo cenário treinado com 50 tópicos.

5.3.2 Avaliação do modelo gerador

Nesta subseção será avaliado a qualidade dos modelos geradores empregados nos métodos testados utilizando uma métrica tradicional em modelagem de tópicos, denominada de perplexidade. Assim como Miao et al. [2016], computou-se a perplexidade por documento por meio da seguinte Equação (5.3):

$$exp\left(-\frac{1}{D}\sum_{n}^{N_d}\frac{1}{N_d}logp(X_d)\right) \tag{5.3}$$

Onde D consiste no número de documentos, N_d no número de palavras contidas no documento d, e $p(X_d)$ é a verossimilhança (*likelihood*) do documento amostrado da base de teste. Utilizou-se toda a base de teste para computar a perplexidade, adotando-se os mesmos parâmetros utilizados nos experimentos de avaliação de tópicos.

Coleção 20newsgroups (50 tópicos)				
	Com Dropout		**Sem Dropout**	
Método	**Com BN**	**Sem BN**	**Com BN**	**Sem BN**
GSDTM	**1126.036**	**884.404**	**1123.764**	**853.691**
LMDTM	1136.989	-	1146.141	981.606*
ProdLDA	1185.553	1113.700*	1134.311	1002.381*
NVDM	1186.689	1060.086	1315.521	904.770
DocNADE	-	-	-	870.505

Tabela 5.6: Resultado de perplexidade obtido da coleção 20newsgroups utilizando 50 tópicos. Valores mais baixos indicam resultados melhores. Asteriscos indicam que houve problema de colapso de tópicos durante o treinamento.

Os valores de perplexidade obtidos são apresentados nas Tabelas 5.6, 5.7, 5.8 e 5.9. Como observado, o método GSDTM alcançou em média os melhores resultados.

Coleção 20newsgroups (200 tópicos)				
	Com Dropout		**Sem Dropout**	
Método	**Com BN**	**Sem BN**	**Com BN**	**Sem BN**
GSDTM	**1120.805**	**795.956**	**1116.674**	861.301
LMDTM	1170.130	-	1204.832	981.606*
ProdLDA	1189.418	1144.174*	1166.472	1122.239*
NVDM	1385.611	1064.448	1483.180	920.497
DocNADE	-	-	-	**851.448**

Tabela 5.7: Resultado de perplexidade obtido da coleção 20newsgroups utilizando 200 tópicos. Valores mais baixos indicam resultados melhores. Asteriscos indicam que houve problema de colapso de tópicos durante o treinamento.

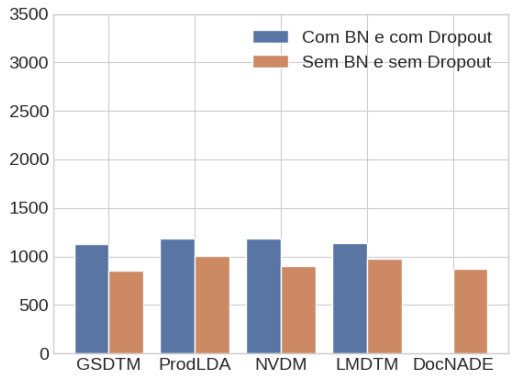
Coleção RCV1-v2 (50 tópicos)				
	Com Dropout		**Sem Dropout**	
Método	**Com BN**	**Sem BN**	**Com BN**	**Sem BN**
GSDTM	**1982.267**	**828.932**	**1964.994**	**429.976**
LMDTM	2169.010	1451.038	2456.187	621.417
ProdLDA	2065.349	1365.887	2270.666	623.260
NVDM	2273.198	1200.769	3507.320	535.439
DocNADE	-	-	-	629.915

Tabela 5.8: Resultado de perplexidade obtido da coleção RCV1-v2 utilizando 50 tópicos. Valores mais baixos indicam resultados melhores. Asteriscos indicam que houve problema de colapso de tópicos durante o treinamento.

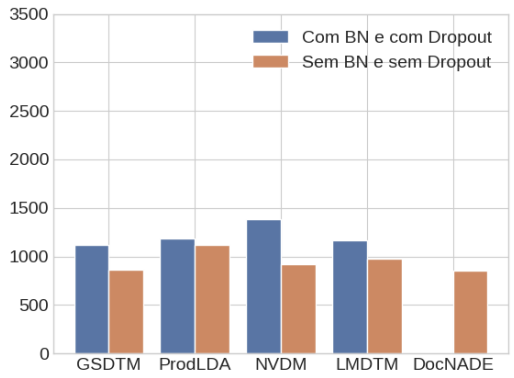
Coleção RCV1-v2 (200 tópicos)				
	Com Dropout		**Sem Dropout**	
Método	**Com BN**	**Sem BN**	**Com BN**	**Sem BN**
GSDTM	**1972.421**	**510.893**	**1957.157**	**269.733**
LMDTM	2213.163	1620.330	2398.209	853.110
ProdLDA	2033.609	1471.019	2231.173	805.310
NVDM	2400.339	1184.080	3428.334	534.592
DocNADE	-	-	-	449.849

Tabela 5.9: Resultado de perplexidade obtido da coleção RCV1-v2 utilizando 200 tópicos. Valores mais baixos indicam resultados melhores. Asteriscos indicam que houve problema de colapso de tópicos durante o treinamento.

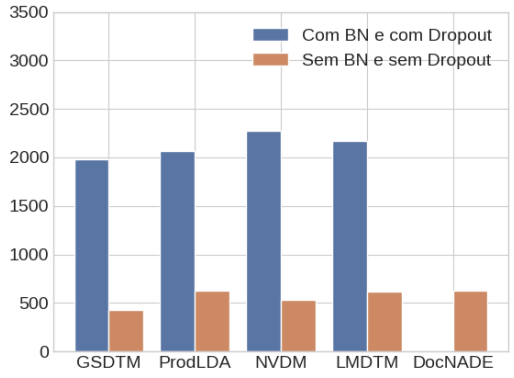
Os maiores ganhos deste método são observados na coleção RCV1-v2, sem o uso das técnicas de *Batch Normalization* e *Dropout*. Em comparação com o método NVDM, que alcançou a segunda melhor colocação nos resultados em termos de perplexidade, o GSDTM foi quase 19,69% melhor (taxa de decaimento da perplexidade) quando usado 50 tópicos. Para 200 tópicos, o ganho registrado é em torno de 40% quando compa-



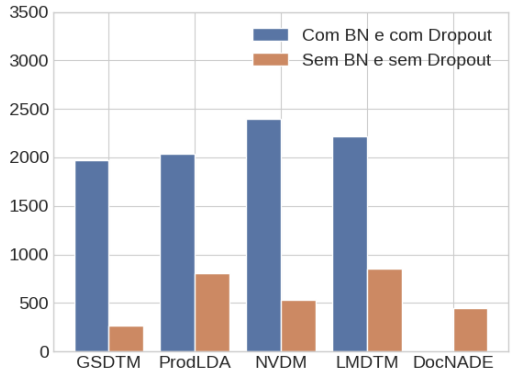
(a) Coleção 20newsgroups com 50 tópicos.



(b) Coleção 20newsgroups com 200 tópicos.



(c) Coleção RCV1-v2 com 50 tópicos.



(d) Coleção RCV1-v2 com 200 tópicos.

Figura 5.4: Comparativo gráfico entre os resultados de perplexidade obtidos utilizando Batch Normalization (BN) e Dropout e os resultados obtidos sem utilizar nenhuma destas técnicas.

rado ao DocNADE, o segundo melhor método neste cenário. Os ganhos na coleção 20newsgroups foram mais modestos do que na coleção RCV1-V2, o que possivelmente indicam que o tamanho do corpus afeta a qualidade do modelo gerador. Por outro lado, o método LMDTM foi superado por outros métodos em termos de perplexidade. Isto indica que o uso de uma mistura de distribuições pode não ser a melhor opção para criar modelos geradores para as bases de dados testadas (20newsgroups e RCV1-v2).

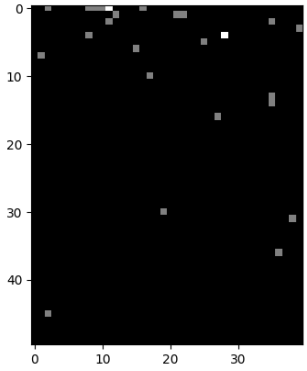
Em geral, todos os métodos obtiveram melhores índices de perplexidade (quanto menor este índice, melhor) quando as duas técnicas auxiliares **não** são empregadas (*Batch Normalization* e *Dropout*), conforme indicado na Figura 5.4. Desta forma, enquanto o uso destas técnicas é benéfico em termos de coerência de tópicos, o mesmo não foi observado na avaliação do modelo gerador, o que indica que o melhor modelo não será necessariamente o que vai gerar tópicos mais coerentes. De fato, isto é uma

observação razoável, uma vez que um documento real não é composto somente por palavras que pertencem puramente a certos tópicos ao longo da coleção de dados. Uma linguagem natural textual pode ser composta for diversos tópicos, com palavras que podem ser compartilhadas entre estes tópicos e algumas palavras que não estão necessariamente restritas a algum tópico em particular. Um bom modelo gerador deve capturar todas estas nuances, o que pode conflitar com o objetivo principal da modelagem de tópicos, que é extrair agrupamentos de termos que são correlacionadas com algum tópico.

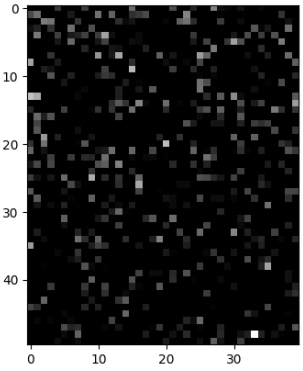
Outro ponto interessante é que o uso do *Batch Normalization* e principalmente do *Dropout* diminui o impacto que palavras com frequência alta possam ter no processo de treino, diminuindo a verossimilhança (*likelihood*) destas palavras que estão no topo de *ranking* de palavras associadas a um certo tópico, aumentando o valor da coerência média de tópicos. Por outro lado, isto pode distorcer a forma com que estes dados estão distribuídos na coleção de dados e, consequentemente, atrapalhar a capacidade de recriação de documentos, aumentando a perplexidade, como observado na Figura 5.5. Nesta figura, tem-se em uma representação em forma de mapa de calor da matriz de frequência utilizada como entrada de dados (representada na Subfigura 5.5a) e das representações de saídas provenientes do método GSDTM e coletadas após treinamento na coleção de dados 20newsgroups com 300 iterações e 50 tópicos. A Figura 5.5b mostra as representações dos documentos quando o *Batch Normalization* e o *Dropout* são utilizados, enquanto que a Figura 5.5c as mostra quando o *Batch Normalization* e o *Dropout* não são utilizados. Pode-se notar que existe mais ruído nos dados reconstruídos no primeiro caso. Isto sugere que o uso destas técnicas causam distúrbios no processo de reconstrução e contribuem para o aumento dos níveis de perplexidade.

5.3.3 Avaliação em Recuperação de Documentos

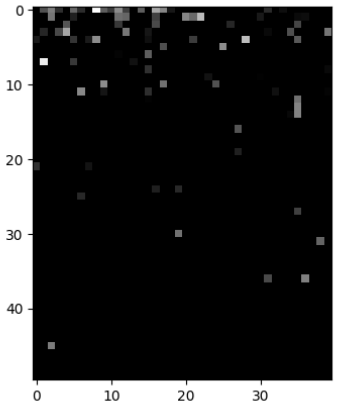
Além das métricas de coerência média de tópicos e perplexidade, também foi avaliada a qualidade dos modelos propostos em aplicações reais. Em particular, seguiu-se a abordagem de Larochelle & Lauly [2012] e foi adotada a tarefa de recuperação de documentos como aplicação alternativa em modelagem de tópicos. Este experimento avalia a tarefa de recuperação de informação baseada na representação vetorial dos documentos que é extraída dos valores de saída provenientes da rede de codificação. Em termos probabilísticos, essa representação é aprendida pela variável latente de tópicos $\mathbf{z}$, que é amostrada da rede de codificação $\mathbf{z} \sim q_\theta(\mathbf{z}|\mathbf{x}, \mathbf{y})$. O objetivo deste experimento é verificar se as representações vetoriais dos documentos no espaço de tópicos latentes conseguem determinar corretamente os rótulos destes documentos.



(a)



(b)



(c)

Figura 5.5: Mapa de calor representando a entrada de dados e duas saídas resultantes do processo de reconstrução.

O experimento foi realizado utilizando as coleções de dados 20Newsgroups e RCV1-V2, usando as instâncias de treinamento como documentos presentes em uma base de dados e instâncias de teste como consultas, além de usar valores fixos para determinar a fração de documentos recuperados da base de treino. Para cada consulta, os documentos no banco de dados são ranqueados de acordo com a similaridade por cosseno em relação às consultas, utilizando suas representações vetoriais. Então, os k documentos com maiores índices de similaridade são recuperados e seus rótulos (que nestas bases representam o tópico ao qual o documento pertence) são comparados. A precisão da consulta é calculada com a proporção de documentos recuperados que compartilham o mesmo rótulo da consulta. Finalmente, realiza-se uma média as precisões de cada consulta para obter o valor final da precisão.

As Figuras 5.6 e 5.7 mostram as curvas de precisão por fração de documentos

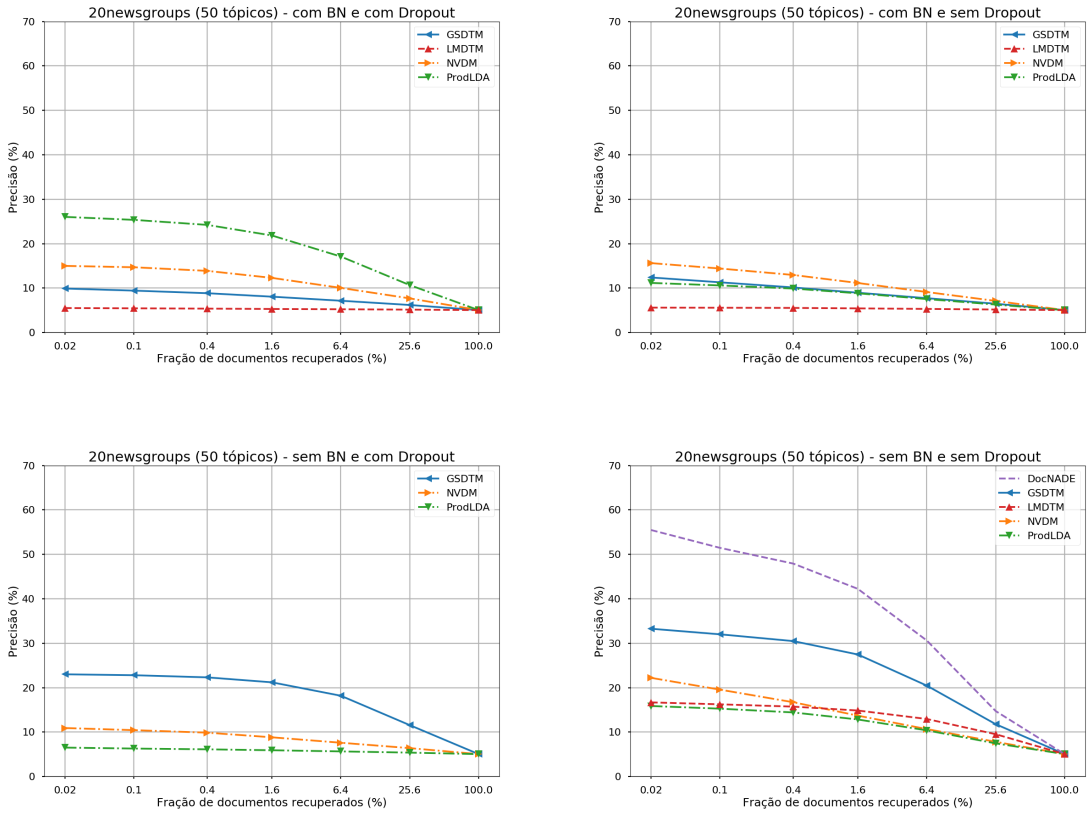


Figura 5.6: Resultado da avaliação de recuperação de documentos na base de dados 20newsgroups com 50 tópicos.

recuperados obtidas usando-se respectivamente 50 e 200 tópicos na coleção de dados 20newsgroups, enquanto que as Figuras 5.8 e 5.9 são curvas de precisão com respectivamente 50 e 200 tópicos usando a coleção de dados RCV1-V2, com 100 documentos usados como base de validação. Os modelos foram treinados usando o otimizador ADAM com o parâmetro β_2 definido com o valor 0.99, igual ao utilizado pela implementação do ProdLDA [Srivastava & Sutton, 2017]. O treino dos modelos foi realizado com e sem uso do *Batch Normalization* e *Dropout*. O intervalo de valores da fração de documentos recuperados é o mesmo adotado em Larochelle & Lauly [2012]. Assim como efetuado no trabalho de Larochelle & Lauly [2012], a representação deste intervalo no gráfico foi linearizado, como forma de melhorar a visualização dos dados.

Nesta tarefa, o GSDTM se mostrou competitivo nos cenários onde o *Batch Normalization* não é utilizado. Por exemplo, no cenário onde a coleção 20newsgroups é treinada com 200 tópicos e sem *Batch Normalization* e com *Dropout*, a diferença de porcentagem é superior a 10% em comparação com os demais métodos. No cenário sem o uso do *Batch Normalization* e *Dropout*, quando o método DocNADE está disponível,

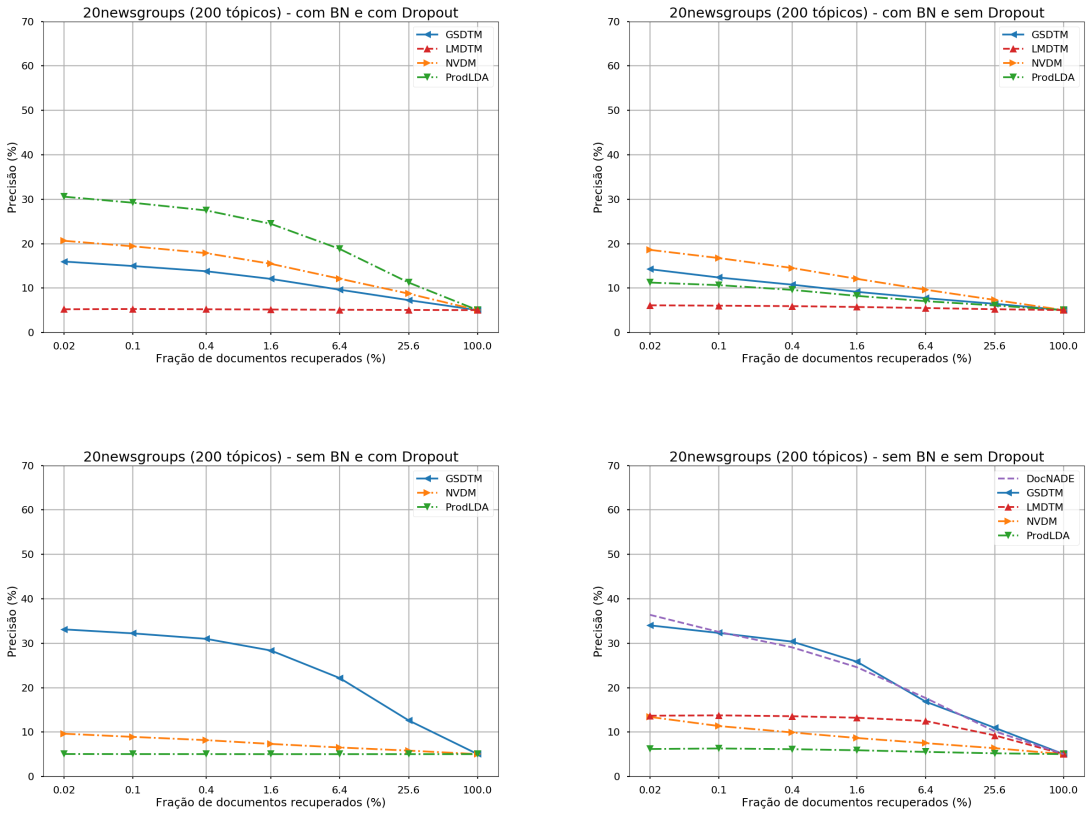


Figura 5.7: Resultado da avaliação de recuperação de documentos na base de dados 20newsgroups com 200 tópicos.

o GSDTM tem um desempenho inferior quando utilizado com 50 tópicos. Entretanto, quando 200 tópicos são treinados, esta vantagem do DocNADE se torna pequena, ficando ambos os métodos empatados. Na coleção de dados RCV1-v2 e no cenário onde o *Batch Normalization* não é utilizado, o GSDTM é competitivo, superando todos os métodos baseados em Autocodificadores Variacionais e tendo qualidade similar ao do método DocNADE. Contudo, em cenários onde o *Batch Normalization* é utilizado, o GSDTM não teve bom desempenho, sendo superado pelos métodos ProdLDA e NVDM na coleção 20newsgroups e por todos os métodos na RCV1-V2. Logo, pode-se concluir que apesar dos bons resultados obtidos nos experimentos de qualidade de tópicos e de avaliação do modelo gerador, o GSDTM não supera todos os métodos comparados, sendo preferível, para esta tarefa, um método focado em representação de documentos, como o DocNADE.

Já o método LMDTM não se mostrou competitivo nesta tarefa na coleção 20newsgroups, uma vez que ele foi superado em todos os cenários possíveis. Entretanto, observa-se que o desempenho deste método se torna altamente competitivo na coleção

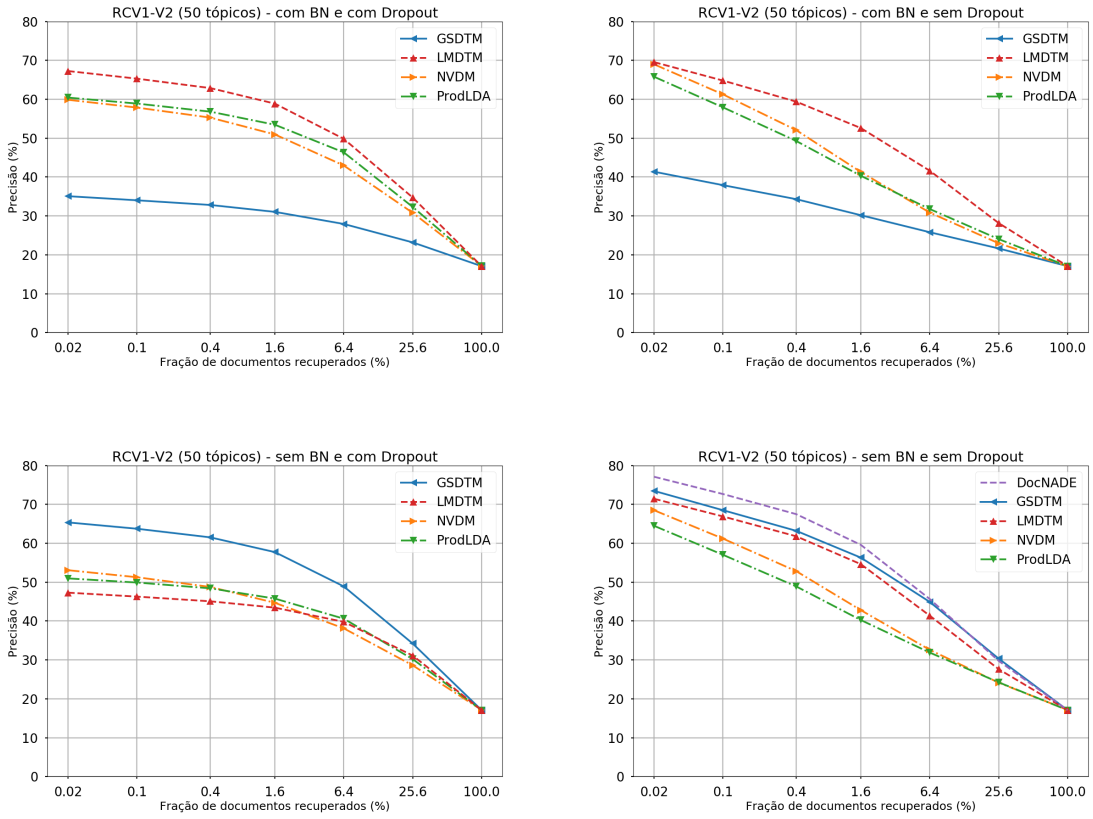


Figura 5.8: Resultado da avaliação de recuperação de documentos na base de dados RCV1-v2 com 50 tópicos.

RCV1-V2, onde ele supera os demais métodos baseados em Autocodificadores Variacionais, nos cenários onde o *Batch Normalization* e 50 tópicos são utilizados. Isto indica que o LMDTM consegue operar bem em duas situações bem definidas: quando a coleção de dados possui um grande número de instâncias e quando o *Batch Normalization* é adotado. Deste modo, a maior complexidade no modelo do LMDTM teve efeitos positivos na tarefa de recuperação de documentos utilizando a coleção RCV1-v2.

Por fim, o método DocNADE obteve o melhor resultado no geral quando considerados os cenários onde o *Batch Normalization* e o *Dropout* não são utilizados. Logo, pode-se conjecturar que modelos baseados em redes neurais estruturadas em modelos probabilísticos não direcionados, como o DocNADE, geram melhores representações vetoriais latentes de documentos do que modelos baseados em Autocodificadores Variacionais, embora experimentos mais detalhados tenham que ser feitos em trabalhos futuros para que haja uma conclusão definitiva. Esta é uma constatação plausível, visto que estes modelos são estruturados para trabalhar com modelagem de documentos, diferentemente dos Autocodificadores Variacionais.

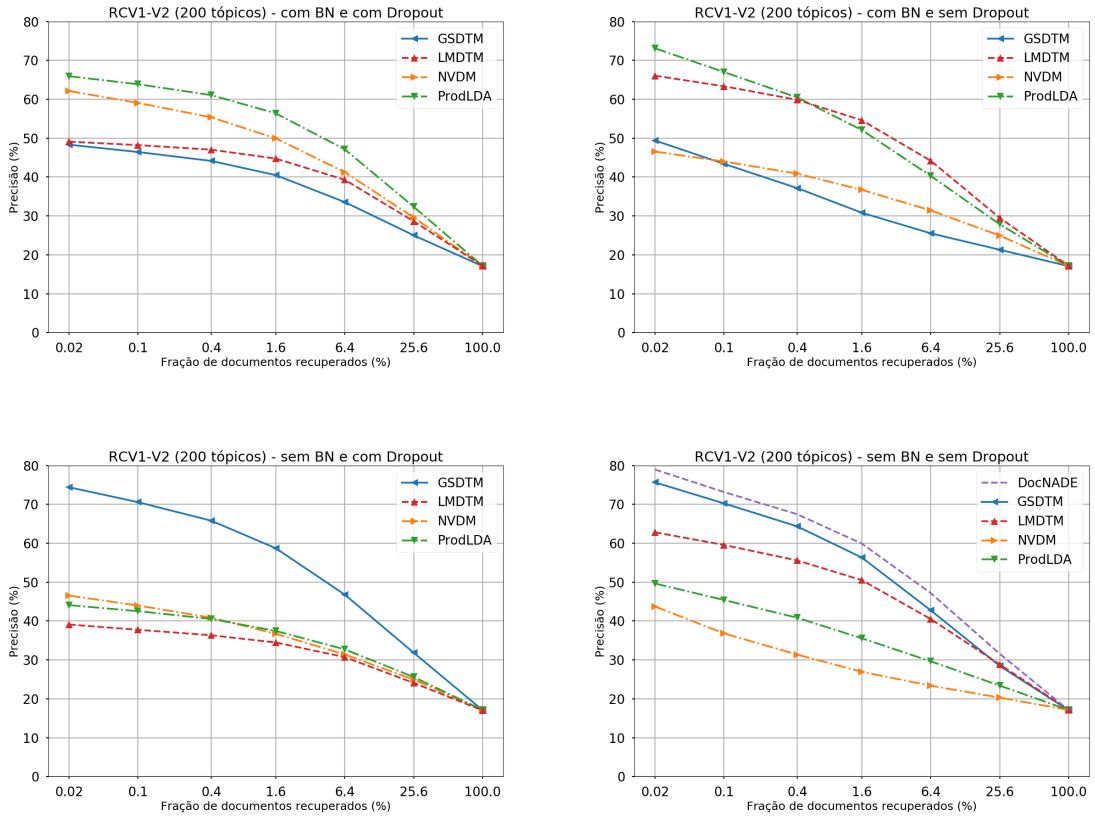


Figura 5.9: Resultado da avaliação de recuperação de documentos na base de dados RCV1-V2 com 200 tópicos.

5.3.4 Inspeção qualitativa

Com a finalidade de complementar as conclusões dos métodos analisados, apresenta-se também um estudo qualitativo com o objetivo de verificar se o GSDTM, o LMDTM e os outros métodos são capazes de aprender relações semânticas úteis. A fim de realizar este estudo, seguiu-se a mesma abordagem descrita na Subseção 5.3.1 e extraiu-se o ranqueamento de tópicos utilizando a matriz de pesos $\mathbf{W}$. Adicionalmente, foram analisados os *embeddings* de termos gerados pelos métodos propostos com o intuito de complementar este estudo.

As Figuras 5.10, 5.11, 5.12, 5.13, 5.14, 5.15, 5.16 e 5.17 mostram os tópicos extraídos dos métodos adotados como *baseline*, sob a forma de nuvem de palavras. Em quase todos os casos, exceto no DocNADE, foram utilizados o *Batch Normalization* e *Dropout*, uma vez que o uso destas técnicas elevou em algum grau o nível da coerência média de tópicos em todos os métodos analisados, como pode ser visto na Seção 5.3.1. Além disso, este experimento foi realizado na coleção de dados 20newsgroups utilizando

50 tópicos, adotando-se o mesmo conjunto de parâmetros utilizados no experimento de avaliação de tópicos. Em seguida, foram extraídos 50 termos de cada tópico que possuem os maiores índices de coerência média de tópicos. Os termos presentes nas nuvens de palavras consistem em palavras presentes no topo do ranqueamento de tópicos, onde o termo com maior tamanho é àquele que possui maior *score*. Este formato de exibição das palavras foi adotado com o intuito de facilitar a visualização da importância de cada termo no tópico analisado.

Em virtude do caráter subjetivo da comparação entre tópicos, não são realizados nesta inspeção qualitativa relatos afirmando que um tópico extraído por um método obteve ganho ou perda em relação a outro. Apenas são apresentados os valores de NPMI do tópico apresentado e um breve relato sobre os tópicos extraídos referente a cada método.

No geral, os métodos propostos GSDTM e LMDTM geram tópicos coerentes, assim como os demais métodos avaliados. Como pode ser visto, existe um tópico bem definido para cada figura. Nas Figuras 5.10 e 5.11, é possível perceber que o tópico é sobre criptografia, embora existam muitos termos dentro desta temática relacionados com outros tópicos, como hardware e software. Neste tópico, os métodos GSDTM e ProdLDA obtiveram os melhores índices de NPMI (0.29 e 0.27, respectivamente), apresentando termos altamente correlacionados. Logo após, o DocNADE obteve um tópico com NPMI igual a 0.15. Também o LMDTM obteve um NPMI igual a 0.10, bem abaixo dos valores apresentados pelo GSDTM e LMDTM. Ainda, pode-se notar que o tópico extraído pelo método LMDTM contém poucas palavras que divergem da temática de criptografia, como "*water*" e "*kings*". No NVDM, que apresentou o menor índice de NPMI, o número de palavras divergentes parece ser maior. Por exemplo, "*missing*" e "*spirit*" claramente não possuem correlação com o tópico analisado.

Nos tópicos de religião (Figuras 5.12 e 5.13) e hardware (Figuras 5.14 e 5.15), observa-se o mesmo comportamento apresentado no de criptografia. Nestes tópicos, o ProdLDA obteve valores de NPMI levemente maiores do que o GSDTM, enquanto que o DocNADE obteve um NPMI mais baixo que o GSDTM e o ProdLDA e o LMDTM manteve um índice de NPMI maior que NVDM. Por fim, o ProdLDA apresentou um valor de NPMI maior do que o GSDTM no tópico de conflitos étnicos (Figuras 5.16 e 5.17), embora as palavras extraídas por ambos os métodos neste tópico sejam similares.

Também é apresentado nesta seção uma análise qualitativa das representações semânticas (*embeddings*) provenientes da matriz de pesos $\mathbf{W}$. Os cenários de teste foram os mesmos utilizados nos experimentos de análise de tópicos. Com o objetivo de representar os *embeddings* em duas dimensões, utilizou-se o algoritmo PCA (*Principal Component Analysis*, ou Análise Principal de Componentes, em língua portuguesa).

Os gráficos 5.18a, 5.18b, 5.19a, 5.19b e 5.20a exibem respectivamente o espaço de *embedding* dos métodos GSDTM, LMDTM, NVDM, ProdLDA e DocNADE, reduzidos a duas dimensões pelo método PCA. Adicionalmente, extraiu-se as cinco palavras mais próximas no *embedding* em relação às representações vetoriais de sete palavras-chaves ("weapons", "medical", "encryption", "space", "religion", "political", "hardware"), cada uma representando um tópico distinto. Para calcular a similaridade entre as representações vetoriais, foi utilizado a distância via cosseno. Finalmente, os termos mais próximos às palavras-chaves são apresentados na Tabela 5.10. Pode-se ver na tabela que todos os métodos foram capazes de aprender relações semânticas úteis entre as palavras. Assim como na inspeção qualitativa de tópicos, não é possível afirmar qual método obteve melhores resultados, uma vez que esta análise possui caráter subjetivo.

(a) GSDTM (NPMI do tópico=0.23)

(b) LMDTM (NPMI do tópico=0.10)

(c) NVDM (NPMI do tópico=0.03)

Figura 5.10: Tópico de criptografia.

(a) ProdLDA (NPMI do tópico=0.20)

(b) DocNADE (NPMI do tópico=0.15)

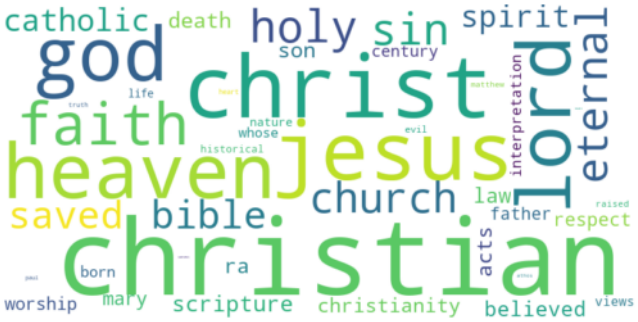
Figura 5.11: Tópico de criptografia (continuação).

(a) GSDTM (NPMI do tópico=0.27)

(b) LMDTM (NPMI do tópico=0.20)

(c) NVDM (NPMI do tópico=0.08)

Figura 5.12: Tópico de religião.

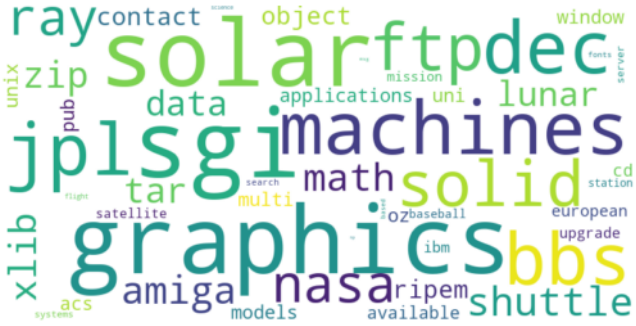


(a) ProdLDA (NPMI do tópico=0.29)

(b) DocNADE (NPMI do tópico=0.22)

Figura 5.13: Tópico de religião (continuação).

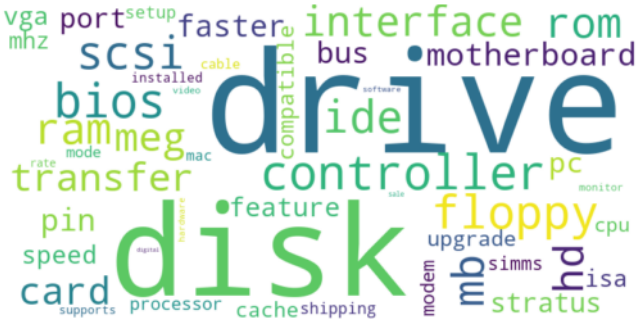
(a) GSDTM (NPMI do tópico=0.37)



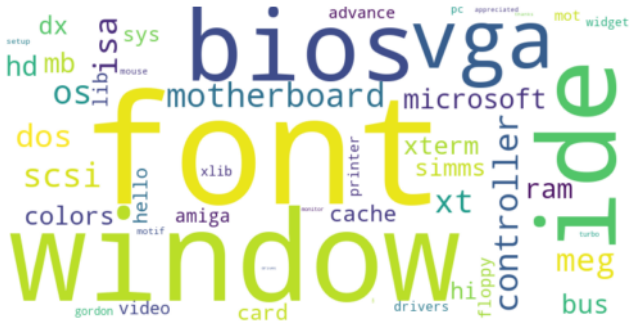
(b) LMDTM (NPMI do tópico=0.15)

(c) NVDM (NPMI do tópico=0.11)

Figura 5.14: Tópico de Hardware.



(a) ProdLDA (NPMI do tópico=0.39)



(b) DocNADE (NPMI do tópico=0.22)

Figura 5.15: Tópico de Hardware (continuação).

(a) GSDTM (NPMI do tópico=0.24)

(b) LMDTM (NPMI do tópico=0.10)

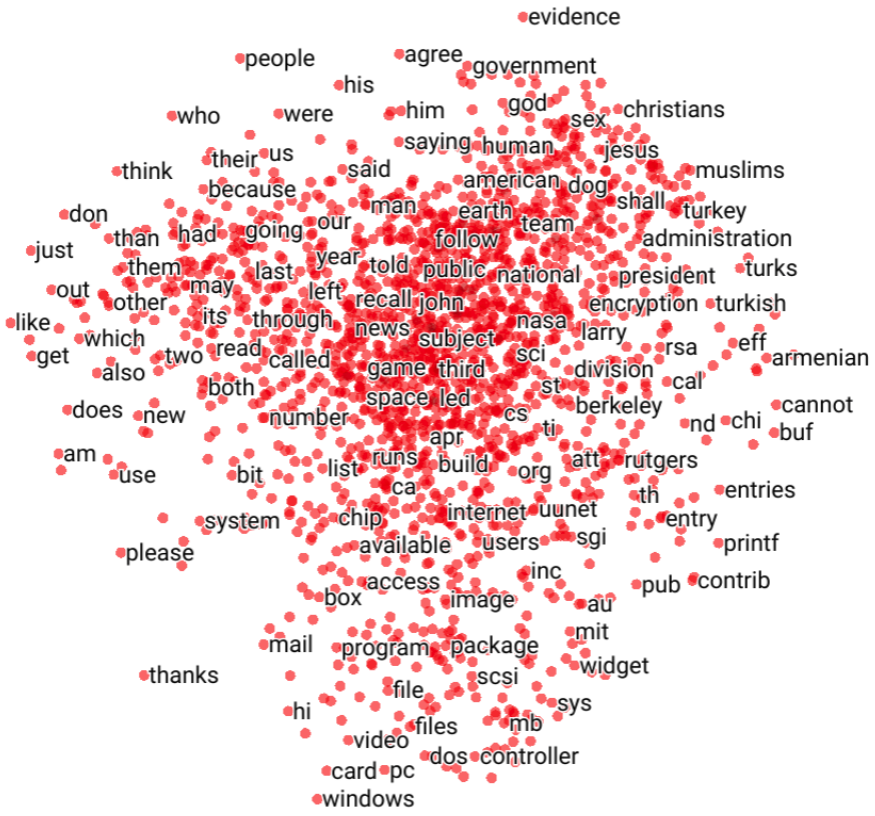
(c) NVDM (NPMI do tópico=0.15)

Figura 5.16: Tópico de Conflitos Étnicos.

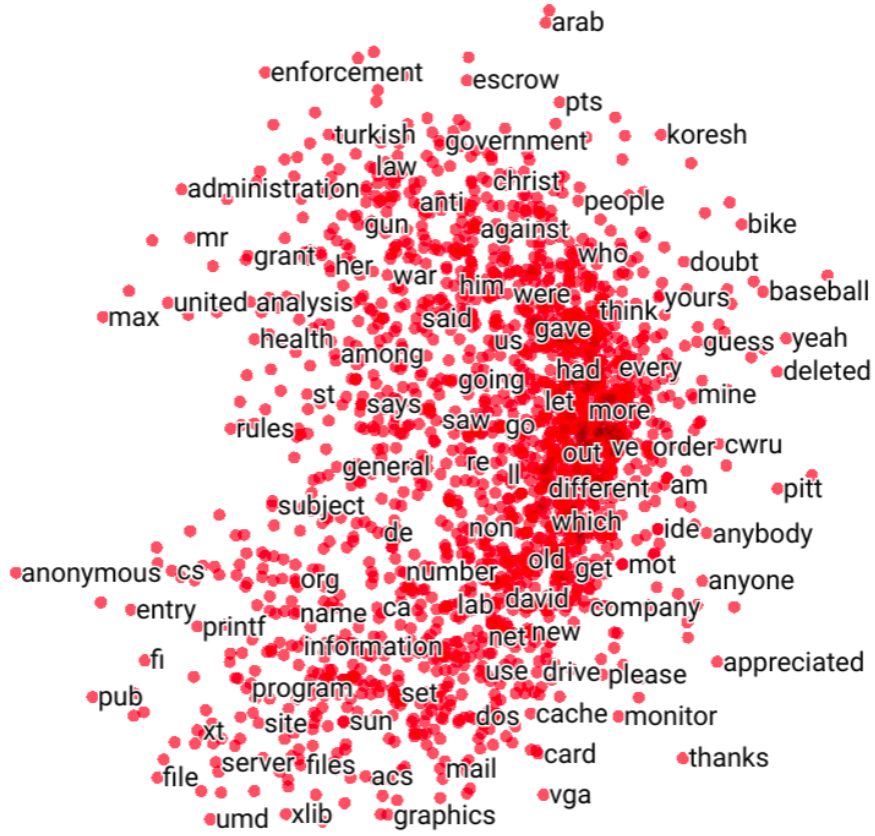
(a) ProdLDA (NPMI do tópico=0.39)

(b) DocNADE (NPMI do tópico=0.24)

Figura 5.17: Tópico de Conflitos Étnicos (continuação).

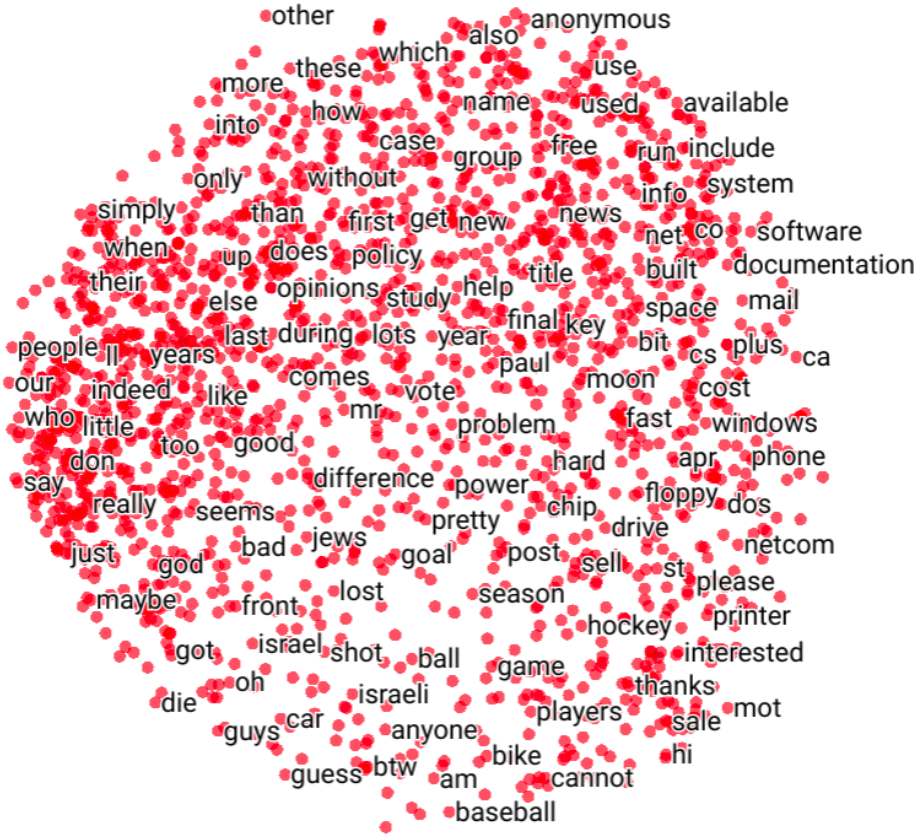


(a) GSDTM

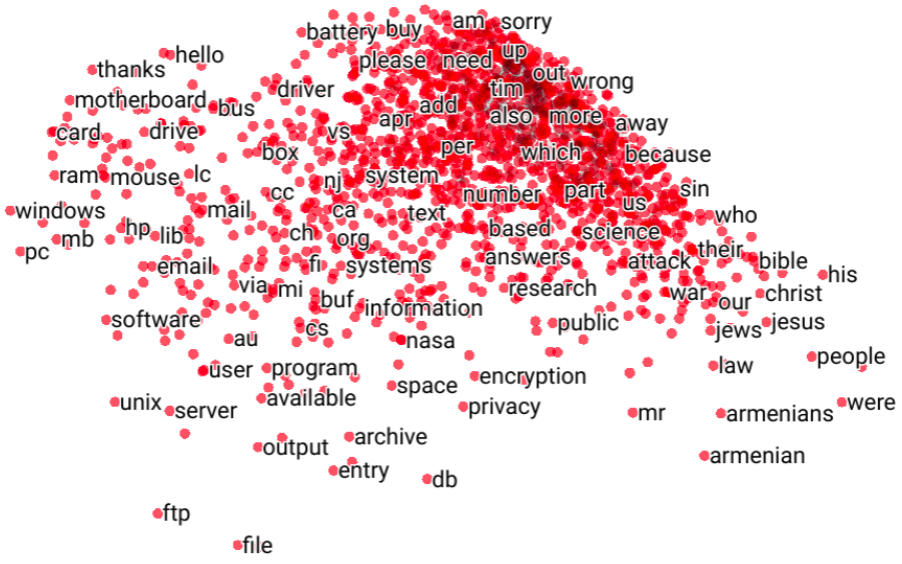


(b) LMDTM

Figura 5.18: Representação dos *Embeddings* dos métodos propostos, utilizando a coleção 20newsgroups e 50 tópicos.



(a) NVDM



(b) ProdLDA

Figura 5.19: Representação dos métodos concorrentes baseados em Autocodificadores Variacionais, utilizando a coleção 20newsgroups e 50 tópicos.

(a) DocNADE

Figura 5.20: Representação do método DocNADE, utilizando a coleção 20newsgroups e 50 tópicos.

	weapons	**medical**	**encryption**	**space**	**religion**	**political**	**hardware**
G	guns	reported	cryptography	satellite	beliefs	military	software
S	armed	health	key	commercial	religious	authorities	color
D	criminal	aids	messages	nasa	christian	nation	bit
T	minority	risk	secure	development	religions	historical	features
M	murder	study	des	lauch	belief	commited	documentation
L	armed	health	secure	orbit	religions	role	mac
M	guns	center	agencies	mission	religious	government	software
D	defense	washington	security	rocket	beliefs	citizens	pc
T	innocent	patients	cryptography	shuttle	christianity	rights	ibm
M	justice	national	rsa	earth	stated	militia	compatible
N	guns	page	escrow	rocket	religious	history	features
V	firearms	health	enforcement	satellite	religions	community	standard
D	armed	patients	agences	shuttle	beliefs	role	allows
M	gun	disease	industry	nasa	belief	anti	faster
	constitution	aids	clipper	mission	christian	states	mac
P.	guns	health	cryptography	satellite	religious	citizens	software
L	armed	congress	escrow	nasa	belief	law	fax
D	criminal	firearms	des	sci	christians	government	mode
A	majority	reported	rsa	technology	christian	military	application
	laws	administration	privacy	shuttle	christianity	policy	machines
D.	armed	effects	crypto	solar	religions	minority	format
N	crime	disease	encrypted	orbit	claim	military	dos
A	gun	dangerous	secure	japanese	evidence	state	mac
D	firearms	medicine	nsa	dc	christians	civil	pc
E	weapons	however	escrow	mission	church	law	interface

Tabela 5.10: Representação das 5 palavras mais próximas no espaço semântico. A palavra no cabeçalho representa uma palavra-chave e os cinco termos seguintes são as palavras cujas representações vetoriais são as mais próximas desta palavra-chave, ordenados pela distância via cosseno. Os nomes dos métodos ProdLDA e DocNADE foram respectivamente abreviados para P.LDA e D.NADE para que não houvesse prejuízo à estética da tabela.

5.4 Considerações Finais

Foram apresentados neste capítulo os resultados provenientes da comparação dos métodos propostos com os demais métodos, bem como uma inspeção qualitativa com o objetivo de verificar os tópicos gerados e a representação distribuída (*embedding*) de palavras. Por meio deste conjunto de experimentos, pode-se definir três conclusões importantes. Em primeiro lugar, o GSDTM supera os demais métodos quando o *Batch Normalization* e o *Dropout* são empregados, em termos de coerência de tópicos. Em outros cenários, o GSDTM obtém, no geral, os melhores resultados quando comparado

com os outros métodos. Em segundo, o LMDTM não superou os demais métodos, porém ele é competitivo em alguns cenários, principalmente quando aplicado na base RCV1-v2. Por último, existe um compromisso (*trade-off*) entre a aplicação do *Batch Normalization* e *Dropout* e a não aplicação destas técnicas, de modo que o uso melhora o índice de coerência de tópicos, mas piora o nível de perplexidade, enquanto que o efeito inverso ocorre quando nenhuma destas técnicas são aplicadas.

Os resultados comparativos do GSDTM e do LMDTM com os métodos adotados no *baseline* (NVDM, ProdLDA e DocNADE), assim como as hipóteses de pesquisas e conclusões de pesquisa, foram publicados no *International Joint Conference on Neural Networks*(IJCNN) 2018 [Silveira et al., 2018].

No próximo capítulo serão abordadas as considerações finais, relacionando as hipóteses de pesquisa com as conclusões obtidas no processo de experimentação. Também serão apresentadas possíveis direções futuras de pesquisa que podem ser tomadas a partir deste trabalho.

6

Considerações Finais

Neste capítulo, são feitas as considerações finais baseadas na hipótese de pesquisa e nos resultados obtidos, além de apresentar possíveis direções futuras para projetos de pesquisa que utilizem Autocodificadores Variacionais em modelagem de tópicos. Desta forma, expõe-se na Seção 6.1 as conclusões obtidas neste trabalho de pesquisa. Além disso, uma discussão sobre as limitações dos métodos propostos é relatada na Seção 6.2. Por fim, mostra-se na Seção 6.3 as possíveis direções futuras deste trabalho de pesquisa.

6.1 Conclusões

Neste trabalho, foi apresentada uma visão geral sobre modelagem de tópicos, além de duas propostas de modelos de tópicos baseados em Autocodificadores Variacionais, denominados GSDTM e LMDTM. Também foram mostrados experimentos comparando a qualidade deles com modelos de tópicos considerados o estado da arte nas bases de dados testadas (20newsgroups e RCV1-v2), que são o NVDM [Miao et al., 2016], ProdLDA [Srivastava & Sutton, 2017] e DocNADE [Larochelle & Lauly, 2012].

Baseado na hipótese de que a utilização de distribuições contínuas capazes de se ajustar a dados categóricos podem codificar melhor os espaço latente de tópicos, uma vez que tópicos são elementos inerentemente categóricos, foi proposto o GSDTM, cuja distribuição latente é denominada de *Gumbel-Softmax* e é capaz de gerar amostras com distribuição de probabilidade mais próxima de uma categórica do que outras distribuições contínuas, como as distribuições Normal e Normal-Logística. Por meio

de experimentos, concluiu-se que há indícios de que a hipótese é verdadeira, já que houve melhorias nos níveis de coerência média de tópicos e perplexidade, superando os métodos considerados estado da arte em modelagem de tópicos, quando considerado o uso das técnicas *Batch Normalization* e *Dropout* em coerência de tópicos e o não uso destas na métrica de perplexidade. Por exemplo, em um dos cenários de teste, houve redução do nível de perplexidade de aproximadamente 40% sobre o método que obteve a segunda melhor colocação. Desta forma, observa-se uma significativa contribuição do método na área de modelagem de tópicos.

Outra hipótese de pesquisa que foi conjecturada é que uma mistura de distribuições pode prover melhores resultados em modelagem de tópicos, uma vez que a complexidade maior deste tipo de distribuição teoricamente contribui para melhorar a aprendizagem em coleções cujos dados são complexos, como é o caso das coleções de texto. Desta forma, foi proposto o método LMDTM, cujo diferencial é a utilização de um modelo de mistura composto por distribuições Normais-Logísticas, com o objetivo de aprender melhor as representações latentes de tópicos assimilando padrões complexos presentes nas coleções de dados. Em contraste com o método GSDTM, os resultados não sustentam esta hipótese de pesquisa, já que a complexidade do modelo se tornou, na prática, prejudicial ao processo de treinamento. Entretanto, é importante mencionar que em alguns cenários de teste, o LMDTM se mostrou competitivo, principalmente na coleção RCV1-v2. Portanto, este método tem potencial para competir com os demais métodos em trabalhos futuros, desde que haja sucesso na aplicação do modelo de mistura de distribuições em Autocodificadores Variacionais de maneira mais simples do que o realizado no LMDTM.

Por meio do processo de experimentação, constatou-se que existe um compromisso (*trade-off*) entre a utilização do *Batch Normalization* e *Dropout* e a não utilização destas técnicas no nível de coerência média de tópicos e na perplexidade. Mais especificamente, quando as duas técnicas são aplicadas em conjunto, a coerência média de tópicos aumenta, da mesma forma que a perplexidade do método aumenta (quanto menor é a perplexidade, melhor a qualidade do modelo gerador). O efeito inverso é registrado quando nenhuma destas técnicas é aplicada. Desta forma, conclui-se que a aplicação de *Batch Normalization* e *Dropout* é benéfica quando o objetivo é obter tópicos mais coesos, ao passo que a não aplicação destas técnicas é benéfica quando o objetivo é obter modelos geradores com melhor qualidade.

Por fim, outras contribuições menores foram realizadas neste trabalho de pesquisa. Por exemplo, identificou-se que modelos de tópicos baseados em Autocodificadores Variacionais não obtêm um bom desempenho na tarefa de recuperação de documentos quando comparados com o DocNADE. Ainda, foi mostrado por meio de

inspeção qualitativa que estes métodos obtêm tópicos com alta relação semântica, ou seja, tópicos que fazem sentido a um observador humano, além de gerar um *embedding* com grande correlação semântica entre as palavras.

6.2 Limitações dos métodos propostos

Assim como todos os métodos presentes na literatura de modelagem de tópicos, os métodos GSDTM e LMDTM possuem limitações. Os problemas específicos do GSDTM se concentram na instabilidade do treinamento do método quando o *Batch Normalization* não é utilizado no cenário composto pela coleção maior (RCV1-v2). Logo, o método depende da adoção do *Batch Normalization* para obter razoáveis índices de coerência de tópicos em coleções mais complexas. Também o método apresenta uma limitação por supor que a variável latente z é independente da variável y, o que torna o modelo probabilístico mais simples no contexto dos Autocodificadores Variacionais, mas que em contrapartida aumenta as suas limitações. Quanto ao *LMDTM*, o método apresenta problemas de instabilidade quando o *Dropout* é utilizado sem o uso de *Batch Normalization*, causado pelo crescimento excessivo do valor do gradiente, problema que é acentuado em coleções de dados menores e mitigado com o uso de *Batch Normalization*. Além disso, o tempo de treinamento cresce linearmente em função do número de componentes. Desta forma, se K componentes forem empregadas, o tempo de treinamento será aproximadamente K vezes maior no LMDTM do que em outros métodos. Por fim, o método LMDTM assume que a distribuição *a priori* $p(y)$ do modelo de mistura é uniforme. Isto possibilita a incorporação de uma aproximação do modelo de mistura de Normais-Logísticas em Autocodificadores Variacionais, mas limita o controle da proporção desta mistura, ocasionando limitações na expressividade do modelo probabilístico proposto.

Outras limitações são compartilhadas com todos os outros métodos baseados em Autocodificadores Variacionais. Por exemplo, não é possível até o momento utilizar distribuições nos dados latentes que não tenham um truque de reparametrização conhecido. Também o modelo probabilístico dos Autocodificadores Variacionais é inflexível quando comparado com outros modelos probabilísticos tal como o LDA, dificultando a adaptação destes modelos para outras tarefas ou tornando inviável o uso de outros métodos de inferência. Além disso, esses modelos têm, no geral, desempenho inferior na tarefa de recuperação de documentos em relação ao DocNADE. A exceção é o método GSDTM, que se mostrou competitivo em grande parte dos cenários analisados.

6.3 Trabalhos futuros

Os métodos propostos, assim como grande parte dos modelos de tópicos existentes, utilizam uma matriz de frequências de documento-termo como entrada de dados. Este tipo de entrada não considera a sequência das palavras no texto, obrigando os métodos descartarem esta característica que pode ser útil ao processo de aprendizagem de modelos de tópicos. Desta forma, busca-se a incorporação da sequência entre as palavras em trabalhos futuros como forma de melhorar a qualidade da modelagem de tópicos.

Também se pretende estudar com maior riqueza de detalhes o comportamento do *Batch Normalization* e do *Dropout* e, assim, propor um novo modelo que seja capaz de balancear a qualidade do modelo gerador com a coerência de tópicos, de modo que se tenham valores altos de coerência de tópicos, ao mesmo tempo que se tenham índices baixos de perplexidade. Adicionalmente, estuda-se no futuro propor um *framework* geral para modelos de tópicos baseados em Autocodificadores Variacionais, o que permitiria explorar diferentes opções de modelagem de forma mais sistemática.

Em relação ao método LMDTM, almeja-se melhorar o seu custo de treinamento e estudar mudanças na estrutura deste método que possibilitem obter resultados mais competitivos em relação aos demais métodos. Uma alternativa para alcançar este objetivo é empregar o processo estatístico denominado *stick-breaking processes*, que permitiria a adoção de mistura de distribuições sem aumentar consideravelmente a complexidade do modelo.

Referências Bibliográficas

Aitchison, J. & Shen, S. M. (1980). Logistic-normal distributions: Some properties and uses. *Biometrika*, 67(2):261--272. ISSN 00063444.

Alghamdi, R. & Alfalqi, K. (2015). A survey of topic modeling in text mining.

Arnab, A.; Zheng, S.; Jayasumana, S.; Romera-Paredes, B.; Larsson, M.; Kirillov, A.; Savchynskyy, B.; Rother, C.; Kahl, F. & Torr, P. H. S. (2018). Conditional random fields meet deep neural networks for semantic segmentation: Combining probabilistic graphical models with deep learning for structured prediction. *IEEE Signal Processing Magazine*, 35(1):37–52. ISSN 1053-5888.

Arora, R. & Ravindran, B. (2008). Latent dirichlet allocation based multi-document summarization. Em *Proceedings of the second workshop on Analytics for noisy unstructured text data*, pp. 91--97. ACM.

Azzopardi, L.; Girolami, M. & van Rijsbergen, C. J. (2004). Topic based language models for ad hoc information retrieval. Em *2004 IEEE International Joint Conference on Neural Networks (IEEE Cat. No.04CH37541)*, volume 4, pp. 3281–3286 vol.4. ISSN 1098-7576.

Bayes, M. & Price, M. (1763). An essay towards solving a problem in the doctrine of chances. by the late rev. mr. bayes, f. r. s. communicated by mr. price, in a letter to john canton, a. m. f. r. s. *Philosophical Transactions (1683-1775)*, 53:370–418. ISSN 02607085.

Beal, M. J. et al. (2003). *Variational algorithms for approximate Bayesian inference*.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. Springer-Verlag New York, Inc., Secaucus, NJ, USA. ISBN 0387310738.

Blei, D. M. (2011). Variational inference. *Lecture from Princeton, https://www. cs. princeton. edu/courses/archive/fall11/cos597C/lectures/variational-inference-i. pdf*.

Blei, D. M. (2012). Probabilistic topic models. *Commun. ACM*, 55(4):77--84. ISSN 0001-0782.

Blei, D. M. & Lafferty, J. D. (2006). Dynamic topic models. Em *Proceedings of the 23rd international conference on Machine learning*, pp. 113--120. ACM.

Blei, D. M. & Lafferty, J. D. (2007). A correlated topic model of science. *The Annals of Applied Statistics*, pp. 17--35.

Blei, D. M.; Ng, A. Y. & Jordan, M. I. (2003a). Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993--1022. ISSN 1532-4435.

Blei, D. M.; Ng, A. Y. & Jordan, M. I. (2003b). Latent dirichlet allocation. *J. Mach. Learn. Res.*, 3:993--1022. ISSN 1532-4435.

Boukadida, H.; Hassen, N.; Gafsi, Z. & Besbes, K. (2011). A highly time-efficient digital multiplier based on the a2 binary representation. 3.

Box, G. E. & Tiao, G. C. (2011). *Bayesian inference in statistical analysis*, volume 40. John Wiley & Sons.

Chen, J.-Y.; Zheng, H.-T.; Jiang, Y.; Xia, S.-T. & Zhao, C.-Z. (2018). A probabilistic model for semantic advertising. *Knowledge and Information Systems*. ISSN 0219-3116.

Chen, X.; Duan, Y.; Houthooft, R.; Schulman, J.; Sutskever, I. & Abbeel, P. (2016). InfoGAN: Interpretable representation learning by information maximizing generative adversarial nets. Em *NIPS'16*, pp. 2172--2180. Curran Associates, Inc.

Chen, X.; Hu, X.; Shen, X. & Rosen, G. (2010). Probabilistic topic modeling for genomic data interpretation. Em *2010 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pp. 149–152. ISSN .

de Oliveira, G. L. & Loschi, R. H. (2013). Inferência bayesiana em modelos de mistura finita: uma aplicação à estimação de densidades. *Matemática e Estatística em Foco*, 1(2).

Deerwester, S.; Dumais, S. T.; Furnas, G. W.; Landauer, T. K. & Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391.

Devroye, L. (1986). Sample-based non-uniform random variate generation. Em *Proceedings of the 18th conference on Winter simulation*, pp. 260--265. ACM.

Dilokthanakul, N.; Mediano, P. A. M.; Garnelo, M.; Lee, M. C. H.; Salimbeni, H.; Arulkumaran, K. & Shanahan, M. (2016). Deep unsupervised clustering with gaussian mixture variational autoencoders. *CoRR*, abs/1611.02648.

DiMaggio, P.; Nag, M. & Blei, D. (2013). Exploiting affinities between topic modeling and the sociological perspective on culture: Application to newspaper coverage of u.s. government arts funding. *Poetics*, 41(6):570 – 606. ISSN 0304-422X. Topic Models and the Cultural Sciences.

Doersch, C. (2016). Tutorial on variational autoencoders. *arXiv preprint arXiv:1606.05908*.

Duchi, J.; Hazan, E. & Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *Journal of Machine Learning Research*, 12(Jul):2121--2159.

Gelman, A.; Carlin, J. B.; Stern, H. S.; Dunson, D. B.; Vehtari, A. & Rubin, D. B. (2013). *Bayesian data analysis*. CRC press.

Goodfellow, I.; Bengio, Y.; Courville, A. & Bengio, Y. (2016). *Deep learning*, volume 1. MIT press Cambridge.

Griffiths, T. L.; Jordan, M. I.; Tenenbaum, J. B. & Blei, D. M. (2004). Hierarchical topic models and the nested chinese restaurant process. Em *Advances in neural information processing systems*, pp. 17--24.

Gumbel, E. J. (1954). The maxima of the mean largest value and of the range. *The Annals Math. Statistics*, 25(1):76--84.

He, X.; Zemel, R. S. & Carreira-Perpinan, M. A. (2004). Multiscale conditional random fields for image labeling. Em *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, volume 2, pp. II–695–II–702 Vol.2. ISSN 1063-6919.

Hinton, G. E. (2002). Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771--1800.

Hinton, G. E. & Zemel, R. S. (1994). Autoencoders, minimum description length and helmholtz free energy. Em *Advances in neural information processing systems*, pp. 3--10.

Hofmann, T. (1999). Probabilistic latent semantic analysis. Em *Proceedings of the Fifteenth conference on Uncertainty in artificial intelligence*, pp. 289--296. Morgan Kaufmann Publishers Inc.

Huang, J. & Malisiewicz, T. (2009). Fitting a hierarchical logistic normal distribution.

I. W. Lang, R.; Warwick, K. & Ay England, R. (2002). A dynamic neural network for continual.

Ioffe, S. & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. Em *International Conference on Machine Learning*, pp. 448--456.

Jang, E.; Gu, S. & Poole, B. (2016). Categorical reparameterization with gumbel-softmax. cite arxiv:1611.01144.

Jensen, J. L. W. V. (1906). Sur les fonctions convexes et les inégalités entre les valeurs moyennes. *Acta mathematica*, 30(1):175--193.

Jordan, M. I. (2004). Graphical models. *Statistical Science*, 19(1):140–155. ISSN 08834237.

Jordan, M. I.; Ghahramani, Z.; Jaakkola, T. S. & Saul, L. K. (1999). An introduction to variational methods for graphical models. *Machine Learning*, 37(2):183--233. ISSN 1573-0565.

Khatoonabadi, S. H. & Bajic, I. V. (2013). Video object tracking in the compressed domain using spatio-temporal markov random fields. *IEEE Transactions on Image Processing*, 22(1):300–313. ISSN 1057-7149.

Kindermann, R. P. (1980). Asymptotic comparisons of functionals of brownian motion and random walk. *The Annals of Probability*, 8(6):1135–1147. ISSN 00911798.

Kingma, D. P. (2015). Variational auto-encoders and extensions.

Kingma, D. P. & Ba, J. (2014). Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Kingma, D. P. & Welling, M. (2013). Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Larochelle, H. & Lauly, S. (2012). A neural autoregressive topic model. Em *NIPS'12*, pp. 2717--2725. Curran Associates, Inc.

Larochelle, H. & Murray, I. (2011). The neural autoregressive distribution estimator. Em *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pp. 29--37.

Larrañaga, P. (2002). An introduction to probabilistic graphical models. Em *Estimation of Distribution Algorithms*, pp. 27--56. Springer.

Lau, J. H.; Newman, D. & Baldwin, T. (2014). Machine reading tea leaves: Automatically evaluating topic coherence and topic model quality. Em Bouma, G. & Parmentier, Y., editores, *EACL'14*, pp. 530--539. The Association for Computer Linguistics.

LeCun, Y.; Touresky, D.; Hinton, G. & Sejnowski, T. (1988). A theoretical framework for back-propagation. Em *Proceedings of the 1988 connectionist models summer school*, pp. 21--28. CMU, Pittsburgh, Pa: Morgan Kaufmann.

Li, W. & McCallum, A. (2006). Pachinko allocation: Dag-structured mixture models of topic correlations. Em *Proceedings of the 23rd international conference on Machine learning*, pp. 577--584. ACM.

Lu, B.; Ott, M.; Cardie, C. & Tsou, B. K. (2011). Multi-aspect sentiment analysis with topic models. Em *ICDMW '11*, pp. 81--88, Washington, DC, USA. IEEE Computer Society.

Luce, R. D. (1959). *Individual Choice Behavior a Theoretical Analysis*. John Wiley and sons.

MacKay, D. J. (2003). *Information theory, inference and learning algorithms*. Cambridge university press.

Maddison, C. J.; Mnih, A. & Teh, Y. W. (2016). The concrete distribution: A continuous relaxation of discrete random variables. *CoRR*, abs/1611.00712.

Maddison, C. J.; Tarlow, D. & Minka, T. (2014). A* sampling. Em *Advances in Neural Information Processing Systems*, pp. 3086--3094.

Miao, Y.; Yu, L. & Blunsom, P. (2016). Neural variational inference for text processing. Em *ICML'16*, pp. 1727--1736. JMLR.org.

Mikolov, T.; Sutskever, I.; Chen, K.; Corrado, G. S. & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. Em *Advances in neural information processing systems*, pp. 3111--3119.

Minksy, M. & Papert, S. (1969). Perceptrons: An introduction to computation geometry. *MIT press*, 200:355--368.

Mohr, J. W. & Bogdanov, P. (2013). Introduction—topic models: What they are and why they matter. *Poetics*, 41(6):545 – 569. ISSN 0304-422X. Topic Models and the Cultural Sciences.

Paisley, J.; Blei, D. M. & Jordan, M. I. (2012). Variational bayesian inference with stochastic search. Em *Proceedings of the 29th International Coference on International Conference on Machine Learning*, pp. 1363--1370. Omnipress.

Pearl, J. (1986). Fusion, propagation, and structuring in belief networks. *Artificial Intelligence*, 29(3):241 – 288. ISSN 0004-3702.

Pennington, J.; Socher, R. & Manning, C. (2014). Glove: Global vectors for word representation. Em *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1532--1543.

Rae, J. W.; Hunt, J. J.; Danihelka, I.; Harley, T.; Senior, A. W.; Wayne, G.; Graves, A. & Lillicrap, T. (2016). Scaling memory-augmented neural networks with sparse reads and writes. Em *NIPS'16*, pp. 3621--3629.

Ranganath, R.; Gerrish, S. & Blei, D. (2014). Black Box Variational Inference. Em Kaski, S. & Corander, J., editores, *Proceedings of the Seventeenth International Conference on Artificial Intelligence and Statistics*, volume 33 of *Proceedings of Machine Learning Research*, pp. 814--822, Reykjavik, Iceland. PMLR.

Resnik, P. (1999). Semantic similarity in a taxonomy: An information based measure and its application to problems of ambiguity in natural language. 11:95–130.

Rezende, D. J.; Mohamed, S. & Wierstra, D. (2014). Stochastic backpropagation and approximate inference in deep generative models. Em Xing, E. P. & Jebara, T., editores, *Proceedings of the 31st International Conference on Machine Learning*, volume 32 of *Proceedings of Machine Learning Research*, pp. 1278--1286, Bejing, China. PMLR.

Rosen-Zvi, M.; Griffiths, T.; Steyvers, M. & Smyth, P. (2004). The author-topic model for authors and documents. Em *Proceedings of the 20th conference on Uncertainty in artificial intelligence*, pp. 487--494. AUAI Press.

Rubin, T. N.; Chambers, A.; Smyth, P. & Steyvers, M. (2012). Statistical topic models for multi-label document classification. *Machine Learning*, 88(1):157--208. ISSN 1573-0565.

Rumelhart, D. E.; Hinton, G. E. & Williams, R. J. (1986). Learning representations by back-propagating errors. *nature*, 323(6088):533.

Salakhutdinov, R. & Hinton, G. E. (2009). Replicated softmax: an undirected topic model. Em *NIPS'09*, volume 22, pp. 1607--1614.

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. *Neural Networks*, 61:85 – 117. ISSN 0893-6080.

Seroussi, Y.; Zukerman, I. & Bohnert, F. (2014). Authorship attribution with topic models. *Comp. Ling.*, 40(2):269--310. ISSN 0891-2017.

Sichel, H. S. (1975). On a distribution law for word frequencies. *Journal of the American Statistical Association*, 70(351):542--547. ISSN 01621459.

Sillitto, G. P. (1969). Derivation of approximants to the inverse distribution function of a continuous univariate population from the order statistics of a sample. *Biometrika*, 56(3):641--650.

Silveira, D.; Carvalho, A.; Cristo, M. & Moens, M. (2018). Topic modeling using variational auto-encoders with gumbel-softmax and logistic-normal mixture distributions. Em *2018 International Joint Conference on Neural Networks, IJCNN 2018, Rio de Janeiro, Brazil, July 8-13, 2018*, pp. 1--8.

Srivastava, A. & Sutton, C. (2017). Autoencoding variational inference for topic models. *arXiv preprint arXiv:1703.01488*.

Srivastava, N.; Hinton, G.; Krizhevsky, A.; Sutskever, I. & Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929--1958.

Srivastava, N.; Salakhutdinov, R. R. & Hinton, G. E. (2013). Modeling documents with deep boltzmann machines. *arXiv preprint arXiv:1309.6865*.

Steyvers, M. & Griffiths, T. (2007). Probabilistic topic models. *Handbook of latent semantic analysis*, 427(7):424--440.

Tian, F.; Gao, B.; He, D. & Liu, T. (2016). Sentence level recurrent topic model: Letting topics speak for themselves. *CoRR*, abs/1604.02038.

Titov, I. & McDonald, R. (2008). A joint model of text and aspect ratings for sentiment summarization. *proceedings of ACL-08: HLT*, pp. 308--316.

van den Oord, A.; Kalchbrenner, N.; Espeholt, L.; kavukcuoglu, k.; Vinyals, O. & Graves, A. (2016). Conditional image generation with pixelcnn decoders. Em Lee, D. D.; Sugiyama, M.; Luxburg, U. V.; Guyon, I. & Garnett, R., editores, *Advances in Neural Information Processing Systems 29*, pp. 4790--4798. Curran Associates, Inc.

Wainwright, M. J.; Jordan, M. I. et al. (2008). Graphical models, exponential families, and variational inference. *Foundations and Trends® in Machine Learning*, 1(1–2):1--305.

Wallach, H. M. (2006). Topic modeling: beyond bag-of-words. Em *Proceedings of the 23rd international conference on Machine learning*, pp. 977--984. ACM.

Wang, C. & Blei, D. M. (2011). Collaborative topic modeling for recommending scientific articles. Em *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 448--456. ACM.

Wang, S. B.; Quattoni, A.; Morency, L. P.; Demirdjian, D. & Darrell, T. (2006). Hidden conditional random fields for gesture recognition. Em *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pp. 1521–1527. ISSN 1063-6919.

Wang, X.; McCallum, A. & Wei, X. (2007). Topical n-grams: Phrase and topic discovery, with an application to information retrieval. Em *Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on*, pp. 697--702. IEEE.

Wei, X. & Croft, W. B. (2006). LDA-based document models for ad-hoc retrieval. Em *SIGIR '06*, pp. 178--185, New York, USA. ACM.

Xie, P. & Xing, E. P. (2013). Integrating document clustering and topic modeling. *CoRR*, abs/1309.6874.

Xu, Y.; Tuttas, S.; Hoegner, L. & Stilla, U. (2018). Voxel-based segmentation of 3d point clouds from construction sites using a probabilistic connectivity model. *Pattern Recognition Letters*, 102:67 – 74. ISSN 0167-8655.

Yang, M.; Cui, T. & Tu, W. (2015). Ordering-sensitive and semantic-aware topic modeling. Em *AAAI'15*, pp. 2353--2359. AAAI Press.

Yellott, J. I. (1977). The relationship between luce's choice axiom, thurstone's theory of comparative judgment, and the double exponential distribution. *Journal of Mathematical Psychology*.

Zipf, G. K. (2013). *The psycho-biology of language: An introduction to dynamic philology*. Routledge.