Leandro Rezende Carneiro de Mendonça

Modelo Neural Recozido para a Representação Semântica de Documentos por meio de Vetores Contínuos

Goiânia

Novembro/2020

UNIVERSIDADE FEDERAL DE GOIÁS
ESCOLA DE ENGENHARIA ELÉTRICA, MECÂNICA E DE COMPUTAÇÃO

TERMO DE CIÊNCIA E DE AUTORIZAÇÃO (TECA) PARA DISPONIBILIZAR VERSÕES ELETRÔNICAS DE TESES

E DISSERTAÇÕES NA BIBLIOTECA DIGITAL DA UFG

Na qualidade de titular dos direitos de autor, autorizo a Universidade Federal de Goiás (UFG) a disponibilizar, gratuitamente, por meio da Biblioteca Digital de Teses e Dissertações (BDTD/UFG), regulamentada pela Resolução CEPEC nº 832/2007, sem ressarcimento dos direitos autorais, de acordo com a Lei 9.610/98, o documento conforme permissões assinaladas abaixo, para fins de leitura, impressão e/ou download, a título de divulgação da produção científica brasileira, a partir desta data.

O conteúdo das Teses e Dissertações disponibilizado na BDTD/UFG é de responsabilidade exclusiva do autor. Ao encaminhar o produto final, o autor(a) e o(a) orientador(a) firmam o compromisso de que o trabalho não contém nenhuma violação de quaisquer direitos autorais ou outro direito de terceiros.

1. Identificação do material bibliográfico

[ ] Dissertação [X] Tese

2. Nome completo do autor

Leandro Rezende Carneiro de Mendonça

3. Título do trabalho

“Modelo Neural Recozido para a Representação Semântica de Documentos por meio de Vetores Contínuos”

4. Informações de acesso ao documento (este campo deve ser preenchido pelo orientador)

Concorda com a liberação total do documento [X] SIM [ ] NÃO[1]

[1] Neste caso o documento será embargado por até um ano a partir da data de defesa. Após esse período, a possível disponibilização ocorrerá apenas mediante:
a) consulta ao(à) autor(a) e ao(à) orientador(a);
b) novo Termo de Ciência e de Autorização (TECA) assinado e inserido no arquivo da tese ou dissertação.
O documento não será disponibilizado durante o período de embargo.
Casos de embargo:
- Solicitação de registro de patente;
- Submissão de artigo em revista científica;
- Publicação como capítulo de livro;
- Publicação da dissertação/tese em livro.

Obs. Este termo deverá ser assinado no SEI pelo orientador e pelo autor.

Documento assinado eletronicamente por **LEANDRO REZENDE CARNEIRO DE MENDONÇA**, **Discente**, em 24/11/2020, às 17:30, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto

nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Gelson Da Cruz Junior**, **Professor do Magistério Superior**, em 25/11/2020, às 14:51, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

A autenticidade deste documento pode ser conferida no site https://sei.ufg.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **1701058** e o código CRC **5ABE0ED3**.

Referência: Processo nº 23070.044552/2020-53 SEI nº 1701058

Leandro Rezende Carneiro de Mendonça

Modelo Neural Recozido para a Representação Semântica de Documentos por meio de Vetores Contínuos

Universidade Federal de Goiás – UFG

Escola de Engenharia Elétrica, Mecânica e de Computação

Programa de Pós-Graduação em Engenharia Elétrica e de Computação

Orientador: Prof. Dr. Gelson da Cruz Júnior

Goiânia

Novembro/2020

Ficha de identificação da obra elaborada pelo autor, através do
Programa de Geração Automática do Sistema de Bibliotecas da UFG.

Rezende Carneiro de Mendonça, Leandro
Modelo Neural Recozido para a Representação Semântica de Documentos por meio de Vetores Contínuos [manuscrito] / Leandro Rezende Carneiro de Mendonça. - 2020.
LXXVIII, 78 f.: il.

Orientador: Prof. Dr. Gelson da Cruz Junior.
Tese (Doutorado) - Universidade Federal de Goiás, Escola de Engenharia Elétrica, Mecânica e de Computação (EMC), Programa de Pós-Graduação em Engenharia Elétrica e de Computação, Goiânia, 2020.
Bibliografia.
Inclui gráfico, tabelas, algoritmos, lista de figuras, lista de tabelas.

1. Representação de Documento. 2. Redes Neurais. 3. Processamento de Linguagem Natural. 4. Recozimento Simulado. 5. Aprendizado de Máquina. I. da Cruz Junior, Gelson, orient. II. Título.

CDU 621.3

UNIVERSIDADE FEDERAL DE GOIÁS

ESCOLA DE ENGENHARIA ELÉTRICA, MECÂNICA E DE COMPUTAÇÃO

ATA DE DEFESA DE TESE

Ata nº **08/2020** da sessão de Defesa de Tese de **Leandro Rezende Carneiro de Mendonça**, que confere o título de **Doutor em Engenharia Elétrica e de Computação**, na área de concentração em **Engenharia de Computação.**

Aos **treze dias do mês de novembro de dois mil e vinte**, a partir das **14h00min.**, realizou-se a sessão pública de Defesa de Tese intitulada **"Modelo Neural Recozido para a Representação Semântica de Documentos por meio de Vetores Contínuos"**. Os trabalhos foram instalados pelo Orientador, Professor Doutor **Gelson da Cruz Junior (EMC/UFG),** com a participação dos demais membros da Banca Examinadora: Professor Doutor **Sérgio Vale Aguiar Campos (DCC/UFMG)**, membro titular externo; Professor Doutor **Fabrízzio Alphonsus Alves de Melo Nunes Soares (Southern Oregon University)**, membro titular externo; Professor Doutor **Marco Antonio Assfalk de Oliveira (EMC/UFG)** membro titular externo; Professora Doutora **Symone Gomes Soares Alcalá (FCT/UFG)** membro titular interno; **cujas participações ocorreram através de videoconferência**. Durante a argüição os membros da banca **não fizeram** sugestão de alteração do título do **trabalho**. A Banca Examinadora reuniu-se em sessão secreta a fim de concluir o julgamento da Tese, tendo sido o candidato **aprovado** pelos seus membros. Proclamados os resultados pelo Professor Doutor **Gelson da Cruz Junior**, Presidente da Banca Examinadora, foram encerrados os trabalhos e, para constar, lavrou-se a presente ata que é assinada pelos Membros da Banca Examinadora, aos **treze dias do mês de novembro de dois mil e vinte**.

TÍTULO SUGERIDO PELA BANCA

Documento assinado eletronicamente por **Sérgio Vale Aguiar Campos**, **Usuário Externo**, em 16/11/2020, às 12:20, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Gelson Da Cruz Junior**, **Professor do Magistério Superior**, em 16/11/2020, às 14:04, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **LEANDRO REZENDE CARNEIRO DE MENDONÇA**, **Discente**, em 16/11/2020, às 14:23, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Marco Antonio Assfalk De Oliveira**, **Professor do Magistério Superior**, em 16/11/2020, às 16:02, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Symone Gomes Soares Alcala**, **Professor do Magistério Superior**, em 17/11/2020, às 10:01, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Fabrízzio Alphonsus Alves De Melo Nunes Soares**, **Professor do Magistério Superior**, em 17/11/2020, às 13:06, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

A autenticidade deste documento pode ser conferida no site https://sei.ufg.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **1663546** e o código CRC **B29740CD**.

Referência: Processo nº 23070.044552/2020-53 SEI nº 1663546

Este trabalho é dedicado a todas as crianças que sonham ser cientistas, especialmente ao meu filho Enzo.

Agradecimentos

A conclusão deste trabalho só foi possível pelas bençãos de Deus. Manifesto os meus agradecimentos e a minha eterna gratidão a todas as pessoas que direta ou indiretamente contribuíram para este trabalho. Agradeço especialmente:

- à minha querida esposa Michelle, pelo apoio, amor e companheirismo em todos os momentos;
- ao meu filho querido, Enzo, pelo amor, e pela motivação em continuar perseverando para superar mais esta etapa da minha vida;
- ao Professor e amigo Gelson da Cruz Junior, pela orientação, compreensão, paciência e amizade, durante todas as etapas do desenvolvimento desta tese;
- aos meus pais, Alan e Maria Lúcia, pelos valores e ensinamentos repassados.

"Ando devagar, porque já tive pressa
Levo esse sorriso, porque já chorei demais
Hoje me sinto mais forte, mais feliz, quem sabe?
Só levo a certeza de que muito pouco eu sei,
Ou nada sei."
Almir Sater

Resumo

Como resultado da crescente produção de dados textuais não estruturados, surgiram técnicas para representar palavras e documentos no espaço vetorial para extração de conhecimento. O Ministério Público brasileiro recebe inúmeras solicitações textuais não estruturadas enviadas por cidadãos com necessidades diversas - violência doméstica contra a mulher, solicitações de internações em unidades de terapia intensiva, entre outras. O tempo gasto na classificação, detecção de similaridades e distribuição para a promotoria competente é essencial para otimização dos recursos públicos. Assim, foi adotado um modelo neural associado ao algoritmo Simulated Annealing (SA), um clássico algoritmo de otimização global com baixa complexidade computacional, de modo a reduzir o tempo de treinamento diário e a proporcionar uma visualização gráfica mais amigável de dados multidimensionais, apoiando o processo de decisão judicial. A analogia física do algoritmo SA associado à representação contínua de documentos no espaço vetorial contribui para a visualização amigável de um conjunto de dados de alta dimensão, mantendo uma acurácia comparável a outros modelos neurais profundos e a outros algoritmos de otimização, como Covariance Matrix Adaptation Evolution Strategy (CMA-ES) e Bayesian Optimization (BO).

Palavras-chave: Representação de Documento, Redes Neurais, Processamento de Linguagem Natural, Análise de Texto, Representação Vetorial, Otimização, Recozimento Simulado, Aprendizado de Máquina.

Abstract

As a result of the growing production of unstructured textual data, techniques for representing words and documents in the vector space have emerged recently. The Brazilian Public Ministry has received several textual requests that are send by citizens with different needs, such as those involved in cases of domestic violence against women, others requesting intensive care unit admissions, and more. The time spent in classifying, detecting similar requests and distributing them is essential to optimize and save public resources. Therefore, we adopted the neural model with the Simulated Annealing (SA), a classic global optimization algorithm with low computational complexity, because of the need to reduce the daily training time, providing a more friendly graphic visualization of data in high dimensions, supporting the judicial decision process. The physical analogy of the SA meta-heuristic associated with the continuous representation of documents in the vector space contribute greatly to the friendly visualization of a high-dimensional dataset, maintaining a comparable score with other deep models and optimization algorithms, such as Covariance Matrix Adaptation Evolution Strategy (CMA-ES) and Bayesian Optimization (BO).

Keywords: Document Representation, Neural Network, Natural Language Process, Text Analysis, Vector Representation, Optimization, Simulated Annealing, Machine Learning.

Lista de ilustrações

Lista de tabelas

Sumário

CAPÍTULO 1

Introdução

Uma grande quantidade de dados é produzida diariamente, sendo, em sua maioria, dados textuais não estruturados. Extrair informações relevantes de documentos é um processo desafiador, envolvendo processamento de linguagem natural (PLN) e redes neurais artificiais profundas, essenciais para o avanço da inteligência artificial (TAN, 1999; CAMBRIA; WHITE, 2014; WHITE; CAMBRIA, 2014; SUN; LUO; CHEN, 2017; ALTINEL; GANIZ, 2018).

Tradicionalmente, o processo de aprendizado de máquina para dados não estruturados requer que os dados de um domínio já estejam mapeados por um especialista. Após o mapeamento dos dados textuais, eles devem ser processados e representados como vetores numéricos e depois computados por algoritmos de aprendizado de máquina. As áreas de aplicação dos algoritmos de aprendizado de máquina são diversas, dentre elas: classificação textual, tradução textual; sumarização de texto; reconhecimento de voz; extração de tópicos; análise de sentimentos; reconhecimento de entidades (LOPEZ; KALITA, 2017).

Um método amplamente utilizado é o *Bag-of-Words(BoW)* (HARRIS, 1954). Nele os documentos são representados como vetores que contêm a frequência dos termos utilizados no documento. Uma desvantagem deste método, é o contexto semântico não ser preservado, já que não captura a ordem de ocorrência dos termos.

Outro método muito utilizado é o *one-hot encoding (OhE)*, que transforma um documento em um vetor numérico de tamanho fixo. Tanto o *BoW* quanto o *OhE* não capturam a relação semântica entre documentos e sofrem com a maldição da alta dimensionalidade (*curse of dimensionality*) e o consequente incremento no consumo de memória, diretamente proporcional ao número de palavras únicas em um conjunto de dados.

Para superar as limitações dos modelos *BoW* e *OhE* em relação à extração do contexto semântico, alguns modelos neurais foram propostos, como o *Word2Vec* (MIKOLOV et al., 2013b; MIKOLOV et al., 2013a; GOLDBERG; LEVY, 2014) e o *Doc2Vec* (LE; MIKOLOV, 2014; LAU; BALDWIN, 2016), buscando representar palavras e documentos como vetores densos. Ambos são escaláveis e permitem a extração de relações semânticas - analogias; sinonímias; hiponímias e polissemias - por meio de medidas de similaridade entre os vetores. As representações vetoriais de palavras e de documentos utilizando os modelos *Word2Vec* e *Doc2Vec* apresentam resultados superiores aos modelos *BoW* e *OhE*, especialmente em processos de classificação textual com grandes volumes de dados (LILLEBERG; ZHU; ZHANG, 2015).

Modelos híbridos, como o *Bag-of-Concepts* (KIM; KIM; CHO, 2017), foram propostos com o intuito de melhor representar os documentos no espaço vetorial, combinando os modelos *BoW* e *Word2Vec*. Os vetores de palavras são gerados com o modelo *Word2Vec* e, posteriormente, os vetores são agrupados por meio de um algoritmo de clusterização. Cada cluster gerado forma um conceito e as frequências são incorporadas aos vetores dos documentos. A incorporação do conhecimento por meio de conceitos requer alto processamento computacional, exigindo muito tempo para convergência, pois, além da representação vetorial dos conceitos, deve ser representado, também, vetorialmente, o corpus de documentos e, em seguida, ambos devem ser adicionados em um único vetor denso (LI et al., 2018).

Apesar de vários esforços para a representação semântica de documentos, o tempo para a convergência dos modelos se torna um obstáculo para a solução de problemas com alto volume de dados e recursos computacionais limitados. Assim, técnicas têm surgido na literatura que combinam meta-heurísticas com os modelos existentes, visando maximizar a representatividade semântica dos documentos e o tempo de convergência dos modelos.

Entre os diversos algoritmos meta-heurísticos existentes, podemos destacar o recozimento simulado (SA) (KIRKPATRICK; GELATT; VECCHI, 1983) - uma técnica robusta para a minimização ou a maximização de funções objetivo em um menor tempo para convergência, evitando, ao mesmo tempo, a convergência prematura para soluções locais. A origem do algoritmo SA é o algoritmo Metropolis (METROPOLIS et al., 1953). Dentre as aplicações do SA em redes neurais profundas, podemos destacar a classificação de imagens (MNIST) usando redes neurais convolucionais (CNN) para otimizar o desempenho do treinamento, minimizando a função de erro na classificação das imagens (RERE; FANANY; ARYMURTHY, 2015) a cada época, por meio da adaptação da taxa de aprendizado do modelo neural (CIFAR-10) (POUYANFAR; CHEN, 2017).

Trabalhos recentes também têm buscado adaptar ciclicamente a taxa de aprendizado da rede neural. Nesta abordagem, a taxa de aprendizado é alternada linearmente entre um limite inferior e um limite superior a cada época de treinamento (SMITH, 2017).

O algoritmo utilizado como referência para a construção do modelo proposto nesta tese é o *Doc2Vec* (LE; MIKOLOV, 2014) que, quando associado à meta-heurística de recozimento simulado (KIRKPATRICK; GELATT; VECCHI, 1983), apresenta ganhos significativos na representação semântica de documentos, acarretando melhores resultados na assertividade e, principalmente, na disposição espacial dos agrupamentos de documentos vetoriais.

Comparamos o SA com a Estratégia de Evolução da Adaptação da Matriz de Covariância (CMA-ES) (HANSEN, 2016) e com a Otimização Bayesiana (BO) (KUSHNER, 1964) que são meta-heurísticas amplamente utilizadas na otimização de funções caixa-preta, não convexas e não lineares. O CMA-ES é um algoritmo evolutivo com uma mutação gaussiana, seleção natural determinística baseada na evolução biológica e é conhecido por seu desempenho de ponta na otimização sem o uso de derivadas. Como o CMA-ES é um algoritmo evolutivo, podemos comparar e avaliar as populações geradas. O BO também é uma abordagem de otimização sem o uso de derivadas, muito utilizada na otimização de funções que demandam tempo para avaliação, tem como princípio o teorema de Bayes para direcionar a pesquisa em problemas de otimização global. Gera-se, inicialmente, uma função probabilística da função objetivo, conhecida como função substituta, que é então utilizada antes das amostras candidatas escolhidas serem avaliadas pela função objetivo real.

1.1 Descrição do problema, motivações e desafios

O tamanho da população do estado de Goiás, estimado em 7.018.354 cidadãos [1], o fácil acesso da população à justiça brasileira e a frequente judicialização das demandas sociais sobrecarregam o sistema de Justiça brasileiro. Assim, as demandas dos atores da Justiça cresceram acima da capacidade de serviço. Atualmente, o Ministério Público do Estado de Goiás conta com 392 promotores de Justiça para atender um território de 340.203.329 km^2 com cidadãos de níveis culturais e linguísticos, dada a complexidade da língua portuguesa. Um promotor recebe de 3 a 6 solicitações por dia. Após o registro, uma solicitação demanda até 48 horas para ser analisada.

O fluxo de atendimento de uma solicitação engloba três atividades críticas (Figura 1): **A** - classificação da solicitação; **B** - existência de similaridades; **C** - distribuição para a promotoria competente. O modelo proposto foi aplicado nessas atividades, automatizando-as de forma a propiciar maior celeridade no atendimento às demandas sociais com o devido cumprimento dos prazos legais, garantindo a distribuição assertiva e igualitária dessas demandas aos promotores.

[1] https://www.ibge.gov.br/cidades-e-estados/go.html

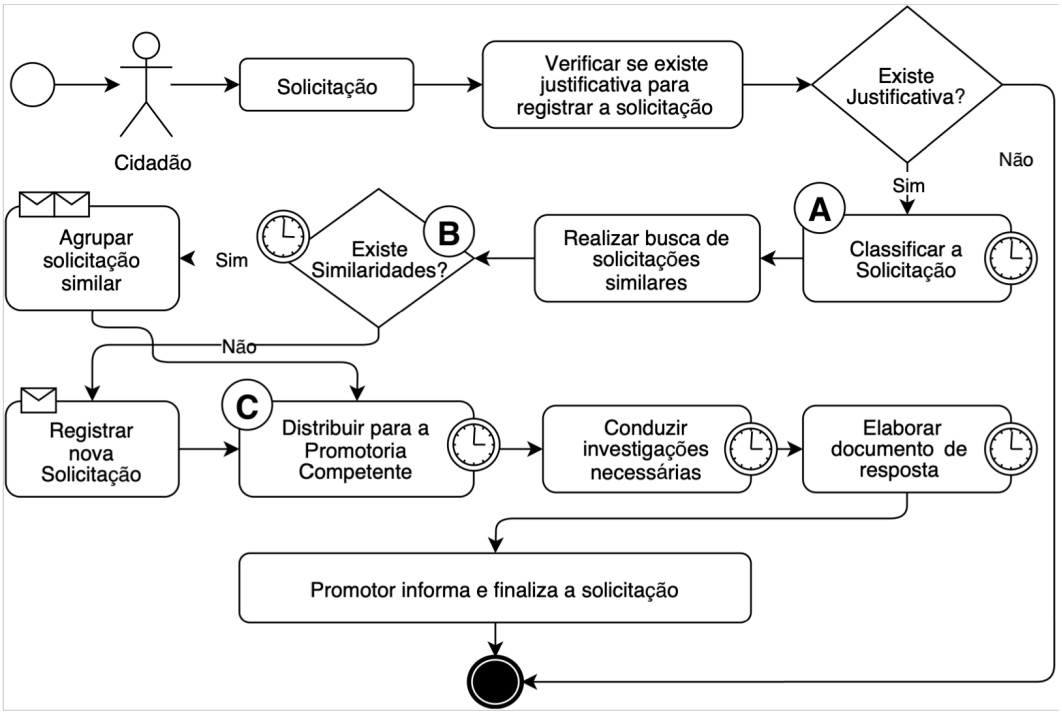


Figura 1 – Fluxo de atendimento de uma solicitação do cidadão

O algoritmo de otimização SA foi adotado devido ao baixo tempo de resposta na fase de treinamento diário do modelo neural com uma acurácia satisfatória na etapa de classificação, proporcionando, ainda, uma visualização amigável de conjuntos de dados de altas dimensões.

1.1.1 Descrição do conjunto de dados

Os dados são essenciais para qualquer algoritmo de aprendizado de máquina. A adoção de um algoritmo está associada ao tipo de dado e forma como ele é estruturado. Os dados devem ser representativos para permitir uma melhor generalização do modelo.

O Ministério Público registra uma média de dez mil documentos por mês. As demandas podem ser encaminhadas por meio do email institucional, presencialmente nos pólos de atendimentos, por telefone ou pelo sistema de registro eletrônico disponível no portal da instituição. Após o registro, os documentos são pré-processados e rotulados para, então, serem distribuídos aos promotores de Justiça conforme as áreas de atuação. A tabela abaixo (Tabela 1) representa um recorte da base de dados, totalizando 100.000 documentos distribuídos em nove áreas de atuação.

Tabela 1 – Conjunto de dados.

ID	Classe	Quantidade
573	Ameaça aos direitos coletivos	9.200
3418	Solicitação de tratamento médico	9.700
3413	Fornecimento de medicamento de alto custo	12.800
1130	Solicitação de pensão alimentícia	9.600
1143	Guarda de criança	10.700
3632	Hospitalização compulsória	10.000
1144	Investigação de paternidade	10.500
3415	Internação em unidade de terapia intensiva	10.200
973	Violência doméstica contra a mulher	17.300
Total		100.000

Cada documento possui um número identificador vinculado à classe, determinada pela narrativa do fato relatado pelo cidadão (Tabela 2). Para o treinamento do modelo proposto nesta tese, utilizamos uma base histórica de documentos com as respectivas classes. Os rótulos (classes) dos documentos foram definidos pelos promotores de Justiça.

Tabela 2 – Exemplo de documento.

ID	Classe	Texto
973	Violência doméstica contra a mulher	William Junior, está frequentemente intimidando Maria, sendo agressivo e não a deixando dormir em casa. Maria tem três filhos e não tem para onde ir. William briga constantemente com Maria agredindo-a e lesionando também os seus filhos. Maria afirma que William é usuário de drogas (cocaína), o que o torna mais violento.

1.2 Objetivos

O principais objetivos são:

- construir um modelo neural associado à meta-heurística de recozimento simulado (KIRKPATRICK; GELATT; VECCHI, 1983) para a representação semântica de documentos no espaço vetorial;
- criar um corpus de vetores de documentos jurídicos na língua portuguesa;
- comparar o modelo proposto com o estado da arte na representação semântica de documentos;
- avaliar algoritmos de otimização global como o CMA-ES e o BO;
- avaliar classificadores para problema multiclasse;
- apresentar resultados que apontam que o processo de recozimento simulado contribui para uma melhor visualização de dados multidimensionais.

1.3 Limitações

As limitações se relacionam a restrição do escopo da pesquisa, tendo em vista que diversos algoritmos e modelos surgem na área de processamento de linguagem natural. Os critérios adotados para esta restrição são:

- referência de modelos clássicos - Word2Vec e o Doc2Vec;
- não avaliação dos modelos GLOVE (Global Vectors for Word Representation) (PENNINGTON; SOCHER; MANNING, 2014), ELMO (Embeddings from Language Models) (PETERS et al., 2018), BERT (Bidirectional Encoder Representations from Transformers) (DEVLIN et al., 2018) e ERNIE (Enhanced Language Representation with Informative Entities) (ZHANG et al., 2019), arquiteturas complexas para a parametrização e o controle dos hiperparâmetros existentes não relacionados ao escopo da tese.

1.4 Resultados esperados

- otimizar as etapas A, B e C (Figura 1), contribuindo para um atendimento mais eficaz ao cidadão;
- identificar os classificadores mais adequados para o processamento de linguagem natural;
- obter uma pontuação estatisticamente similar, a outros modelos neurais profundos;
- otimizar o tempo de convergência.
- propiciar uma melhor análise visual de dados multidimensionais.

1.5 Estrutura da tese

No capítulo 2, apresentamos os trabalhos mais utilizados para a representação de documentos. No capítulo 3, relacionamos os trabalhos que utilizam o algoritmo SA em redes neurais. No capítulo 4, apresentamos o modelo proposto, e, no capítulo 5, os resultados das simulações. No capítulo 6, as conclusões da pesquisa e os trabalhos futuros.

CAPÍTULO 2

Material e métodos

Neste capítulo, apresentaremos os métodos clássicos existentes mais utilizados para a representação vetorial de documentos.

2.1 *Bag-of-words*

O método bag-of-words (HARRIS, 1954), o método mais utilizado para a representação de documentos (HUANG, 2008), representa um vetor de documento por suas frequências de palavras (f_{w_i}), apresentando muitos resultados na mineração de texto. Os vetores de documentos gerados são computacionalmente simples, facilmente interpretáveis e a similaridade entre os documentos pode ser determinada quando os vetores compartilham o mesmo número de ocorrências de palavras. A perda do contexto semântico limita a aplicabilidade deste método em problemas mais avançados, pois cada palavra é analisada independentemente uma da outra, podendo causar a maldição da alta dimensionalidade quando o número de palavras ou documentos únicos excede o espaço de memória disponível.

Considerando um *corpus* de documentos textuais $C = [d_1, d_2, d_3, ..., d_n]$ (Tabela 3), representado por k atributos ou termos extraídos dos documentos, cada documento d_i é um vetor $d_i = (t_{i1}, t_{i2}, t_{i3}, t_{ik})$, em que t_{ij} é o valor do atributo j_{th} no documento i.

Tabela 3 – Bag-of-words de documentos.

	t_1	t_2	t_3	t_k	Y
d_1	t_{11}	t_{12}	t_{13}	t_{1k}	y_1
d_2	t_{21}	t_{22}	t_{23}	t_{2k}	y_2
d_3	t_{31}	t_{32}	t_{33}	t_{3k}	y_3
d_n	t_{n1}	t_{n2}	t_{n3}	t_{nk}	y_n

Os seguintes índices discriminantes podem ser utilizados para representação do termo t_{ij}:

- Boolean: o valor de t_{ij} é 1 se o termo t_j estiver presente no documento d_i, caso contrário será 0 (Equation 2.1).

$$bool(t_j, d_i) = \begin{cases} 1 & t_j \in d_i \\ 0 & t_j \ni d_i \end{cases} \tag{2.1}$$

- tf-idf: o índice tf-idf (*Term Frequency, Inverse Document Frequency*) (SOUCY; MINEAU, 2005) assume que os termos não possuem a mesma relevância em documentos distintos. O cálculo do peso de relevância de cada termo é definido como frequência inversa no documento (idf) (Equação 2.2).

  Na equação 2.2, N identifica o número de documentos e $d(t_j)$ o número de documentos nos quais o termo t_j ocorre pelo menos uma vez. Desse modo, se um termo estiver presente em vários documentos seu peso será próximo a $0(zero)$, já se estiver presente em poucos documentos, terá um peso maior. Com o $idf(t_j)$ calculado, podemos calcular o $tfidf$ (Equação 2.3):

$$idf(t_j) = \log(\frac{N}{d(t_j)}) \tag{2.2}$$

$$tfidf(t_j, d_i) = freq(t_j, d_i).idf(t_j) \tag{2.3}$$

2.2 *One-hot-encoding*

Esta representação identifica a ocorrência dos termos contidos nos documentos. Caso uma palavra esteja presente no documento, o valor assumido é 1, caso contrário, é 0. Na tabela abaixo (Tabela 4), observamos a ocorrência dos termos ($t_1 ... t_6$) nos documentos: d_1 ("Adoro programar computadores") e d_2 ("Programar é muito divertido").

Tabela 4 – Documento representado como one-hot vector.

	Adoro	*programar*	*computadores*	é	*muito*	*divertido*
d_1	1	1	1	0	0	0
d_2	0	1	0	1	1	1

2.3 *Word and Document Embeddings*

Estudos recentes têm apresentado resultados que são o estado da arte na representação de palavras e documentos no espaço vetorial. Ao incorporar seu contexto semântico (MIKOLOV et al., 2013b; MIKOLOV et al., 2013a; PENNINGTON; SOCHER; MANNING, 2014; LE; MIKOLOV, 2014; GOLDBERG; LEVY, 2014; KUSNER et al., 2015), estes estudos tornaram-se os mais utilizados para a representação da proximidade semântica entre palavras e documentos no espaço vetorial (KIM; KIM; CHO, 2017; KIM et al., 2019).

O modelo Word2Vec (MIKOLOV et al., 2013b) possui dois algoritmos - *Continuous Bag Of Words* (CBOW) (Equação 2.4) e Skip-gram (Equação 2.5). O CBOW recebe como entrada para o treinamento uma sequência de palavras do documento D no corpus C e visa maximizar a probabilidade de *log* prevista para projeção de uma palavra t, considerando os vetores de palavras ao redor em uma janela de contexto $c_t(v_{t-s}, ..., v_{t-1}, v_{t+1}, ..., v_{t+s})$ com tamanho s.

$$CBOW = \sum_{D \in C} \sum_{t \in D} \log p(t|c_t) \tag{2.4}$$

O algoritmo *Skip-gram* projeta uma janela de contexto $c_t = v_{t-s}, ..., v_{t-1}, v_{t+1}, ..., v_{t+s}$ dada uma palavra t.

$$Skipgram = \sum_{D \in C} \sum_{t \in D} \sum_{c_t' \in c_t} \log p(c_t'|t) \tag{2.5}$$

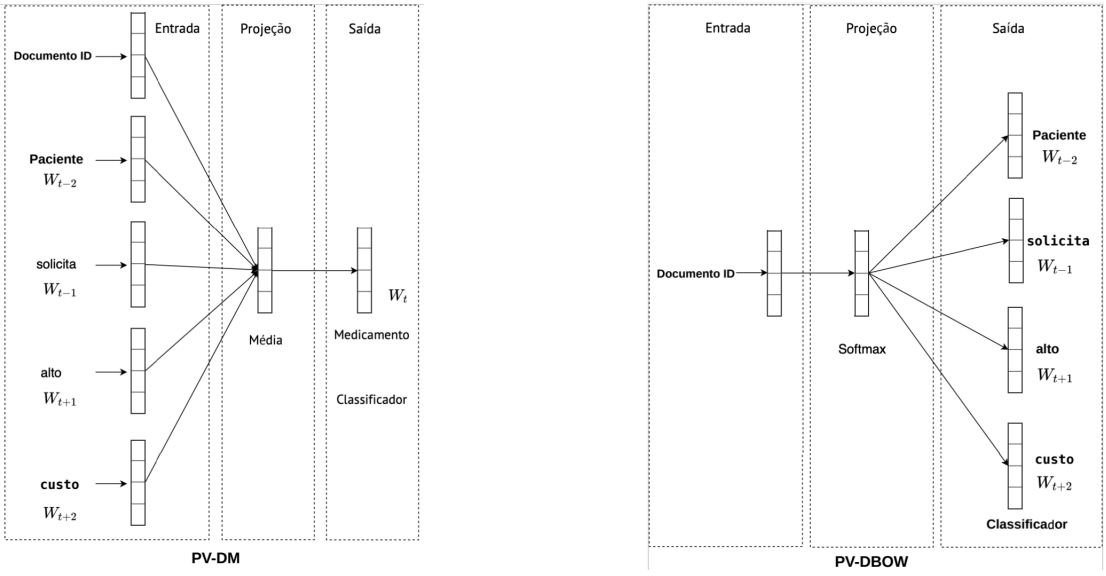
A função *Softmax* (Equação 2.6) é uma extensão da função de regressão logística para problemas de classificação multiclasse, atribuindo uma probabilidade decimal a cada classe, cuja a soma totaliza 1. A probabilidade é obtida por meio da transformação das saídas da rede neural ϕ para as probabilidades y através da normalização da função exponencial:

$$\begin{aligned} y_c &= \frac{\exp(\phi_c)}{\sum_j^c \exp(\phi_j)} \\ onde\ \phi &= w_0 x_0 + w_1 x_1 + ... + w_c x_c \\ e\ c\ &é\ o\ número\ de\ classes. \end{aligned} \tag{2.6}$$

O modelo base Doc2Vec (LE; MIKOLOV, 2014; KUSNER et al., 2015), estende o modelo Word2Vec (MIKOLOV et al., 2013b) gerando a representação vetorial $v_D^{d2v} \in \mathbb{R}^d$ para um documento D por meio de dois algoritmos distintos. Os algoritmos *Distributed Memory (DM)* (Figura 2a) e *Distributed Bag Of Words (DBOW)* (Figura 2b) são semelhantes aos algoritmos do modelo Word2Vec descritos acima, em que cada palavra é mapeada como um único vetor representado como uma coluna na matriz w e cada identificador de documento como um vetor exclusivo na coluna da matriz D.

O algoritmo DM preserva a ordem das palavras em um documento (Figura 2a), é baseado no algoritmo CBOW do modelo Word2Vec (Equação 2.4) e visa maximizar a predição de uma palavra t, dado um conjunto de palavras contextualmente vizinhas c_t no documento D, $\log p(t|c_t, D)$. Por exemplo, no documento *[Paciente solicita medicamento alto custo para tratamento]*, o modelo CBOW buscaria prever a palavra *[alto]*, dadas as palavras de contexto *[paciente solicita medicamento custo para tratamento]* e uma janela de tamanho $s = 3$.

O algoritmo DBOW não preserva a ordem das palavras no documento (Figura 2b), assim como o algoritmo Skip-gram do modelo Word2Vec (Equação 2.5). Diferentemente do algoritmo Skip-gram, que representa os vetores de palavras individualmente, o DBOW representa o documento D_i como um vetor único, com as palavras e o identificador do documento $\sum_{c_t' \in c_t} \log p(c_t'|D)$.



(a) Arquitetura DM. (b) Arquitetura DBOW.

Figura 2 – Arquiteturas para estratégias de treinamento.

2.4 Técnicas de Otimização

Apesar dos resultados positivos provenientes do uso das redes neurais profundas nas áreas de processamento de imagens e de processamento de linguagem natural, as pesquisas com foco na otimização dos hiperparâmetros desses modelos são incipientes.

A grande quantidade de hiperparâmetros e de camadas desencoraja a investigação em processos de otimização das redes neurais profundas, pois a cada camada adicionada ou hiperparâmetro modificado, há um aumento exponencial da complexidade em rastrear a origem dos resultados da rede. Contudo, devido ao aumento também exponencial dos dados não estruturados e acesso limitado a recursos computacionais para processamento, é imprescindível a construção de modelos otimizados, para que retornem os resultados em tempo hábil e com maior assertividade.

É notório que o principal parâmetro de uma rede neural é a taxa de aprendizado, sendo determinante para a convergência ou divergência da rede.

> "A taxa inicial de aprendizado [...] Esse geralmente é o hiperparâmetro mais importante e deve-se sempre verificar se foi ajustado [...] Se houver tempo apenas para otimizar um hiperparâmetro e este utiliza o gradiente

> estocástico descendente, então este é o hiper-parâmetro que vale a pena ser ajustado."(BENGIO, 2012)

2.4.1 *Simulated Annealing*

O algoritmo de recozimento simulado (SA) (KIRKPATRICK; GELATT; VECCHI, 1983) é um algoritmo meta-heurístico para resolver problemas de otimização combinatória, tendo como predecessor o algoritmo da metrópole (METROPOLIS et al., 1953).

O princípio do SA (Algoritmo 1) compreende uma analogia que expõe o processo de recozimento de metais a uma temperatura alta para sua deformação e resfriamento a um estado de baixa energia para obter sólidos mais homogêneos com melhor qualidade. O algoritmo procura soluções em torno da solução atual. Se a nova solução for melhor que a solução atual, a nova solução se tornará a solução atual. No entanto, a característica fundamental do algoritmo SA é a sua capacidade de aceitar soluções piores. Um fator de probabilidade baseado na distribuição de probabilidade de Boltzmann determina se uma solução poior que a atual é aceita. Isso evita a convergência prematura do algoritmo para soluções locais e permite que o algoritmo encontre soluções globais ideais. A função RANDOM([0, 1)) adota uma distribuição uniforme contínua.

Algorithm 1 Algoritmo Simulated Annealing - Problema de Minimização

function SIMULATEDANNEALINGMIN()
  $T \leftarrow T_{max}$
  $S_{atual} \leftarrow$ **SOLUÇÃO INICIAL()**
  while $T > T_{min}$ **do**
    $S_{nova} \leftarrow$ **SOLUÇÃO VIZINHA**(T, S_{atual})
    $\Delta E \leftarrow$ **ENERGIA**(S_{nova}) − **ENERGIA**(S_{atual})
    if $\Delta E < 0$ **then**
      $S_{atual} \leftarrow S_{nova}$
    else if RANDOM$([0, 1)) < \exp(-\Delta E/t)$ **then**
      $S_{atual} \leftarrow S_{nova}$
    $T \leftarrow$ **RESFRIAR**(T, S_{atual})
  return $S_{current}$

O processo de resfriamento começa com uma alta probabilidade de o algoritmo aceitar soluções piores devido a alta temperatura do sistema (t). À medida que o sistema esfria, o algoritmo se comporta como o algoritmo de escalada (*Hill Climbing*) (KHARI; KUMAR, 2017), convergindo em um espaço de busca local.

2.4.2 *Covariance Matrix Adaptation Evolution Strategy (CMA-ES)*

Proposta pela primeira vez em 2001 (HANSEN; OSTERMEIER, 2001), a Estratégia de Evolução da Adaptação da Matriz de Covariância (CMA-ES) (Algoritmo 2) é uma

técnica de otimização iterativa e evolutiva. Uma nova população com possíveis soluções candidatas é produzida a cada iteração, utilizando uma distribuição normal multivariada, o que possibilita que a próxima população seja mais propensa a produzir melhores resultados (HANSEN, 2016). Os algoritmos evolutivos são estocásticos e buscam o melhor valor da função objetivo. A característica básica desses algoritmos é gerar um conjunto de soluções (população), geralmente aleatórias. As novas populações são produzidas por meio da aplicação de operadores genéticos, como a reprodução, a mutação e a seleção. O melhor conjunto da população é mantido para a próxima iteração, ocorrendo um novo ciclo de operações genéticas. Assim, o algoritmo prossegue de forma evolutiva, em que os indivíduos mais capazes sobrevivem. Embora os algoritmos evolutivos sejam simplistas quando comparados a um biológico, eles são suficientemente complexos para fornecer mecanismos de busca de forma adaptativa e robusta (LIGHTWEIGHT...,).

Algorithm 2 Algoritmo CMA-ES

function CMAES()
  $D \leftarrow$ **Dimensão Inicial()**
  $\lambda \leftarrow$ **Tamanho da População Inicial**$(4.0 + 3.0\log(D))$
  $\mu \leftarrow$ **Population Descendente**$(\lfloor(\lambda/2)\rfloor)$
  $\sigma_{start} \leftarrow$ **Desvio Padrão Inicial()**
  $cov_{start} \leftarrow$ **Taxa de aprendizagem de covariância inicial()**
  while não terminar do
    $C^{(g+1)} \leftarrow$ **Atualizar Matriz de Covariância**(cov_{start})
    $\sigma^g \leftarrow$ **Atualizar Passo de Adaptação Sigma**(σ_{start})
    $x_{k=1,\ldots,\lambda}^{(g+1)} \leftarrow$ **Gerar População Normal Multivariada**(λ)
    $\overline{m}^{(g+1)} \leftarrow$ **Atualizar Média da População**$(\sum_{i=1}^{\mu} w_i x_{i:\lambda}^{(g+1)})$
  return $\overline{m}$

2.4.3 Otimização Bayesiana

A Otimização Bayesiana (BO) foi proposta em 1964 para encontrar o ponto máximo de uma curva *multi peak* com ruído (KUSHNER, 1964), no entanto, tornou-se popular na otimização global das funções de caixa preta (JONES; SCHONLAU; WELCH, 1998). A maioria dos problemas de otimização envolvendo aprendizagem de máquina são relacionados à otimização das funções de caixa preta, nas quais a função objetivo $f : X_n \rightarrow \mathbb{R}$ requer um alto custo computacional de processamento e tempo para avaliação.

Devido à complexidade computacional substancial, é essencial minimizar o número de amostras X_n do espaço de busca para avaliação pela função $f(X)$ objetivo. A otimização Bayesiana é aplicada para simplificar o espaço de busca, buscando encontrar o número mínimo de etapas (SHAHRIARI et al., 2016) para alcançar o melhor ponto global.

A otimização Bayesiana (Algoritmo 3) (FRAZIER, 2018; BAYESIAN...,) utiliza um modelo para se aproximar da função objetivo, denominado, modelo substituto, baseando-

se no processo gaussiano (GP), o qual incorpora crenças anteriores sobre a função objetivo. As funções de distribuição posterior $P(f|X)$ têm um custo computacional baixo de avaliação e são usadas para propor melhores soluções.

Algorithm 3 Algoritmo BO

function BO()
    Inserir um processo Gaussiano antes de f.
    Observe f no ponto n_0.
    $n \leftarrow n_0$
    while $n \leq Número de Avaliações$ **do**
        Obter distribuição de probabilidades posteriores condicionado nas amostras atuais de f.
        Encontrar novos pontos x_n maximizando a função de aquisição.
        Observar $y_n \leftarrow f(x_n)$
        $n \leftarrow n + 1$
    return y_n

2.4.4 *Grid Search*

A pesquisa *Grid Search* é uma abordagem amplamente utilizada para a otimização de hiperparâmetros (BERGSTRA; BENGIO, 2012). Ela avalia um modelo para cada combinação de valores possíveis especificados em uma grade ou vetor. Avaliamos os modelos construídos para cada combinação de valores. Essa pesquisa é denominada um método de força bruta, tendo em vista que avalia as combinações entre os modelos e os hiperparâmetros. O produto escalar entre os n hiperparâmetros e os modelos resulta em uma pontuação V_k (Equação 2.7), possibilitando a seleção do modelo mais adequado.

$$\bigtimes_{k=1}^{n} V_k \tag{2.7}$$

Quando o espaço de busca não é bem delimitado, o método se torna ineficiente, sendo necessário paralelizar a busca.

2.5 Métricas de avaliação

Na avaliação dos vetores otimizados e não otimizados foram utilizados alguns classificadores multiclasse, extraindo as suas respectivas métricas de acurácia. A acurácia (Equação 2.8) é uma métrica comumente utilizada para avaliar a performance de classificadores e não deve ser utilizada quando as classes estão desbalanceadas. Neste caso, é recomendável o uso da métrica F1-Score (Equação 2.11). Utilizamos um conjunto da dados balanceado, em que as classes possuem quantidades de documentos equivalentes.

2.5.1 Acurácia

É a proximidade dos resultados da predição com os valores reais ou a porcentagem de documentos que são corretamente classificados.

$$A = \frac{t_p + t_n}{(t_p + t_n + f_p + f_n)} \tag{2.8}$$

Tal que:

t_p = *true positive* - número de exemplos positivos classificados como positivos.
t_n = *true negative* - número de exemplos negativos classificados como negativos.
f_p = *false positive* - número de exemplos negativos classificados como positivos.
f_n = *false negative* - número de exemplos positivos classificados como negativos.

2.5.2 Precisão

A precisão determina o percentual de documentos classificados como positivos $(t_p + f_p)$ que são verdadeiramente positivos (t_p). Não inclui exemplos negativos.

$$P = \frac{t_p}{(t_p + f_p)} \tag{2.9}$$

2.5.3 *Recall*

Recall determina o percentual de documentos classificados como positivos entre todos os documentos positivos do conjunto de dados.

$$R = \frac{t_p}{(t_p + f_n)} \tag{2.10}$$

2.5.4 *F-measure, F1 ou F-score*

F-Measure é a média harmônica entre a precisão (P) e o *recall* (R).

$$Fmeasure = \frac{1}{\frac{1}{2}(\frac{1}{P} + \frac{1}{R})} = \frac{2PR}{P + R} \tag{2.11}$$

2.5.5 Validação cruzada

Para validar os modelos implementados, foi utilizado o método de validação cruzada estratificada, dividindo o conjunto de dados em dez *fold's* $(k = 10)$ (STRATIFIED...,) proporcionais ao número de classes, minimizando a possibilidade de *overfitting* dos modelos.

A cada etapa de validação um *fold* é selecionado para teste e os demais para treinamento. A média das pontuações das etapas são consideradas para definir a pontuação final do classificador ($\overline{S}_{C_n}$) (Figura 3).

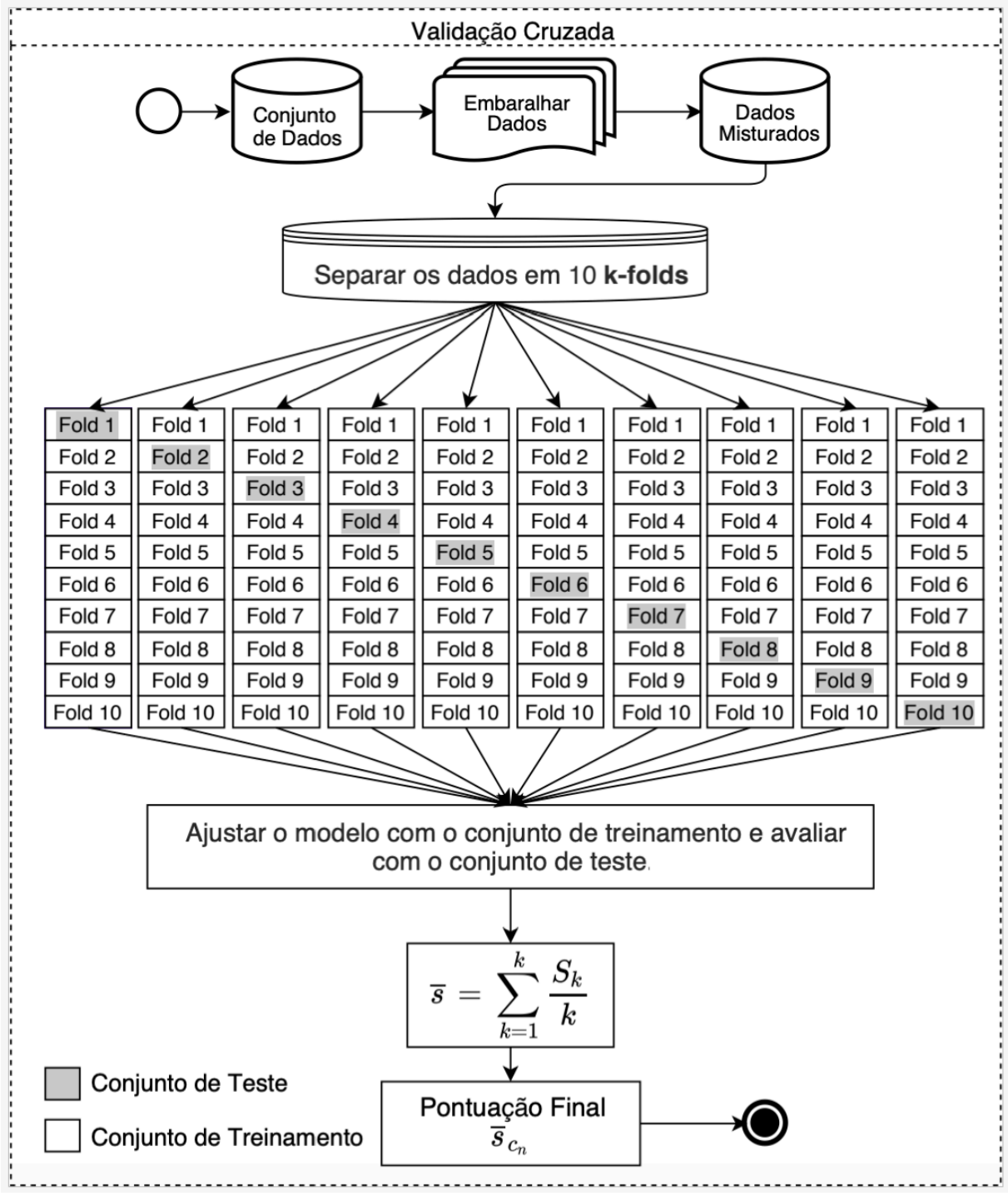


Figura 3 – Método de validação cruzada

2.5.6 *t-distributed Stochastic Neighbor Embedding*

O *t-distributed Stochastic Neighbor Embedding* (t-SNE) (MAATEN; HINTON, 2008b) é uma técnica para visualizar dados multidimensionais em espaço bidimensional

ou tridimensional. É um algoritmo variante do *Stochastic Neighbor Embedding* (SNE) (HINTON; ROWEIS, 2003), computacionalmente mais simples e com visualizações significativamente melhores, reduzindo a tendência de aglomeração de pontos no centro do gráfico. O t-SNE é a técnica mais adequada para visualizar um conjunto de dados de alta dimensão, apresentando os dados como uma matriz de similaridade.

2.6 Testes Estatísticos

Os testes estatísticos nos permitem realizar uma avaliação crítica e numérica de experimentos científicos. É possível testar hipóteses estatísticas por meio da análise dos dados para verificar o nível de significância das evidências nos resultados da pesquisa.

O Teste de Significância de Hipóteses Nulas (NHST) ou teste frequentista é ainda adotado no aprendizado de máquina para a comparação de modelos. Os resultados desse teste podem rejeitar ou não a hipótese nula em apoio à hipótese alternativa. No teste NHST, a única probabilidade (p-valor) calculada, é a de rejeitar a hipótese nula. Como alternativa para complementar as lacunas da abordagem frequentista, podemos utilizar a abordagem Bayesiana (CARRASCO et al., 2020). Nessa abordagem, é gerada a distribuição de probabilidades para cada modelo, permitindo a comparação entre eles. Por essa razão adotamos o teste T-pareado (KALPIC; HLUPIC; LOVRIC, 2011) em conjunto com o teste T-pareado Bayesiano (BENAVOLI et al., 2016), minimizando o grau de incerteza dos testes estatísticos.

2.6.1 Teste T pareado (*T-Student)*

O teste T pareado (KALPIC; HLUPIC; LOVRIC, 2011) é um método estatístico utilizado para determinar se a diferença média entre dois conjuntos de observações é zero. Existem duas hipóteses: a hipótese nula e a hipótese alternativa. A hipótese nula (H_0) pressupõe que a diferença média entre as amostras pareadas é zero. A hipótese alternativa (H_A) assume que não é igual a zero (Equação 2.12).

$$\begin{aligned} H_0 &: h_1(t) = h_0(t), \qquad \text{para todo } t \in [0, \tau] \\ H_A &: h_1(t) \neq h_0(t), \qquad \text{para alguns } t \in [0, \tau] \end{aligned} \tag{2.12}$$

Comparamos o *t-Statistic* obtido com os valores *t-Critical* na tabela do teste T [1], usando graus de liberdade e o nível de significância selecionado (α). Se o valor absoluto *t-Statistic* for maior que o *t-Critical*, rejeitamos a hipótese nula.

[1] http://www.epi.uff.br/wp-content/uploads/2015/05/Tabela-T.pdf

2.6.2 Teste Bayesiano Pareado

Atualmente, a abordagem bayesiana oferece uma alternativa às estatísticas frequentistas para comparar modelos em problemas de otimização e classificação (BENAVOLI et al., 2016). Essa abordagem calcula a incerteza ou o grau de crença por probabilidade. As crenças prévias são atualizadas por meio dos dados para produzir crenças posteriores, o que permite a quantificação de evidências para H_0 versus H_1 e vice-versa. A precisão dos modelos não é avaliada somente com a verificação dos dados empíricos, pois a significância estatística não implica em significância prática. Os fatores de bayes (BFs) complementam o NHST e quantificam a força das evidências em favor da hipótese alternativa em relação à hipótese nula (BF_{10}) ou a favor da hipótese nula em relação à hipótese alternativa (BF_{01}), dado o conjunto de dados D (Equação 2.13) (CARRASCO et al., 2020).

$$BF_{10} = \frac{p(D|H_1)}{p(D|H_0)} \qquad BF_{01} = \frac{p(D|H_0)}{p(D|H_1)} \tag{2.13}$$

2.7 Considerações finais

Apresentamos as técnicas clássicas e os algorimos mais utilizados no processamento de linguagem natural, abordando os métodos para a representação de palavras e documentos, bem como a metaheurística de recozimento simulado, as métricas e os testes estatísticos utilizados para a avaliação do modelo proposto. No capítulo seguinte apresentaremos os trabalhos relacionados à pesquisa.

CAPÍTULO 3

Trabalhos relacionados

O algoritmo de retropropagação (BP) é usado para simular o processo de aprendizagem nas redes neurais artificiais. A cada iteração (t) do processo, o peso do neurônio (θ) é ajustado pela taxa de aprendizagem (lr_t) da camada de saída para as camadas intermediárias, por meio do gradiente descendente estocástico (SGD), minimizando a função de erro (L) por $\theta^t = \theta^{t-1} - lr_t \frac{\partial L}{\partial \theta}$.

Uma alternativa ao algoritmo BP é utilizar metaheurísticas para o processo de aprendizagem das redes. Estudos para a classificação dos conjuntos de dados *Iris flower, National Institute of Diabetes and Digestive and Kidney Diseases*, *Teaching Assistant Evaluation*, e *Glass Identification* (DUA; GRAFF, 2017) demonstram que a metaheurística SA apresenta resultados superiores ao algoritmo BP (BEHERA; CHATTOPADHYAY, 2012). A associação do SA com uma rede neural convolucional também apresentou resultados efetivos na classificação de imagens do conjunto de dados *Modified National Institute of Standards and Technology (MNIST)* (LECUN et al., 1998) (RERE; FANANY; ARYMURTHY, 2015).

O algoritmo SA permite a ampliação do espaço de busca, minimizando a possibilidade de se obter soluções locais. A cada iteração do algoritmo, a temperatura é reduzida, gradativamente, e piores soluções que a atual podem ser aceitas por meio da distribuição de probabilidade de Boltzmann (Algoritmo 1).

A otimização de redes neurais profundas é um processo desafiador devido à grande quantidade de hiperparâmetros e de camadas existentes, o que aumenta a complexidade no rastreamento dos resultados obtidos. Apesar de estudos científicos relevantes proporem novas arquiteturas de modelos neurais profundos com a adoção de um número maior

de camadas, o uso de técnicas de otimização de hiperparâmetros pode proporciar resultados satisfatórios nos modelos existentes, dispensando a adoção de novas arquiteturas (BERGSTRA et al., 2011).

A taxa de aprendizado é o hiperparâmetro com maior impacto no processo de aprendizagem da rede. Uma alta taxa pode levar à convergência prematura do modelo, resultando em uma solução local, já uma taxa de aprendizado baixa aumenta, consideravelmente, o tempo de convergência do modelo, podendo o mesmo até não convergir.

Estudos recentes para a classificação de imagens associam as redes neurais convolucionais ao algoritmo SA para ajustar a taxa de aprendizado, produzindo resultados eficazes (KRIZHEVSKY, 2012) (POUYANFAR; CHEN, 2017). Técnicas híbridas também foram propostas (FISCHETTI; STRINGHER, 2019) utilizando o SA com o SGD para mover o gradiente para um espaço de busca mais amplo. Outro estudo para a otimização da taxa de aprendizado aplicada a problemas de classificação de imagens (KRIZHEVSKY, 2012; RUSSAKOVSKY et al., 2014) propõe um modelo empírico que tem o objetivo de variar ciclicamente a taxa de aprendizado (SMITH, 2017). Essa técnica alterna a taxa de aprendizado entre um limite superior e um inferior em um tamanho de passo. O tamanho do passo equivale a meio ciclo. A cada ciclo a taxa de aprendizado é incrementada linearmente do limite inferior ao limite superior e vice-versa, movimento denominado política de aprendizagem triangular.

Outras técnicas de otimização que devem ser destacadas são as metaheurísticas CMA-ES e BO, que são aplicadas na otimização de hiperparâmetros de funções de caixa preta. O ajuste dos hiperparâmetros por meio do CMA-ES é muito utilizado por se tratar de um algoritmo considerado o estado da arte na otimização de funções. O CMA-ES pode ser aplicado na otimização de hiperparâmetros (LOSHCHILOV; SCHOENAUER; SEBAG, 2012; LOSHCHILOV; HUTTER, 2016) e na otimização de sinais digitais, como o reconhecimento de voz (WATANABE; ROUX, 2014). O custo computacional do CMA-ES é sensível à escala dos problemas, tornando impraticável sua aplicação em problemas de altas dimensões (TONG; YUAN; LI, 2019).

A técnica de otimização bayesiana (BO) também apresenta resultados satisfatórios em otimizar hiperparâmetros de funções de caixa preta (SNOEK; LAROCHELLE; ADAMS, 2012; KLEIN et al., 2016; TOSCANO-PALMERIN; FRAZIER, 2018; SHIN et al., 2020). Destacamos o estudo de um modelo neural convolucional aplicado à classificação de imagens, em que os hiperparâmetros da rede são otimizados, obtendo-se um resultado 3% superior em relação a outras técnicas de otimização (SNOEK; LAROCHELLE; ADAMS, 2012).

3.1 Considerações finais

Apresentamos os trabalhos relevantes que aplicam o algoritmo SA para a otimização de hiperparâmetros de modelos neurais. No capítulo seguinte apresentaremos o modelo proposto.

CAPÍTULO 4

Modelo Proposto

A motivação em propor um modelo neural recozido para a representação semântica de documentos (Figura 4) advém da observação de estudos que adotaram com sucesso o SA em conjunto com os modelos neurais para a classificação de imagens. No entanto, não encontramos estudos relevantes que aplicam o SA a problemas de classificação textual.

A otimização da taxa de aprendizado pode ser definida em problemas de classificação por uma função objetivo definida por $f : l_{ri} \rightarrow \mathbb{R}$, buscando maximizar a pontuação do modelo. Um vetor l_{ri} representa a taxa de aprendizado das soluções candidatas e a função $f(l_{r1}) \geqslant f(l_{r0})$, a solução ótima encontrada.

Para a criação do modelo proposto, utilizamos os mesmos hiperparâmetros (Tabela 5) sugeridos pelo modelo base usado como referência (MIKOLOV et al., 2013b; MIKOLOV et al., 2013a) nos modelos neurais otimizado e não otimizado.

Tabela 5 – Hiperparâmetros utilizados no modelo base (Doc2Vec)

Tamanho do Conjunto de Dados (n)	100.000 documentos
Tamanhos dos vetores (v)	[100, 300, 600, 1000]
Estratégias de treinamento (s)	[pvdbow, pvdm/averaging, pvdm/concat]
Tamanho da Janela	5
Frequência mínima (*threshold)*	1
Sub-amostragem	10^{-5}
Amostragem Negativa	5
Épocas de treinamento	30
Taxa de aprendizado inicial (lr_0)	0.025
Limite inferior para otimização da taxa de aprendizado	10^{-7}
Limite superior para otimização da taxa de aprendizado	10^{-2}
Decremento linear da taxa de aprendizado	0.001

Nos hiperparâmetros elencados acima (Tabela 5), o tamanho da janela define a distância máxima entre a palavra atual e a palavra predita dentro de um documento - janela de contexto semântico em torno de cada palavra. A frequência mínima descarta as palavras com frequência inferior ao valor definido, não sendo incorporadas ao modelo. A sub-amostragem tem a função de reduzir o impacto das palavras frequentes na pontuação do modelo. A amostragem negativa modifica uma amostra dos pesos a cada iteração da etapa de treinamento.

Na representação dos vetores otimizados, o SA ($D_{v1}^{s^1}...D_{vn}^{s^1}$) (Algoritmo 1), o CMA-ES ($D_{v1}^{s^2}...D_{vn}^{s^2}$) (Algoritmo 2) e os vetores BO ($D_{v1}^{s^3}...D_{vn}^{s^3}$) (Algoritmo 3), foi otimizada apenas a taxa de aprendizado do modelo neural dentro de um espaço de busca contínuo e delimitado (Tabela 5). Nos vetores não otimizados, a taxa de aprendizado foi reduzida linearmente a cada iteração do modelo (Tabela 5).

O modelo tem início na etapa de pré-processamento (Figura 4), que consiste na sanitização dos dados com a eliminação de valores vazios (NaN), exclusão de palavras com pouca significância semântica (*Stopwords*) - preposições, artigos, pontuações e caracteres especiais. Na etapa de treinamento, associamos a cada iteração o algoritmo de recozimento simulado (SA). Essa associação permite ao modelo ampliar a busca por soluções melhores $f(l_{r1})$ que a solução atual $f(l_{r0})$. Soluções piores também podem ser aceitas, considerando um fator de probabilidade (p) dependente da temperatura do sistema $t_i > 0$.

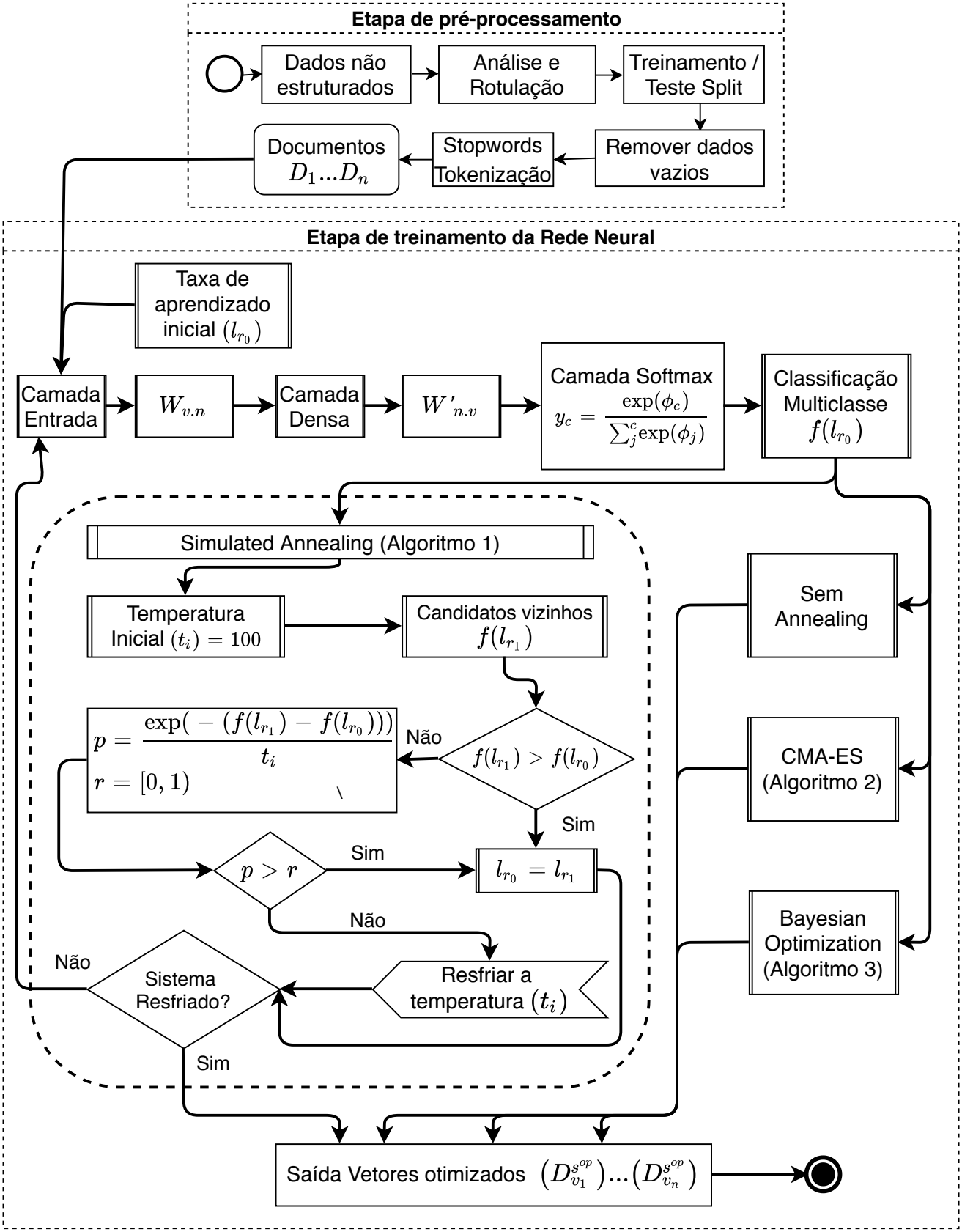


Figura 4 – Fluxo geral do modelo proposto

A arquitetura base do modelo neural (Figura 5) possui como saída os vetores de documentos otimizados e não otimizados. No fluxo geral (Figura 5), os documentos

são representados por $D_{v_n}^{s^{op}}$, em que $op = 1$ é o vetor otimizado, $op = 0$ é o vetor não otimizado, $v = [100, 300, 600, 1000]$ é o tamanho do vetor, n é o número de documentos e $s = [pvdbow, pvdm/concat, pvdm/averaging]$ são as estratégias de treinamento.

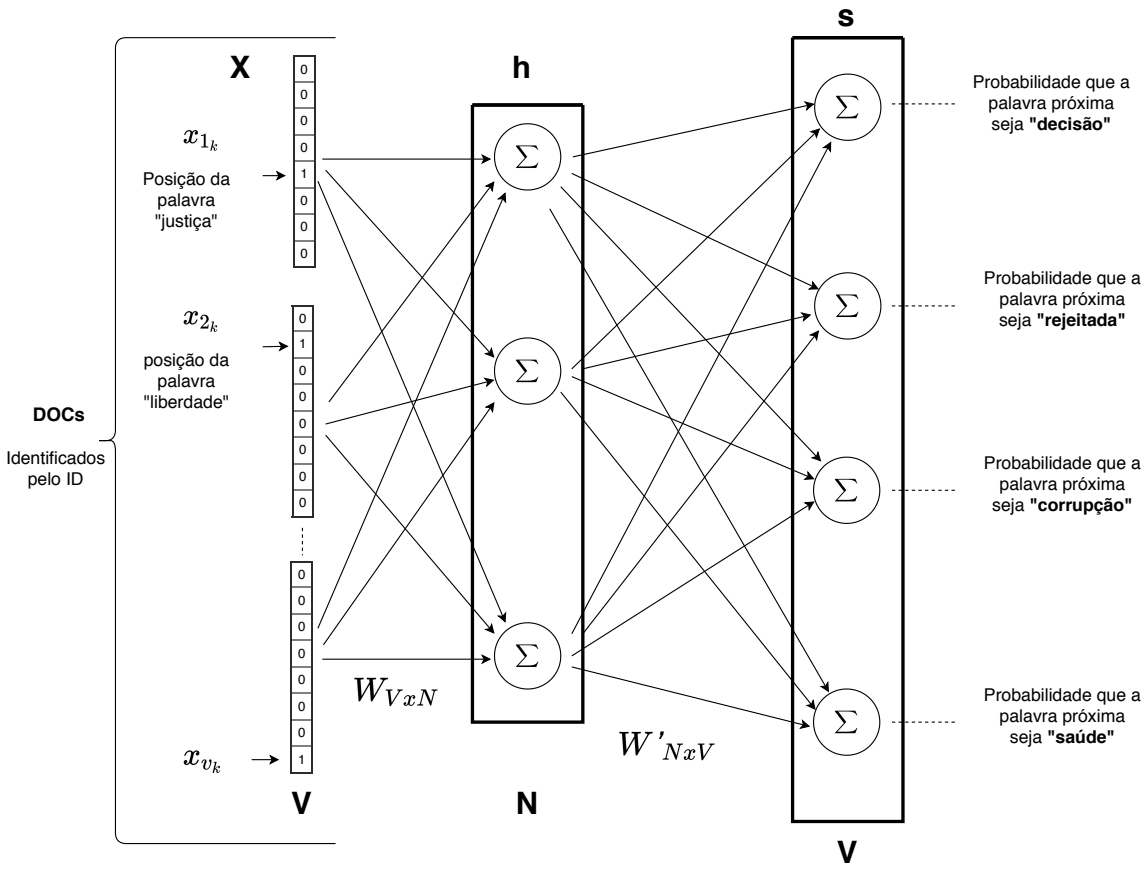


Figura 5 – Arquitetura geral da rede neural

A camada de saída da rede neural calcula a probabilidade condicional usando a função *Softmax* (Equação 2.6). Na avaliação dos vetores otimizados e não otimizados (Figura 6), adotamos os classificadores mais utilizados na literatura com o objetivo de compará-los (Tabela 6). A comparação considerou a dimensionalidade vetorial, as estratégias de treinamento e os algoritmos de otimização com o objetivo de obter as melhores representações vetoriais selecionadas pela etapa de classificação. Nos problemas de classificação, assumimos uma função desconhecida $f : D \rightarrow C$, em que $f(d)$ é o valor do atributo da classe $d \in D$, cujo objetivo é encontrar uma função $h : D \mapsto C$ que se aproxima da função desconhecida f, $h(d) \approx f(d)$.

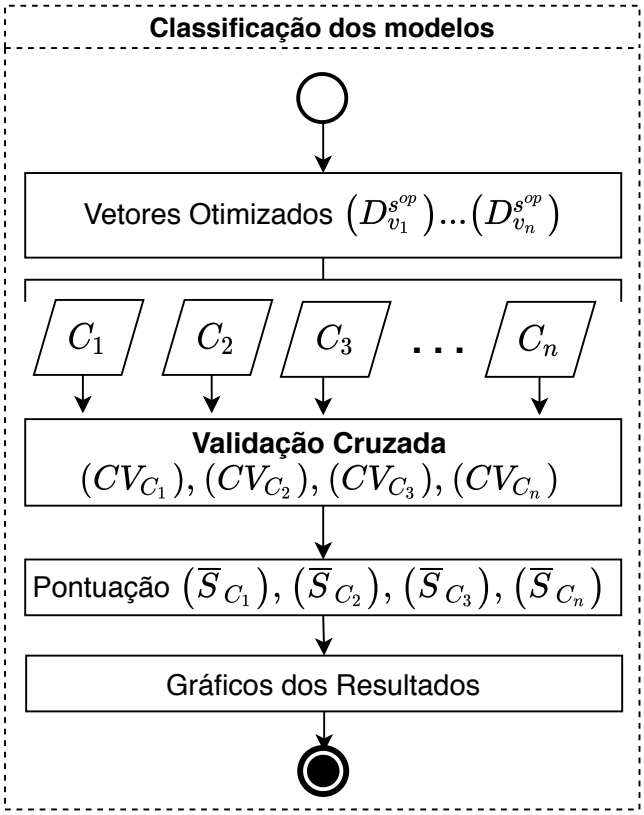


Figura 6 – Processo de Classificação

Utilizamos o método de pesquisa em grade (Grid Search) (Seção 2.4.4) para encontrar a melhor taxa de aprendizado para os classificadores (Tabela 6). Selecionamos esse método por ser amplamente utilizado e de fácil implementação para a otimização de funções em um espaço de busca discreto. A busca em grade consiste no produto cartesiano de dois conjuntos $C \bigtimes L$, em que C são os classificadores C = [XGBC, SVC, SGD, GBC, KNN, GNB, MLP, ADA, LSVC, RFC, DTC, LGR] e L, as taxas de aprendizado L = $[10^{-7}, 10^{-6}, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}, 10^{0}, 10^{+1}, 10^{+2}, 10^{+3}, 10^{+4}, 10^{+5}, 10^{+6}, 10^{+7}]$.

Tabela 6 – Classificadores utilizados nos vetores de documentos

ID	**Classificadores**	**Referências**
XGBC	Extreme Gradient Boosting	(FRIEDMAN, 2001)
SVC	Support Vector Machine with LibSVM	(CHANG; LIN, 2011)
SGD	Linear with Stochastic Gradient Descent	(ZHANG, 2004)
GBC	Stochastic Gradient Boosting	(FRIEDMAN, 2002)
KNN	Kernel Nearest Neighbor	(YU; JI; ZHANG, 2002)
GNB	Gaussian Naive Bayes	(JOHN; LANGLEY, 1995)
MLP	Multi Layer Perceptron	(HINTON, 1989)
ADA	Ada Boost	(KIM; LEE; CHO, 2005)
LSVC	Support Vector Machine with Linear SVC	(CORTES; VAPNIK, 1995)
RFC	Random Forest	(BREIMAN, 2001)
DTC	Decision Tree	(QUINLAN, 1986)
LGR	Logistic Regression	(HOSMER; LEMESHOW, 1989)

Na visualização dos resultados (Figura 7), reduzimos a dimensionalidade de todos

os vetores de documentos $[D^{s^{op}}_{100_n}, D^{s^{op}}_{300_n}, D^{s^{op}}_{600_n}, D^{s^{op}}_{1000_n}]$ para cinquenta dimensões utilizando o método da decomposição de valor singular. A redução é necessária porque o algoritmo t-SNE (MAATEN; HINTON, 2008a) escala quadraticamente na visualização de dados multidimensionais.

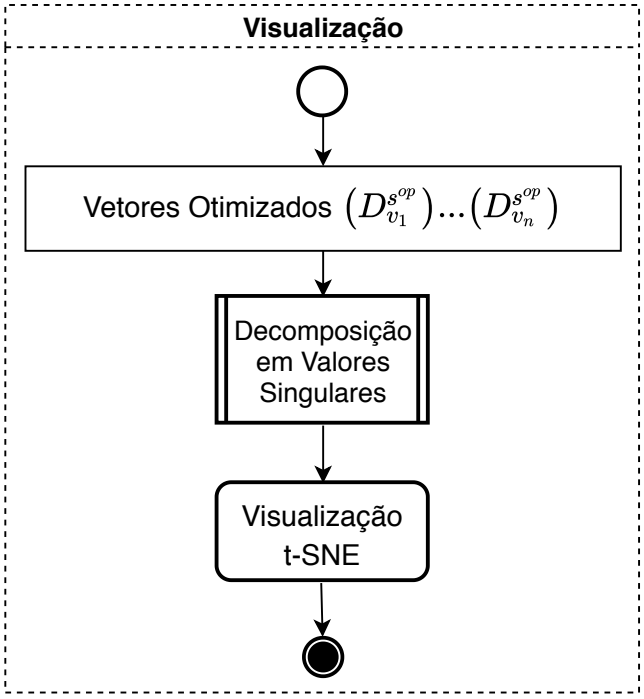


Figura 7 – Etapa de Visualização

Dentre os hiperparâmetros utilizados no algoritmo t-SNE (Tabela 7), o número de iterações é responsável pelo aumento linear da complexidade computacional. O valor escolhido se mostra adequado para a convergência do algoritmo, considerando o número de vetores do conjunto de dados. O valor recomendado para a taxa de aprendizado está entre 10 e 1000, um valor alto produz visualizações esparsas e um valor baixo gera visualizações difusas. A dimensão final para a visualização dos vetores é determinada pelo hiperparâmetro "componentes". O valor de perplexidade recomendado está entre 5 e 50 e determina o número de vizinhos próximos a um ponto (MAATEN; HINTON, 2008b; WATTENBERG; VIÉGAS; JOHNSON, 2016).

Tabela 7 – t-SNE hiperparâmetros

Hiperparâmetros	**Valor**
Iterações	50.000
Taxa de Aprendizado	100
Perplexidade	50
Componentes	2

4.1 Considerações finais

Apresentamos o modelo proposto associado aos otimizadores dos vetores de documentos, detalhando as etapas, os hiperparâmetros e os classificadores utilizados para a comparação dos modelos, bem como as técnicas para a visualização dos vetores. No capítulo seguinte, apresentaremos os resultados obtidos.

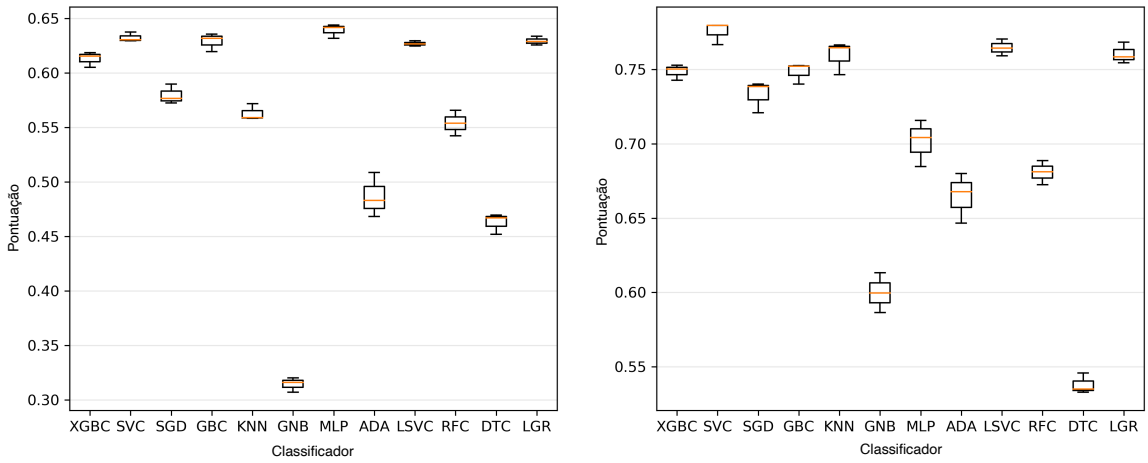
CAPÍTULO 5

Resultados

5.1 PVDBOW

5.1.1 Vetores com 100 dimensões

Na estratégia de treinamento com cem dimensões, os vetores não otimizados (Figura 8a) tiveram um desempenho inferior aos vetores recozidos (Figura 8b) em todos os classificadores. Os vetores CMA-ES (Figura 8c) e BO (Figura 8d) obtiveram resultados próximos aos vetores recozidos (Figura 8b).



(a) Vetores não otimizados

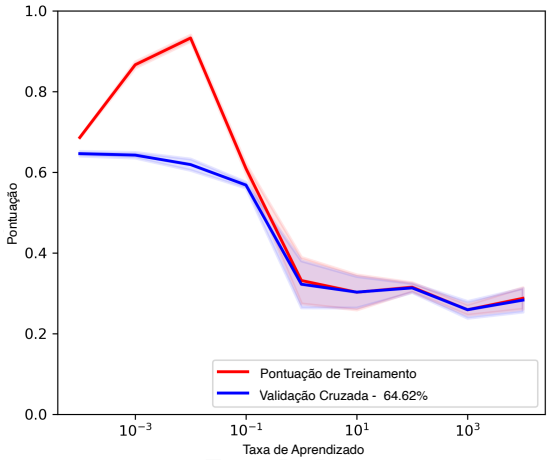
(b) Vetores recozidos

(c) Vetores CMA-ES

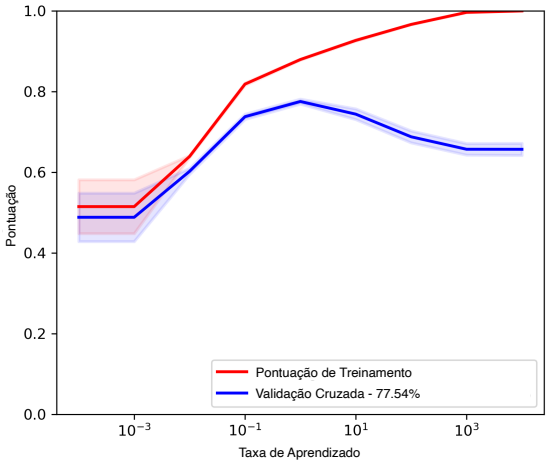
(d) Vetores BO

Figura 8 – Vetores PVDBOW com 100 dimensões

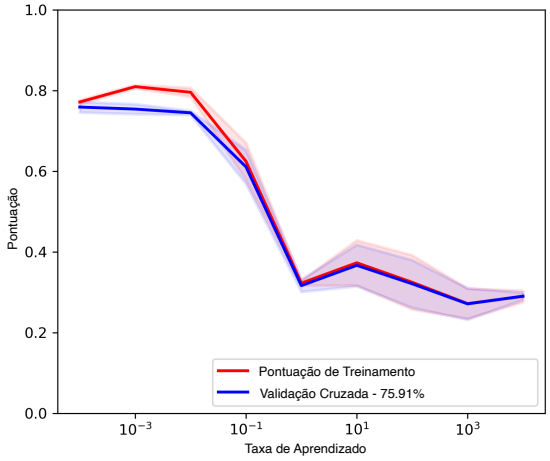
No experimento com os vetores não otimizados, o classificador de melhor pontuação foi o MLP (Figura 9a). Já no experimento com os vetores recozidos, o melhor pontuado foi o SVC (Figura 9b), mostrando-se superior, também ao MLP ($\Delta_{Score} = |Score_{(MLP)} - Score_{(SVC)}| = |64.62 - 77.54| = 12.92\%$). Com o otimizador CMA-ES o classificador MLP (Figura 9c) obteve um desempenho marginalmente inferior ao SVC ($\Delta_{Score} = |Score_{(MLP)} - Score_{(SVC)}| = |75.91 - 77.54| = 1.63\%$). O otimizador bayesiano com o classificador XGBC (Figura 9d) obteve resultados semelhantes aos vetores recozidos ($\Delta_{Score} = |Score_{(XGBC)} - Score_{(SVC)}| = |77.60 - 77.54| = 0.06\%$).



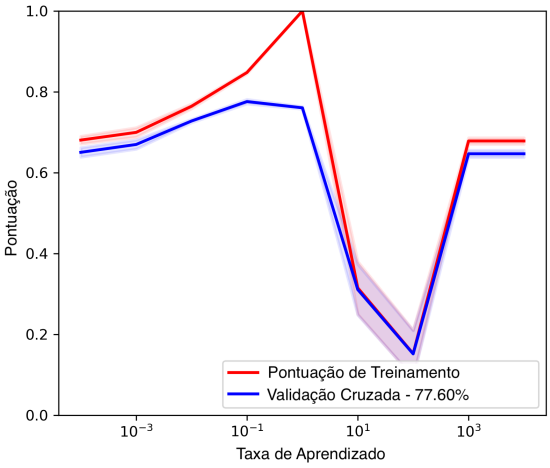
(a) Vetores não otimizados - Classificador MLP



(b) Vetores recozidos - Classificador SVC



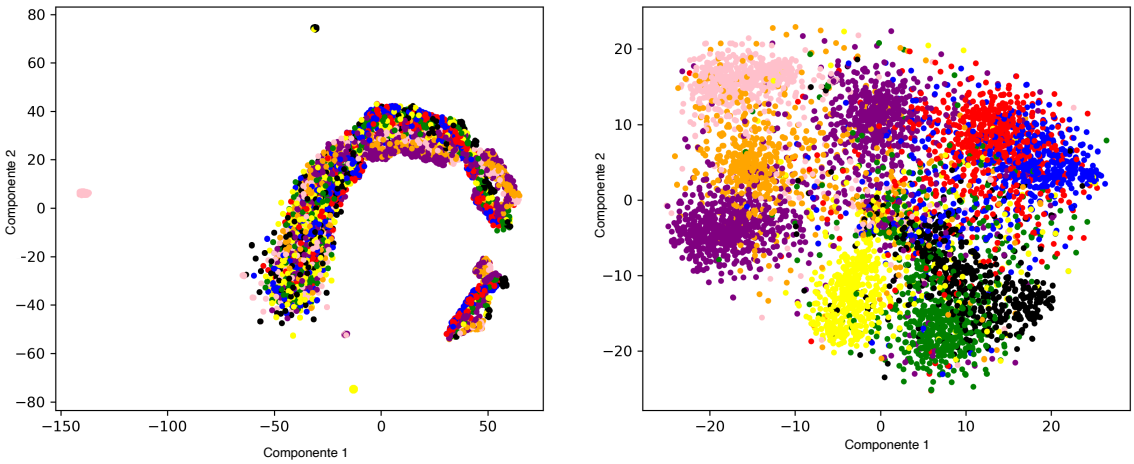
(c) Vetores CMA-ES - Classificador MLP



(d) Vetores BO - Classificador XGBC

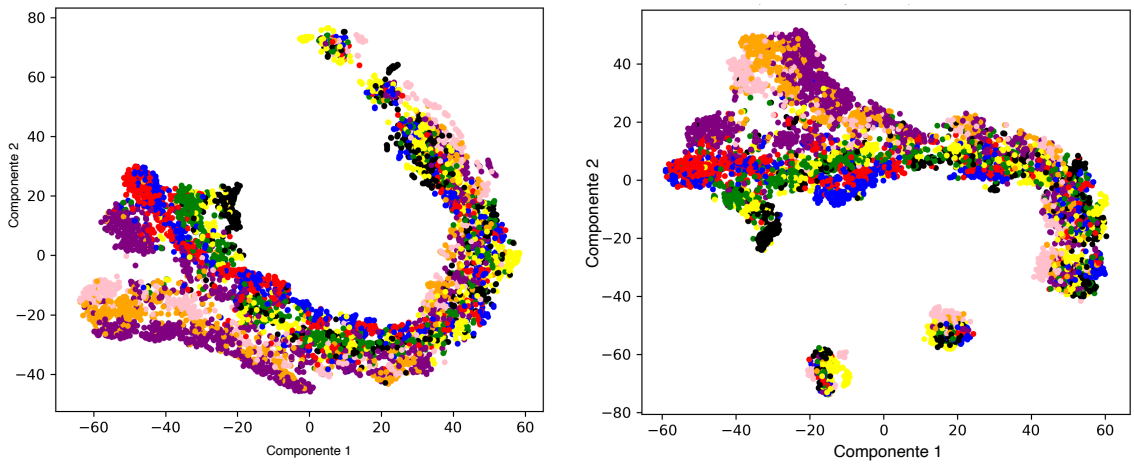
Figura 9 – Vetores PVDBOW com 100 dimensões

A visualização dos agrupamentos de documentos fica comprometida sem o processo de recozimento (Figura 10a) pela não dispersão das classes durante o processo de aquecimento e resfriamento do modelo. Assim, constatamos que o processo de recozimento simulado contribui para melhores agrupamentos de dados em altas dimensões (Figura 10b), sendo visivelmente superior aos vetores otimizados com CMA-ES (Figura 10c) e BO (Figura 10d).



(a) Vetores não otimizados

(b) Vetores recozidos



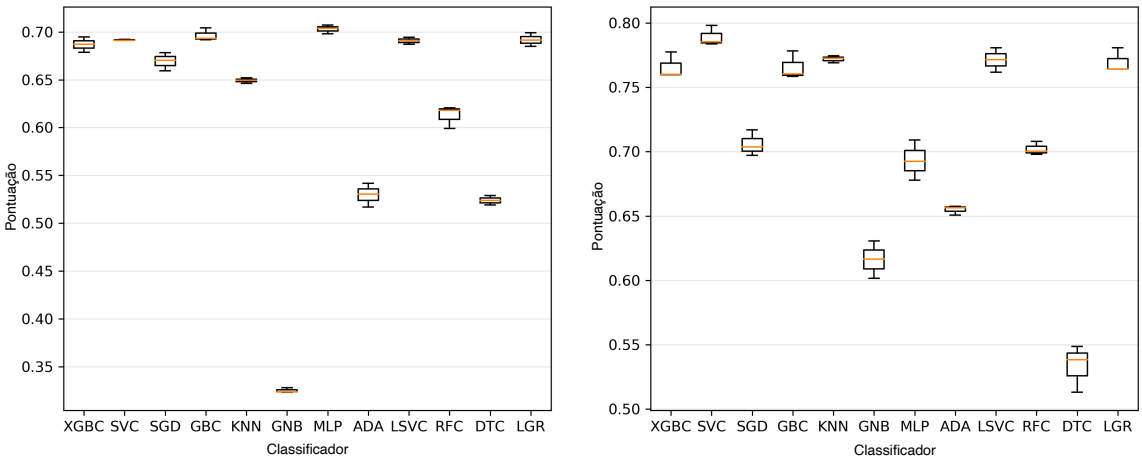
(c) Vetores CMA-ES

(d) Vetores BO

Figura 10 – t-SNE - Vetores PVDBOW com 100 dimensões

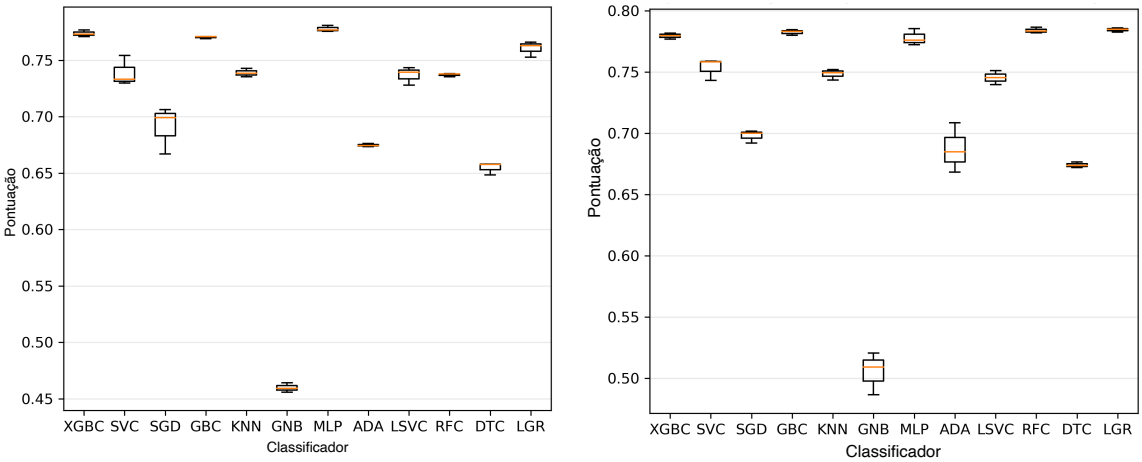
5.1.2 Vetores com 300 dimensões

Na estratégia com trezentas dimensões, os vetores não otimizados (Figura 11a) foram novamente superados pelos vetores recozidos (Figura 11b), pelos vetores CMA-ES (Figura 11c) e pelos vetores BO (Figura 11d).



(a) Vetores não otimizados

(b) Vetores recozidos

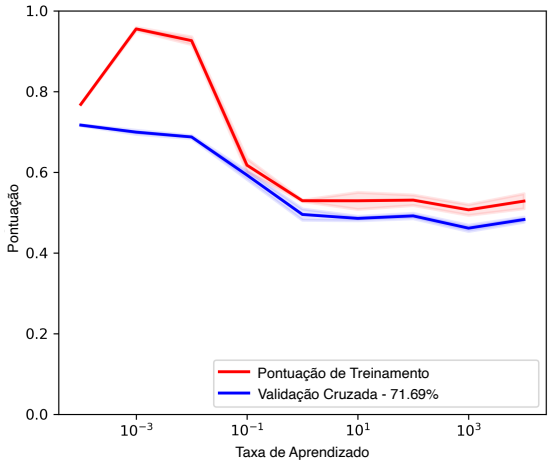


(c) Vetores CMA-ES

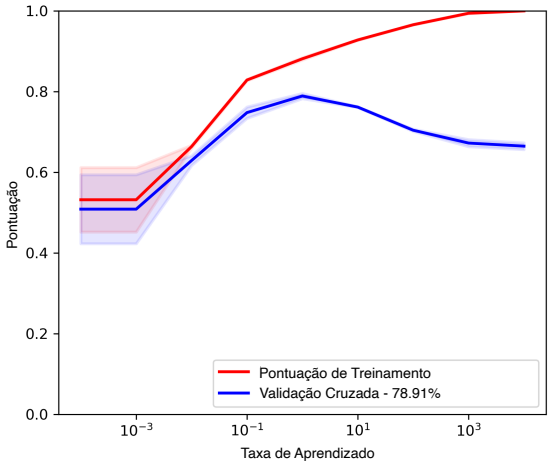
(d) Vetores BO

Figura 11 – Vetores PVDBOW com 300 dimensões

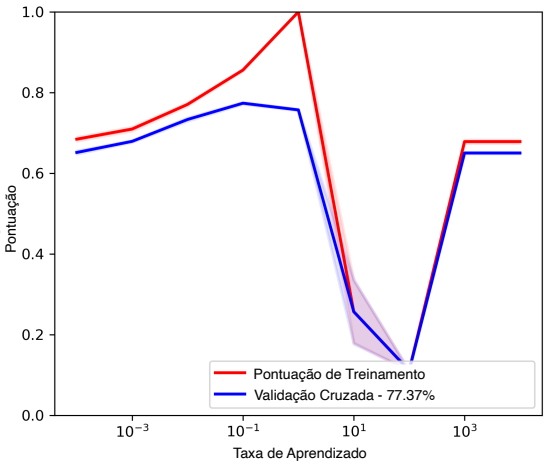
No experimento com trezentas dimensões, o classificador MLP obteve novamente a melhor pontuação (Figura 12a) alcançando, também, uma pontuação superior à observada no experimento com cem dimensões (Figura 9a) ($\Delta_{Score} = |Score_{(MLP)} - Score_{(MLP)}| = |71.69 - 64.62| = 7.07\%$). Não observamos diferenças significativas entre os vetores recozidos (Figura 12b) CMA-ES (Figura 12c) e BO (Figura 12d).



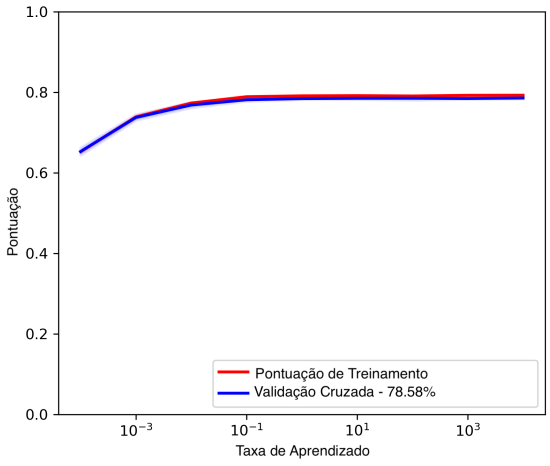
(a) Vetores não otimizados - Classificador MLP



(b) Vetores recozidos - Classificador SVC



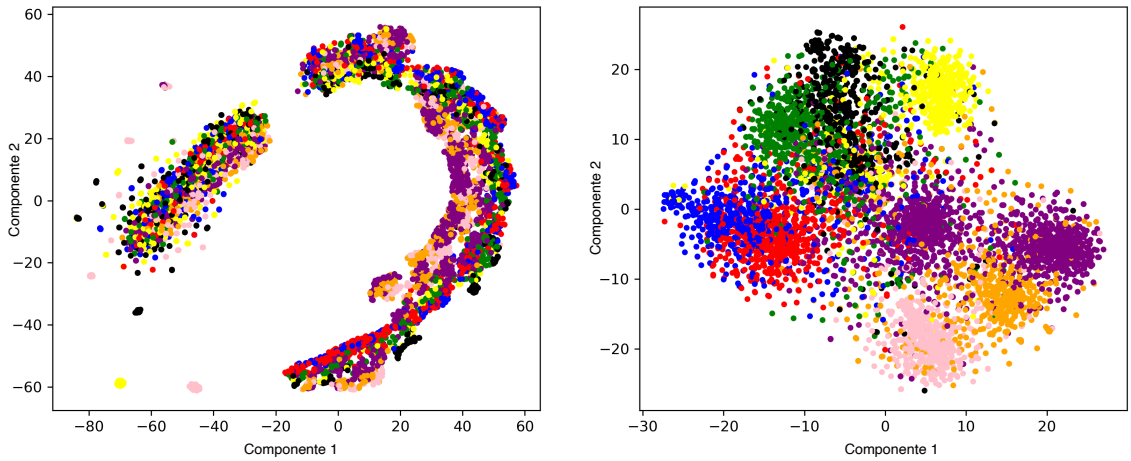
(c) Vetores CMA-ES - Classificador XGB Classifier



(d) Vetores BO - Classificador LGR

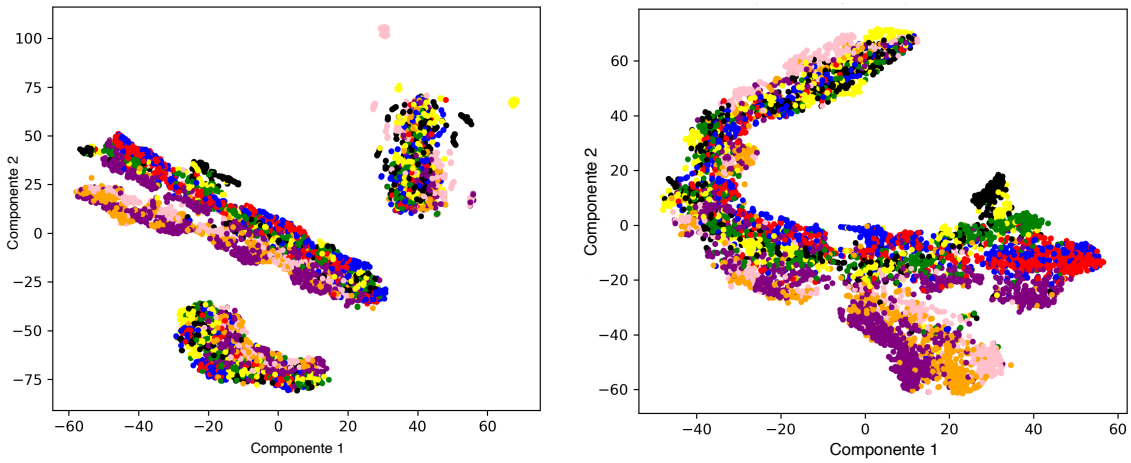
Figura 12 – PVDBOW vectors with 300 dimensions

Observamos que o processo de recozimento simulado promove um melhor agrupamento das classes dos vetores, conforme demonstrado nas figuras abaixo (Figuras 13a, 13b, 13c, 13d).



(a) Vetores não otimizados

(b) Vetores recozidos



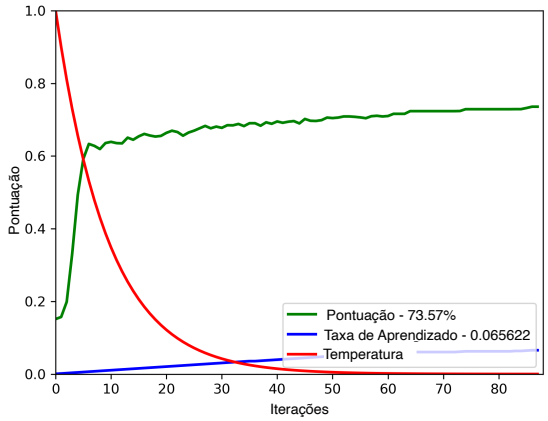
(c) Vetores CMA-ES

(d) Vetores BO

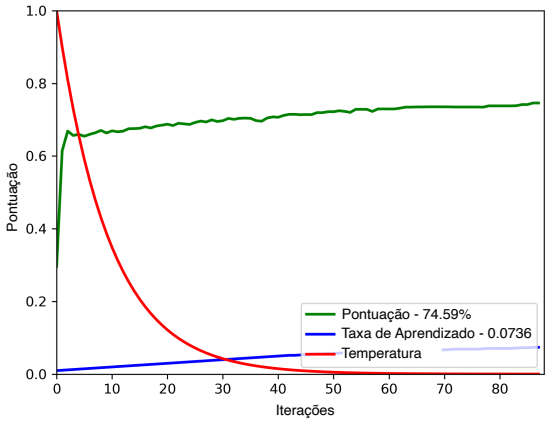
Figura 13 – t-SNE - Vetores PVDBOW com 300 dimensões

5.1.3 Resultados das iterações do algoritmo de recozimento simulado

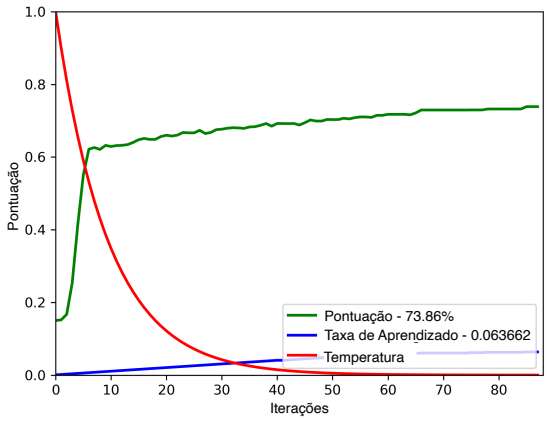
Após os experimentos de recozimento simulado (SA) dos vetores com $100, 300, 600$ e 1000 dimensões (Figuras 14a, 14b, 14c, 14d), foi verificado que a melhor representatividade semântica ocorre com 300 dimensões (Figura 14b).



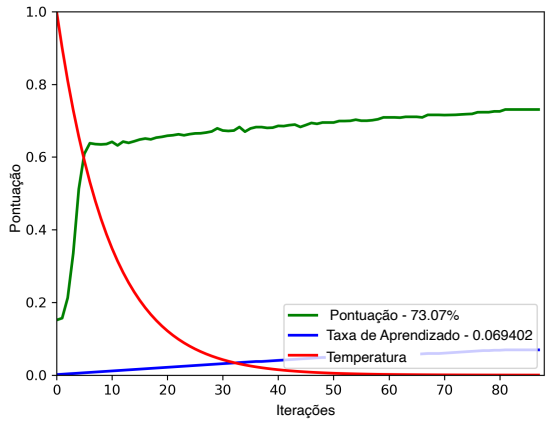
(a) 100 dimensões



(b) 300 dimensões



(c) 600 dimensões

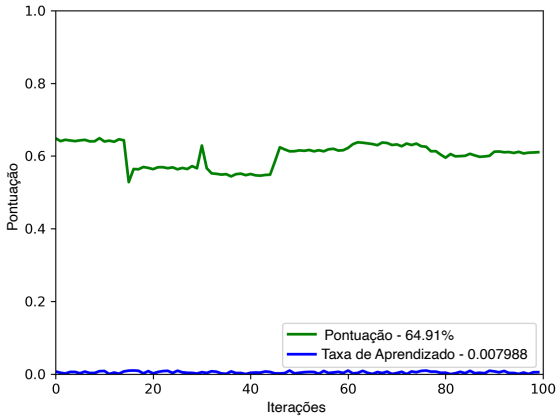


(d) 1000 dimensões

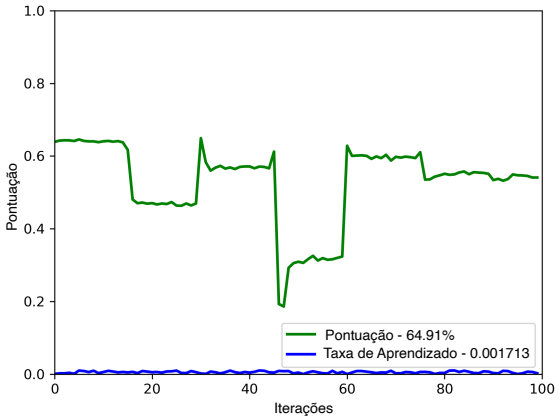
Figura 14 – Iterações do algoritmo SA

5.1.4 Resultados das iterações do algoritmo da estratégia evolutiva de adaptação da matriz de covariância

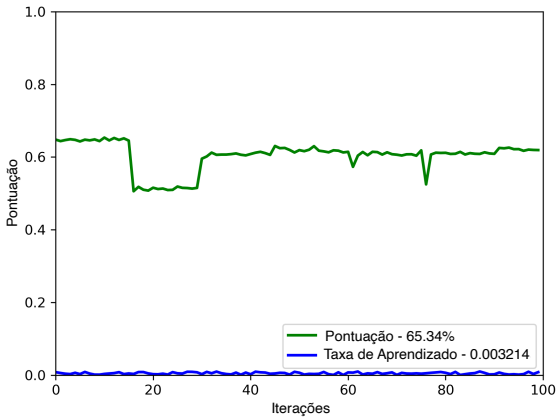
Os vetores otimizados com Estratégia Evolutiva de Adaptação da Matriz de Covariância (CMA-ES) obtiveram um aumento marginal na pontuação com o aumento das dimensões (Figuras 15a, 15b, 15c, 15d).



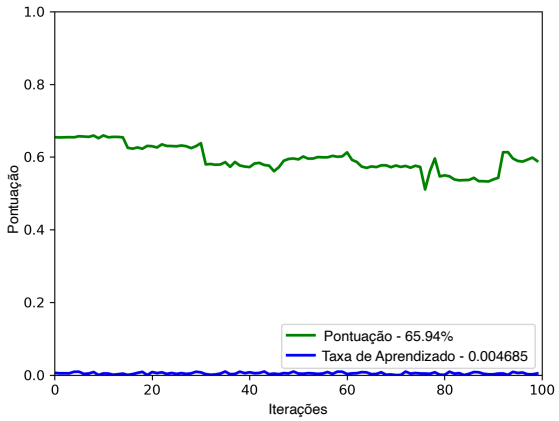
(a) 100 dimensões



(b) 300 dimensões



(c) 600 dimensões

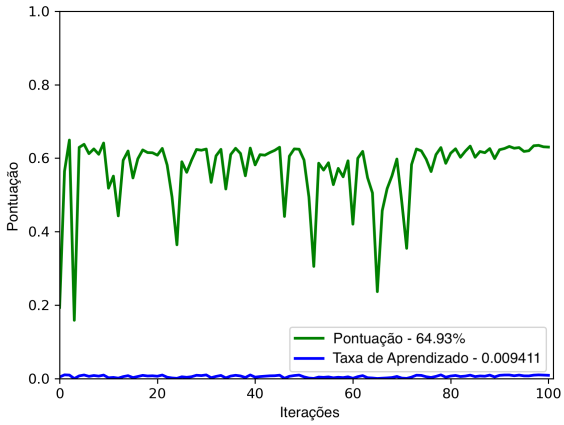


(d) 1000 dimensões

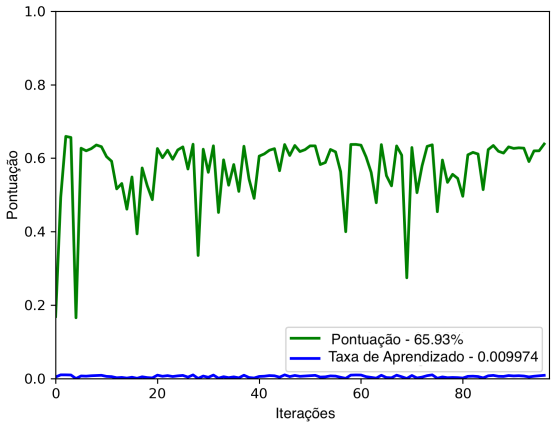
Figura 15 – Iterações do algoritmo CMA-ES

5.1.5 Resultados das iterações do algoritmo de otimização bayesiana

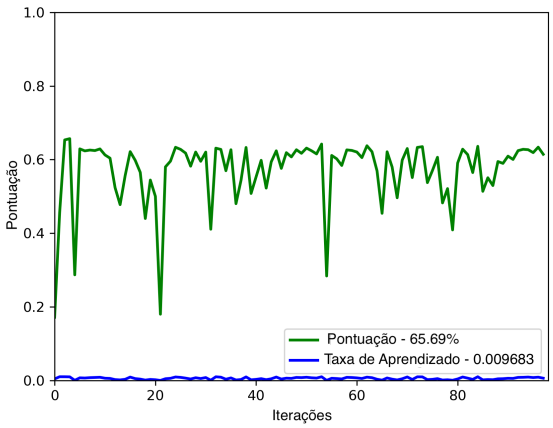
A otimização bayesiana (BO) não propiciou um aumento significativo na pontuação dos vetores com o aumento da dimensionalidade (Figuras 16a, 16b, 16c, 16d).



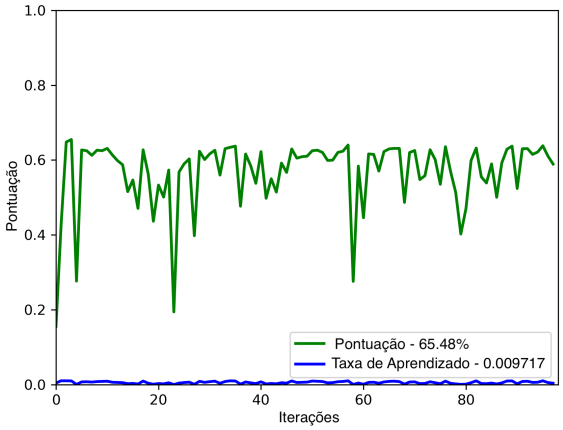
(a) 100 dimensões



(b) 300 dimensões



(c) 600 dimensões



(d) 1000 dimensões

Figura 16 – Iterações do algoritmo BO

5.1.6 Testes Estatísticos

A estatística descritiva (Tabela 8) sinaliza que houve um aumento na pontuação média de todos os classificadores com o aumento da dimensionalidade de 100 para 300.

Tabela 8 – Estatística descritiva

	Não otimizado		SA		CMA-ES		BO	
	100	300	100	300	100	300	100	300
Válido	12	12	12	12	12	12	12	12
Ausente	0	0	0	0	0	0	0	0
Média	56.5	62.7	71.0	71.2	68.9	71.0	71.9	72.7
Desvio Padrão	9.8	11.3	7.2	7.5	9.3	8.7	7.6	7.8
Mínimo	31.9	32.5	54.6	53.9	42.6	46.0	51.3	51.4
Máximo	64.6	71.7	77.5	78.9	75.9	77.4	77.6	78.6

Quando agrupamos as pontuações médias das dimensões avaliadas, observamos que os otimizadores obtiveram pontuações aproximadas (Tabela 9).

Tabela 9 – Estatística descritiva

	N	Média	SD	SE
SA	24	71.1	7.2	1.5
Não otimizado	24	59.6	10.9	2.2
CMA-ES	24	70.0	8.9	1.8
BO	24	72.3	7.5	1.5

Considerando um nível de significância 0.05, não observamos diferença estatística significativa entre os otimizadores SA, CMA-ES e BO (Tabela 10), o que nos direciona a não rejeitar a hipótese nula.

Tabela 10 – Teste t pareado

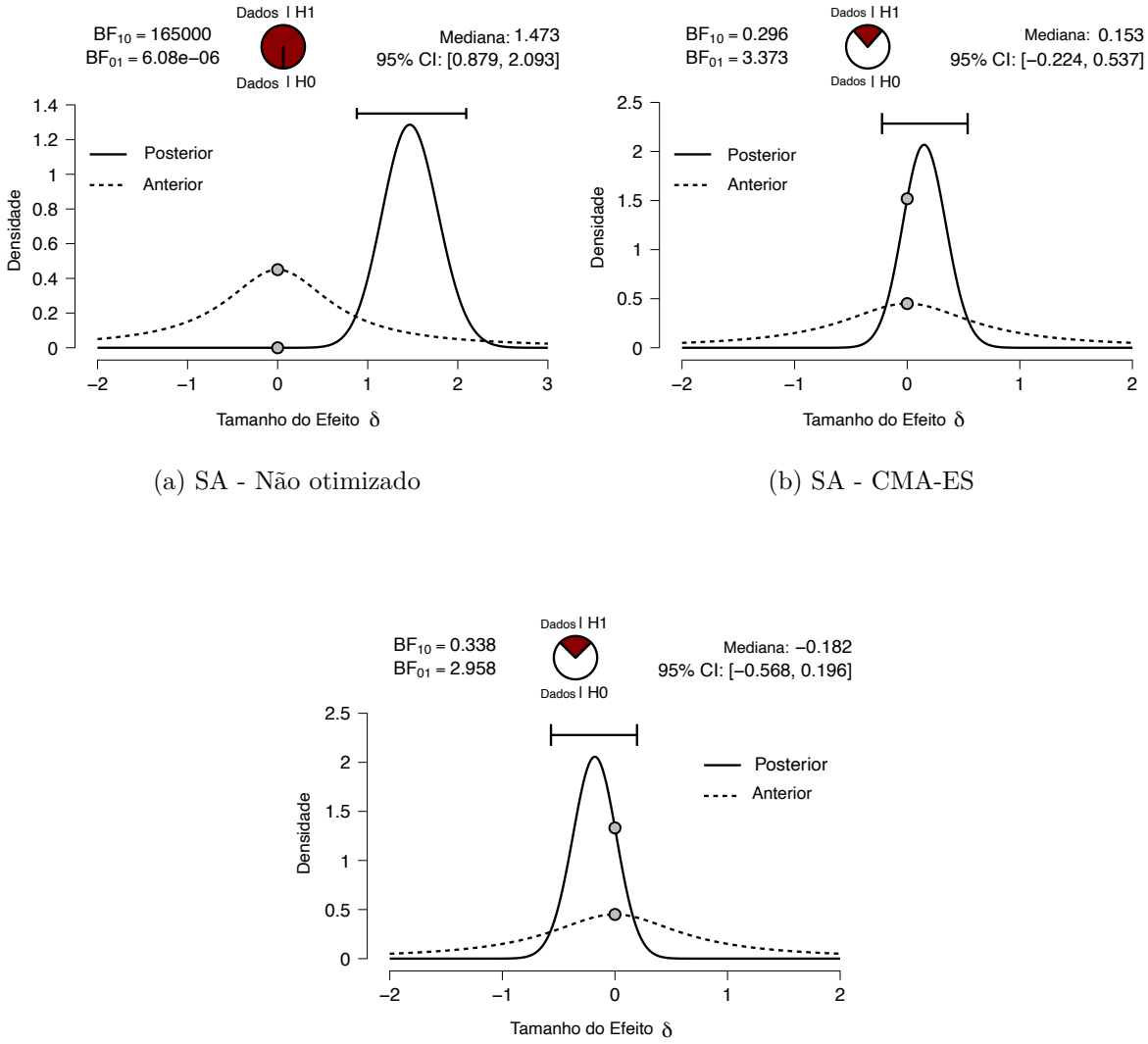
Medida 1		Medida 2	t	df	p	Dif. Médias	Diferença SE
SA	-	Não otimizado	7.7	23	$< .001$	11.5	1.5
	-	CMA-ES	0.8	23	0.4	1.1	1.3
	-	BO	-1.0	23	0.3	-1.2	1.2

Utilizando o teste Bayesiano (Tabela 11) para comparar o modelo recozido com o não otimizado, verificamos que o fator de Bayes (BF_{10}) é aproximadamente 164545.9 vezes mais favorável à hipótese alternativa, o que significa que a hipótese alternativa prevê os dados 164545.9 vezes mais que a hipótese nula. Comparando o modelo recozido com os modelos CMA-ES e BO, a hipótese alternativa prevê os dados 0.3 vezes mais ($BF_{10} = 0.3$) que a hipótese nula em ambos, isto é, não há diferença estatística significativa entre os otimizadores.

Tabela 11 – Teste t Bayesiano pareado

Medida 1		Medida 2	BF_{10}	erro %
SA	-	Não otimizado	164545.9	8.3×10^{-10}
	-	CMA-ES	0.3	3.7×10^{-3}
	-	BO	0.3	1.2×10^{-2}

Analisando as distribuições anterior e posterior para comparar os modelos recozido e não otimizado, observamos que o ponto da distribuição anterior é superior ao da posterior (Figura 17a). Desse modo, o fator de Bayes apóia a hipótese alternativa. Já comparando o modelo recozido com os otimizadores CMA-ES e BO, foi observado que o ponto da distribuição posterior é superior ao ponto da distribuição anterior nos dois otimizadores (Figuras 17b, 17c), o que indica que o fator de Bayes apóia a hipótese nula, não havendo diferença estatística entre os otimizadores.



(a) SA - Não otimizado

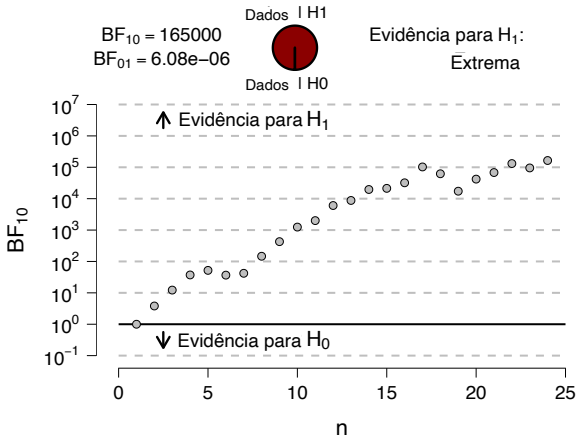
(b) SA - CMA-ES

(c) SA - BO

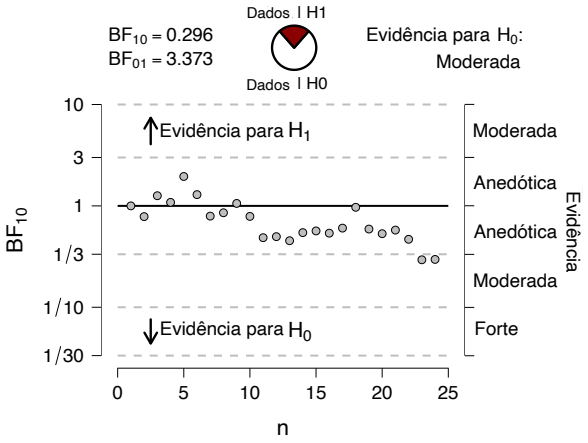
Figura 17 – Distribuição Anterior e Posterior - PVDBOW

A análise sequencial permite avaliar o grau de evidência para cada uma das hipóteses (nula ou alternativa). Um fator de Bayes acima de 1 é uma evidência favorável à hipótese alternativa e abaixo de 1 é favorável à hipótese nula.

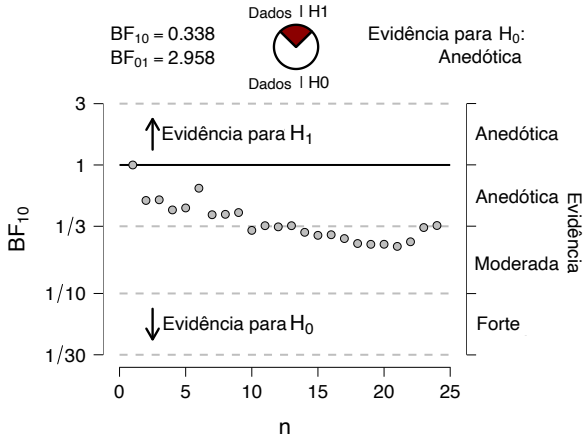
Na comparação do modelo recozido com o modelo não otimizado, constatamos uma evidência extremamente favorável à hipótese alternativa (Figura 18a), indicando que o recozido se difere, estatísticamente, do não otimizado. Já comparando o modelo recozido com os modelos otimizados CMA-ES (Figura 18b) e BO (Figura 18c) foram constatadas evidências moderada e anedótica, respectivamente, favoráveis à hipótese nula.



(a) SA - Não otimizado



(b) SA - CMA-ES



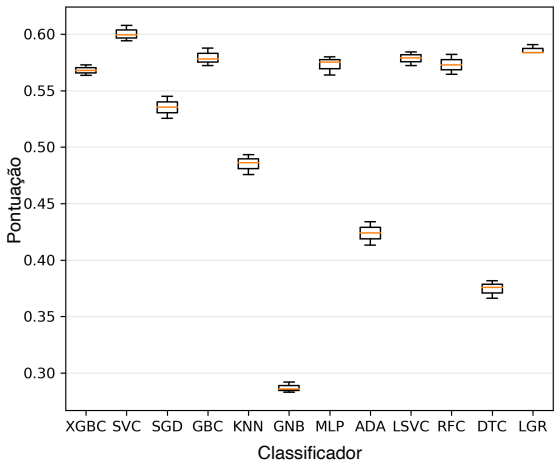
(c) SA - BO

Figura 18 – Análise sequencial - PVDBOW

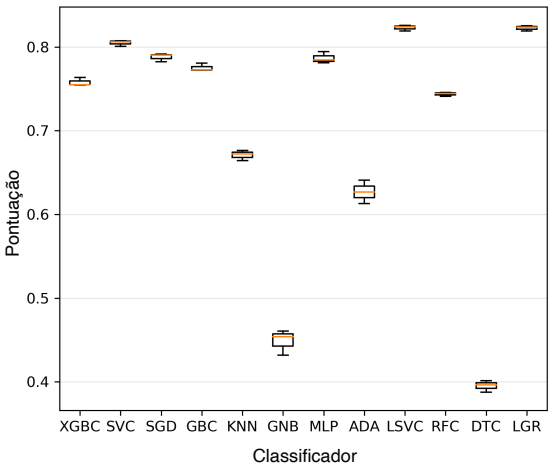
5.2 PVDM/*Averaging*

5.2.1 Vetores com 100 dimensões

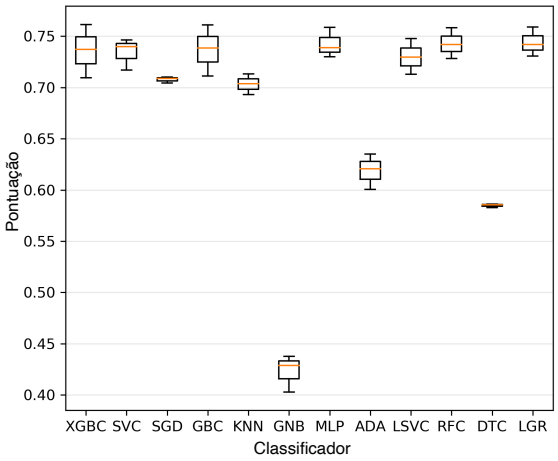
Na estratégia de treinamento com 100 dimensões, os vetores não otimizados (Figura 19a) tiveram um desempenho inferior aos vetores recozidos (Figura 19b). As pontuações dos classificadores nos vetores recozidos (Figura 19b) indicam valores superiores aos vetores CMA-ES (Figura 19c). Nessa estratégia de treinamento, os vetores BO (Figura 19d) não obtiveram um desempenho satisfatório.



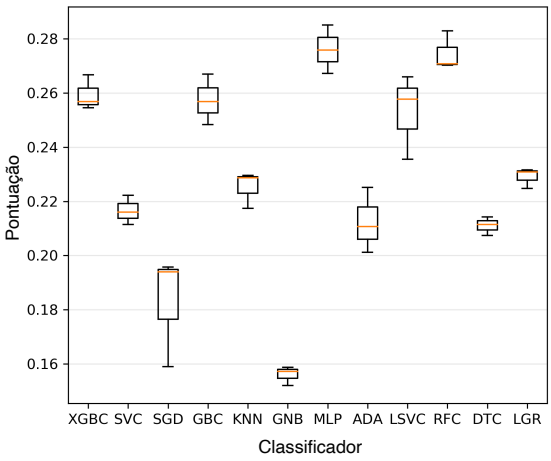
(a) Vetores não otimizados



(b) Vetores recozidos



(c) Vetores CMA-ES



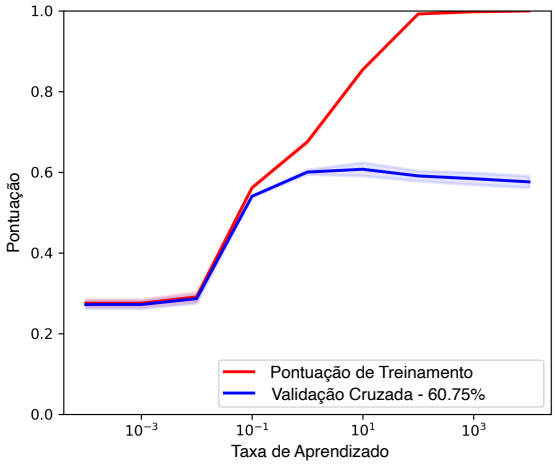
(d) Vetores BO

Figura 19 – Vetores PVDM-AVERAGING com 100 dimensões

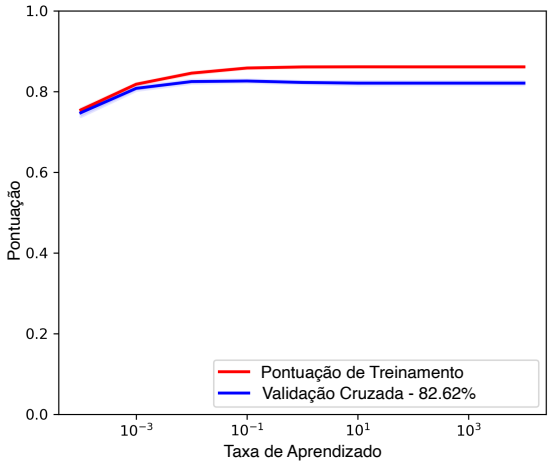
No experimento com os vetores não otimizados (Figura 20a), o classificador de melhor pontuação foi o SVC. Nos vetores recozidos, no entanto, o classificador LGR obteve melhor pontuação que o SVC (Figura 20b) ($\Delta_{Score} = |Score_{(SVC)} - Score_{(LGR)}| = |60.75 - 82.62| = 21.87\%$). Comparando o classificador LGR dos vetores recozidos com o mesmo classificador dos vetores CMA-ES (Figura 20c), constatamos uma superioridade para os recozidos ($\Delta_{Score} = |Score_{(LGR)} - Score_{(LGR)}| = |82,62 - 74,44| = 8,18\%$). Já os vetores BO tiveram um desempenho insatisfatório em todos os classificadores (Figura 20d).

Nos experimentos observamos que os vetores recozidos com a estratégia de treina-

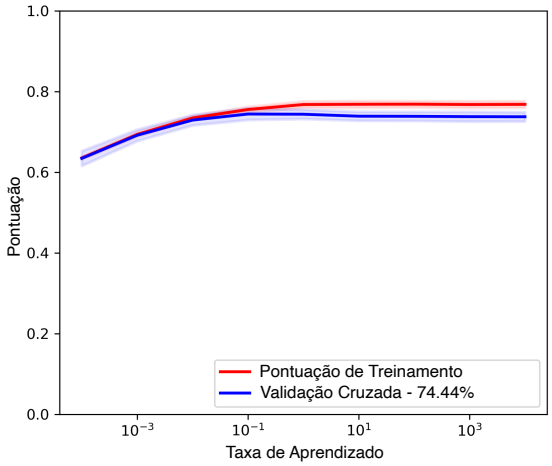
mento PVDM-Averaging superou os vetores recozidos com a estratégia PVDBOW. Isso corrobora o que foi constatado por Mikolov e Le (LE; MIKOLOV, 2014), em que o modelo PVDM demonstrou ser consistentemente superior ao modelo PVDBOW. A superioridade é justificada pela preservação da ordem das palavras no documento, sendo fundamental para o contexto semântico.



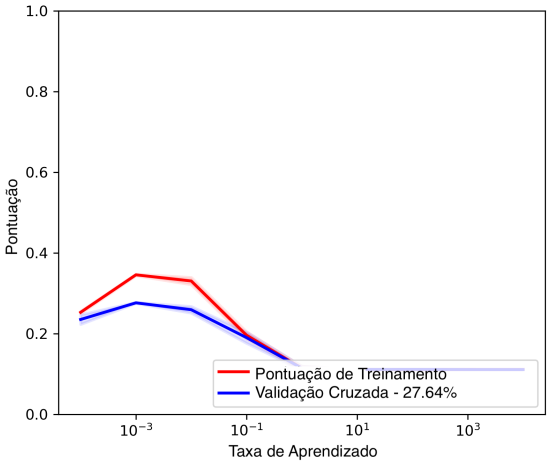
(a) Vetores não otimizados - Classificador SVC



(b) Vetores recozidos - Classificador LGR



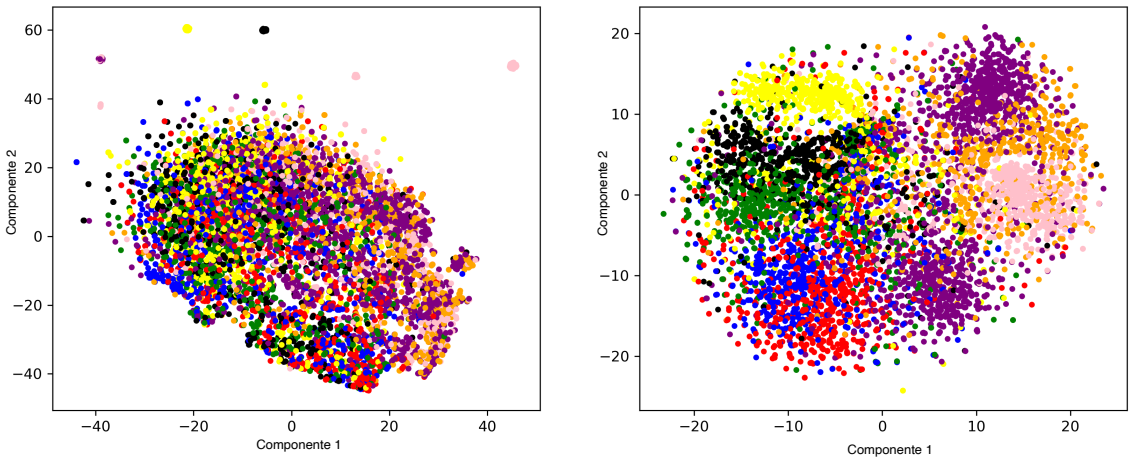
(c) Vetores CMA-ES - Classificador LGR



(d) Vetores BO - Classificador MLP

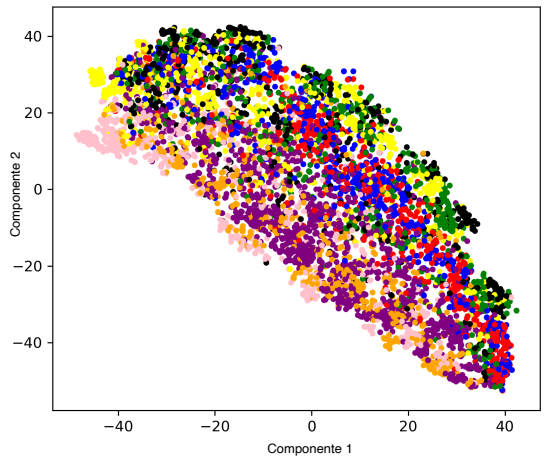
Figura 20 – Vetores PVDM-AVERAGING com 100 dimensões

Na visualização dos vetores não otimizados em 100 dimensões, os agrupamentos se apresentam nebulosos (Figura 21a), assim como, nos vetores CMA-ES e BO. Já nos vetores recozidos, a visualização dos agrupamentos é mais nítida (Figuras 21b, 21c, 21d).

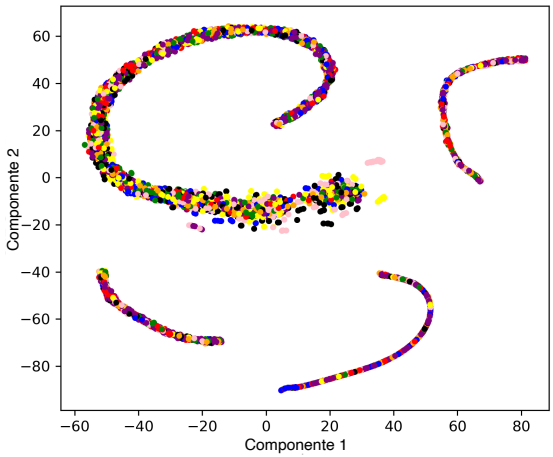


(a) Vetores não otimizados

(b) Vetores recozidos



(c) Vetores CMA-ES

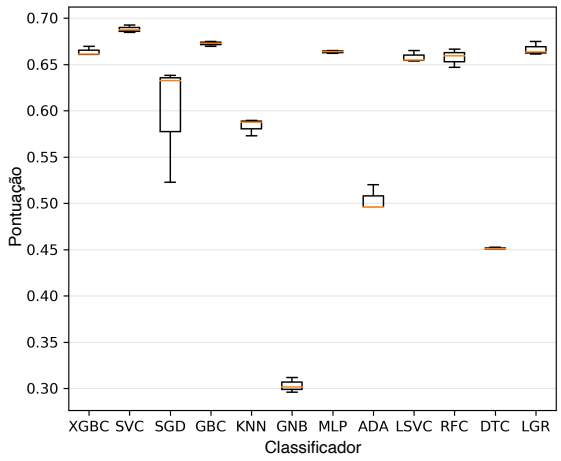


(d) Vetores BO

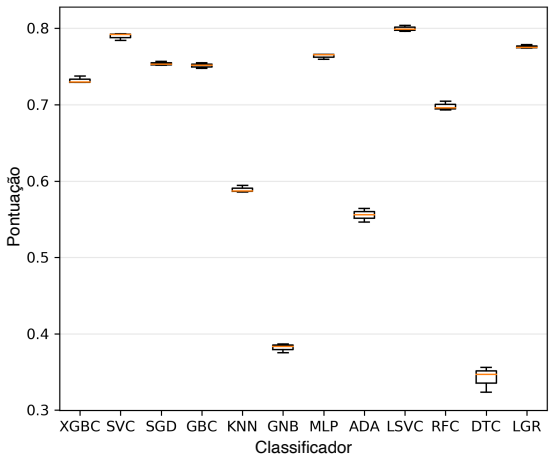
Figura 21 – t-SNE - Vetores PVDM-AVERAGING com 100 dimensões

5.2.2 Vetores com 300 dimensões

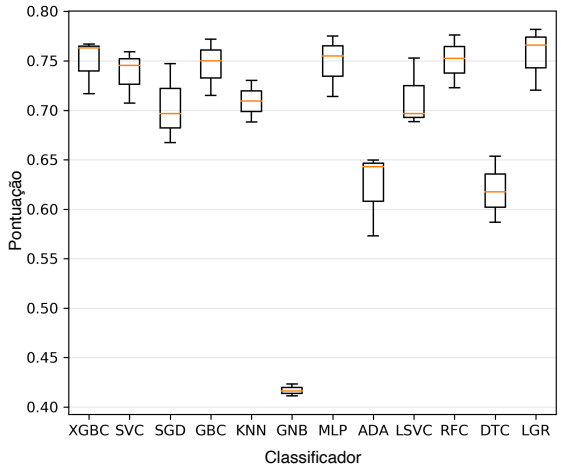
Na estratégia com trezentas dimensões, a pontuação dos vetores não otimizados (Figura 22a) foi inferior à pontuação dos vetores recozidos (Figura 22b) em todos os classificadores. Comparando os vetores recozidos e CMA-ES (Figura 22c), constatamos uma pontuação próxima em ambos os vetores. As pontuações dos vetores BO (Figura 22d) foram insatisfatórias em todos os classificadores.



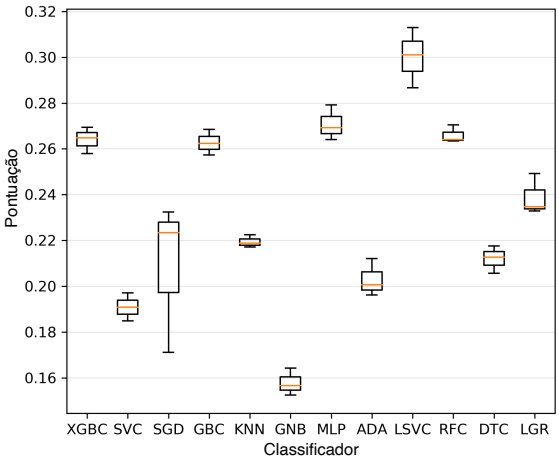
(a) Vetores não otimizados



(b) Vetores recozidos



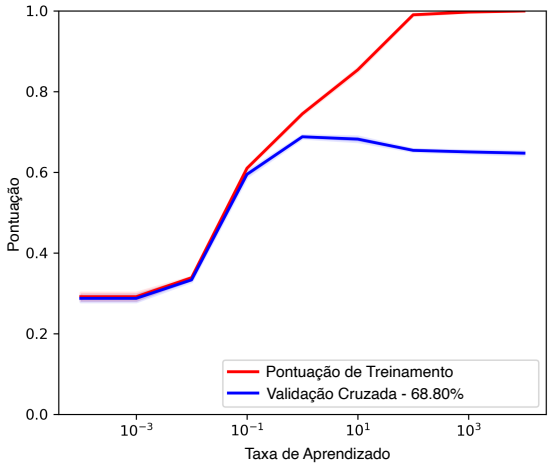
(c) Vetores CMA-ES



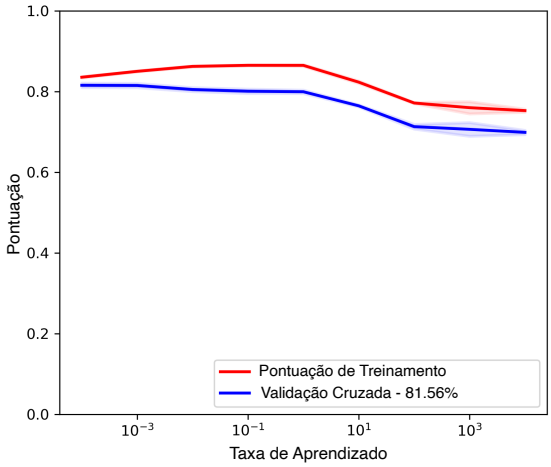
(d) Vetores BO

Figura 22 – Vetores PVDM-AVERAGING com 300 dimensões

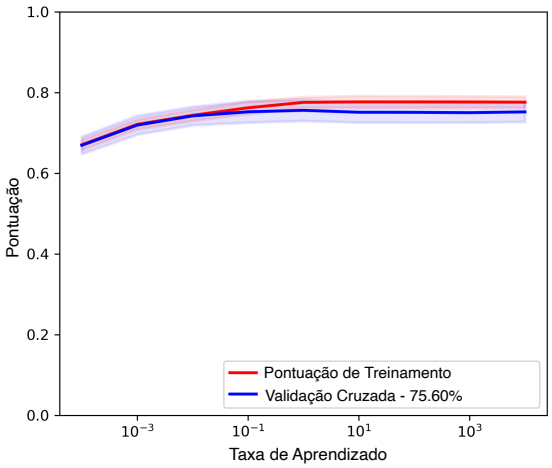
No modelo não otimizado, o classificador SVC obteve a melhor pontuação (Figura 23a), mas foi superado pelo classificador LSVC dos vetores recozidos (Figura 23b) ($\Delta_{Score} = |Score_{(SVC)} - Score_{(LSVC)}| = |68.80 - 81.56| = 12.76\%$). Observamos que houve um aumento na pontuação média de todos os classificadores dos vetores CMA-ES (Figura 22c), destacando-se o LGR (Figura 23c) como o de melhor pontuação. Os resultados obtidos pelos vetores BO foram novamente insatisfatórios (Figura 23d).



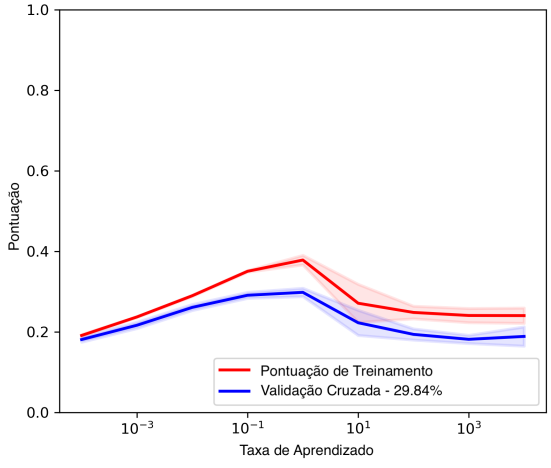
(a) Vetores não otimizados - Classificador SVC



(b) Vetores recozidos - Classificador LSVC



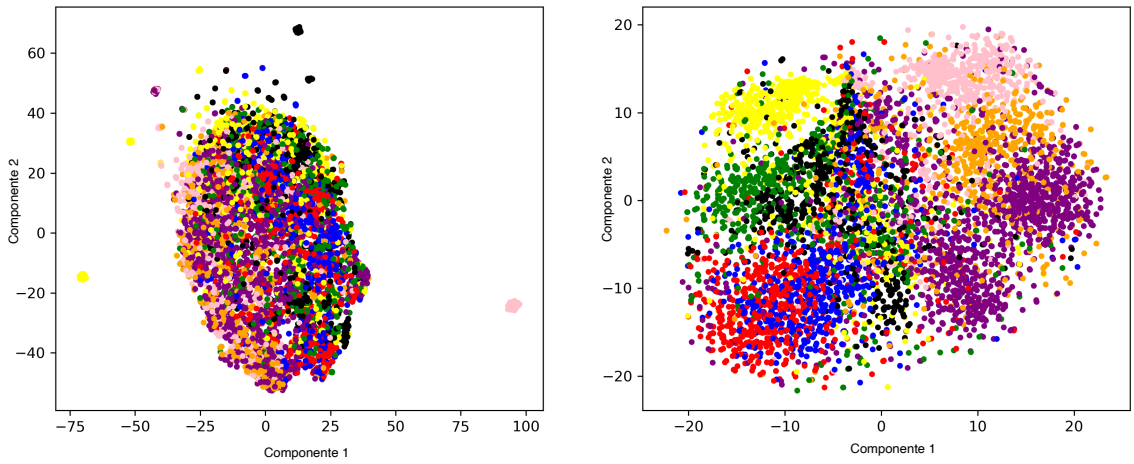
(c) Vetores CMA-ES - Classificador LGR



(d) Vetores BO - Classificador LSVC

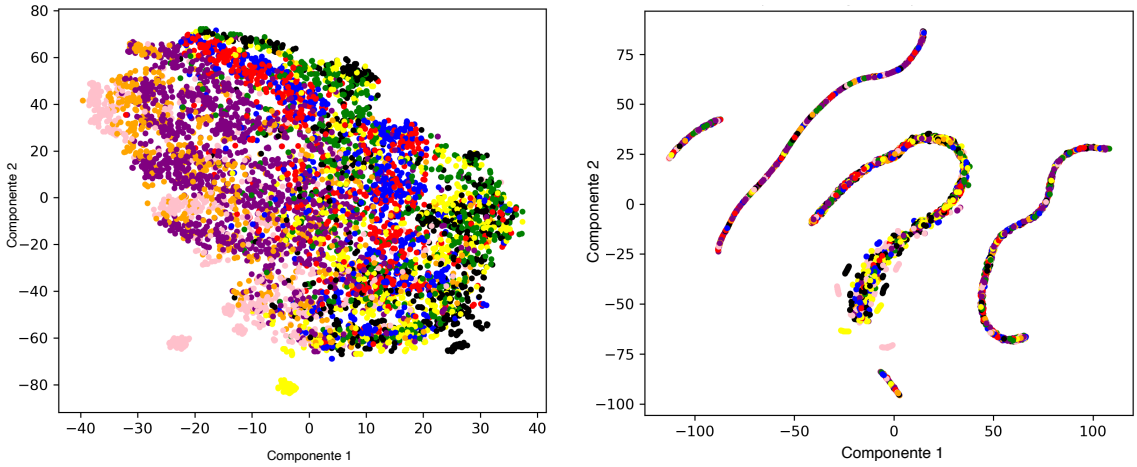
Figura 23 – Vetores PVDM-AVERAGING com 300 dimensões

Uma melhor representação visual dos agrupamentos de documentos também foi observada em 300 dimensões. Os agrupamentos dos vetores não otimizados (Figura 24a) se mostraram difusos quando comparados aos agrupamentos dos vetores recozidos (Figura 24b), mais evidentes. Com o otimizador CMA-ES, os vetores que apresentaram os agrupamentos mais nítidos foram os com 300 dimensões (Figura 24c). Quando comparados aos vetores recozidos, no entanto, mostraram-se mais difusos. Os vetores BO, mais uma vez, apresentaram agrupamentos insatisfatórios (Figura 24d).



(a) Vetores não otimizados

(b) Vetores recozidos



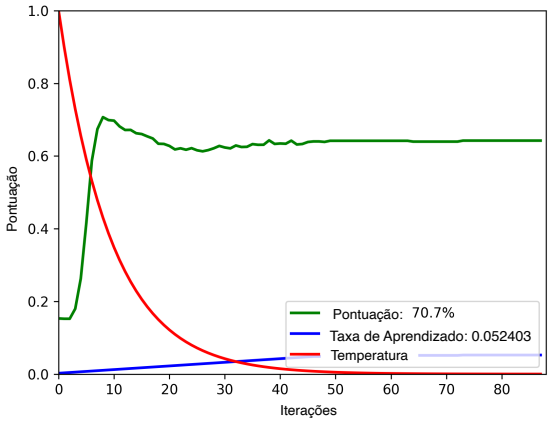
(c) Vetores CMA-ES

(d) Vetores BO

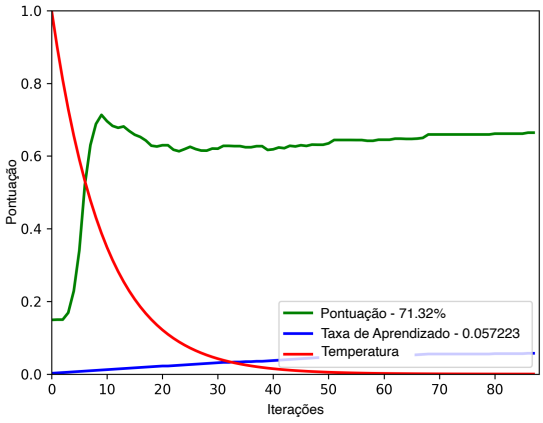
Figura 24 – t-SNE - Vetores PVDM-AVERAGING com 300 dimensões

5.2.3 Resultados das iterações do algoritmo de recozimento simulado

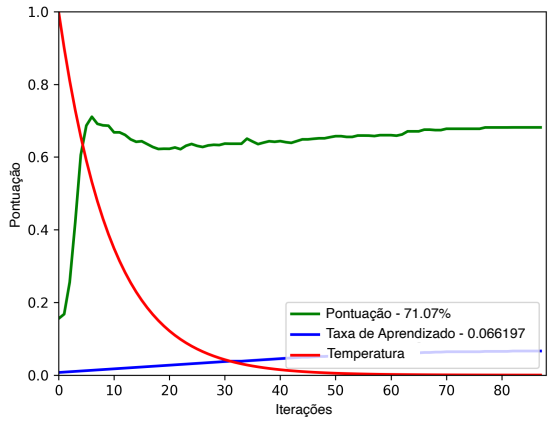
As simulações de recozimento simulado (SA) dos vetores com 100, 300, 600 e 1000 dimensões demonstraram que não houve aumento na pontuação com a dimensionalidade superior a 300 (Figuras 25a, 25b, 25c, 25d). Os experimentos sinalizam, assim, que em 300 dimensões atingimos a melhor representatividade semântica dos vetores de documentos.



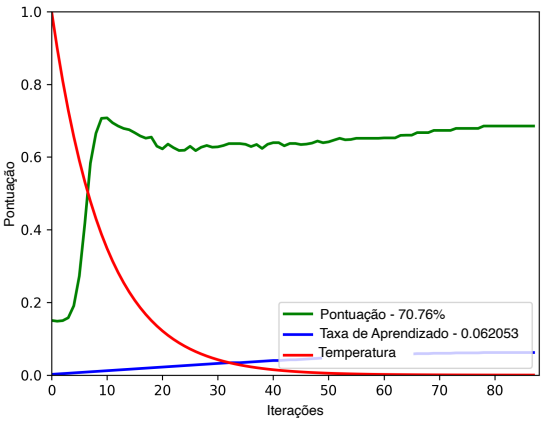
(a) 100 dimensões



(b) 300 dimensões



(c) 600 dimensões

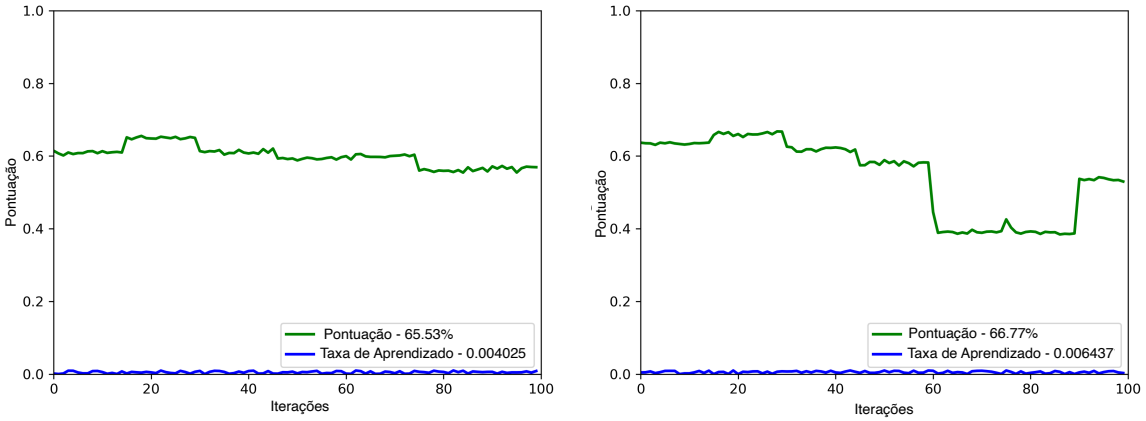


(d) 1000 dimensões

Figura 25 – Iterações do algoritmo SA

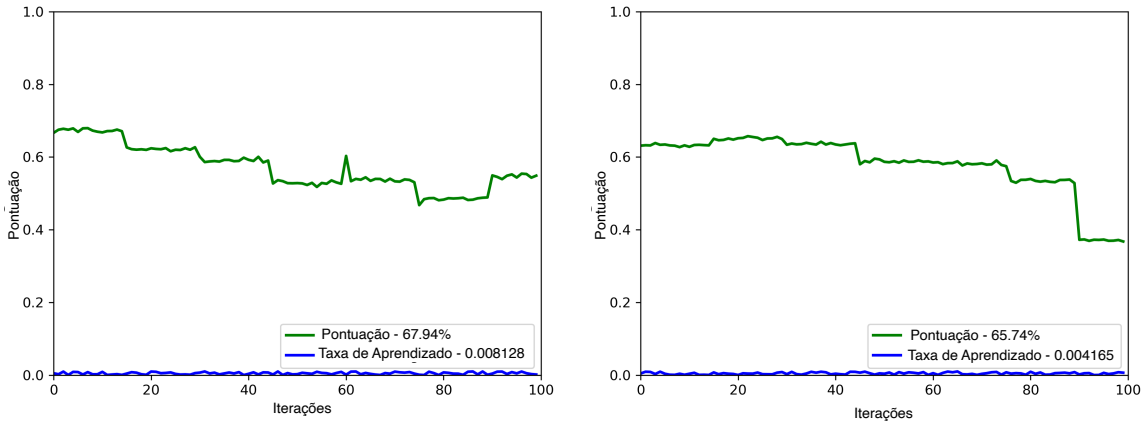
5.2.4 Resultados das iterações do algoritmo da estratégia evolutiva de adaptação da matriz de covariância

Constatamos pelos experimentos que o otimizador CMA-ES não é performático com o aumento da dimensionalidade $(100, 300, 600, 1000)$ (Figuras 26a, 26b, 26c, 26d).



(a) 100 dimensões

(b) 300 dimensões



(c) 600 dimensões

(d) 1000 dimensões

Figura 26 – Iterações do algoritmo CMA-ES

5.2.5 Testes Estatísticos

Na tabela com a estatística descritiva, os vetores BO tiveram uma pontuação média inferior à obtida pelos modelos não otimizados, recozidos e CMA-ES. Os vetores recozidos atingiram uma média próxima aos vetores CMA-ES em 100 e 300 dimensões (Table 12).

Tabela 12 – Estatística descritiva

	Não otimizado		SA		CMA-ES		BO	
	100	300	100	300	100	300	100	300
Válido	12	12	12	12	12	12	12	12
Ausente	0	0	0	0	0	0	0	0
Média	51.6	59.5	70.7	66.7	68.5	69.2	23.2	23.6
Desvio Padrão	10.2	11.9	14.2	16.1	9.6	9.7	3.5	4.1
Mínimo	28.4	30.1	40.0	34.5	42.7	41.8	15.8	15.6
Máximo	60.8	68.8	82.6	81.6	74.5	75.6	27.6	29.8

Considerando o agrupamento das pontuações médias das dimensões avaliadas, verificamos que os vetores recozidos apresentaram uma pontuação média próxima à dos vetores CMA-ES. Os vetores BO não apresentaram um desempenho satisfatório, com pontuações inferiores a de todos os modelos avaliados (Tabela 13).

Tabela 13 – Estatística descritiva

	N	Média	SD	SE
SA	24	68.7	15.0	3.1
Não otimizado	24	55.5	11.6	2.4
CMA-ES	24	68.8	9.4	1.9
BO	24	23.4	3.7	0.8

Na tabela abaixo (Tabela 14), não observamos diferença estatística significativa entre os otimizadores SA e CMA-ES, levando em consideração um nível de significância $\alpha = 0.05$, o que aponta para a não rejeição da hipótese nula.

Tabela 14 – Paired Samples T-Test

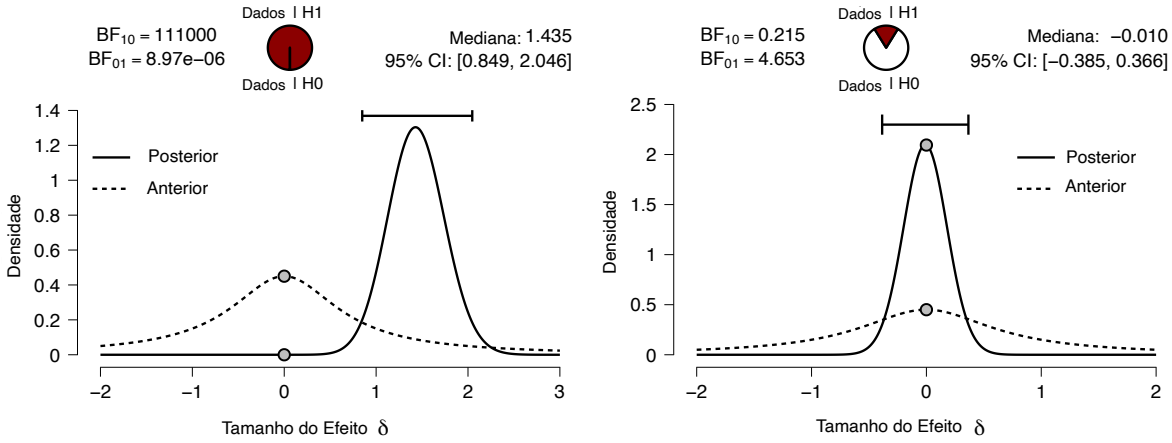
Measure 1		Measure 2	t	df	p	Mean Difference	SE Difference
SA	-	Não otimizado	7.5	23	$< .001$	13.2	1.8
	-	CMA-ES	-5.3×10^{-2}	23	1.0	-9.7×10^{-2}	1.8
	-	BO	16.9	23	$< .001$	45.3	2.7

Comparando o modelo recozido com o não otimizado (Tabela 15), verificamos que o fator de Bayes BF_{10} é mais favorável à hipótese alternativa (aproximadamente 111474.4 vezes), isto é, a hipótese alternativa prevê os dados 111474.4 vezes mais que a hipótese nula. Comparando o modelo recozido com o modelo BO, o fator de Bayes BF_{10} é mais favorável à hipótese alternativa (3.3×10^{11} vezes) e com o modelo CMA-ES, a hipótese alternativa prevê os dados 0.2 vezes mais ($BF_{10} = 0.2$) que a hipótese nula. Desse modo, não há diferença estatística significativa entre os dois otimizadores.

Tabela 15 – Teste t Bayesiano pareado

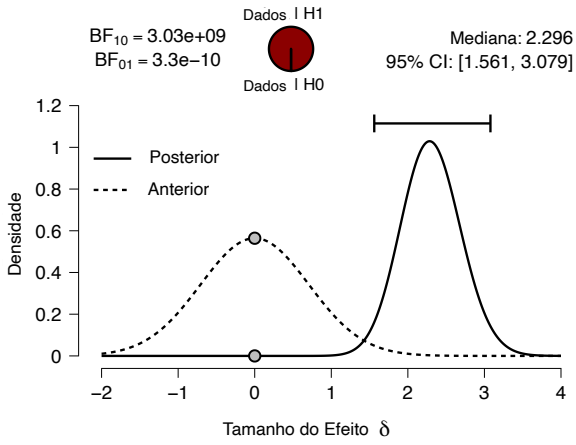
Medida 1		Medida 2	BF_{10}	erro %
SA	-	Não otimizado	111474.4	2.9×10^{-8}
	-	CMA-ES	0.2	3.2×10^{-2}
	-	BO	3.3×10^{11}	3.5×10^{-17}

Comparando o modelo recozido com o modelo não otimizado, o ponto da distribuição anterior é superior ao da distribuição posterior (Figura 27a), o que significa que o fator de Bayes apóia a hipótese alternativa. O modelo recozido também não apresenta diferença estatística em relação ao modelo CMA-ES, pois o ponto da distribuição posterior é superior ao ponto da anterior (Figura 27b). Comparando o modelo recozido com o BO, o fator de Bayes apóia a hipótese alternativa, indicando que o modelo recozido é superior ao modelo BO (Figura 27c).



(a) SA - Não otimizado

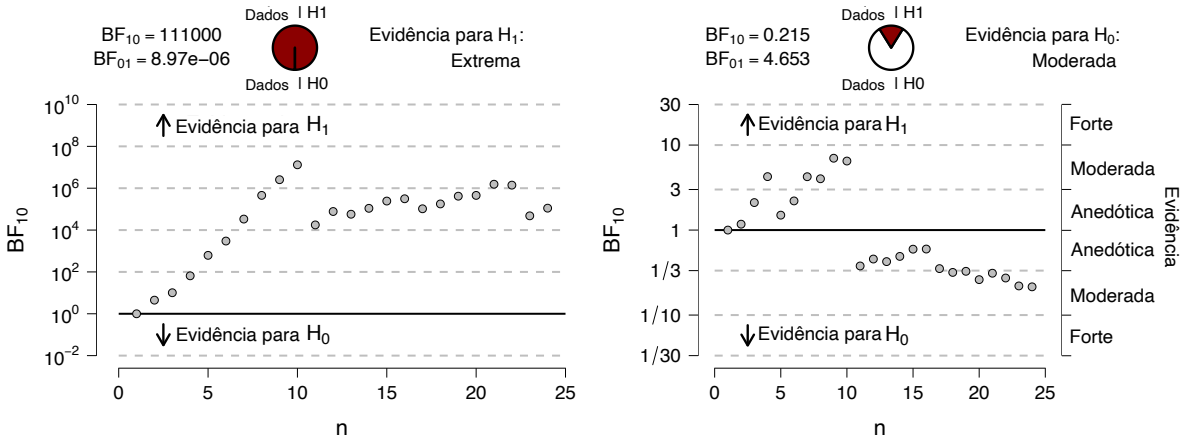
(b) SA - CMA-ES



(c) SA - BO

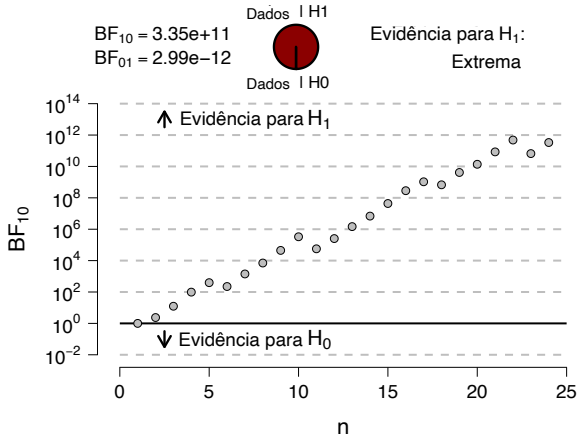
Figura 27 – Distribuição Anterior e Posterior - PVDM-Averaging

Verificamos pelo gráfico sequencial (Figura 28a) que há uma evidência extremamente favorável à hipótese alternativa, indicando que o modelo recozido se difere, estatísticamente, do não otimizado. Quando comparamos o modelo recozido com o CMA-ES (Figura 28b), constatamos uma evidência moderadamente favorável à hipótese nula. Os vetores BO tiveram um desempenho insatisfatório quando comparados aos vetores recozidos, com evidências extremas a favor da hipótese alternativa (Figura 28c).



(a) SA - Não otimizado

(b) SA - CMA-ES



(c) SA - BO

Figura 28 – Análise Sequencial - PVDM-Averaging

5.2.6 Comparação com outros modelos neurais profundos

Utilizamos nos experimentos um computador com processador Intel Xeon W-2145 16 CPU(s) 3.70 GHz, 64 GB de memória RAM, GPU NVIDIA Quadro P2000 5GB GDDR5. A linguagem adotada foi a Python 3.7.4 com a biblioteca CUDA 10.2 e Tensorflow 1.13.0.

Na tabela abaixo (Tabela 16), o modelo proposto é comparado com o estado da arte de modelos para a classificação e representação semântica de documentos.

Tabela 16 – Comparação com outros modelos neurais profundos

Modelo	**Referência**	**Pontuação (%)**	**Tempo**
Modelo Neural Recozido	Figura 23b	81.56	1.51
Modelo Neural CMA-ES	Figura 12c	77.37	2.51
Modelo Neural BO	Figura 12d	78.58	2.63
Recurrent CNN	(LAI et al., 2015)	82.91	3.37
Very Deep CNN	(CONNEAU et al., 2016)	77.51	4.55
Attention-Based Bidirectional RNN	(ZHOU et al., 2016)	81.77	6.44
Character-level CNN	(ZHANG; ZHAO; LECUN, 2015)	78.93	7.11

Considerando que o modelo desenvolvido, quando comparado aos modelos avaliados neste estudo (Tabela 16), apresenta tempo de execução significativamente menor com pontuação estatisticamente equivalente, concluímos que a sua aplicabilidade é promissora para a representação semântica de documentos em instituições que utilizam um sistema de gestão eletrônica de documentos.

CAPÍTULO 6

Conclusões

Tendo em vista a existência de um grande número de formatos e mídias, a extração de conhecimento de textos não estruturados é um processo desafiador. Extrair contexto semântico de palavras e documentos é fundamental para o processamento de linguagem natural. Esta tese apresentou um modelo neural para a representação contínua de documentos que, associado a um algoritmo clássico de otimização (SA), apresentou resultados promissores no processo de classificação textual, com um tempo de resposta menor em relação a outros modelos neurais e otimizadores.

O modelo desenvolvido pode auxiliar o promotor de Justiça como ferramenta de apoio à decisão, com menor tempo de resposta e melhor visualização de dados multidimensionais, permitindo a distribuição e balanceamento dos processos judiciais por promotoria de Justiça. Possibilita, assim, maior celeridade no atendimento ao cidadão. Os experimentos mostraram que a estratégia PVDM-*Averaging*, é a mais adequada, por preservar a ordem das palavras no documento, mantendo o contexto semântico. O modelo se mostrou escalável em diferentes dimensões, mantendo-se performático e atingindo pontuações superiores a outros modelos.

Como resultado das pesquisas realizadas durante o desenvolvimento desta tese, publicamos um artigo na revista *Engineering Applications of Artificial Intelligence* (MENDONÇA; JUNIOR, 2020), comparando o modelo desenvolvido com outros modelos neurais profundos e otimizadores globais de funções.

6.1 Trabalhos Futuros

O algoritmo *Simulated Annealing (SA)* demonstrou eficiência na otimização de modelos para melhor representação semântica de documentos no espaço vetorial. Essa técnica de otimização pode ser aplicada em outros modelos de incorporação semântica - *Global Vectors for Word Representation (GLOVE)* (PENNINGTON; SOCHER; MANNING, 2014), *Embeddings from Language Models (PETERS et al., 2018)*, *Bidirectional Encoder Representations from Transformers (BERT)* (DEVLIN et al., 2018), *Enhanced Language Representation with Informative Entities (ERNIE)* (ZHANG et al., 2019) - com resultados promissores. Outra análise a ser considerada é a substituição do algoritmo SA por outros algoritmos de otimização - *Adaptive Differential Evolution With Optional External Archive (JADE)* (ZHANG; SANDERSON, 2009), L-SHADE (PIOTROWSKI, 2018), *Multi-objective Bayesian Optimization (MOBOpt)* (GALUZIO et al., 2020) e *Coyote Optimization Algorithm (COA)* (PIEREZAN; COELHO, 2018) - para ajustes dos hiperparâmetros - dimensão vetorial, número de camadas, janela de contexto, entre outros (PATEL; BHATTACHARYYA, 2017).

Referências

ALTINEL, B.; GANIZ, M. C. Semantic text classification: A survey of past and recent advances. *Inf. Process. Manage.*, v. 54, n. 6, p. 1129–1153, 2018. Citado na página 15.

BAYESIAN Optimization (BO). https://github.com/fmfn/BayesianOptimization. Citado na página 27.

BEHERA, S. S.; CHATTOPADHYAY, S. A comparative study of back propagation and simulated annealing algorithms for neural net classifier optimization. *Procedia Engineering*, v. 38, p. 448–455, 2012. ISSN 18777058. Citado na página 33.

BENAVOLI, A.; CORANI, G.; DEMSAR, J.; ZAFFALON, M. Time for a change: A tutorial for comparing multiple classifiers through bayesian analysis. *Journal of Machine Learning Research*, v. 18, 06 2016. Citado 2 vezes nas páginas 31 e 32.

BENGIO, Y. Practical recommendations for gradient-based training of deep architectures. *CoRR*, abs/1206.5533, 2012. Citado na página 26.

BERGSTRA, J.; BARDENET, R.; BENGIO, Y.; KéGL, B. Algorithms for hyper-parameter optimization. In: SHAWE-TAYLOR, J.; ZEMEL, R. S.; BARTLETT, P. L.; PEREIRA, F. C. N.; WEINBERGER, K. Q. (Ed.). *NIPS*. [S.l.: s.n.], 2011. p. 2546–2554. Citado na página 34.

BERGSTRA, J.; BENGIO, Y. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, v. 13, p. 281–305, 2012. Citado na página 28.

BREIMAN, L. Random forests. *Machine Learning*, v. 45, n. 1, p. 5–32, oct 2001. Citado na página 40.

CAMBRIA, E.; WHITE, B. Jumping NLP curves: A review of natural language processing research. *IEEE Computational Intelligence Magazine*, v. 9, n. 2, p. 48–57, 2014. ISSN 1556603X. Citado na página 15.

CARRASCO, J.; GARCíA, S.; RUEDA, M.; DAS, S.; HERRERA, F. Recent trends in the use of statistical tests for comparing swarm and evolutionary computing algorithms: Practical guidelines and a critical review. *Swarm and Evolutionary Computation*, v. 54, p. 100665, 02 2020. Citado 2 vezes nas páginas 31 e 32.

CHANG, C.-C.; LIN, C.-J. LIBSVM: a library for support vector machines. *ACM Transactions on Intelligent Systems and Technology (TIST)*, ACM, v. 2, n. 3, p. 27, 4 2011. Citado na página 40.

CONNEAU, A.; SCHWENK, H.; BARRAULT, L.; LECUN, Y. Very deep convolutional networks for natural language processing. *CoRR*, abs/1606.01781, 2016. Disponível em: <http://arxiv.org/abs/1606.01781>. Citado na página 69.

CORTES, C.; VAPNIK, V. Support vector networks. *Machine Learning*, v. 20, n. 3, p. 273–297, 1995. Citado na página 40.

DEVLIN, J.; CHANG, M.-W.; LEE, K.; TOUTANOVA, K. Bert: Pre-training of deep bidirectional transformers for language understanding. *CoRR*, abs/1810.04805, 2018. Citado 2 vezes nas páginas 20 e 71.

DUA, D.; GRAFF, C. *UCI Machine Learning Repository*. 2017. Disponível em: <http://archive.ics.uci.edu/ml>. Citado na página 33.

FISCHETTI, M.; STRINGHER, M. Embedded hyper-parameter tuning by simulated annealing. *CoRR*, abs/1906.01504, 2019. Citado na página 34.

FRAZIER, P. I. A tutorial on bayesian optimization. *CoRR*, abs/1807.02811, 2018. Citado na página 27.

FRIEDMAN, J. Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, v. 29, 2001. Citado na página 40.

FRIEDMAN, J. H. Stochastic gradient boosting. *Computational Statistics & Data Analysis*, v. 38, n. 4, p. 367–378, fev. 2002. Citado na página 40.

GALUZIO, P. P.; SEGUNDO], E. H. [de V.; COELHO, L. dos S.; MARIANI, V. C. Mobopt — multi-objective bayesian optimization. *SoftwareX*, v. 12, p. 100520, 2020. ISSN 2352-7110. Disponível em: <http://www.sciencedirect.com/science/article/pii/S2352711020300911>. Citado na página 71.

GOLDBERG, Y.; LEVY, O. word2vec explained: deriving mikolov et al.'s negative-sampling word-embedding method. *CoRR*, abs/1402.3722, 2014. Citado 2 vezes nas páginas 16 e 23.

HANSEN, N. The cma evolution strategy: A tutorial. *CoRR*, 2016. Disponível em: <http://arxiv.org/abs/1604.00772>. Citado 2 vezes nas páginas 17 e 27.

HANSEN, N.; OSTERMEIER, A. Completely derandomized self-adaptation in evolution strategies. *Evolutionary Computation*, v. 9, n. 2, p. 159–195, 2001. Citado na página 26.

HARRIS, Z. Distributional structure. *Word*, v. 10, n. 23, p. 146–162, 1954. Citado 2 vezes nas páginas 15 e 21.

HINTON, G.; ROWEIS, S. Stochastic neighbor embedding. *Advances in neural information processing systems*, Citeseer, v. 15, p. 833–840, 2003. Citado na página 31.

HINTON, G. E. Connectionist learning procedures. *Artificial Intelligence*, v. 40, p. 185–234, 1989. Citado na página 40.

HOSMER, D.; LEMESHOW, S. *Applied Logistic Regression*. [S.l.]: John Wiley, SonsInterscience, 1989. Citado na página 40.

HUANG, A. Similarity measures for text document clustering. *NZCSRSC2008*, p. 49–56, 2008. Citado na página 21.

JOHN, G. H.; LANGLEY, P. Estimating continuous distributions in Bayesian classifiers. *11th Conference on Uncertainty in Artificial Intelligence*, p. 338–345, 1995. Citado na página 40.

JONES, D. R.; SCHONLAU, M.; WELCH, W. J. Efficient global optimization of expensive black-box functions. *J. Global Optimization*, v. 13, n. 4, p. 455–492, 1998. Citado na página 27.

KALPIC, D.; HLUPIC, N.; LOVRIC, M. Student's t-tests. In: LOVRIC, M. (Ed.). *International Encyclopedia of Statistical Science*. [S.l.]: Springer, 2011. p. 1559–1563. ISBN 978-3-642-04898-2. Citado na página 31.

KHARI, M.; KUMAR, P. Empirical evaluation of hill climbing algorithm. *Int. J. of Applied Metaheuristic Computing*, v. 8, n. 4, p. 27–40, 2017. Citado na página 26.

KIM, D.; SEO, D.; CHO, S.; KANG, P. Multi-co-training for document classification using various document representations: Tf-idf, lda, and doc2vec. *Inf. Sci.*, v. 477, p. 15–29, 2019. Citado na página 23.

KIM, H. I.; LEE, S. H.; CHO, N. I. An efficient multicategory classifier based on adaboosting. In: WANI, M. A.; MILANOVA, M. G.; KURGAN, L. A.; REFORMAT, M.; HAFEEZ, K. (Ed.). *ICMLA*. [S.l.]: IEEE Computer Society, 2005. p. 5. ISBN 0-7695-2495-8. Citado na página 40.

KIM, H. K.; KIM, H.; CHO, S. Bag-of-concepts: Comprehending document representation through clustering words in distributed representation. *Neurocomputing*, v. 266, p. 336–352, 2017. Citado 2 vezes nas páginas 16 e 23.

KIRKPATRICK, S.; GELATT, C. D.; VECCHI, M. P. Optimization by simulated annealing. *Science*, v. 220, p. 671–680, 1983. Citado 4 vezes nas páginas 16, 17, 19 e 26.

KLEIN, A.; FALKNER, S.; BARTELS, S.; HENNIG, P.; HUTTER, F. Fast bayesian optimization of machine learning hyperparameters on large datasets. *CoRR*, abs/1605.07079, 2016. Citado na página 34.

KRIZHEVSKY, A. Learning multiple layers of features from tiny images. *University of Toronto*, 05 2012. Citado na página 34.

KUSHNER, H. J. A new method of locating the maximum point of an arbitrary multipeak curve in the presence of noise. *Journal of Basic Engineering*, ASME International, v. 86, n. 1, p. 97–106, mar 1964. Disponível em: <https://doi.org/10.1115%2F1.3653121>. Citado 2 vezes nas páginas 17 e 27.

KUSNER, M. J.; SUN, Y.; KOLKIN, N. I.; WEINBERGER, K. Q. From word embeddings to document distances. In: *Proceedings of the 32nd International Conference on Machine Learning (ICML 2015)*. [S.l.: s.n.], 2015. p. 957–966. Citado 2 vezes nas páginas 23 e 24.

LAI, S.; XU, L.; LIU, K.; ZHAO, J. Recurrent convolutional neural networks for text classification. In: BONET, B.; KOENIG, S. (Ed.). *AAAI*. [S.l.: s.n.], 2015. v. 333, p. 2267–2273. Citado na página 69.

LAU, J. H.; BALDWIN, T. An Empirical Evaluation of doc2vec with Practical Insights into Document Embedding Generation. *Proceedings of the 1st Workshop on Representation Learning for NLP*, Proceedings of the 1st Workshop on Representation Learning for NLP, n. 2014, p. 78–86, 2016. Disponível em: <http://aclweb.org/anthology/W16-1609>. Citado na página 16.

LE, Q. V.; MIKOLOV, T. Distributed representations of sentences and documents. *CoRR*, abs/1405.4053, 2014. Citado 5 vezes nas páginas 16, 17, 23, 24 e 58.

LECUN, Y.; BOTTOU, L.; BENGIO, Y.; HAFFNER, P. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, v. 86, n. 11, p. 2278–2324, 1998. ISSN 0018-9219. Citado na página 33.

LI, Y.; WEI, B.; LIU, Y.; YAO, L.; CHEN, H.; YU, J.; ZHU, W. Incorporating knowledge into neural network for text representation. *Expert Syst. Appl.*, v. 96, p. 103–114, 2018. Citado na página 16.

LIGHTWEIGHT Covariance Matrix Adaptation Evolution Strategy (CMA-ES). `https://github.com/CyberAgent/cmaes`. Citado na página 27.

LILLEBERG, J.; ZHU, Y.; ZHANG, Y. Support vector machines and word2vec for text classification with semantic features. In: GE, N.; LU, J.; WANG, Y.; HOWARD, N.; CHEN, P.; TAO, X.; ZHANG, B.; ZADEH, L. A. (Ed.). *ICCI*CC*. [S.l.]: IEEE Computer Society, 2015. p. 136–140. ISBN 978-1-4673-7290-9. Citado na página 16.

LOPEZ, M. M.; KALITA, J. Deep learning applied to nlp. *CoRR*, abs/1703.03091, 2017. Citado na página 15.

LOSHCHILOV, I.; HUTTER, F. Cma-es for hyperparameter optimization of deep neural networks. *CoRR*, abs/1604.07269, 2016. Citado na página 34.

LOSHCHILOV, I.; SCHOENAUER, M.; SEBAG, M. Self-adaptive surrogate-assisted covariance matrix adaptation evolution strategy. *CoRR*, abs/1204.2356, 2012. Citado na página 34.

MAATEN, L. v. d.; HINTON, G. Visualizing Data using t-SNE. *Journal of Machine Learning Research*, v. 9, n. Nov, p. 2579–2605, 2008. ISSN ISSN 1533-7928. Citado na página 41.

MAATEN, L. van der; HINTON, G. E. Visualizing high-dimensional data using t-sne. *Journal of Machine Learning Research*, v. 9, p. 2579–2605, 2008. Citado 2 vezes nas páginas 30 e 41.

MENDONÇA, L. R. C. d.; JUNIOR, G. d. C. Deep neural annealing model for the semantic representation of documents. *Engineering Applications of Artificial Intelligence*, v. 96, 2020. Citado na página 70.

METROPOLIS, N.; ROSENBLUTH, A.; ROSENBLUTH, M.; TELLER, A.; TELLER, E. Equations of state calculations by fast computing machines. *Journal of Chemical Physics*, v. 21, p. 1087–1091, 1953. Citado 2 vezes nas páginas 16 e 26.

MIKOLOV, T.; CHEN, K.; CORRADO, G.; DEAN, J. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013. Citado 3 vezes nas páginas 16, 23 e 36.

MIKOLOV, T.; SUTSKEVER, I.; CHEN, K.; CORRADO, G. S.; DEAN, J. Distributed representations of words and phrases and their compositionality. In: BURGES, C. J. C.; BOTTOU, L.; GHAHRAMANI, Z.; WEINBERGER, K. Q. (Ed.). *NIPS*. [S.l.: s.n.], 2013. p. 3111–3119. Citado 4 vezes nas páginas 16, 23, 24 e 36.

PATEL, K.; BHATTACHARYYA, P. Towards lower bounds on number of dimensions for word embeddings. In: KONDRAK, G.; WATANABE, T. (Ed.). *IJCNLP(2)*. Asian Federation of Natural Language Processing, 2017. p. 31–36. ISBN 978-1-948087-01-8. Disponível em: <http://dblp.uni-trier.de/db/conf/ijcnlp/ijcnlp2017-2.htmlPatelB17>. Citado na página 71.

PENNINGTON, J.; SOCHER, R.; MANNING, C. D. Glove: Global vectors for word representation. In: *EMNLP2014*. [S.l.: s.n.], 2014. v. 14, p. 1532–1543. Citado 3 vezes nas páginas 20, 23 e 71.

PETERS, M. E.; NEUMANN, M.; IYYER, M.; GARDNER, M.; CLARK, C.; LEE, K.; ZETTLEMOYER, L. Deep contextualized word representations. *CoRR*, abs/1802.05365, 2018. Citado 2 vezes nas páginas 20 e 71.

PIEREZAN, J.; COELHO, L. dos S. Coyote optimization algorithm: A new metaheuristic for global optimization problems. In: *CEC*. [S.l.]: IEEE, 2018. p. 1–8. ISBN 978-1-5090-6017-7. Citado na página 71.

PIOTROWSKI, A. P. L-shade optimization algorithms with population-wide inertia. *Inf. Sci.*, v. 468, p. 117–141, 2018. Citado na página 71.

POUYANFAR, S.; CHEN, S.-C. T-lra: Trend-based learning rate annealing for deep neural networks. In: *BigMM*. [S.l.]: IEEE Computer Society, 2017. p. 50–57. ISBN 978-1-5090-6549-3. Citado 2 vezes nas páginas 16 e 34.

QUINLAN, J. R. Induction of decision trees. *Machine Learning*, v. 1, n. 1, p. 81–106, March 1986. Citado na página 40.

RERE, L. R.; FANANY, M. I.; ARYMURTHY, A. M. Simulated annealing algorithm for deep learning. *Procedia Computer Science*, v. 72, p. 137–144, 2015. Citado 2 vezes nas páginas 16 e 33.

RUSSAKOVSKY, O.; DENG, J.; SU, H.; KRAUSE, J.; SATHEESH, S.; MA, S.; HUANG, Z.; KARPATHY, A.; KHOSLA, A.; BERNSTEIN, M. S.; BERG, A. C.; LI, F.-F. Imagenet large scale visual recognition challenge. *CoRR*, abs/1409.0575, 2014. Citado na página 34.

SHAHRIARI, B.; SWERSKY, K.; WANG, Z.; ADAMS, R. P.; FREITAS, N. de. Taking the human out of the loop: A review of bayesian optimization. *Proceedings of the IEEE*, v. 104, n. 1, p. 148–175, 2016. Citado na página 27.

SHIN, S.; LEE, Y.; KIM, M.; PARK, J.; LEE, S.; MIN, K. Deep neural network model with bayesian hyperparameter optimization for prediction of no x at transient conditions in a diesel engine. *Engineering Applications of Artificial Intelligence*, v. 94, p. 103761, 09 2020. Citado na página 34.

SMITH, L. N. Cyclical learning rates for training neural networks. In: *WACV*. [S.l.]: IEEE Computer Society, 2017. p. 464–472. ISBN 978-1-5090-4822-9. Citado 2 vezes nas páginas 16 e 34.

SNOEK, J.; LAROCHELLE, H.; ADAMS, R. P. Practical bayesian optimization of machine learning algorithms. *CoRR*, abs/1206.2944, 2012. Citado na página 34.

SOUCY, P.; MINEAU, G. W. Beyond tfidf weighting for text categorization in the vector space model. In: KAELBLING, L. P.; SAFFIOTTI, A. (Ed.). *IJCAI*. [S.l.]: Professional Book Center, 2005. p. 1130–1135. ISBN 0938075934. Citado na página 22.

STRATIFIED Cross Validation. `https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.StratifiedKFold.html`. Citado na página 29.

SUN, S.; LUO, C.; CHEN, J. A review of natural language processing techniques for opinion mining systems. *Information Fusion*, v. 36, p. 10–25, 2017. Citado na página 15.

TAN, A. hwee. Text mining: The state of the art and the challenges. In: *In Proceedings of the PAKDD 1999 Workshop on Knowledge Disocovery from Advanced Databases*. [S.l.: s.n.], 1999. p. 65–70. Citado na página 15.

TONG, X.; YUAN, B.; LI, B. Model complex control cma-es. *Swarm Evol. Comput.*, v. 50, 2019. Citado na página 34.

TOSCANO-PALMERIN, S.; FRAZIER, P. I. Bayesian optimization with expensive integrands. *CoRR*, abs/1803.08661, 2018. Citado na página 34.

WATANABE, S.; ROUX, J. L. Black box optimization for automatic speech recognition. In: *ICASSP*. [S.l.]: IEEE, 2014. p. 3256–3260. Citado na página 34.

WATTENBERG, M.; VIÉGAS, F.; JOHNSON, I. How to use t-sne effectively. *Distill*, 2016. Disponível em: <http://distill.pub/2016/misread-tsne>. Citado na página 41.

WHITE, B.; CAMBRIA, E. Jumping NLP curves : a review of natural language processing research. *IEEE Computational Intelligence Magazine*, v. 9, p. 2, 2014. Citado na página 15.

YU, K.; JI, L.; ZHANG, X. Kernel nearest neighbor algorithm. *Neural Processing Letters*, v. 15, n. 2, p. 147–156, 2002. Citado na página 40.

ZHANG, J.; SANDERSON, A. C. Jade: Adaptive differential evolution with optional external archive. *IEEE Trans. Evol. Comput.*, v. 13, n. 5, p. 945–958, 2009. Citado na página 71.

ZHANG, T. Solving large scale linear prediction problems using stochastic gradient descent algorithms. In: BRODLEY, C. E. (Ed.). *ICML*. [S.l.]: ACM, 2004. (ACM International Conference Proceeding Series, v. 69), p. 116. Citado na página 40.

ZHANG, X.; ZHAO, J. J.; LECUN, Y. Character-level convolutional networks for text classification. *CoRR*, abs/1509.01626, 2015. Disponível em: <http://arxiv.org/abs/1509.01626>. Citado na página 69.

ZHANG, Z.; HAN, X.; LIU, Z.; JIANG, X.; SUN, M.; LIU, Q. Ernie: Enhanced language representation with informative entities. *CoRR*, abs/1905.07129, 2019. Citado 2 vezes nas páginas 20 e 71.

ZHOU, P.; SHI, W.; TIAN, J.; QI, Z.; LI, B.; HAO, H.; XU, B. Attention-based bidirectional long short-term memory networks for relation classification. In: *ACL (2)*. [S.l.]: The Association for Computer Linguistics, 2016. p. 207—212. ISBN 978-1-945626-01-2. Citado na página 69.