Patrick Marques Ciarelli

Rede Neural Probabilística para a Classificação de Atividades Econômicas

Dissertação apresentada ao Programa de Pós-Graduação em Engenharia Elétrica do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do Grau de Mestre em Engenharia Elétrica, na área de concentração em Automação.

Orientador:
Dr. Evandro Ottoni Teatini Salles

Co-orientador:
Dr. Elias Silva de Oliveira

Vitória – ES

2008

Dados Internacionais de Catalogação-na-publicação (CIP)
(Biblioteca Central da Universidade Federal do Espírito Santo, ES, Brasil)

Ciarelli, Patrick Marques, 1982 -

C566r Rede neural probabilística para a classificação de atividades econômicas / Patrick Marques Ciarelli. - 2008.

82f.: il.

Orientador: Evandro Ottoni Teatini Salles.

Co-Orientador: Elias Silva de Oliveira.

Dissertação (mestrado) - Universidade Federal do Espírito Santo, Centro Tecnológico.

1. Redes neurais (Computação). 2. Reconhecimento de padrões. 3. Textos - Classificação. 4. Inteligência artificial. I. Salles, Evandro Ottoni Teatini. II. Oliveira, Elias Silva de. III. Universidade Federal do Espírito Santo. Centro Tecnológico. IV. Título.

CDU: 621.3

PATRICK MARQUES CIARELLI

REDE NEURAL PROBABILISTICA PARA A CLASSIFICAÇÃO DE ATIVIDADES ECONÔMICAS

Dissertação submetida ao programa de Pós-Graduação em Engenharia Elétrica do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisição parcial para a obtenção do Grau de Mestre em Engenharia Elétrica - Automação.

Aprovada em 22 de fevereiro de 2008.

COMISSÃO EXAMINADORA

Dr. Evandro Ottoni Teatini Salles
Universidade Federal do Espírito Santo
Orientador

Dr. Elias Silva de Oliveira
Universidade Federal do Espírito Santo
Co-orientador

Dr. Felipe Maia Galvão França
Universidade Federal do Rio de Janeiro

Dr. Renato Antônio Krohling
Universidade Federal do Espírito Santo

Dedico esta Dissertacao a minha fam lia,
pelo apoio, compreensao e incentivo
que foram indispensaveis para a conclusao
de mais esta etapa da minha vida.

Agradecimentos

Dedico meus sinceros agradecimentos

– ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) e a Receita Federal do Brasil pelo apoio dado através da bolsa de estudos durante o período de realização desta Dissertação;

– aos professores do PPGEE da UFES, principalmente aos meus orientadores Dr. Evandro Ottoni Teatini Salles e Dr. Elias Silva de Oliveira que sempre foram prestativos e dispostos a ajudar;

– à todos os amigos da UFES, em especial do Cisne e do LCAD, que me ajudaram a realizar mais este passo importante da minha vida.

\Seja você quem for, seja qual for a posição social que você tenha na vida, a mais alta ou a mais baixa, tenha sempre como meta muita força, muita determinação e sempre faça tudo com muito amor e com muita fé em Deus, que um dia você chega lá. De alguma maneira você chega lá."

Ayrton Senna

Sumário

Lista de Tabelas

Lista de Figuras

Resumo

Este trabalho apresenta uma abordagem baseada em Redes Neurais Artificiais para problemas de classificação multi-rotulada. Em particular, foi empregada uma versão modificada da Rede Neural Probabilística para tratar de tais problemas. Em experimentos realizados em várias bases de dados conhecidas na literatura, a Rede Neural Probabilística proposta apresentou um desempenho comparável, e algumas vezes até superior, a outros algoritmos especializados neste tipo de problema.

Como o foco principal deste trabalho foi o estudo de estratégias para classificação automática de texto de atividades econômicas, foram realizados também experimentos utilizando uma base de dados de atividades econômicas. No entanto, diferente das bases de dados utilizadas anteriormente, esta base de dados apresenta um número extenso de categorias e poucas amostras de treino por categoria, o que aumenta o grau de dificuldade deste problema. Nos experimentos realizados foram utilizados a Rede Neural Probabilística proposta, o classificador k-Vizinhos mais Próximos Multi-rotulado, e um Algoritmo Genético para otimização dos parâmetros dos mesmos. Nas métricas utilizadas para avaliação de desempenho, a Rede Neural Probabilística mostrou resultados superiores e comparáveis aos resultados obtidos pelo k-Vizinhos mais Próximos Multi-rotulado, mostrando que a abordagem utilizada neste trabalho é promissora.

Abstract

This work presents an approach based on Artificial Neural Networks for problems of multi-label classification. In particular, was used a modified version of Probabilistic Neural Network to handle such problems. In experiments carried out in various databases known in the literature, the Probabilistic Neural Network proposal presented a performance comparable, and sometimes even superior to other algorithms specialized in this type of problem.

As the main focus of this work was the study of strategies for automatic text classification of economic activities then were also conducted experiments using a database of economic activities. However, unlike of databases used previously, this database shows a huge number of categories and few samples of training by category, which increases the degree of difficulty this problem. In the experiments were used to Probabilistic Neural Network proposal, the classifier Multi-label k-Nearest Neighbor and a Genetic Algorithm for optimization of the parameters. The metrics used to evaluation of performance have shown that the results of Probabilistic Neural Network were superior and comparable to the results obtained by the Multi-label k-Nearest Neighbor, showing that the approach used in this work is promising.

1 Introdução

A capacidade de aprendizado, raciocínio, dedução e reconhecimento de padrões são algumas das características mais fundamentais do ser humano. Devido a essas características o ser humano é capaz de inventar, criar e montar objetos que ajudem-no nas tarefas do dia-a-dia. De forma similar, ele pode modificar o ambiente em que vive para torná-lo menos hostil e mais confortável. Além disso, graças a essas características nos destacamos dos outros animais, bem como elas permitiram que nós, seres humanos, saíssemos da Idade da Pedra para plena era da informatização.

Com o passar do tempo surgiram várias tentativas de reproduzir a inteligência de um ser humano, criando assim uma máquina inteligente ou com inteligência artificial. É curioso notar que o conceito de inteligência artificial varia com o tempo. Na Grécia antiga, por exemplo, um distribuidor de água era considerado **inteligente** por simplesmente fornecer água em função do peso da moeda que era colocado nele. Mais recentemente, no início do século passado, um sistema que era realimentado e mantinha a saída estável era considerado **inteligente** [1].

Por volta da década de 40 surgiu, inspirado no cérebro humano, um novo campo da inteligência artificial conhecido como Redes Neurais Artificiais (RNA). Apesar de até hoje não ser conhecido por completo o funcionamento do cérebro, esta nova área da ciência é baseada no neurônio biológico e na estrutura das redes neurais que compõem o cérebro humano. Além da sua base na neurociência, as Redes Neurais Artificiais possuem raízes em outras áreas como Matemática, Estatística, Física, Ciência da Computação e Engenharia [2].

Ao longo das últimas décadas as Redes Neurais Artificiais têm mostrado ser um conjunto de técnicas com potencial para resolver uma variedade de problemas, tais como modelagem, predição, reconhecimento de padrões, controle de sistemas, processamento de sinais e análise de séries temporais [2–10]. A capacidade das Redes Neurais de adaptação ao problema através de exemplos, de realizar mapeamento não linear entre a entrada e

saída, de generalizar informações ruidosas e/ou incompletas, garantindo às Redes Neurais um certo grau de robustez, e de obter resultados tão bons ou melhores do que outras técnicas particulares, têm despertado o interesse de muitos pesquisadores [2].

No entanto, apesar da versatilidade, a abordagem clássica de Redes Neurais apresentam algumas limitações. O uso da técnica de descida de gradiente (que depende do cálculo do gradiente) para o treinamento de alguns tipos de redes pode ser computacionalmente custosa e sujeita a cair em mínimos locais da superfície de erro. Além disso, a definição do número de neurônios das camadas ocultas não é uma tarefa trivial, sendo que um número grande sacrifica a capacidade de generalização da rede e um número pequeno interfere na capacidade de aprendizado [11–13]. Como se não bastasse, se for necessário apresentar novas amostras de treinamento, caso típico em aplicações **on-line**, ou incluir novas classes às Redes Neurais é necessário um processo de re-treinamento da rede, que pode acabar sendo um procedimento custoso. Isto indica que em geral as Redes Neurais não possuem uma estrutura flexível para acomodação de novas classes ou amostras [14, 15].

Além disso, com relação a problemas de reconhecimento de padrões, mais especificamente ao de classificação, o treinamento das Redes Neurais pode se tornar muito custoso e também ser muito sensível aos dados usados no treinamento, especialmente quando o problema apresenta muitas classes [16, 17]. Para evitar estas desvantagens uma das abordagens utilizadas é a criação de um conjunto de redes para classificação. Sendo assim, cada rede é responsável por um conjunto de classes ou por uma classe, onde no segundo caso a rede deve ser capaz de prever se a amostra pertence ou não a esta classe [18–20]. Uma outra variante desta técnica acontece quando cada rede usa somente um pequeno conjunto do espaço de entrada para classificação [21]. No entanto, embora esta abordagem torne o processo de treinamento menos custoso, ela pode acabar tornando o problema mais complexo. Um dos fatores que influenciam é que normalmente neste procedimento as redes são treinadas de forma independente, e dessa forma não levam em consideração a correlação entre as classes ou regiões de amostra [22].

No estudo de caso proposto nesta Dissertação este problema se torna ainda mais evidente. Além de se tratar de um problema de classificação de texto, que de forma geral são problemas que apresentam alta dimensionalidade e esparsidade dos dados [23], a classificação de atividades econômicas engloba várias centenas de categorias além de ser uma classificação multi-rotulada. Diferente da usual classificação uni-rotulada, na qual é associada somente uma classe por amostra, na classificação multi-rotulada uma amostra pode ser classificada em uma ou mais classes. Ademais, no problema proposto não existe

uma definição da quantidade de classes a serem associados à amostra, e tão pouco existe um limite máximo da quantidade de classes que podem ser associados à amostra. Estas características somadas tornam o problema mais difícil de ser resolvido [24].

Apesar das desvantagens mencionadas anteriormente das Redes Neurais, existem vários trabalhos que utilizaram Redes Neurais para a classificação, inclusive de texto, cujos resultados obtidos foram considerados animadores pelos autores [17, 25–30]. Contudo, muitos destes trabalhos empregaram Redes Neurais para a classificação uni-rotulada, sendo muito pouco o uso desta abordagem para a classificação multi-rotulada.

A partir dos resultados obtidos pelas Redes Neurais em vários campos de pesquisa e, em especial, para a classificação de texto, este trabalho propõe o uso de tal abordagem para a classificação de atividades econômicas. No entanto, para contornar parte das dificuldades apresentadas anteriormente, este trabalho propõe a abordagem de uma classificação multi-rotulada baseada na Rede Neural Probabilística [31]. Esta Rede Neural possui as vantagens de possuir um treinamento rápido, que não depende de cálculos numéricos como o gradiente, além de possuir poucos parâmetros para serem selecionados. Além disso, graças ao seu treinamento rápido existe uma facilidade maior em criar uma única rede para todas as classes, ao invés de ser feito um conjunto de pequenas redes como citado anteriomente, além de tornar mais plausível uma aplicação **on-line** desta rede. Entretanto, a Rede Neural Probabilística proposta neste trabalho é uma versão modificada da original de forma que ela possa resolver problemas de classificação multi-rotulada, uma vez que a versão original desta rede foi projetada para o caso de classificação uni-rotulada.

O classificador proposto será avaliado através de um conjunto de bases de dados e seu desempenho será comparado ao de outros algoritmos especialmente projetados para tal problema. Entre os algoritmos usados para comparação será utilizado o MLkNN [32], que é uma versão modificada do tradicional classificador k-Vizinhos mais Próximos e que apresentou bons resultados para várias bases de dados multi-rotuladas [32, 33]. Após verificar a eficácia da Rede Neural proposta, será realizada uma bateria de experimentos com a Rede Neural Probabilística e com o MLkNN para a base de dados de atividades econômicas, que é o foco deste trabalho.

Para obter um desempenho melhor na classificação de atividades econômicas será utilizado um Algoritmo Evolucionário, mais precisamente um Algoritmo Genético, para a otimização parcial da Rede Neural Probabilística e do MLkNN. Um dos Algoritmos Evolucionários foi selecionado pois, como mencionado em [12, 13], existem vários motivos para se utilizar tais algoritmos para otimização de soluções de problemas ao invés de

outros algoritmos. Alguns destes motivos são: eles não dependem do cálculo do gradiente, realizam busca global, são robustos às condições iniciais, pouco propensos a caírem em mínimos locais, possuem capacidade de tratar problemas em superfícies de busca grandes, complexas, não diferenciáveis e multimodais, que são os típicos casos encontrados no mundo real.

A seguir será apresentado brevemente algumas características do problema proposto, assim como a importância em se conseguir uma solução prática para o mesmo. Maiores detalhes deste problema serão vistos no Capítulo 2.

1.1 De nicao do Problema

Atualmente, a burocracia para se abrir uma empresa no Brasil demora em média até 152 dias, sendo um dos piores países da América Latina para a abertura de empresas [34]. Parte deste longo tempo de espera é porque o empreendedor precisa passar por uma longa cadeia de processos manuais, às vezes em diferentes momentos nos três níveis do governo: Municipal, Estadual e Federal. O Governo Federal, numa tentativa de reduzir este tempo, estuda uma maneira de criar uma interface única entre os cidadãos e todos os três níveis do governo brasileiro. Para isto se concretizar, um dos problemas que o Governo precisa atacar é a classificação das atividades econômicas das empresas. A descrição da atividade econômica de uma empresa, que deste ponto em diante também será chamado de objeto social, informa os ramos de atividade que a empresa atua. O objeto social pode ser classificado em uma ou mais categorias de um total de mais de 1000 categorias [35]. Por essa tarefa ser realizada de forma manual ela acaba sendo extremamente lenta e sujeita a subjetividade, além de apresentar a falta de mão-de-obra devidamente qualificada para esta tarefa. Um outro agravante da situação é a imensa demanda de objetos a serem classificados: nos últimos 5 anos teve-se, em média por ano, mais de 1 milhão e 300 mil empresas que surgiram ou alteraram seu objeto social [36], necessitando dessa forma de uma classificação ou re-classificação.

De acordo com o nosso conhecimento, devido à quantidade de categorias, este é um problema incomum na literatura [24]. O trabalho encontrado com o maior número de classes foi em [37], onde os autores trabalharam em (ao)7ura

Mesmo se fosse aplicado tal procedimento, ele necessitaria de uma grande quantidade de amostras por categoria para poderem ser aferidas estatísticas mais precisas dos dados [38]. No entanto, para o caso em questão não se tem disponível tal quantidade de dados. Por outro lado, uma abordagem usando Redes Neurais não necessita de tal procedimento, além disso as capacidades de adaptação e generalização das Redes Neurais podem proporcionar relativamente numa boa classificação das atividades econômicas.

1.2 Estrutura da Dissertacao

Esta Dissertação está estruturada da seguinte forma:

- **Capítulo 1: Introdução**

  É descrito o tema desta Dissertação, em linhas gerais. São apresentados o problema, o procedimento a ser utilizado e os objetivos aos quais se propõe este trabalho.

- **Capítulo 2: Classificação Nacional de Atividades Econômicas - CNAE**

  A Classificação Nacional de Atividades Econômicas (CNAE), sua importância, seu contexto histórico, a tabela e como é realizada a classificação dos objetos sociais são apresentados no Capítulo 2.

- **Capítulo 3: Revisão Bibliográfica de Técnicas**

  Neste capítulo é realizado uma revisão de Redes Neurais Artificiais e Algoritmos Evolucionários, com uma ênfase maior para a Rede Neural Probabilística e Algoritmo Genético, respectivamente. Também será descrito sobre o MLkNN e outros classificadores utilizados na seção de experimentos.

- **Capítulo 4: Resultados e Discussões**

  Neste capítulo são apresentadas as bases de dados utilizadas, a preparação das bases de dados, os testes realizados, os resultados obtidos e discussão dos mesmos.

- **Capítulo 5: Conclusões**

  Nesta parte é feita uma análise conclusiva do trabalho realizado e são propostos trabalhos futuros.

2 Classificação Nacional de Atividades Econômicas - CNAE

2.1 Importância da CNAE

Antes de prosseguirmos para uma melhor detalhamento da **Classificação Nacional de Atividades Econômicas** - CNAE, é de importância definirmos primeiro o que é uma **atividade econômica**. Segundo [39, 40] a "atividade econômica é a combinação de recursos: mão-de-obra, capital, matérias primas e serviços, associada a um processo produtivo, que permite a produção de bens ou serviços, num determinado período".

A CNAE é uma classificação das atividades econômicas projetada sob a coordenação do IBGE (Instituto Brasileiro de Geografia e Estatística) usando como referência a **International Standard Industrial Classification** - ISIC. Por sua vez, a ISIC é uma padronização internacional definida pelas Nações Unidas para harmonização da produção e disseminação das estatísticas econômicas no globo [40].

A CNAE é usada com o objetivo de padronizar a identificação das diversas atividades econômicas no Brasil junto às três esferas do poder (Municipal, Estadual e Federal), de forma a contribuir na melhoria da qualidade dos sistemas de informação que auxiliam nas decisões e ações do Estado [40]. Com esta classificação é possível representar estatisticamente o parque produtivo do País e classificar as unidades segundo a sua atividade principal, permitindo fazer uma análise da estrutura de organização da economia e assim obter uma visão geral deste setor no país. Empresas podem também utilizar de informações obtidas da CNAE para investigar o ambiente onde estão inseridas e assim descobrir novas oportunidades e minimizar os riscos [41]. Com esta importante função, fica evidenciado o interesse de uma correta classificação dos objetos sociais para evitar que o Governo obtenha uma informação distorcida da economia e tome decisões inadequadas para o setor econômico, o que pode ocasionar prejuízos aos cofres públicos e afetar de forma direta e indireta a população brasileira.

Uma vez que esta tabela é padronizada no país inteiro e apresenta, até certo grau, uma correspondência com a ISIC, é possível fazer comparações do setor econômico não só entre municípios e estados, mas também com outros países [39, 40].

2.2 Contexto Historico

O período pós-guerra apresentou um forte crescimento econômico-industrial até o final dos anos 60 ou o início dos anos 70. Este período caracterizou-se por uma produção de bens materiais em série e distribuição em massa. Com isso, o comércio se intensificou além das fronteiras dos países. Assim, a partir dos anos 70, começou a surgir no mundo a necessidade de uma padronização das classificações para servir de referência mundial e facilitar o comércio entre os povos [42].

Seguindo essa tendência, no Brasil houve a necessidade de realizar uma padronização dos códigos de atividades econômicas utilizados pelos diversos órgãos da administração tributária e, dessa forma, evitar possíveis confusões e desorganização da informação. Então, na década de 80, foi criado uma **Tabela de Atividades Econômicas** - TAE, que foi uma primeira tentativa de padronizar os códigos no país.

Porém, o processo de padronização só apresentou avanços em nível nacional com a definição da tabela CNAE. No entanto, somente os órgãos federais utilizavam esta tabela, enquanto que os estados e municípios continuavam a trabalhar com suas próprias tabelas para a classificação das atividades econômicas, que tinham sido definidas em momentos diferentes e que apresentavam um maior grau de especificação em relação à tabela CNAE.

Devido a este impasse foi necessário criar uma tabela mais detalhada para satisfazer as necessidades dos estados e municípios. Neste contexto foi definido a CNAE-Fiscal. A CNAE-Fiscal é um detalhamento da CNAE que mantém a sua estrutura e possui mais um nível de desagregação, surgindo deste modo as subclasses, não presentes na tabela do CNAE original. A primeira versão da tabela de códigos da CNAE-Fiscal apresentava um total de 1094 subclasses e entrou em vigor em 25 de junho de 1998. Após uma série de revisões e versões chegou-se a versão 1.1 da tabela. A versão 1.1 da tabela entrou em vigor desde de 1º de abril de 2003 e apresenta 1183 subclasses [39]. Recentemente foi elaborada a versão 2.0 e esta tabela esta em vigor desde de janeiro de 2007 [40]. Devido à recente mudança da tabela de classificação, todas as demais informações são referentes à versão 1.1 da tabela, uma vez que a base de dados obtida é referente a esta classificação.

2.3 A Tabela CNAE

Devido a impossibilidade de representar todas as características de todas as atividades econômicas, foi decidido, pelos membros que elaboraram a tabela, utilizar somente as características mais relevantes das atividades e agrupa-las de acordo com certos critérios:

- Similaridade de funções produtivas;
- Características em comum;
- Finalidade de uso.

Dessa forma, a tabela CNAE foi estruturada em 5 níveis hierárquicos: Seção, Divisão, Grupo, Classe e Subclasse. A versão 1.1 da tabela está dividida em 17 Seções, 59 Divisões, 222 Grupos, 580 Classes e 1183 Subclasses. A divisão desta tabela segue uma estrutura lógica de códigos, onde cada código é formado por 7 dígitos, sendo que os 5 primeiros dígitos definem a Classe da atividade (códigos do CNAE) e os 2 últimos definem as Subclasses. Na Tabela 1 são apresentadas todas as Seções, suas denominações e divisões por Seção da tabela. A tabela pode ser dividida de forma grosseira em atividades de manejo de recursos naturais (Seções A, B e C), atividades de transformação, tratamento, montagem e construção (Seções D, E e F), atividades de compra e venda (Seção G), serviços de uso genérico voltados a empresa e/ou famílias (Seções H, I, J, K, L, M, N e O), serviços domésticos (Seção P) e atividades de organismos internacionais e instituições extraterritoriais (Seção Q) [39].

Para ser possível fazer uma relação entre a situação do país no setor econômico e o resto do mundo, as duas primeiras hierarquias da tabela (Seção e Divisão) apresentam uma certa semelhança com a ISIC. A terceira e quarta categorias no entanto, não apresentam uma correspondência com a ISIC.

Para fins ilustrativos, na Tabela 2 é apresentado um exemplo da hierarquia da Subclasse **cultivo de milho**. Observa-se neste exemplo que para cada nível da hierarquia está associada uma denominação.

SEÇÕES	DENOMINAÇÃO DA SEÇÃO	DIVISÕES
A	Agricultura, pecuária, silvicultura e exploração florestal	01 a 02
B	Pesca	05
C	Indústrias extrativas	10 a 14
D	Indústria de transformação	15 a 37
E	Produção e distribuição de eletricidade, gás e água	40 a 41
F	Construção	45
G	Comércio, reparação de veículos automotores, objetos pessoais e domésticos	50 a 52
H	Alojamento e alimentação	55
I	Transporte, armazenagem e comunicações	60 a 64
J	Intermediação financeira, seguros, previdência complementar e serviços relacionados	65 a 67
K	Atividades imobiliárias, aluguéis e serviços prestados às empresas	70 a 74
L	Administração pública, defesa e seguridade social	75
M	Educação	80
N	Saúde e serviços sociais	85
O	Outros serviços coletivos, sociais e pessoais	90 a 93
P	Serviços domésticos	95
Q	Organismos internacionais e outras instituições extraterritoriais	99

Tabela 1: Seções e denominações da tabela

SEÇAO	DIVISÃO	GRUPO	CLASSE	SUBCLASSE	DENOMINAÇÃO
A					Agricultura, pecuária, silvicultura e exploração florestal
	01				Agricultura, pecuária e serviços relacionados
		011			Produção de lavouras temporárias
			0111-2		Cultivo de cereais para grãos
				0111-2/02	Cultivo de milho

Tabela 2: Exemplo de hierarquia da tabela CNAE

No entanto a denominação apresenta uma informação muito limitada e ineficiente a respeito da categoria. Então, para auxiliar na classificação foram elaboradas notas explicativas para os códigos de classe e subclasse da tabela CNAE. Essas notas explicativas se constituem num instrumento de interpretação da CNAE com o objetivo de definir o conteúdo e a abrangência dessas duas categorias, apontando os casos limites quando necessário e os casos de excessão quando existentes. Porém, ela não é um instrumento que cobre a definição do conteúdo como um todo: ela está ali apenas para esclarecer alguns conceitos que geram dúvida, como indicar atividades que aparentemente não pertencem à categoria, mas que são classificados como tal e, nos casos inversos, atividades que parecem pertencer a categoria mas que não pertencem. Um exemplo dessas notas explicativas é ilustrado na Tabela 3.

CNAE-Fiscal		
Hierarquia		
Seção:	B	PESCA
Divisão:	05	PESCA, AQÜICULTURA E SERVIÇOS RELACIONADOS
Grupo:	051	PESCA, AQÜICULTURA E SERVIÇOS RELACIONADOS
Classe:	0511-8	PESCA E SERVIÇOS RELACIONADOS
Subclasse	0511-8/01	**PESCA DE PEIXES**
Notas Explicativas:		
Esta Subclasse compreende:		
– A pesca de peixes em águas marítimas e em águas continentais		
Esta Subclasse compreende tambem:		
– A preparação e conservação do peixe no próprio barco		
Esta Subclasse nao compreende:		
– A captura de crustáceos e moluscos (0511-8/02)		
– A preparação do peixe (frigorificado, congelado, salgado, seco) e a fabricação de conservas de peixe em estabelecimentos fabris, inclusive em barcos-fabrica (1514-8/00)		
– A preparação de qualquer tipo de farinha de peixe (1514-8/00)		
– A criação e cultivo de peixes (0512-6/01)		

Tabela 3: Notas explicativas da tabela CNAE-Fiscal para a Subclasse **Pesca de Peixe**

2.4 A Classificação dos Objetos Sociais

Atualmente a classificação dos objetos sociais é realizada da seguinte forma: uma empresa preenche um questionário e, entre outras informações, ela fornece a descrição

dos seus vários ramos de atividades e, com base nessa descrição é verificado manualmente em quais subclasses da tabela CNAE a empresa deve ser classificada. A prática indica que raramente um profissional especializado toma parte nesse processo. A inexistência de profissionais devidamente treinados para a realização dessa tarefa leva à distorções que poderão prejudicar em muito as futuras análises estatísticas realizadas pelos órgãos públicos. Outra dificuldade intrínseca ao problema é o grau de subjetividade existente em processo de indexação/classificação dessa natureza [43]. Pessoas diferentes podem tomar decisões distintas sobre os possíveis códigos a serem associados a um certo texto que descreve o objeto social de uma empresa. Acrescenta-se ao conjunto de dificuldades desse problema que um objeto social que descreve as atividades econômicas de uma empresa pode ser classificado em muitas subclasses ao mesmo tempo. Na base de dados que utilizamos para nossos experimentos foi encontrado um caso com até 109 subclasses associadas a um único objeto social e, em média, há cerca de 4 subclasses por objeto social, com um desvio padrão de 5,51. Para melhor detalhes desta base de dados veja a Tabela 6 da Seção . Vale ressaltar, entretanto, que a primeira subclasse é dita ser a subclasse principal em que a empresa se enquadraria, ou seja, é o seu principal ramo de atividade.

A regra geral para se decidir o principal ramo de atividade de uma empresa é baseada na atividade que gera a maior receita de venda para mesma. Sendo assim é possível empresas possuírem o mesmo objeto social mas, no entanto, possuírem subclasses principais diferentes. As áreas de maior interesse para o governo são utilizadas para identificar o principal ramo de atividade também. Normalmente não é possível descobrir com apenas uma iteração com a empresa o seu principal ramo de atividade, necessitando muitas vezes, após uma prévia classificação, obter uma ou mais informações junto ao cliente.

OBJETO SOCIAL:	Serviços de lavagem, lubrificação e polimento de veículos, lanchonete e comércio de bebidas. (BAR)
CLASSIFICAÇÃO:	– Serviços de lavagem, lubrificação e polimento de veículos. (5020-2/03) – Comércio varejista de bebidas. (5224-8/00) – Lanchonete, casas de chá, de sucos e similares. (5522-0/00)

Tabela 4: Exemplo de um objeto social e sua classificação

Na Tabela 4 é ilustrado um exemplo de objeto social e sua classificação. O exemplo de objeto social é típica de um posto de gasolina. Como as atividades de lavagem, lubrificação e polimento de veículos geram a maior receita para o estabelecimento, então esta subclasse é definida como a subclasse principal do objeto social e por isso aparece como primeiro na classificação. A ordem das demais subclasses associadas ao objeto não apresenta

relevância.

3 Métodos de Classificação

3.1 Redes Neurais Artificiais

3.1.1 Introdução

O homem é o ser mais inteligente entre todos os seres vivos existentes sobre a face da Terra. O ser humano é capaz de raciocinar, aprender, interpretar, deduzir entre outras diversas atividades intelectuais. O cérebro é o órgão responsável por todas estas tarefas e ele é composto por bilhões de células conhecidas por neurônios, que são as menores unidades de processamento do cérebro, e cada neurônio está conectado a milhares de outros neurônios.

Os neurônios são divididos em três partes: dendritos, corpo (ou soma) e axônio, como mostrado na Figura 1. Basicamente, os dendritos recebem os impulsos nervosos (informações) de outros neurônios e transporta-os para dentro do corpo da célula. Esses impulsos são processados no corpo e na rede dentrítica da célula e um novo impulso é transmitido para outros neurônios através do axônio. Este e outros axônios de outros neurônios estão conectados aos dendritos de várias outras células. Por sua vez, os axônios destas células estão conectados a vários outros neurônios e estes a outros e assim por diante, formando uma rede chamada de rede neural. O ponto de contato entre o axônio de uma célula e o dentrito de uma outra é chamado de sinapse ou junção sináptica, que é a unidade básica para a construção de circuitos neurais biológicos [44, 45].

Baseado no neurônio biológico e no conhecimento da estrutura das redes neurais, um novo campo da ciência surgiu com o objetivo de tentar reproduzir o funcionamento das redes neurais. Este campo ficou conhecido por Redes Neurais Artificiais e o seu marco inicial foi realizado por McCulloch e Pitts [46] que descreveram o funcionamento de um neurônio artificial que, assim como o neurônio na rede neural biológica, é a unidade básica de processamento das Redes Neurais Artificiais.

Essencialmente um neurônio artificial é composto por várias entradas, uma função de

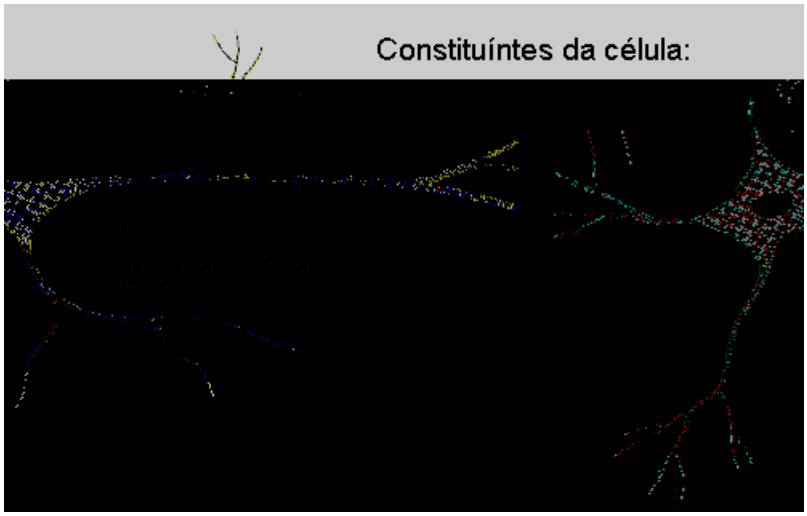


Figura 1: Esquema de um neurônio biológico

transferência (ou função de ativação) e uma saída. Conforme pode ser visto na Figura 2, para cada entrada do neurônio j está associado um valor numérico (w_{j0}, w_{j1}, ..., w_{jn}), conhecido como peso. Quando é apresentada uma informação ao neurônio (x_{j0}, x_{j1}, ..., x_{jn}), cada elemento desta informação é multiplicado pelo peso correspondente a entrada da mesma e o resultado é somado. Desta forma é realizada uma soma ponderada da informação de entrada. O resultado da soma passa por uma função de transferência e a saída do neurônio j será o resultado obtido na função de transferência [2].

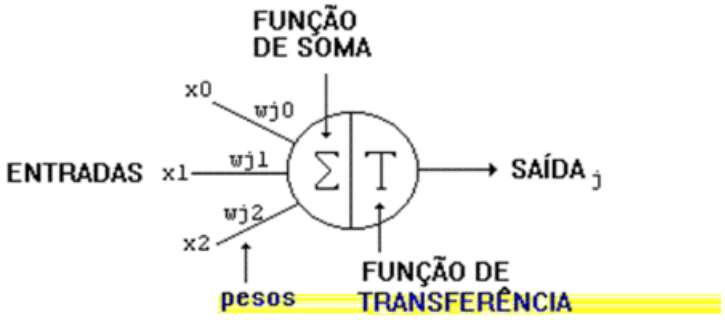


Figura 2: Esquema de um neurônio artificial

A arquitetura (topologia) da rede é que define como os neurônios estão interligados entre si. Este é um parâmetro de suma importância, pois ela define o tipo de problema que a Rede Neural Artificial pode tratar.

As Redes Neurais Artificiais podem ser classificadas em função da quantidade de camadas e os tipos de conexões entre os neurônios. Uma camada é formada por neurônios

que recebem informação ao mesmo instante e uma rede neural pode possuir uma única camada (quando a entrada e saída da rede são separados por um único nó (neurônio)), ou por multíplas camadas (quando existe camadas ocultas, ou seja, que não estão conectadas nem com a saída e nem com a entrada da rede). Estas camadas podem ser completamente conectadas (quando todos os neurônios da camada anterior estão conectados a todos neurônios da camada seguinte) ou parcialmente conectadas (quando os neurônios da camada anterior estão conectadas a apenas alguns neurônios da camada seguinte).

O arranjo das conexões das redes podem ser do tipo:

- Diretas (inglês: **Feedforward**): Não existe realimentação na rede, ou seja, a saída da rede não é utilizada como entrada, tendo portanto um fluxo unidirecional dos dados.
- Recorrentes (inglês: **Feedback**): Apresenta realimentação na rede, ou seja, os dados de saída são utilizados na entrada.

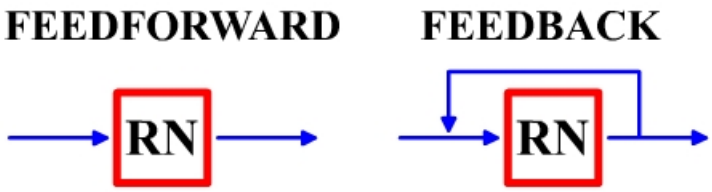


Figura 3: Arranjo de conexões das Redes Neurais.

As Redes Neurais Artificiais não são programadas para executar uma função. Elas "aprendem" a realizar as tarefas a partir de amostras de exemplo e com essas amostras elas vão se adaptando ao problema. Ela

Existem mais outros dois tipos de aprendizagem que são derivados dos dois primeiros [47]:

- Aprendizado por reforço: derivado do aprendizado supervisionado. Neste aprendizado é dito somente à rede se uma saída está certa ou não, diferente do supervisionado que já diz qual é a saída correta.
- Aprendizado por competição: derivado do aprendizado não supervisionado. Neste aprendizado é apresentado um conjunto de entradas e os neurônios disputam entre si os recursos. O vencedor tem seus pesos atualizados e direito a saída ativada. Os demais permanecem desativados.

Alguns tipos de Redes Neurais Artificiais são:

- Perceptron Multi Camada: provavelmente o tipo de Rede Neural mais amplamente utilizada. São Redes Neurais diretas com uma ou mais camadas ocultas, cujos neurônios possuem uma função de transferência não linear, normalmente uma função sigmoidal. São utilizados para problemas de classificação, aproximação de funções, entre outros [2, 48];
- Mapas Auto-Organizáveis: Mapas Auto-Organizáveis são redes cujo aprendizado é não supervisionado, possuem uma única camada de neurônios, cujos neurônios estão conectados a outros neurônios da mesma camada. Usualmente utilizado para agrupamento de dados [2];
- Hopfield: a Rede de Hopfield consiste de um conjunto de neurônios numa arquitetura recorrente, cuja a saída de um neurônio realimenta a entrada dos outros neurônios (com exceção a ele mesmo) com uma unidade de atraso. Uma das aplicações é a recuperação de informação incompleta ou ruidosa [2];
- Redes de Função de Base Radial: são Redes Neurais diretas cuja ativação de cada neurônio da camada oculta é determinada pela distância entre um vetor de entrada e um protótipo de vetor contido no neurônio, na qual a distância normalmente é a Euclidiana. A camada oculta é não linear, enquanto que a camada de saída é linear. Podem ser utilizadas para aproximação de funções e classificação [2, 48].

A seguir será descrito a Rede Neural Probabilística, que é uma rede que faz parte da família das Redes Neurais de Função de Base Radial.

3.1.2 Rede Neural Probabil stica

A Rede Neural Probabilística foi inicialmente proposta por Specht em 1990 [31]. Ela é uma rede neural de arquitetura direta, multi-camadas com mapeamento não linear da entrada para a saída.

A classificação realizada por esta rede é baseada na teoria de decisão Bayesiana e ela realiza a estimação da função de densidade de probabilidade a partir da utilização do método não paramétrico de Parzen [49, 50].

A Rede Neural Probabilística é composta por quatro camadas como mostrado na Figura 4: a camada de entrada, a camada de padrões, a camada de soma e a camada de decisão. Onde X é o vetor de entrada (amostra que se deseja classificar), $W_{i,j}$ é a j-éssima amostra de treino da classe i, c é a quantidade de classes, $p_i(X)$ representa a probabilidade da amostra pertencer à classe i e $C(X)$ é a classe em que X foi classificado. Ambos vetores X e $W_{i,j}$ são de dimensão d.

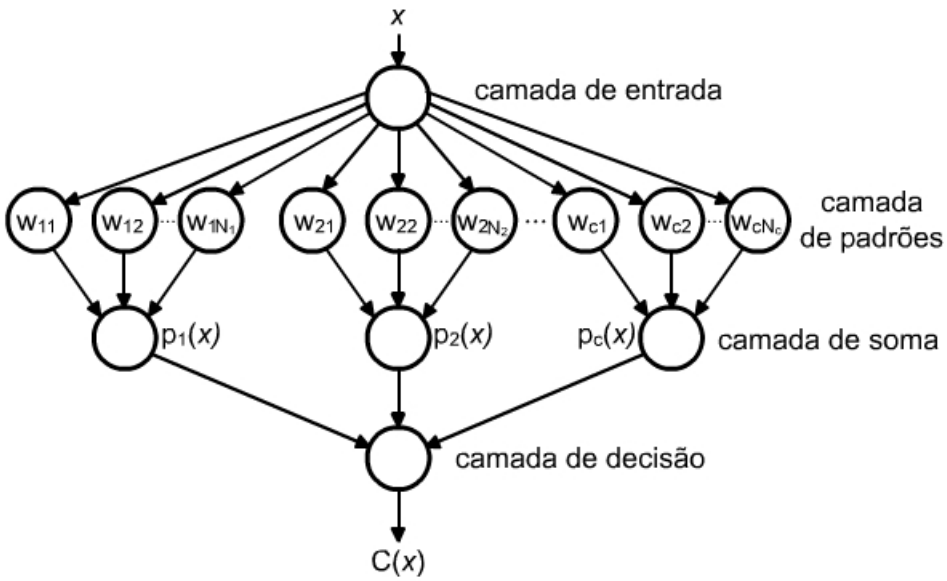


Figura 4: Arquitetura da Rede Neural Probabilística.

Na camada de entrada não é realizado nenhum cálculo, ela simplesmente transmite o vetor de entrada para a próxima camada: a camada de padrões. Nos neurônios da camada de padrões a entrada passa por uma função de transferência:

$$ft_{i,j}(X) = \frac{1}{(2\pi)^{d/2}\sigma_i^d} \exp\left[\frac{-(X - W_{i,j})^T(X - W_{i,j})}{2\sigma_i^2}\right] \tag{3.1}$$

Se tanto X e $W_{i,j}$ forem normalizados de forma que $X^T X = W_{i,j}^T W_{i,j} = 1$, podemos obter a seguinte função de transferência:

$$ft_{i,j}(X) = \frac{1}{(2\pi)^{d/2}\sigma_i^d} \exp\left[\frac{-(X - W_{i,j})^T (X - W_{i,j})}{2\sigma_i^2}\right]$$

$$ft_{i,j}(X) = \frac{1}{(2\pi)^{d/2}\sigma_i^d} \exp\left[\frac{-(\overbrace{X^T X}^{1} + \overbrace{W_{i,j}^T W_{i,j}}^{1} - 2X^T W_{i,j})}{2\sigma_i^2}\right]$$

$$ft_{i,j}(X) = \frac{1}{(2\pi)^{d/2}\sigma_i^d} \exp\left[\frac{-(2 - 2X^T W_{i,j})}{2\sigma_i^2}\right]$$

$$ft_{i,j}(X) = \frac{1}{(2\pi)^{d/2}\sigma_i^d} \exp\left[\frac{X^T W_{i,j} - 1}{\sigma_i^2}\right] \tag{3.2}$$

O σ (sigma) representa o desvio padrão da Gaussiana e σ^2 a variância. Este é o único parâmetro a ser configurado nas funções de transferência. Sigma muito pequeno causa uma aproximação muito ruidosa e pode não generalizar bem, enquanto que para sigma muito grande a Gaussiana é mais suave e causa perda de detalhes [51].

Dependendo da escolha dos sigmas da rede surgem duas versões:

1. Se é utilizado um único sigma σ para toda a rede então esta rede é chamada de Rede Neural Probabilística Básica, sendo usada a Equação 3.1, onde $\sigma_i = \sigma$, para i = 1, 2, ..., c;

2. Se são utilizados sigmas diferentes para cada classe [52] ou sigmas diferentes para cada dimensão dos vetores de amostras [53] então esta rede é chamada de Rede Neural Probabilística Adaptativa. Neste caso se utiliza a Equação 3.3.

$$ft_{i,j}(X) = \frac{1}{(2\pi)^{d/2}|\Sigma|^{1/2}} \exp\left[-\frac{1}{2}(X - W_{i,j})^T \Sigma^{-1} (X - W_{i,j})\right] \tag{3.3}$$

onde $|\Sigma|$ é o determinante da matriz Σ, onde Σ é igual a:

$$\Sigma = \begin{bmatrix} \sigma_1^2 & 0 & \cdots & 0 \\ 0 & \sigma_2^2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \sigma_d^2 \end{bmatrix} \tag{3.4}$$

Em seguida, o resultado obtido na camada de padrões passa para a camada de soma. Nesta camada é calculada a probabilidade da entrada X pertencer a classe i, sendo que cada neurônio desta camada corresponde a uma única classe. Assim, a saída do i-ésimo neurônio desta camada é dada por:

$$p_i(X) = \frac{h_i}{n_i} \sum_{j=1}^{n_i} ft_{i,j}(X) \tag{3.5}$$

Onde n_i é o número de neurônios na camada de padrões da classe i e h_i é a probabilidade a priori da classe i. Se h_i for igual a $\frac{n_i}{\sum n}$ (Onde)Tj /R158 11.9551 Tj 28.2657 0 Tm (n)Tj /R

mais complexas e custosas [49, 56–60].

No entanto, esta rede possui algumas desvantagens. Uma delas é a necessidade de armazenar as amostras de cada classe, o que pode ser um problema para grandes quantidades de amostras de grande dimensões [54]. Uma outra desvantagem é a possibilidade de amostras redundantes, que pode ocasionar não somente um aumento do esforço computacional como também tornar a rede muito sensível para os dados de treinamento e apresentar baixa capacidade de generalização [61].

Embora não tenham sido encontrados muitos trabalhos utilizando a Rede Neural Probabilística para a classificação de textos, em [25] foi relatado o uso de uma variante da Rede Neural Probabilística com este propósito. Neste trabalho, esta Rede Neural foi utilizada para identificar e classificar páginas da Internet relacionadas a área de comércio eletônico. Se a página fosse identificada como de comércio eletônico ela seria classificada em uma de 11 possíveis categorias definidas no artigo, caso contrário seria classificada como não sendo de comércio eletônico, dessa forma o problema apresentava 12 categorias distintas. Nos experimentos realizados foi usada uma rede com 5958 neurônios na camada de padrões e 12 neurônios na camada de soma, onde cada amostra de treinamento era um vetor com 432 dimensões. Apesar da grande quantidade de neurônios usada na rede, os autores relataram que limitações de requerimento de memória e velocidade de execução não aconteceram. Como resultados dos experimentos foram obtidos a correta classificação em uma das 12 categorias e a correta identificação de páginas da área de comércio eletônico com índices de acerto de aproximadamente 80% e 92%, respectivamente. Estes resultados alcançados foram considerados como satisfatórios e suficientes para o problema proposto.

No entanto, em [25] o problema apresentando possui uma quantidade de classes muito pequena em relação ao problema apresentado nesta Dissertação. Além disso, naquele artigo a seleção do valor de sigma foi realizado de maneira exaustiva, na base de tentativa e erro.

Uma vez que o desempenho da Rede Neural Probabilística é fortemente influenciado pelo sigma da função de transferência, neste trabalho será evitado a obtenção do valor do sigma através do método usado em [25] e de métodos matemáticos, como ilustrados em [2]. O motivo principal é porque tais métodos não apresentam garantia nenhuma de fornecer o valor ótimo para o parâmetro. A abordagem utilizada aqui para a seleção do sigma será o uso do Algoritmo Genético (AG), mesma abordagem adotada por [61], cujos resultados obtidos usando esta abordagem apresentaram ser encorajadores. Detalhes sobre Algoritmo Genético serão mencionados na Subseção 3.4.2.

A versão original da Rede Neural Probabilística foi projetada para realizar classificação uni-rotulada, ou seja, associar somente uma única classe por amostra. Entretanto, o problema apresentado neste trabalho é referente a uma classificação multi-rotulada, na qual uma amostra pode ser associada a uma ou mais classes. Então, para permitir que esta rede possa classificar em mais de uma classe, a última camada da rede (a camada de decisão) foi eliminada. Assim sendo, a saída da rede será a saída da camada de soma e, para decidir se a amostra foi classificada ou não em uma classe, será empregado um limiar de saída. Se o neurônio i da camada de soma apresentar uma saída maior que este limiar então a amostra é classificada na classe i, caso contrário a classe i não é associada à amostra. Neste trabalho foi usado um limiar único para todas as classes, no entanto também é possível fazer um limiar adaptado para cada classe. Além disso, como problemas de classificação de textos normalmente são representados por vetores de alta dimensão, foi considerado $d = 2$ na Equação 3.2 para evitar que a função de transferência tenda para zero quando σ for maior do que um.

A seguir serão descritos os classificadores cujos os resultados, obtidos para as bases de dados utilizadas neste trabalho, foram usados como referência para a avaliação da eficácia da Rede Neural Probabilística proposta.

3.2 k Vizinhos Mais Proximos

O método dos k Vizinhos Mais Próximos foi inicialmente proposto por Loftsgaarden e Quesenbery [62] em 1965 como um método não paramétrico para estimação da função de densidade de probabilidade. Uma extensão desse método para o problema de classificação é conhecida como a regra dos k Vizinhos Mais Próximos [63].

A regra dos k Vizinhos Mais Próximos, ou simplesmente kNN em inglês (**k-Nearest Neighbor**), é uma técnica sub-ótima de classificação que guia para uma taxa de erro maior do que o mínimo possível, que é a taxa Bayesiana. No entanto, o classificador Bayesiano requer a função de densidade de probabilidade das classes para ser utilizado, o que de fato muitas vezes não é conhecida. Por não precisar desta informação, o kNN acaba se tornando uma técnica mais simples e fácil de ser utilizada [54].

Seja W um conjunto de n amostras de treinamento de c classes e através do uso de uma métrica sejam selecionadas k amostras de treinamento mais próximas da amostra X que se deseja classificar. Ambas as amostras de treinamento e amostra X são representadas por vetores de dimensão d. Sendo $k_i(X)$ a quantidade de amostras pertencentes à classe

i que estão entre as k amostras mais próximas à X, então pode-se utilizar duas regras de decisão [64]:

- Regra 1: esta regra é a mais tradicional. A amostra X é associada à classe com mais amostras de treinamento entre os k vizinhos mais próximos:

$$C(X) = argmax\,[k_i(X)] \quad i = 1, 2, \ldots, c \tag{3.7}$$

- Regra 2: neste modo são somadas as similaridades das $k_i(X)$ amostras em relação a X, a classe mais próxima é associada à X:

$$C(X) = argmax[f_i(X)] \quad i = 1, 2, \ldots, c \quad onde \tag{3.8}$$

$$f_i(X) = \sum_{j=1}^{k_i(X)} sim(X, Wk_{i,j})$$

  Onde Wk é o subconjunto das k amostras de treino mais próximas da amostra X e $Wk_{i,j}$ é a j-éssima amostra da classe i deste subconjunto. Por sua vez, a função $sim(X, Wk_{i,j})$ é uma métrica de similaridade, logo quanto mais próxima (similar) à amostra X for de $Wk_{i,j}$ maior será o valor retornado.

Uma versão popular deste classificador é quando $k = 1$, conhecida como Vizinho Mais Próximo. Neste caso, a amostra X é classificada na classe associada à amostra de treinamento mais próxima de X.

Existe também algumas versões do kNN que levam em consideração a probabilidade a priori das classes durante a classificação, além da informação disponível das amostras de treinamento [32, 33, 63].

Algumas das métricas possíveis de serem utilizadas para avaliar a proximidade das amostras são a distância Euclidiana (Equação 3.9) [54], provavelmente a mais utilizada, a distância de Tanimoto (Equação 3.10) [54] e a medida de cosseno (Equação 3.11) [65]. Enquanto a distância Euclidiana e a medida de cosseno utilizam diretamente os valores contidos nas amostras X e W_j, onde W_j é a j-ésima amostra de W, a distância de Tanimoto utiliza a quantidade de elementos contidos em cada amostra. Dessa forma, v_X e v_{W_j} são o número de elementos contidos nas amostras X e W_j, respectivamente, e v_{X,W_j} é a quantidade de elementos em comum em ambas as amostras.

Quanto menor for o valor de $dist\_euclid(X, W_j)$ e $dist\_tanim(X, W_j)$, mais próxima a

reduzir o esforço computacional, além da já citada implementação paralela [54, 68].

3.2.1 Classi cador dos k-Vizinhos mais proximos Multi-rotulado

Os k-Vizinhos mais próximos Multi-rotulado (**Multi-label k-Nearest Neighbor** - (MLkNN) em inglês) é um classificador baseado no kNN especialmente projetado para aprendizado multi-rotulado [32].

Suponha que desejamos classificar uma amostra X dentre de c possíveis classes usando para isto o MLkNN. Inicialmente o MLkNN irá identificar os k vizinhos mais próximos de X dentro do conjunto de treinamento W usando a métrica de distância Euclidiana. Seja $T_{i,1}(X)$ o evento em que X é rotulado por i e $T_{i,0}(X)$ o evento em que X não é rotulado por i. Além disso, considere que $E_{i,k_i}(X)$ ($k_i \in \{0,1,...,k\}$) denota o evento em que existe exatamente k_i amostras da classe i entre os k vizinhos mais próximos de X. Então, baseado na regra Bayesiana, a probabilidade de classificar X na classe i é dada por [32]:

$$C_i(X) = argmax_b[P(T_{i,b}(X))P(E_{i,k_i}(X)|T_{i,b}(X))], \quad b = 0,1 \quad i = 1,2,...,c \tag{3.12}$$

Na Equação 3.12 $C_i(X)$ representa a classificação de X na classe i. Se X for classificada em i então $C_i(X)$ recebe 1, caso contrário recebe 0. $P(T_{i,b}(X))$ representa a probabilidade a priori da amostra X ser ou não classificada em i e $P(E_{i,k_i}(X)|T_{i,b}(X))$ representa a probabilidade a posteriori, sendo que estas duas probabilidades podem ser estimadas diretamente do conjunto de treinamento W.

Os cálculos de $P(T_{i,b}(X))$ e $P(E_{i,k_i}(X)|T_{i,b}(X))$ são explicados a seguir. Primeiramente são calculados os valores dos eventos $P(T_{i,b}(X))$ para cada classe i, de acordo com a Equação 3.13.

$$P(T_{i,1}) = \frac{\delta + n_i}{2\delta + n}, \quad P(T_{i,0}) = 1 - P(T_{i,1}) \quad i = 1,2,...,c \tag{3.13}$$

Considere n_i o número de amostras de treinamento classificadas na classe i, n o número total de amostras de treinamento e δ é um parâmetro de suavização da probabilidade.

Após isto, é calculado para cada amostra W_j do conjunto de treinamento os k vizinhos mais próximos dela, e são calculados para cada classe i e amostra W_j o valor de k_i. Se W_j é rotulada na classe i então é somado um a $L_i(k_i)$, senão é somado um a $\overline{L_i(k_i)}$. $L_i(k_i)$

e $\overline{L_i(k_i)}$ contam quantas amostras de treinamento associadas à classe i e não associadas à classe i, respectivamente, incluem, entre os k vizinhos mais próximos, exatamente k_i amostras classificadas na classe i. Com estes passos efetivados, os valores dos eventos $P(E_{i,k_i}(X)|T_{i,b}(X))$ são calculados para cada classe i usando a Equação 3.14.

$$P(E_{i,t}|T_{i,1}) = \frac{\delta + L_i(t)}{\delta(k+1) + \sum_{o=0}^{k} L_i(o)} \quad P(E_{i,t}|T_{i,0}) = \frac{\delta + \overline{L_i(t)}}{\delta(k+1) + \sum_{o=0}^{k} \overline{L_i(o)}} \tag{3.14}$$

$$t = 0, 1, ..., k \quad i = 1, 2, ..., c$$

Os parâmetros que precisam ser selecionados para o MLkNN são o número de vizinhos k e a suavização δ. Como pode ser visto pelas Equações 3.13 e 3.14 o valor de δ altera ligeiramente as probablidades a priori ($P(T_{i,b}(X))$) e posteriori ($P(E_{i,k_i}(X)|T_{i,b}(X))$). Para $\delta = 1$ é alcançada a suavização Laplaciana [32].

3.3 Outros Classi cadores

Nesta Dissertação foram realizadas comparações de desempenho dos classificadores mencionados anteriormente com outros classificadores. No entanto os resultados destes classificadores foram obtidos a partir de [32], portanto os algoritmos dos mesmos não foram utilizados neste trabalho. Deste modo, será dada somente uma breve descrição de cada um destes classificadores sem entrar em detalhes.

3.3.1 ADTBoost.MH

Derivado dos algoritmos ADTboost e do AdaBoost.MH, o ADTBoost.MH é um algoritmo de decisão em árvore [69]. A idéia central deste algoritmo consiste na combinação de um conjunto de regras do tipo apresentadas na Tabela 3.3.1 que serão usadas para a classificação das amostras.

Nesta Tabela, Cb é um conjunto de condições básicas necessárias, Cs é um conjunto de condições secundárias, A e B são vetores de pesos de dimensões c formados por números reais e 0 é um vetor de zeros de dimensão c, na qual c corresponde ao número de classes.

Quando uma amostra é classificada neste algoritmo, são testadas as condições básicas e condições secundárias. Para cada condição, atendida ou não, cada classe recebe um

```
se Cb então
    se Cs então
        A
    senão
        B
    fim
senão
    0
fim
```

Tabela 5: Regra genérica do ADTBoost.MH.

valor. Depois de serem analisadas todas as condições, os valores obtidos por cada classe são somados. Se o valor da soma for positivo, a classe é associada à amostra com uma taxa de confiança igual ao módulo da soma. Caso o resultado da soma seja negativo, a classe não é associada à amostra com uma taxa de confiança igual ao módulo da soma.

O parâmetro a ser informado para este algoritmo é o número de **rounds** e, na fase de treinamento, a cada **round** o algoritmo pode selecionar e adicionar uma nova regra, além de ajustar os valores dos pesos para melhorar o desempenho da classificação. Logo, se for usado um grande número de **rounds**, a quantidade de regras pode crescer muito [69].

3.3.2 BoosTexter

O BoosTexter é um algoritmo projetado especialmente para a tarefa de classificação de texto multi-rotulado. Este algoritmo é embutido por quatro outros algoritmos: três versões do AdaBoost.MH e uma versão do AdaBoost.MR [70].

Semelhante ao ADTBoost.MH, citado anteriormente, o BoosTexter é um algoritmo baseado em regras. No entanto, a árvore de decisão deste algoritmo é formada por somente um nível, sendo diferente portanto do ADTBoost.MH, que pode possuir vários níveis na árvore de decisão. A não ser por esta diferença, as regras possuem um formato semelhante ao apresentado na Tabela 3.3.1.

Assim como o ADTBoost.MH, o parâmetro necessário de ser informado para este algoritmo é o número de **rounds** e, a cada **round**, o algoritmo atualiza os pesos das regras além de procurar uma regra que possa melhorar o desempenho do mesmo.

Um dos problemas deste algoritmo é que a busca por uma boa regra pode consumir muito tempo se o conjunto de treinamento for grande [70].

3.3.3 Rank-SVM

O Rank-SVM é um classificador projetado para classificação multi-rotulada baseado nas Máquinas de Suporte Vetorial, conhecidas também como SVM em inglês **Support Vector Machines** [71]. Em ambos os classificadores são otimizados hiperplanos de separação entre as classes e a partir dos mesmos é realizada a classificação.

No entanto, existe uma diferença com relação a abordagem de classificação. No SVM a abordagem clássica é a binária, na qual o algoritmo classifica ou não a amostra X na classe i dependendo de qual lado do hiperplano de i a amostra se encontra. Para uma explicação melhor, consideremos um hiperplano linear (no caso uma reta) dado por $a_iX + b_i$, onde a_i e b_i são vetores de mesma dimensão de X. Se $a_iX + b_i > 0$ então a amostra X é classificada na classe i, caso contrário a amostra não é classificada em i.

Já no Rank-SVM a abordagem utilizada é baseada no **ranking** das amostras. Assim sendo, para o caso anterior, é calculado $r_i = a_iX + b_i$, onde r_i é o valor do **rank** da classe i. Para realizar a classificação é utlizado um valor de limiar. Classes que possuem um **rank** maior do que o limiar são associadas à amostra X.

Para a classificação de algum problema de separação não linear é aplicada uma função chamada **kernel**, que transforma a dimensão do espaço do problema para um espaço de alta dimensão, onde o mesmo pode se comportar como se possuísse uma separação linear.

3.4 Algoritmos Evolucionarios

3.4.1 Introducao

Ao longo de milhões de anos as formas de vida existentes sobre a face da Terra alteraram muito através do processo de evolução. Neste processo os seres vivos tiveram que passar por profundas mudanças, sejam elas físicas ou comportamentais, para poder se adaptarem, sobreviverem e perpetuarem a espécie diante de condições ambientais adversas [72]. Darwin foi o primeiro a explicar o processo de evolução por seleção natural através de mecanismos como capacidade de sobrevivência (adaptabilidade ao meio ambiente, vigor, competição entre outros organismos, etc) e capacidade de reprodução (tamanho da prole, tempo de gestação, etc) [73].

Inspirado na teoria de Darwin de evolução por seleção natural foram desenvolvidas, há algumas décadas, um grupo de métodos estocásticos empregados para otimização con-

hecido como Algoritmos Evolucionários.

Algoritmos Evolucionários trabalham com uma população de indivíduos, que representam soluções para um problema, aplicando o princípio de sobrevivência do melhor adaptado para assim produzir indivíduos melhores. A cada geração uma nova população é gerada na qual um processo de evolução guia os indivíduos desta para uma adaptação melhor ao problema [74]. Sendo assim, os Algoritmos Evolucionários se diferenciam de outros métodos, tais como **Hill-Climbing** [75] e **Simulated Annealing** [76], por utilizarem uma estratégia de busca da otimização baseada numa população formada por soluções com potencial para resolver um problema, ao invés de utilizar somente uma solução [77].

Basicamente os Algoritmos Evolucionários funcionam da seguinte maneira: inicialmente é gerada uma população aleatória de indivíduos, onde cada indivíduo representa uma possível solução para o problema. Uma função objetivo é empregada para medir o desempenho de cada indivíduo em resolver o problema. Em seguida é realizado um processo de seleção dos indivíduos com melhor desempenho para fazerem parte de uma nova população. Estes indivíduos passam por transformações unitárias (mutação), que criam novos indivíduos a partir de uma pequena mudança do indivíduo original, e transformações de mais alta ordem (cruzamento), que criam novos indivíduos através da combinação de dois ou mais indivíduos da população original. Estes indivíduos formam uma nova população que é avaliada e o procedimento continua até que ocorra a convergência dos resultados ou até que um outro critério de parada seja satisfeito. Desta forma é esperado que a melhor solução encontrada seja próxima da solução ótima [77–79].

Uma estrutura de um Algoritmo Evolucionário é ilustrada na Figura 5 [12].

1. Gera inicialmente uma população aleatória $G(0)$, e associa $i = 0$;
2. REPETIR:
   (a) Avalia cada indivíduo na população;
   (b) Seleciona os indivíduos da população $G(i)$ baseado no desempenho deles;
   (c) Submete estes indivíduos a transformações e produz uma nova população $G(i+1)$;
   (d) $i = i + 1$;
3. REPETIR até critério de parada seja satisfeito.

Figura 5: Estrutura de um Algoritmo Evolucionário.

Existem quatro principais algoritmos evolucionários:

- Programação Genética: usada para a busca do programa com melhor desempenho para resolver um determinado problema, algo como "programação automática". A representação dos indivíduos é realizada em forma de árvore [80, 81];
- Programação Evolucionária: apresenta ser muito eficiente para otimização de funções de valores reais contínuos. Apesar de utilizar os operadores de mutação e de seleção ela não utiliza o operador de cruzamento [82, 83];
- Estratégias Evolucionárias: usada para otimização de funções de valores reais. Ela incorpora os próprios parâmetros dele nos indivíduos a serem avaliados, dessa forma ele passa por um processo de auto aprendizado [79, 84];
- Algoritmo Genético: provavelmente a mais difundida técnica evolucionária. Geralmente aplicado para vários tipos de problemas de otimização. Embora utilize os operadores de cruzamento e mutação, além de seleção, inicialmente ele foi prosposto para utilizar somente o operador de cruzamento [78, 85].

Algoritmos Evolucionários são frequentemente usados para encontrar boas soluções em problemas complexos de otimização cuja a solução ótima é custosa ou inatingível e se pode abrir mão de uma solução ótima em prol de uma solução próxima da ótima em tempo hábil [86]. Algoritmos Evolucionários são menos sujeitos a cair em mínimos locais e apresentam ser mais robustos do que muitos outros algoritmos de busca [12]. O uso de Algoritmos Evolucionários em Redes Neurais Artificiais têm alcançado bons resultados conforme relatado em [4, 12, 13, 18, 80, 87]. Por outro lado, sob certas circunstâncias, eles podem falhar em conseguir bons resultados em problemas cujo número de iterações necessárias para resolvê-los é uma função exponencial da dimensão do espaço de busca [86], além do fato que definir uma função de avaliação dos indivíduos nem sempre costuma ser uma tarefa trivial.

3.4.2 Algoritmo Genetico

Provavelmente o Algoritmo Genético é a técnica de otimização mais difundida da família de Algoritmos Evolucionários. Apresentado inicialmente por Holland [88] em 1962, o Algoritmo Genético utilizava somente os operadores de seleção e cruzamento. No entanto logo foi incorporado o operador de mutação e, com o surgimento de mais pesquisadores interessados na técnica, ela ganhou ao longo do tempo tantas versões e modificações que não ficou mais claro a distinção entre Algoritmo Genético e outros Algoritmos Evolucionários [79].

Assim como mencionado a respeito dos Algoritmos Evolucionários, o Algoritmo Genético é formado por uma população de indivíduos que representam soluções e que tendem a se aperfeiçoar nas futuras gerações.

O uso de Algoritmo Genético requer a especificação de seis procedimentos fundamentais [89]:

1. Representação da solução: antes de resolver qualquer problema, o Algoritmo Genético tipicamente representa cada indivíduo da população como um vetor, onde cada vetor é formado por elementos de um certo tipo de alfabeto. Um alfabeto pode consistir de dígitos binários, números reais, inteiros, símbolos, etc [89]. A representação clássica é realizada por vetores preenchidos por zeros e uns. Este esquema de representação discreta é ligeiramente mais próxima do modelo natural da cadeia de DNA do que a maioria das técnicas de Algoritmos Evolucionários [79]. No entanto, Michalewicz em [78] realizou vários experimentos comparando a representação binária e valores reais e chegou a conclusão que a representação por valores reais é computacionalmente mais rápido além de obter uma maior precisão no resultado;

2. Inicialização: é necessário a geração de uma população inicial para que o Algoritmo Genético possa ser usado e assim começar o procedimento de otimização. O método mais comumente empregado é a geração aleatória dos indivíduos da população com os valores dentro dos limites do espaço de busca previamente definidos. No entanto, se for conhecido por antecedência uma solução boa do problema, ela pode ser incorporada à população inicial ou como parte dela [89];

3. Função objetivo: a função objetivo é a responsável por avaliar o desempenho de cada indivíduo da população. Funções objetivos de várias formas podem ser usadas em Algoritmo Genético, assim como os critérios de otimização do tipo minimização ou maximização, dadas as modificações necessárias;

4. Operador de seleção: logo que os indivíduos são avaliados, eles passam por um processo de seleção para formarem uma nova população. Uma seleção baseada em probabilidade é realizada, na qual indivíduos que apresentaram maior desempenho têm mais changes de serem selecionados. Existem vários métodos de seleção:

    - Roleta russa, **ranking** linear e **ranking** geométrico: estes métodos de seleção associam uma probabilidade do indíviduo ser selecionado baseado no valor do desempenho dele. O método da roleta russa [78] é o mais comumente usado e foi o primeiro método de seleção criado [89];

- **Ranking**: o método do **ranking** associa uma probabilidade ao indivíduo de acordo com a posição do desempenho dele quando o desempenho de todos os indivíduos são ordenados [89];
- Torneio: neste método não é associado probabilidade aos indivíduos. Ele realiza uma seleção de um conjunto de indivíduos e aqueles que apresentaram melhor desempenho são escolhidos para fazer parte da nova população [89].

5. Operadores genéticos: os operadores genéticos são os que garantem o mecanismo básico de busca do Algoritmo Genético, eles são aplicados sobre os indivíduos selecionados para gerar novas soluções:

   - Mutação: mutação é um operador de segundo plano no Algoritmo Genético. Mutação altera somente um indivíduo, mudando aleatoriamente um elemento dele para produzir uma nova solução. Nas versões de representação binária das soluções ele trabalha invertendo os elementos do vetor, e sua probabilidade de ocorrer é muito pequena [79]. A utilidade da mutação surge para restaurar a diversidade de soluções do Algoritmo Genético quando este converge de forma prematura para um ótimo local (mínimo ou máximo) [90];
   - Cruzamento: este é o operador de maior ênfase no Algoritmo Genético. Normalmente a probabilidade de ocorrência dele é relativamente alta. Este operador tem a função de recombinar segmentos de diferentes indivíduos para gerar um novo indivíduo, e assim uma nova solução [79].

6. Critério de parada: o Algoritmo Genético percorrerá geração após geração até alcançar um critério de parada. Alguns dos critérios de parada podem ser: quando é atingido um número máximo de gerações, quando ocorre convergência das soluções e não há melhora significativa nas soluções de uma geração para outra, quando o Algoritmo Genético alcança uma solução considerável aceitável. Estes e outros critérios podem também serem utilizados juntos.

A estrutura do Algoritmo Genético é bem similar ao apresentado na Figura 5.

Algoritmo Genético tem sido uma técnica robusta de otimização aplicada em problemas complicados de serem resolvidos. Algumas das diferentes áreas em que ela foi aplicada com sucesso são: reconhecimento de padrões, vida artficial, aplicações biológicas, projeto de motores, entre outros [91–93]

4 Resultados Experimentais

Neste capítulo serão expostos os experimentos realizados e os resultados alcançados. Para uma melhor exposição e organização da metodologia aplicada sobre os experimentos, este capítulo foi dividido nas seguintes seções:

- Seção 4.1: as bases de dados utilizadas nos experimentos e o método de pré-processamento das mesmas estão informadas nesta seção. Uma ênfase maior foi dada na base de dados de atividades econômicas, que é o estudo de caso apresentado;
- Seção 4.2: nesta seção são mencionadas as métricas utilizadas para avaliação do desempenho dos classificadores;
- Seção 4.3: os experimentos realizados, seus resultados e discussões foram feitos nesta seção.

4.1 Bases de Dados

Nos experimentos foram utilizados 12 bases de dados diferentes: 11 bases de dados da **Internet** em inglês do domínio **Yahoo** obtidas com os autores de [32] e uma base de dados de atividades econômicas, nomeada aqui de base CNAE, obtida junto com a Prefeitura de Vitória.

As bases de dados do **Yahoo** já estavam no formato de matriz numérica, no entanto, segundo [32], cada base de dados passou por um processo de seleção de termos baseado no número de documentos que contém um termo específico. Quanto em mais documentos aparecesse um termo maior seria a frequência do mesmo. Dessa forma, foram selecionados 2% dos termos com a mais alta frequência nos documentos. Depois da seleção dos termos, cada documento na base de dados foi representado na forma de um vetor, onde cada dimensão z do vetor corresponde ao número de vezes que o termo z apareceu no

documento. Se o termo z não apareceu no documento, então é associado zero à dimensão z. No entanto, em [32] não é comentado se foram usadas **stop words** para retirar palavras com pouco valor semântico [65], como artigo, preposições e pronomes, ou se foi realizado algum processamento de linguagem natural para redução das palavras à forma canônica ou extração do radical das palavras [94]. Cada base de dados é formada por 2000 documentos de treino, 3000 documentos de teste e todas são do tipo multi-rotuladas.

A base de dados CNAE é composta por 3264 descrições em texto livre de atividades de empresas brasileiras e mais uma descrição para cada uma das categorias presentes na base de dados, totalizando 764 descrições de categorias.

Para representar estas descrições em forma de matriz numérica, e assim poderem ser processadas, foram seguidos alguns passos. No primeiro passo as descrições passaram por um processo de retirada de **stop words**, números, palavras com uma única letra, caracteres sem valor semântico, como símbolos de pontuação, e foram retirados os acentos das palavras. A seguir, foram tratados casos triviais de plural e gênero, reduzindo as flexões das palavras para um único termo. Após isto, foram considerados somente os termos presentes nas descrições das 764 categorias, uma vez que a princípio elas serviriam como treino para os classificadores. Desta forma, foram selecionados 1001 termos para a representação das descrições. Com os termos selecionados, a representação de cada descrição foi semelhante ao realizado para as bases do **Yahoo**: cada descrição foi representada em forma de vetor, onde cada dimensão z do vetor corresponde ao número de vezes que o termo z apareceu na descrição.

A Tabela 6 mostra algumas estatísticas tanto das bases de dados do **Yahoo** como também da base de dados CNAE. As informações a respeito da base CNAE foram desmembradas nos cinco níveis de hierarquia da tabela CNAE: Seções, Divisões, Grupos, Classes e Subclasses. No entanto, a não ser que seja informado o contrário, toda vez que for referenciado a base CNAE será em relação as Subclasses. Além disto, para evitar confusões com a hierarquia Classe da base de dados CNAE, o termo "classe" será evitado e em seu lugar será utilizado a palavra "categoria".

Chamaremos agora a atenção para algumas destas estatísticas:

1. A quantidade de categorias presentes na base de dados CNAE é muitas vezes superior a quantidade de categorias do **Yahoo**, sendo que a base de dados do **Yahoo** com mais categorias é a base **Science** com 40 categorias;

Base de Dados	*Categorias*	*Termos*	*Conjunto*	*QA*	*PMC*	*MC*	*NMC*	*VC*	*PCR*
CNAE									
Seção	*15*	*1001*	*Treino*	*764*	*0,00%*	*1,00*	*1*	*0,00*	*20,00%*
			Teste	*3264*	*39,77%*	*1,62*	*9*	*0,90*	*33,33%*
Divisão	*56*	*1001*	*Treino*	*764*	*0,00%*	*1,00*	*1*	*0,00*	*53,57%*
			Teste	*3264*	*46,29%*	*1,86*	*13*	*1,65*	*57,14%*
Grupos	*167*	*1001*	*Treino*	*764*	*0,00%*	*1,00*	*1*	*0,00*	*81,44%*
			Teste	*3264*	*62,53%*	*2,63*	*23*	*4,77*	*65,27%*
Classes	*364*	*1001*	*Treino*	*764*	*0,00%*	*1,00*	*1*	*0,00*	*97,25%*
			Teste	*3264*	*69,88%*	*3,42*	*35*	*11,58*	*74,73%*
Subclasses	***764***	***1001***	***Treino***	***764***	***0,00%***	***1,00***	***1***	***0,00***	***100,00%***
			Teste	***3264***	***74,48%***	***4,27***	***109***	***30,37***	***85,21%***
Yahoo									
Arts	*26*	*462*	*Treino*	*2000*	*44,50%*	*1,63*	*11*	*0,78*	*19,23%*
			Teste	*3000*	*43,63%*	*1,64*	*14*	*0,92*	*19,23%*
Business	*30*	*438*	*Treino*	*2000*	*42,20%*	*1,59*	*10*	*0,71*	*50,00%*
			Teste	*3000*	*41,93%*	*1,59*	*12*	*0,72*	*43,33%*
Computers	*33*	*681*	*Treino*	*2000*	*29,60%*	*1,49*	*17*	*1,18*	*39,39%*
			Teste	*3000*	*31,27%*	*1,52*	*17*	*1,10*	*36,36%*
Education	*33*	*550*	*Treino*	*2000*	*33,50%*	*1,47*	*7*	*0,58*	*57,58%*
			Teste	*3000*	*33,73%*	*1,46*	*6*	*0,57*	*57,58%*
Entertainment	*21*	*640*	*Treino*	*2000*	*29,30%*	*1,43*	*9*	*0,87*	*28,57%*
			Teste	*3000*	*28,20%*	*1,42*	*17*	*0,98*	*33,33%*
Health	*32*	*612*	*Treino*	*2000*	*48,05%*	*1,67*	*7*	*0,73*	*53,13%*
			Teste	*3000*	*47,20%*	*1,66*	*13*	*0,81*	*53,13%*
Recreation	*22*	*606*	*Treino*	*2000*	*30,20%*	*1,41*	*13*	*0,66*	*18,18%*
			Teste	*3000*	*31,20%*	*1,43*	*17*	*0,75*	*18,18%*
Reference	*33*	*793*	*Treino*	*2000*	*13,75%*	*1,16*	*5*	*0,18*	*51,52%*
			Teste	*3000*	*14,60%*	*1,18*	*12*	*0,29*	*54,55%*
Science	*40*	*743*	*Treino*	*2000*	*34,85%*	*1,49*	*7*	*0,62*	*35,00%*
			Teste	*3000*	*30,57%*	*1,43*	*9*	*0,57*	*40,00%*
Social	*39*	*1047*	*Treino*	*2000*	*20,95%*	*1,27*	*9*	*0,41*	*56,41%*
			Teste	*3000*	*22,83%*	*1,29*	*10*	*0,38*	*58,97%*
Society	*27*	*636*	*Treino*	*2000*	*41,90%*	*1,71*	*13*	*1,45*	*25,93%*
			Teste	*3000*	*39,97%*	*1,68*	*16*	*1,55*	*22,22%*
(média dos dados)	*30,55*	*655,27*	*Treino*	*2000*	*33,53%*	*1,48*	*9,82*	*0,74*	*39,54%*

2. Foram encontradas na base CNAE descrições classificadas em mais de 40 categorias, inclusive foi encontrada uma associada a 109 categorias. Por sua vez, nas bases de dados do **Yahoo** foram encontrados documentos associados a um número máximo de apenas 17 categorias;

3. A variância e a média de categorias associadas às descrições da base de dados CNAE é maior do que de qualquer base do **Yahoo**;

4. A porcentagem de amostras classificadas a mais de uma categoria do conjunto de dados de teste do CNAE é maior do que de qualquer base do **Yahoo**;

5. Na base CNAE existe uma porcentagem grande de categorias raras, ou seja, que possuem menos do que 1% de amostras na base da dados, 4. 1encouconjun

de documendo que qu

vetor C_j a partir do vetor P_j. Categorias com probabilidades acima ou igual ao ponto de corte pc são associadas à amostra j, e esta classificação é armazenada em C_j. Quando uma amostra j é classificada na categoria i então $C_j(i) = 1$, caso contrário $C_j(i) = 0$, o mesmo é aplicável para o vetor Y_j. Tanto os vetores Y_j, P_j e C_j são de dimensão c, onde cada dimensão está associada a uma categoria. Além disso, os valores de P_j podem variar entre 0 e 1. Desta forma, podemos definir as métricas usadas:

- **Perda de Hamming** (inglês: **Hamming Loss**): avalia quantas vezes uma amostra rotulada foi mal classificada, isto é, quando uma amostra é classificada numa categoria errada assim como quando não é classificada numa categoria certa. Logo, quanto menor for o valor de p_ham melhor será o desempenho do classificador, sendo que o desempenho será perfeito quando $p\_ham = 0$.

$$p\_ham = \frac{1}{m}\sum_{j=1}^{m}\frac{q_j}{c}, \quad onde \quad q_j = \sum_{i=1}^{c} xor(Y_j(i), C_j(i)) \tag{4.1}$$

  Onde a função xor realiza uma operação lógica binária, se são fornecidos dois valores iguais ele retorna zero, caso contrário retorna um.

  Para o caso uni-rótulo esta métrica se reduz a $\frac{2}{c}$ vezes a métrica de erro de classificação usual.

  Para se utilizar esta métrica é necessário estabeler um ponto de corte para evitar que o classificador possa classificar a amostra em todas as classes existentes.

  Para uma amostra j classificada, a menor **perda de hamming** possível de ser obtida é dada por:

$$\underset{min}{p\_ham_j} = min\left\{\frac{s_j - 2u_j + F_j(u_j)}{c}\right\} \quad u_j = 0, 1, \ldots, s_j \quad onde \tag{4.2}$$

$$s_j = \sum_{t=1}^{c} Y_j(t), \quad F_j = ordenar(R_j)$$
$$R_j = \{pos(P_j, i) \quad | \quad Y_j(i) = 1\}, \quad i = 1, 2, \ldots, c$$

  Onde $F(0) = \emptyset$, a função $ordenar$ ordena de forma crescente os valores do vetor R, min retorna o menor valor do vetor e pos retorna a posição da categoria i no vetor P_j ordenado de forma decrescente. O valor de pos da categoria com a maior probabilidade é igual a um.

- **Um Erro** (inglês: **One Error**): avalia quantas vezes a categoria retornada pelo classificador com a maior probabilidade de ser associada à amostra não pertence ao conjunto de categorias corretas da amostra. Portanto, quanto menor for o valor de *um_erro* melhor o desempenho do classificador, sendo que o desempenho será perfeito quando $um\_erro = 0$.

$$um\_erro = \frac{1}{m}\sum_{j=1}^{m} erro_j, \quad onde \quad erro_j = \begin{cases} 0, & se \quad Y_j(argmax(P_j)) = 1 \\ 1, & se \quad Y_j(argmax(P_j)) = 0 \end{cases} \tag{4.3}$$

  Onde a função *argmax* retorna a categoria com a maior probabilidade.

  Para um problema uni-rótulo esta métrica é idêntica a métrica de erro de classificação.

- **Cobertura** (inglês: **Coverage**): esta métrica avalia até qual ponto é necessário alcançar em uma lista de categorias ordenadas pelas probabilidades até cobrir todas as categorias que pertencem a amostra. Assim sendo, quanto menor for o valor de *cobertura* melhor será o desempenho do classificador. O desempenho será perfeito para uma amostra quando o *cobertura* da amostra for igual ao número de categorias associadas à amostra menos um.

$$cobertura = \frac{1}{m}\sum_{j=1}^{m}(max(R_j)) - 1, \quad onde \tag{4.4}$$

$$R_j = \{pos(P_j, i) \quad | \quad Y_j(i) = 1\} \quad i = 1, 2, \ldots, c$$

  Onde a função *max* retorna o valor máximo do vetor R_j e, como dito anteriormente, a função *pos* retorna a posição da categotia i no vetor P_j ordenado de forma decrescente. O valor de *pos* da categoria com a maior probabilidade é igual a um.

- **Perda de Ordenacao** (inglês: **Ranking Loss**): avalia a fração de categorias que são rersamente ordenadas para a amostra. Quanto menor for o valor de *p_ord* melhor o desempenho do classificador. O desempenho será perfeito quando $p\_ord = 0$.

$$p\_ord = \frac{1}{m}\sum_{j=1}^{m}\frac{\sum_{u=1}^{s_j}(R_j(u) - u)}{s_j * (c - s_j)}, \quad onde \tag{4.5}$$

$$s_j = \sum_{t=1}^{c} Y_j(t), \quad R_j = \{pos(P_j, i) \quad | \quad Y_j(i) = 1\}, \quad i = 1, 2, \ldots, c$$

- **Precisao Media** (inglês: **Average Precision**): avalia uma fração das categorias ordenadas pela probabilidade até cobrir todas as categorias corretas da amostra. Dessa forma, quanto maior for o valor de *prec_med* melhor o desempenho do classificador. O desempenho será perfeito quando $prec\_med = 1$.

$$prec\_med = \frac{1}{m}\sum_{j=1}^{m}\frac{1}{s_j}\sum_{u=1}^{s_j}\frac{u}{F_j(u)}, \quad onde \tag{4.6}$$

$$F_j = ordenar(R_j), \quad s_j = \sum_{t=1}^{c} Y_j(t)$$
$$R_j = \{pos(P_j, i) \quad | \quad Y_j(i) = 1\}, \quad i = 1, 2, \ldots, c$$

- **Categoria Principal**: esta métrica foi criada para o caso especial da classificação de atividades econômicas, por isto ela estará presente somente nos testes realizados com a base de dados de atividades econômicas. Como descrito no Capítulo 2, dentre as categorias associadas a uma determinada atividade econômica, existe uma categoria que apresenta uma maior relevância para os estudos estatísticos e esta categoria é conhecida como categoria principal. A finalidade desta métrica é avaliar o quanto é necessário descer uma lista de categorias ordenadas pelas probalidades até alcançar a categoria principal. Logo, quanto menor for o valor de *cat_princ* melhor será o desempenho do classificador. O desempenho será perfeito quando $cat\_princ = 0$.

$$cat\_princ = \frac{1}{m}\sum_{j=1}^{m}(pos(P_j, cp_j)) - 1 \tag{4.7}$$

Onde cp_j contém a categoria principal da amostra j.

Para uma melhor compreensão das métricas, um exemplo fictício é ilustrado na Tabela 7. O exemplo proposto é composto por uma amostra podendo ser classificada em até 5 categorias.

Dado os valores calculados na Tabela 7 podemos calcular o valor das métricas:

- **perda de hamming**:

$p\_ham = \frac{q}{c} = \frac{2}{5} = 0,4$
$p\_ham_{min} = min\left\{\frac{s-2u+F(u)}{c}\right\}, \quad u = 0, 1, \ldots, s$
$s = \sum_{t=1}^{c} Y(t) = 2$
$p\_ham_{min} = min[\frac{2-2\times0+0}{5}, \frac{2-2\times1+1}{5}, \frac{2-2\times2+4}{5}]$

EXEMPLO FICTÍCIO DE CLASSIFICAÇÃO DE UMA AMOSTRA	
Quantidade de Amostras	$m = 1$
Quantidade de Classes	$c = 5$
Correta Classificação	$Y = [1\ 0\ 0\ 1\ 0]$
Classe Principal	$cp = 4$
Probabilidades do Classificador	$P = [0{,}3\ 0{,}4\ 0{,}6\ 0{,}8\ 0]$
Ponto de Corte	$pc = 0{,}5$
Classificação do Classificador	$C = [0\ 0\ 1\ 1\ 0]$

Valores Calculados
$q = \sum_{i=1}^{c} xor(Y(i), C(i)) = 1 + 0 + 1 + 0 + 0 = 2$
$argmax(P) = 4$
$R = \{pos(P, i) \mid Y(i) = 1, \forall\ i \in \aleph \mid i = 1, 2, \ldots, c\} = [4\ 1]$
$max(R) = 4$
$F = ordenar(R) = [1\ 4]$

Tabela 7: Exemplo fictício de classificação de uma amostra.

$p\_ham_{min} = min[\frac{2}{5}, \frac{1}{5}, \frac{2}{5}]$
$p\_ham_{min} = \frac{1}{5} = 0,2$

ponto de corte ideal: $0,6 < pc \leq 0,8$

- **um erro**:

$$um\_erro = erro = \begin{cases} 0, & se \quad Y(argmax(P)) = 1 \\ 1, & se \quad Y(argmax(P)) = 0 \end{cases}$$

$Y(argmax(P)) = Y(4) = 1, \quad logo \quad erro = 0 \quad e \quad um\_erro = 0$

- **cobertura**:

$cobertura = max(R) - 1 = 3$

- **perda de ordenacao**:

$p\_ord = \frac{\sum_{u=1}^{s}(R(u)-u)}{s*(c-s)}$
$s = \sum_{t=1}^{c} Y(t) = 2$
$p\_ord = \frac{(R(1)-1)+(R(2)-2)}{2*(5-2)}$
$p\_ord = \frac{(4-1)+(1-2)}{6} = \frac{2}{6} = 0,33$

- **precisao media**:

$prec\_med = \frac{1}{s}\sum_{u=1}^{s}\frac{u}{F(u)}$

$$s = \sum_{t=1}^{c} Y(t) = 2$$
$$prec\_med = \frac{1}{s} \sum_{u=1}^{s} \frac{u}{F(u)} = \frac{1}{2} \sum_{u=1}^{2} \frac{u}{F(u)}$$
$$prec\_med = \frac{1}{2} * (\frac{1}{1} + \frac{2}{4}) = \frac{1}{2} * \frac{6}{4} = 0,75$$

- **categoria principal**:

  $cat\_princ = pos(P, cp) - 1 = 0$

Entendemos que, das métricas apresentadas aqui, a métrica de **perda de hamming** é a que apresenta maior relevância. Essa relevância acontece porque esta é a única métrica apresentada que compara, para cada amostra, as categorias que o classificador automático associou com as que o ser humano informou. Embora as outras métricas realizam medidas de desempenho em relação ao ordenamento das categorias, não há garantia nenhuma que tais categorias sejam associadas às amostras. No entanto a **perda de hamming** depende do ponto de corte utilizado, e a seleção da mesma pode não ser uma tarefa muito fácil. Porém as demais métricas podem auxiliar na percepção de que a **perda de hamming** obtida é um bom valor ou não. Além destas métricas existem outras métricas para problemas de classificação multi-rotulada que trabalham no mesmo nível que a **perda de hamming** [64, 65, 67]. Porém elas não serão utilizadas aqui, uma vez que parte dos resultados foram extraídos de [32], que usou as métricas anteriormente citadas para avaliação dos desempenhos dos classificadores.

4.3 Resultados e Discussoes

As 11 bases de dados do domínio **Yahoo** foram utilizadas para avaliar o desempenho da Rede Neural Probabilística (RNP) proposta neste trabalho frente aos classificadores MLkNN, BoosTexter, ADTBoost.MH e Rank-SVM, que são classificadores projetados especificamente para o problema de classificação multi-rotulada. A partir da avaliação dos resultados obtidos, serão selecionados os classificadores que farão a classificação automática da base CNAE.

Para a base de dados do **Yahoo** foram usadas todas as métricas apresentadas na Seção 4.2, com exceção da métrica **categoria principal**, e os resultados dos classificadores foram citados diretamente de [32], com exceção dos resultados para a RNP. Os parâmetros utilizados em [32] para o MLkNN, BoosTexter, ADTBoost.MH e Rank-SVM e o parâmetro selecionado para a RNP foram:

- MLkNN: número de vizinhos igual a 10 e suavização (δ) igual a 1;
- BoosTexter: número de **rounds** igual a 500;
- ADTBoost.MH: número de **rounds** igual a 50;
- Rank-SVM: **kernels** polinomiais com 8 graus;
- RNP: variância (σ^2) igual a 0,1.

Em [32] não foi utilizado nenhum processo de busca exaustiva para otimização dos classificadores, sendo que os parâmetros do MLkNN foram obtidos a partir de uma base de dados de imagem, o do BoosTexter e ADTBoost.MH foram selecionados valores que, segundo os autores de [32], não alterariam significantemente o desempenho dos classificadores para as bases de dados utilizadas, e para o Rank-SVM foram utilizados os parâmetros que alcançaram o melhor resultado em [22].

Para tornar numa comparação justa com as outras técnicas, para a RNP foi somente selecionada a ordem de grandeza da variância. Para isto foi utilizada a base de dados **Arts** do **Yahoo** e testado os valores de variância igual 10, 1 e 0,1. O ponto de corte utilizado para a métrica **perda de hamming** foi de 0,5, mesmo valor utilizado para o MLkNN, portanto este parâmetro não foi otimizado para a RNP.

Os resultados das métricas para as bases de dados do **Yahoo**, exceto para a base de dados **Arts**, estão ilustrados nas Figuras 6 a 10. Cada um dos gráficos representa uma métrica, e em cada gráfico os termos "**Business**", "**Computers**", "**Education**", "**Entertainment**", "**Health**", "**Recreation**", "**Reference**", "**Science**", "**Social**" e "**Society**" são referentes as bases de dados do **Yahoo** com os mesmos nomes. O termo "Média" é o valor médio da métrica obtido por cada classificador para todas as bases de dados. As barras estão representando, da esquerda para direita, os classificadores MLkNN (vermelho), Boostexter (verde), ADTBoost.MH (azul), Rank-SVM (amarelo) e RNP (preto).

A Figura 6 mostra os resultados obtidos para a métrica **perda de hamming** e pode ser observado que não houve grande discrepância entre os classificadores, com exce(para)Tj 26.5187 0 T

para as bases de dados que apresentaram mais categorias. Por exemplo, bases de dados como **Reference**, **Science** e **Social**, que apresentam um número maior de categorias, os classificadores apresentaram valores de **perda de hamming** menor do que para bases de dados como **Entertainment**, **Recreation** e **Society**, que possuem um número menor de categorias.

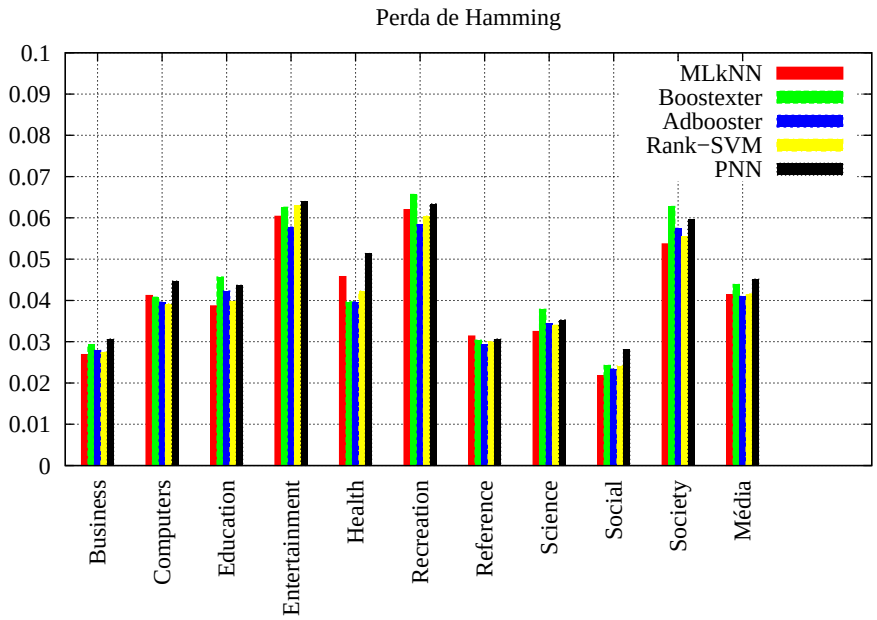


Figura 6: Resultado experimental de cada uma das base de dados do **Yahoo** em termos da **perda de hamming**.

Os resultados obtidos pelos classificadores para a métrica de **um erro** estão ilustrados na Figura 7. Neste caso é possível observar que o Rank-SVM apresentou o melhor desempenho em uma boa parte das bases de dados e mantendo-se sempre próximo do melhor **um erro** das demais bases. Dessa forma ele apresentou o melhor desempenho na média. Por outro lado, o MLkNN apresentou um resultado muito ruim para a base de dados **Recreation**, aproximadamente 0,7 de **um erro**. Nesta métrica os classificadores usados apresentaram um resultado surpreendente para a base **Business**, pois os mesmos obtiveram um **um erro** médio quase 3 vezes melhor que o segundo melhor resultado.

Se por um lado o Rank-SVM foi o melhor na média em **um erro**, por outro ele foi o pior em absoluto nas métricas **cobertura** e **perda de ordenacao** apresentadas nas Figuras 8 e 9, respectivamente. O Rank-SVM apresentou o pior resultado em todas as bases de dados testadas, chegando a ter, em algumas bases de dados, uma **cobertura** e uma **perda de ordenacao** cerca de duas vezes maior do que o melhor resultado obtido. Em ambas as


0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0

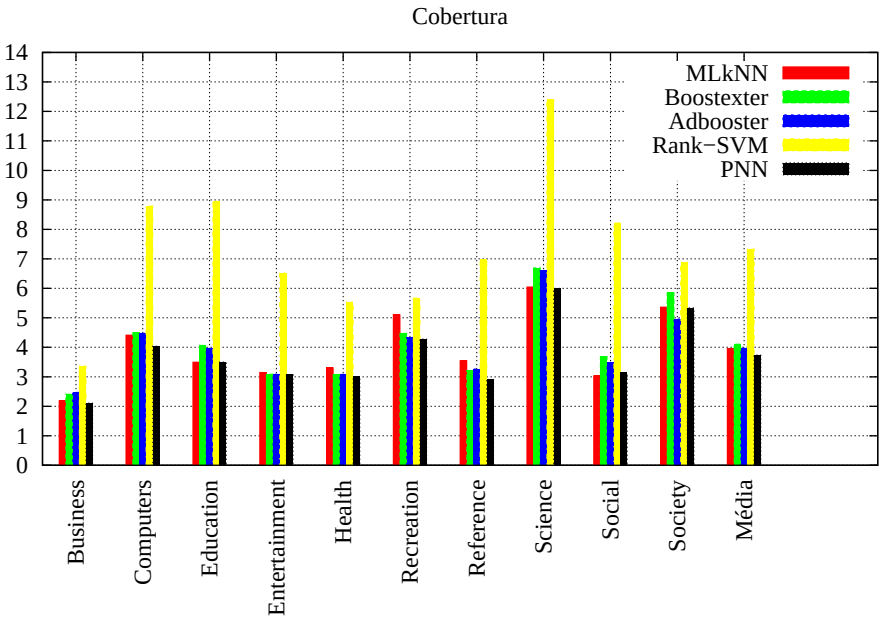


Figura 8: Resultado experimental de cada uma das base de dados do **Yahoo** em termos da **cobertura**.

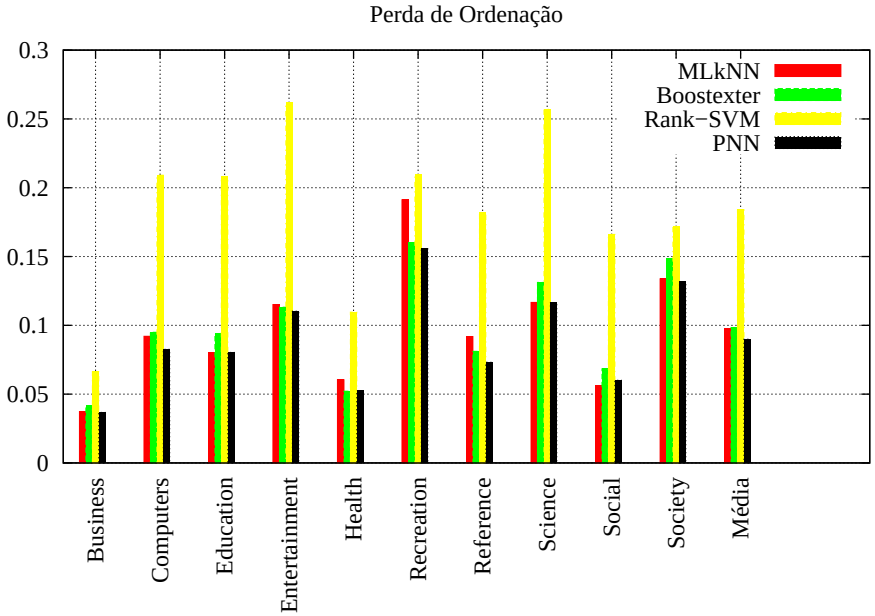


Figura 9: Resultado experimental de cada uma das base de dados do **Yahoo** em termos da **perda de ordenacao**.

enquanto que para as bases **Recreation** e **Science** foram obtidos resultados não muito conclusivos.

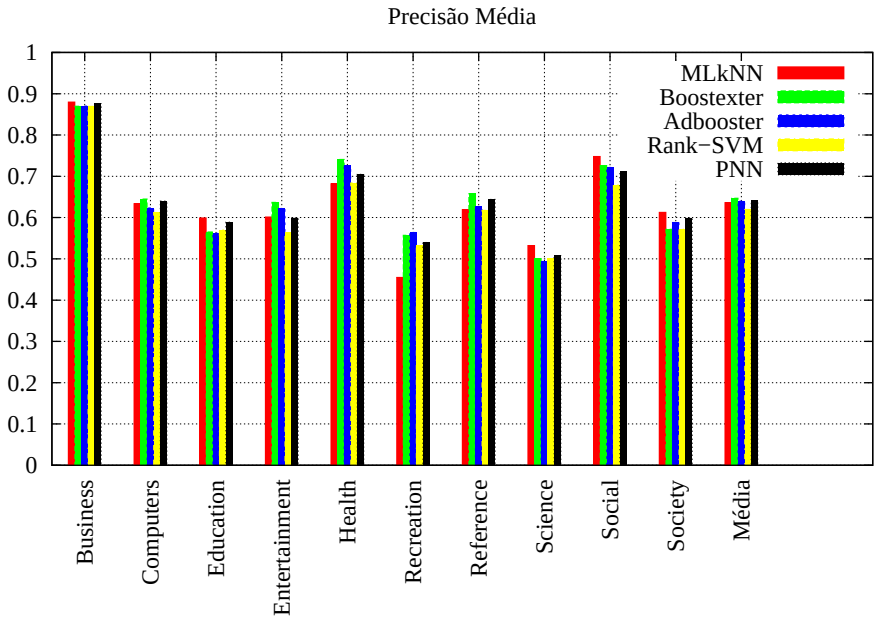


Figura 10: Resultado experimental de cada uma das base de dados do **Yahoo** em termos da **precisao media**.

Uma outra análise é quanto ao desempenho dos classificadores em relação às bases de dados. Por exemplo, o MLkNN obteve, de uma forma geral, melhores resultados que os demais classificadores nas bases **Business**, **Education** e **Social** e piores desempenhos nas bases **Health**, **Recreation** e **Reference**. O Boostexter obteve melhores resultados para a base **Health** e piores para as bases **Education** e **Society**. O ADTBoost.MH apresentou bom desempenho para a base **Recreation** e um desempenho não muito bom para a base **Science**. O Rank-SVM não apresentou grande desempenho para nenhuma base, além disso ele obteve resultados piores para as bases **Entertainment** e **Social**. Por último, a RNP apresentou um resultado equilibrado para todas as bases. O desempenho médio de cada classificador para cada base de dados é informado na Tabela 8. A faixa de valores está entre 1 e 5 e representa a média dos desempenhos dos classificadores para cada base, portanto quanto mais próximo de 1 for o valor melhor o desempenho.

Para realizar uma avaliação mais clara dos classificadores foram adotados dois critérios [32]. O primeiro critério é o de criar uma ordem parcial "$\succ$" que avalie o desempenho entre dois classificadores para cada métrica. Dessa forma, se o classificador A_1 tem um desempenho melhor que A_2 para uma determinada métrica, então temos $A_1 \succ A_2$. Para realizar esta avaliação estatística foi realizado o **t-test** bicaudal e emparelhado com um nível de significância $p < 0{,}05$. O valor de p se extende por uma faixa entre 0 e 1, e quanto

Desempenho dos Classificadores por Base de Dados					
	Classificadores				
Base de Dados	MLkNN	BoosTexter	ADTBoost.MH	Rank-SVM	RNP
Business	1,4	3,4	3,8	3,6	2,6
Computers	2,8	2,6	3,5	3,2	2,6
Education	1,2	4,0	4,0	3,0	2,6
Entertainment	3,2	2,0	2,0	4,0	3,2
Health	4,2	1,2	2,3	3,6	3,0
Recreation	4,0	2,8	1,3	3,6	2,6
Reference	3,8	2,0	3,0	3,4	2,4
Science	1,6	3,8	4,0	3,0	2,4
Social	1,0	3,0	3,0	4,0	3,6
Society	1,8	4,2	2,8	3,2	2,4

Tabela 8: Desempenho de cada classificador para cada base de dados do **Yahoo**.

MÉTRICAS	*CLASSIFICADORES*
	A_1 – *MLkNN;* A_2 – *BoosTexter;* A_3 – *ADTBoost.MH;* A_4 – *Rank-SVM;* A_5 – *RNP*
Perda de Hamming	$A_1 \succ A_5$, $A_3 \succ A_2$, $A_4 \succ A_2$, $A_3 \succ A_5$, $A_4 \succ A_5$
Um Erro	$A_4 \succ A_1$, $A_2 \succ A_3$, $A_2 \succ A_5$, $A_4 \succ A_3$, $A_4 \succ A_5$
Cobertura	$A_1 \succ A_4$, $A_5 \succ A_1$, $A_2 \succ A_4$, $A_5 \succ A_2$, $A_3 \succ A_4$, $A_5 \succ A_3$, $A_5 \succ A_4$
Perda de Ordenação	$A_1 \succ A_4$, $A_2 \succ A_4$, $A_5 \succ A_2$, $A_5 \succ A_4$
Precisão Média	$A_2 \succ A_4$, $A_3 \succ A_4$, $A_5 \succ A_4$
ORDEM TOTAL	*RNP(2) > {MLkNN(1),ADTBoost.MH(1),BoosTexter(1)} > Rank-SVM(-5)*

Tabela 9: Desempenho relativo dos classificadores para a base de dados **Yahoo**.

menor for o valor de p menor a probabilidade que a diferença entre as médias de duas amostras seja mera coincidência [95]. O algoritmo usado para calcular o **t-test** foi obtido em [96].

No entanto, o critério apresentado mede somente o desempenho relativo entre dois classificadores para uma determinada métrica. Para obter o desempenho do classificador como um todo, é aplicado um segundo critério baseado em recompensas e punições. Por exemplo, para o caso de $A_1 \succ A_2$ o classificador A_1 é recompensado com +1 e o classificador A_2 é punido com -1. No final, são somadas as recompensas e punições de cada classificador e aqueles que apresentarem a maior soma serão considerados os melhores classificadores. Um sinal "$>$" será usado para definir a ordem total dos classificadores. Assim, para o caso $A_1 > A_2$ significa que o classificador A_1 apresentou melhor desempenho que o classificador A_2. Os resultados obtidos utilizando os critérios de avaliação estão ilustrados na Tabela 9.

A ordem dos classificadores mostrada na Tabela 9 indica que os classificadores RNP, MLkNN, ADTBoost.MH e BoosTexter são competitivos, com a RNP apresentando um de-

sempenho ligeiramente superior. Logo, o classificador proposto neste trabalho apresentou, nas bases de dados do **Yahoo**, desempenho comparável e até ligeiramente supeior a outros algoritmos especialmente projetados para o problema de classificação multi-rotulada.

Além da RNP, será escolhido o MLkNN para a classificação da base CNAE pois, apesar de empatar com ADTBoost.MH e BoosTexter, ele apresentou resultados superiores a estes em outras bases de dados [32, 33].

Agora iremos apresentar os resultados dos experimentos realizados com a base de dados CNAE, descrita na Seção 4.1, usando todas as métricas apresentadas na Seção 4.2.

Para os experimentos com a base de dados CNAE, a base foi dividida em 5 partes:

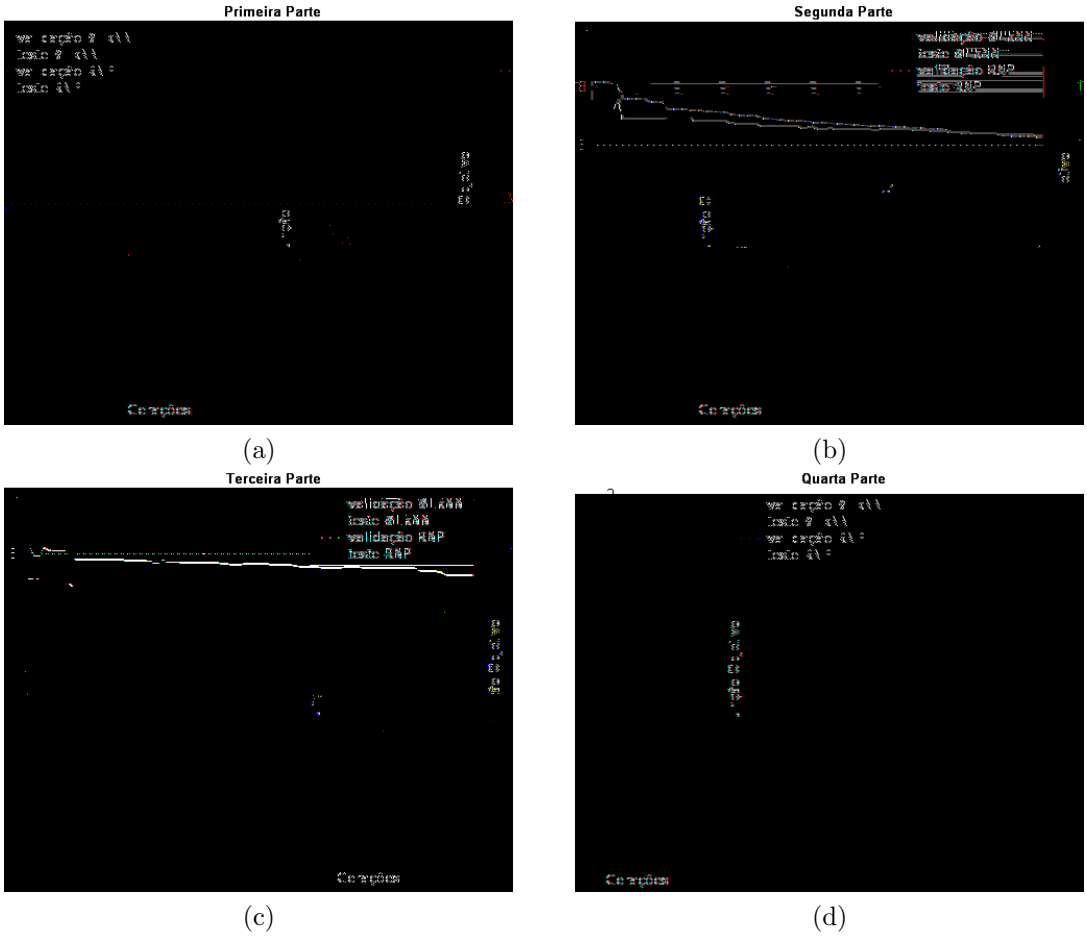
- Primeira parte: subconjunto do conjunto de teste, na faixa da amostra 1 a 816 da base de dados;
- Segunda parte: subconjunto do conjunto de teste, na faixa da amostra 817 a 1632 da base de dados;
- Terceira parte: subconjunto do conjunto de teste, na faixa da amostra 1633 a 2448 da base de dados;
- Quarta parte: subconjunto do conjunto de teste, na faixa da amostra 2449 a 3264 da base de dados;
- Quinta parte: formada pelas 764 descrições das categorias que compõem o conjunto de treino. Esta parte é unicamente utilizada para treino.

Para o treinamento dos classificadores foram utilizadas as 764 descrições das categorias, uma parte dos objetos sociais para validação e as outras 3 partes foram utilizadas para o teste. A seguir uma outra parte dos objetos sociais foi utilizada para a validação e o restante foi usado para teste. Essa rotina foi seguida de forma que as 4 partes dos objetos sociais fossem utilizadas para validação, formando assim uma validação cruzada de 4 partes. A validação cruzada de 4 partes foi realizada 3 vezes, totalizando 12 experimentos dessa forma.

Para a otimização dos parâmetros do MLkNN e da RNP, foi utilizado o **toolbox** de Algoritmo Genético desenvolvido por Houck et al em [89]. Esse **toolbox** em especial apresentou bons resultados em problemas de otimização em [74, 97–99]. Neste trabalho foram utilizados os parâmetros padrões do **toolbox** de Algoritmo Genético.

Algoritmo Genético	
parâmetros	*valores*
tamanho da população	*80*
número de gerações	*100*
representação dos indivíduos	*núm. reais*
MLkNN	
parâmetros	*limites*
ponto de corte da perda de hamming	*[0 1]*
número de vizinhos (0,01 para cada vizinho)	*[0,01 0,25]*
suavização (δ) para cada categoria	*[0,5 1,5]*
métricas (0 a 0,25: dist. Euclidiana, 0,25 a 0,5: cosseno, 0,5 a 0,75: dist. Tanimoto)	*[0 0,75]*
RNP	
parâmetros	*limites*
ponto de corte da perda de hamming	*[0 1]*
variância (σ^2) para cada categoria	*[0,5 1,5]*
função de transferência (0 a 0,25: função desnormalizada, 0,25 a 0,5: função normalizada)	*[0 0,5]*

Tabela 10: Parâmetros de otimização do Algoritmo Gen´



(a) (b)

(c) (d)

Figura 11: Validação e teste do MLkNN e RNP. Validação primeira (a), segunda (b), terceira (c) e quarta (d) partes.

sentados os resultados obtidos na validação (azul tracejado) e teste (vermelho contínuo) do MLkNN e validação (verde tracejado) e teste (preto contínuo) da RNP.

Em todos os gráficos da Figura 11 a RNP apresentou um desempenho nitidamente superior ao MLkNN. Nos quatro gráficos foi observada uma diferença entre os resultados de validação e entre os resultados de teste dos dois classificadores em torno de 0,45. Nem os resultados obtidos para a validação do MLkNN não foram, em nenhum momento, melhores que os resultados obtidos para o teste da RNP.

Além disso, observando-se a Figura 11(a) notamos que algo inesperado aconteceu: para a primeira parte de validação os conjuntos de teste de ambos os classificadores apresentaram resultados melhores que os conjuntos de validação. Com uma observação mais atenta dos gráficos notamos que a medida que percorremos as partes de validação o desempenho da validação é incrementada. O que indica que é capaz de existir alguma diferença subtancial entre uma parte e outra.

Outro aspecto notável é o desempenho do teste da RNP e do MLkNN. Apesar do desempenho da validação de ambos os classificadores alterar à medida em que são aplicados novas partes, o desempenho do teste dos mesmos permaneceu bastante estável nas últimas gerações, não ocorrendo grandes variações de um gráfico para outro. Considerando os valores obtidos na última geração para cada classificador, a RNP apresentou, para estes gráficos, uma variação máxima de 0,0261 entre o melhor resultado e o pior e na média dos três conjuntos de validação cruzada obteve uma variação máxima de 0,0735. Já o MLkNN apresentou uma variação máxima de 0,0960 e na média dos três conjuntos de validações cruzadas obteve 0,0990. Portanto a RNP apresentou uma estabilidade levemente superior em relação ao MLkNN.

Por fim, foi observado para ambos os classificadores, nas três validações cruzadas, que houve um desempenho maior na parte de validação quando foi utilizada a quarta parte como validação, seguida pela terceira, segunda e primeira partes. Com relação a parte de teste os classificadores obtiveram, em média, mais êxito quando foi usada a segunda parte para validação, seguida pela primeira, terceira e quarta partes.

Olhando nos parâmetros selecionados para cada classificador, foi observado alguns valores que foram mais constantes. Para o MLkNN o número de vizinhos oscilou entre 5 e 9 vizinhos, sendo 7 vizinhos o valor mais selecionado. Além disso, a métrica selecionada mais frequentemente foi a distância de Tanimoto e o valor médio do ponto de corte da **perda de hamming** foi de 0,5168 com variância de 0,0380. Para a RNP foi selecionada, em todas as vezes, a função de transferência normalizada (Equação 3.2) e o valor médio do ponto de corte da **perda de hamming** foi de 0,4139 com variância de 0,0264.

As Figuras 12, 13, 14, 15, 16 e 17 mostram o desempenho em gráficos dos classificadores para as métricas **perda de hamming**, **um erro**, **cobertura**, **perda de ordenacao**, **precisao media** e **categoria principal**, respectivamente. As barras em cada gráfico representam, da esquerda para direita, os resultados das validações (azul) e testes (vermelho) do MLkNN e os resultados das validações (verde) e testes (preto) da RNP. Os termos "1ª Parte", "2ª Parte" em diante são referentes a parte utilizada para a validação do treinamento e o conjunto utilizado para teste foram as outras três partes do conjunto de teste, lembrando que a base de dados foi dividida em cinco partes, mas que uma parte foi usada exclusivamente para treino. Os resultados nos gráficos são os valores médios obtidos nos três conjuntos de validação cruzada de 4 partes. Por último, o termo "Média" é o valor médio obtido nas quatro partes.

Na métrica **perda de hamming**, ilustrada na Figura 12, não houve nenhuma diferença

estatisticamente significante entre o MLkNN e a RNP, além disso, é mantido um valor quase que constante entre uma parte de validação e outra. Um provável motivo para não ter sido observada nenhuma grande variação entre os valores obtidos pode ser devido à própria natureza intrínsica da **perda de hamming**. Por seu valor ser normalizado pelo fator $\frac{1}{c}$, onde c é o número de categorias, ela pode não ser uma métrica muito útil para avaliação quando se é usada uma base de dados com muitas categorias, como no caso presente. Talvez esta métrica deveria ser melhorada para poder atender melhor casos onde são utilizadas bases de dados com um grande número de categorias.

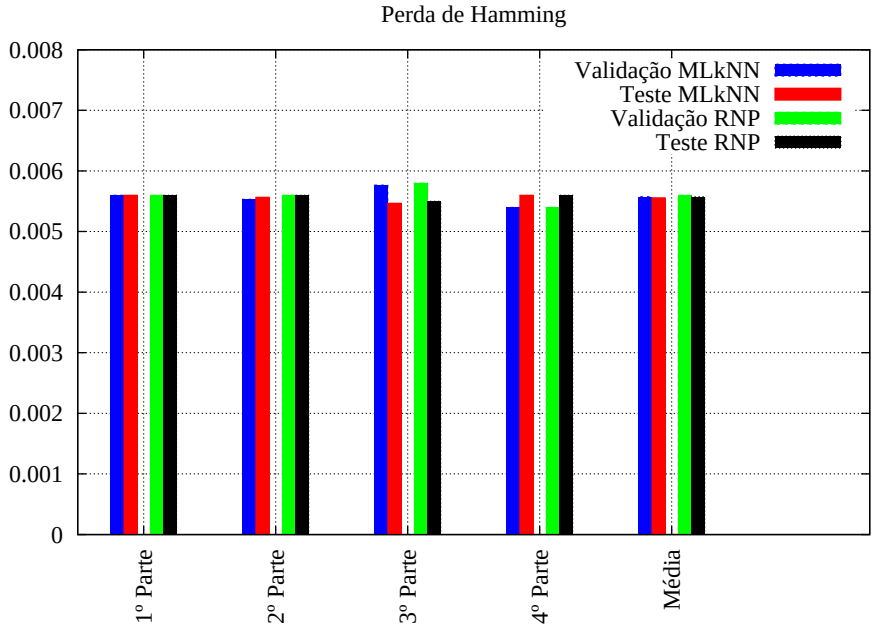


Figura 12: Resultado experimental de cada parte da base de dados CNAE em termos da **perda de hamming**.

Nas demais métricas fica claro observar que a RNP superou em todas elas o MLkNN, algumas delas ficando em torno de duas vezes melhor. Também é possível observar nos gráficos um efeito "escada" do resultado da validação, com o desempenho dela aumentando a cada vez que uma parte consecutiva do conjunto de teste era utilizada como validação. Dessa forma, o mesmo efeito que aconteceu na função objetivo foi observado na maioria das métricas.

A Figura 17 é referente a métrica **categoria principal**. Uma vez que a **categoria principal** está relacionada à ordem da categoria principal do objeto social, é esperado que este valor seja próximo de zero. Entretanto, o resultado obtido foi relativamente alto, tanto para o MLkNN quanto para a RNP. No entanto, embora não tenham sido encontrados

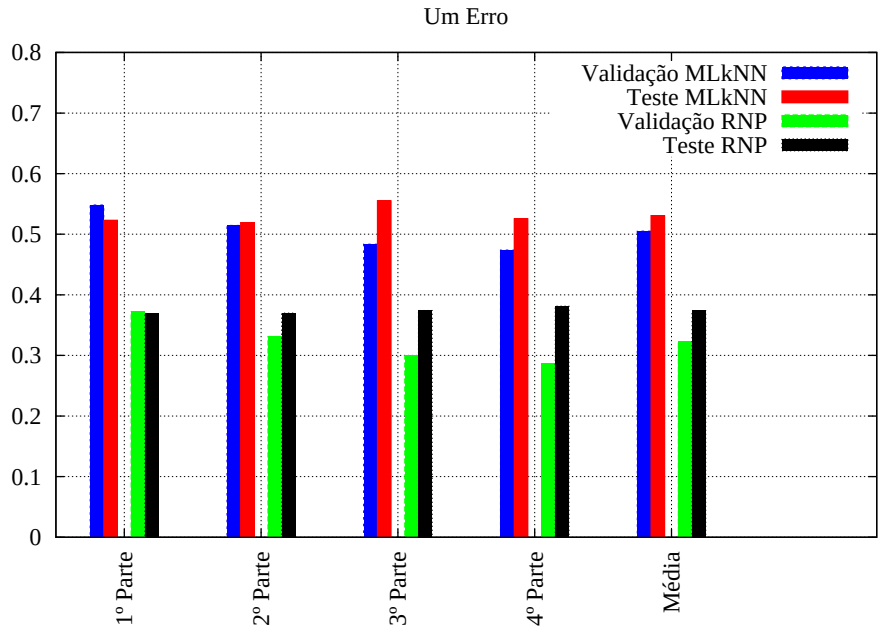


Figura 13: Resultado experimental de cada parte da base de dados CNAE em termos do **um erro**.

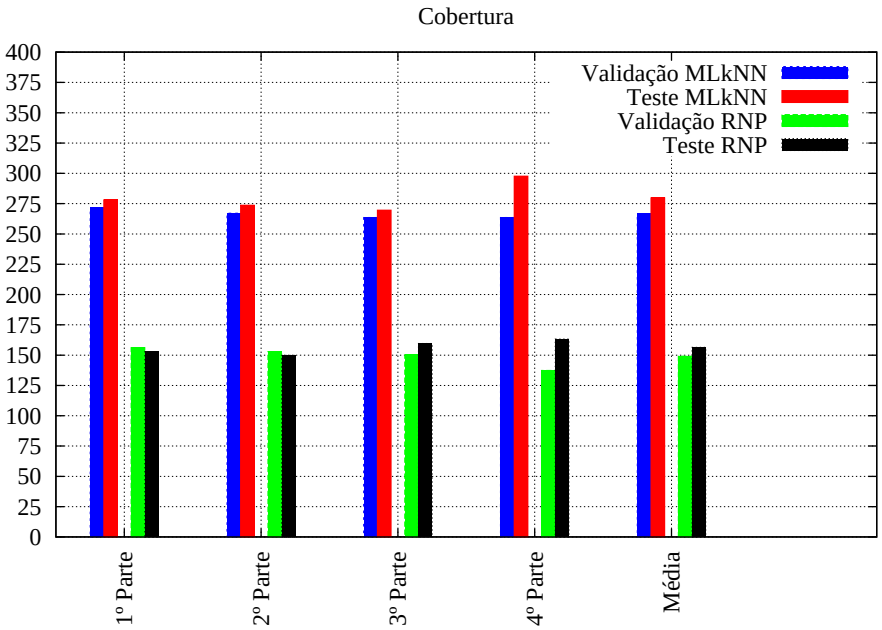


Figura 14: Resultado experimental de cada parte da base de dados CNAE em termos da **cobertura**.

bons resultados para esta métrica, a base de dados utilizada não fornece informações suficientes para a determinação da categoria principal, o que pode ser o motivo para terem sido obtidos valores tão altos.

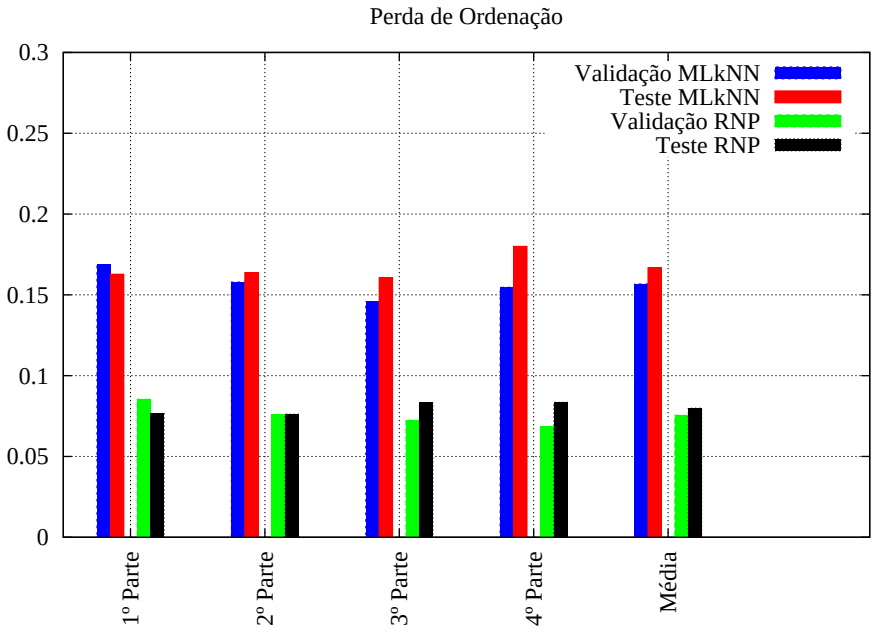


Figura 15: Resultado experimental de cada parte da base de dados CNAE em termos da **perda de ordenacao**.

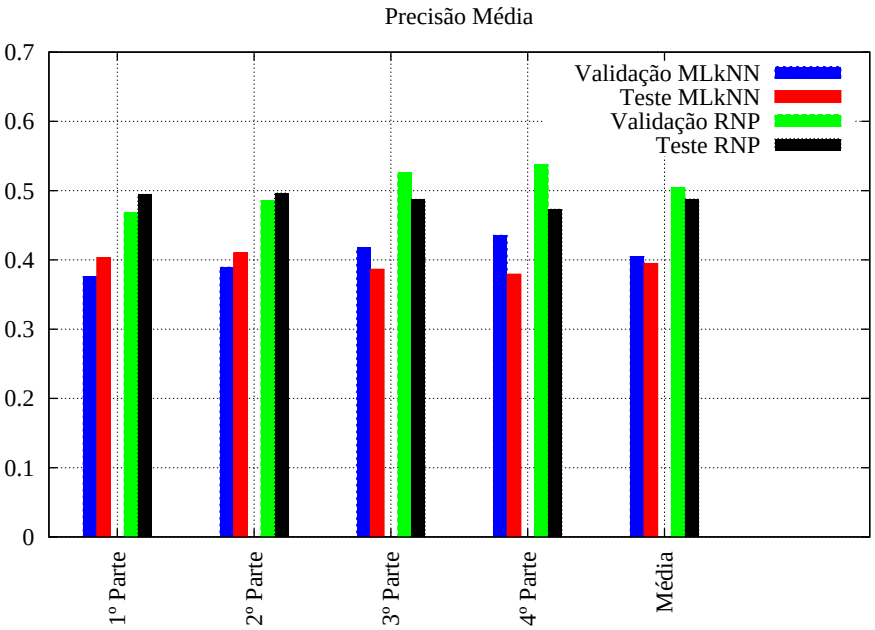


Figura 16: Resultado experimental de cada parte da base de dados CNAE em termos da **precisao media**.

Uma representação mais compacta dos resultados obtidos pode ser realizada na forma geométrica apresentada na Figura 18. Em um único gráfico foram representados todos os valores médios das métricas obtidos na parte de teste, tanto para o MLkNN quanto

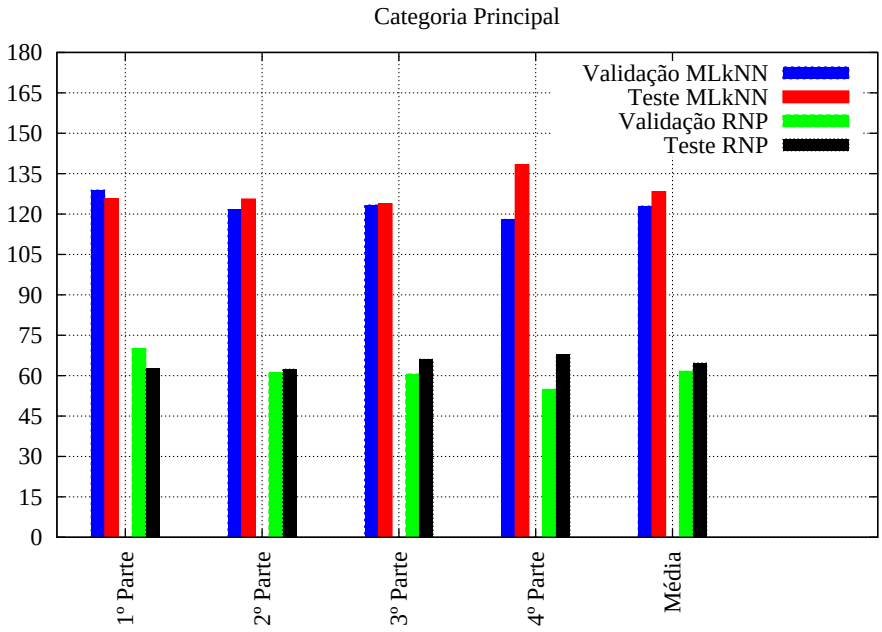


Figura 17: Resultado experimental de cada grupo da base de dados CNAE em termos da **categoria principal**.

para a RNP. Os círculos na figura servem para dar uma noção do valor das métricas, sendo o círculo menor de raio 0,333, o do meio de 0,667 e o maior 1,000. Cada eixo do gráfico corresponde a uma métrica e, para facilitar a visualização, a métrica **cobertura** foi normalizada ($cobertura/(c-1)$) e a métrica **precisao media** foi mudada para 1 - **precisao media**. Dessa forma, depois dessa normalização, o valor máximo para cada métrica é 1 e o mínimo é 0, e quanto menor for este valor melhor. Na Figura 18 é possível ver que os resultados da RNP estão quase que totalmente compreendidos dentro dos resultados do MLkNN, o que demonstra que a RNP foi superior ao MLkNN. Além disso, é interessante notar que o formato geométrico dos resultados da RNP está numa "escala" menor do que o do MLkNN. Isto indica que ambos os classificadores apresentaram melhores resultados nas mesmas métricas e piores também nas mesmas métricas. A visualização deste fato foi mais facilmente observado neste gráfico do que nos gráficos em barra apresentados anteriormente.

Os mesmos critérios de avaliação dos classificadores utilizados nas bases de dados do **Yahoo** foram utilizados para esta base de dados. Os resultados obtidos estão ilustrados na Tabela 11 e eles mostram ser coerentes com o que foi observado anteriormente nos resultados das métricas para a base CNAE. Os resultados obtidos para a métrica **categoria principal** foram desconsiderados nos critérios de avaliação por acreditarmos que deva

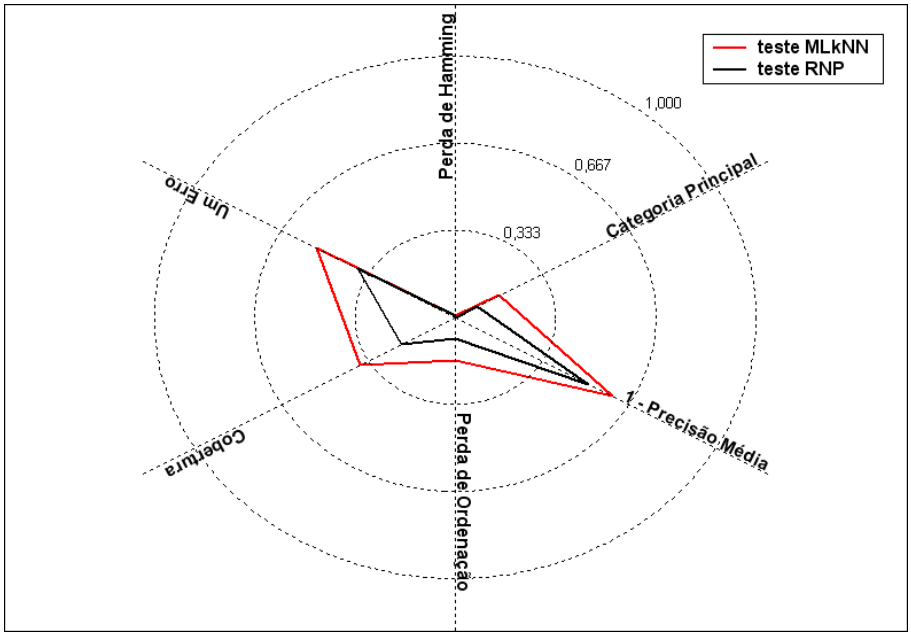


Figura 18: Gráfico geométrico das métricas para o MLkNN e RNP

MÉTRICAS	*CLASSIFICADORES*
	A_1 – *MLkNN*; A_2 – *RNP*
Perda de Hamming	*(estatisticamente não houve diferenças)*
Um Erro	$A_2 \succ A_1$
Cobertura	$A_2 \succ A_1$
Perda de Ordenação	$A_2 \succ A_1$
Precisão Média	$A_2 \succ A_1$
ORDEM TOTAL	*RNP(4) > MLkNN(-4)*

Tabela 11: Desempenho relativo dos classificadores para a base de dados CNAE.

existir alguma forte correlação com a métrica **cobertura**.

A partir deste ponto a métrica **um erro** será utilizada para uma análise mais detalhada dos resultados obtidos na base de dados CNAE. O motivo para a escolha desta métrica é devido a que, se considerarmos que o classificador retornará pelo menos uma categoria para cada amostra, então o **um erro** é a métrica capaz de informar se esta categoria está correta ou não.

A Figura 19 mostra o resultado da métrica **um erro** para os cinco níveis da hierarquia da base de dados CNAE. A medida que é descida a hierarquia surgem mais categorias e menos amostras de treinamento, o que reflete na taxa de **um erro** alcançada.

A Tabela 12 mostra, para cada hierarquia e classificador, quais são as categorias em que mais frequentemente as amostras são classificadas quando ocorre **um erro**. A coluna "Cat." mostra os códigos das categorias conforme definidos na tabela CNAE,

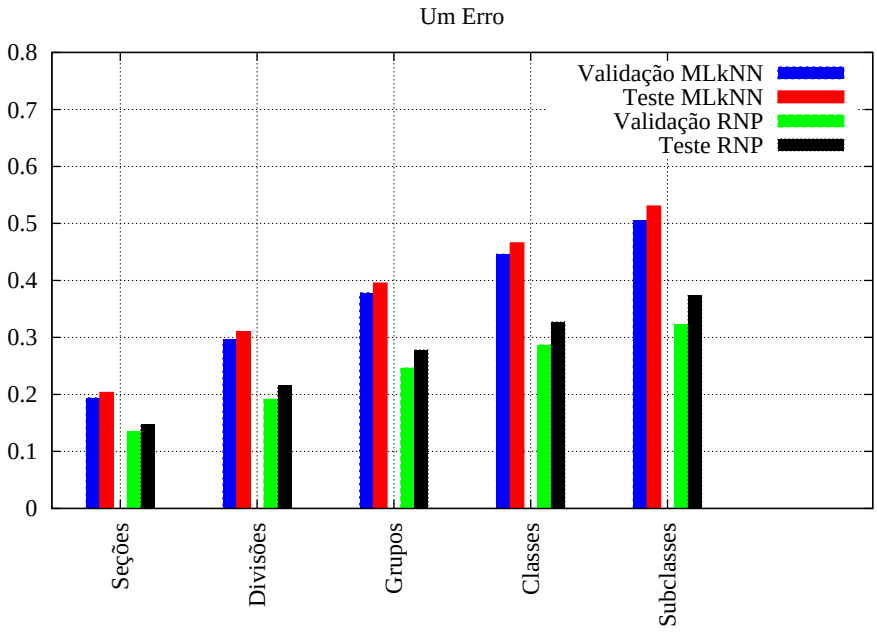


Figura 19: Resultado experimental para cada etapa da hierarquia da base de dados CNAE em termos do **um erro**.

a coluna "Clas. Err." aponta o número de amostras classificadas erradas na categoria correspondente e "Total" mostra o total de classificações erradas para cada hierarquia. Embora exista uma pequena alteração na ordem, muitas categorias aparecem em ambos classificadores, tanto na parte de validação como na de teste.

Analisando as categorias apresentadas na Tabela 12 observou-se que na hierarquia da Subclasse as categorias em que o MLkNN mais comete erros apresentam algumas diferenças em relação às categorias em que a RNP erra mais. Neste ponto é bom lembrar que só existe uma amostra para cada Subclasse no conjunto de treino e que parte do conjunto de teste é utilizado para validação. De forma geral, as categorias do MLkNN apresentam mais termos na definição das categorias ou um valor equilibrado entre a quantidade de amostras no conjunto de teste e número de termos. Por outro lado, a RNP apresentou a categoria 6025-9/01 que possui somente uma única amostra no conjunto de teste e uma no conjunto de treino e somente dois termos definem esta categoria. Além disso, grande parte das categorias que a RNP errou possui em torno de dois termos somente.

Já no nível de Grupo, os dois classificadores tendem a ficar polarizados em torno das categorias que apresentam mais amostras, como no caso da categoria 524 que apresentou ter 45 amostras no conjunto de treino e mais de 2000 amostras no conjunto de teste além

Hierarquia	MLkNN				RNP			
	Validação		Teste		Validação		Teste	
	Cat.	Clas. Err.	Cat.	Clas. Err.	Cat.	Clas. Err.	Cat.	Clas. Err.
Seção	K	33,83	K	99,83	D	17,08	D	55,83
	D	22,08	D	76,58	F	15,83	K	48,58
	F	19,17	F	61,25	K	15,75	F	47,67
	I	13,5	N	42,75	I	13,33	I	44,92
	N	13,5	J	41,17	N	11,33	N	42,25
	Total	158,42	Total	498,50	Total	110,00	Total	360,08
Divisão	74	34,33	74	101,75	45	15,83	74	51,75
	52	29,25	52	86,25	74	15,58	45	47,67
	51	26,17	51	79,33	51	13,00	51	44,42
	45	19,17	45	61,25	85	11,33	85	42,25
	85	13,5	85	42,75	52	10,5	52	37,75
	Total	241,67	Total	759,92	Total	156,25	Total	527,92
Grupo	749	32,92	749	98,25	749	12,17	749	42,00
	524	22,33	524	68,08	602	10,42	851	38,83
	455	16,5	455	53,50	851	10,25	524	33,83
	514	13,67	851	42,50	524	9,83	602	32,58
	851	13,08	514	42,42	452	8,5	661	27,5
	Total	308,75	Total	968,50	Total	201,25	Total	680,92
Classe	7499-3	29,58	7499-3	89,17	6025-9	9,58	6025-9	29,33
	4550-0	16,50	4550-0	53,50	7499-3	6,83	7499-3	25,33
	5242-6	12,58	5249-3	38,33	6611-7	6,58	6611-7	23,50
	5249-3	12,42	5242-6	36,67	7411-0	6,58	4550-0	21,25
	5212-4	9,08	5212-4	27,83	7491-8	6,5	7491-8	21,17
	Total	363,92	Total	1141,50	Total	234,00	Total	800,83
	7499-3/12	16,25	7499-3/12	47,75	6025-9/01	9,33	6025-9/01	28,17
	5212-4/00	9,08						

1. Na parte de validação calcular a quantidade de amostras classificadas erradamente e corretamente para cada categoria i e associar os valores obtidos a $Errados_i$ e $Corretos_i$, respectivamente, na qual a categoria i é a categoria com a maior probabilidade para as amostras;

2. Calcular o fator de depreciação para cada categoria i de acordo com a Equação 4.9;

$$Deprec_i = \frac{Corretos_i - Errados_i}{Corretos_i} \quad i = 1, 2, ..., c \tag{4.9}$$

   Para categorias que apresentam $Errados = 0$ e $Corretos = 0$ o fator de depreciação é igual a 1. Categorias que apresentam $Errados > 0$ e $Corretos = 0$ o fator de depreciação tende para menos infinito.

3. Calculado o fator de depreciação de cada categoria, então se calcula a nova suavização δ (para o MLkNN) ou a variância σ^2 (para a RNP) para cada categoria de acordo com a Equação 4.10.

$$\sigma_i^2 = \sigma_i^2 + alt_i \cdot \sigma_i^2 \quad ou \quad \delta_i = \delta_i - alt_i \cdot \delta_i \tag{4.10}$$

$$alt_i = (1 - Deprec_i) \cdot 0,05 \qquad alt_i \in \Re \quad | \quad alt_i \in \{0, ..., 0,5\} \quad i = 1, 2, ..., c$$

   Segundo a Equação 4.10 o valor de alt_i é limitado ao máximo de 0,5.

4. Utilizar os novos valores de δ e σ^2 para a classificação.

Os resultados obtidos utilizando este procedimento estão ilustrados na Figura 20. No gráfico foram postos os resultados do **um erro** obtidos com a modificação dos δs e σs pelo procedimento (azul claro) sobrepostos sobre os resultados originais do **um erro** (azul escuro) para ser visualizado melhor as melhoras alcançadas. Embora não se tenha alcançado uma melhora muito significativa nos resultados, este procedimento alcançou melhoras em todos os experimentos realizados na RNP (tanto nas partes de validação quanto de teste) e na grande maioria dos experimentos realizados no MLkNN. Na média, a RNP apresentou uma melhora de 2,5% nas partes de validação e aproximadamente 2% nas partes de teste. O MLkNN apresentou uma melhora de 1,3% nas partes de validação e aproximadamente 1% nas partes de teste. O MLkNN apresentou uma mudança menor nos resultados, um dos motivos pode ser que ele não seja tão sensível à mudança dos valores dos δs como a RNP é em relação aos σs. Talvez para resultados mais efetivos seriam necessários uma variação maior dos parâmetros, principalmente para o MLkNN.

É importante notar que o procedimento apresentado é um tanto ingênuo, pois ele só altera os valores dos parâmetros em uma direção, aumentando os σs das categorias, no caso da RNP, ou diminuindo os δs das categorias, no caso do MLkNN. Talvez uma versão adaptativa deste procedimento, disponibilizando as informações dele para um Algoritmo Genético ou qualquer outro Algoritmo Evolucionário, possa resultar em resultados com melhor desempenho.

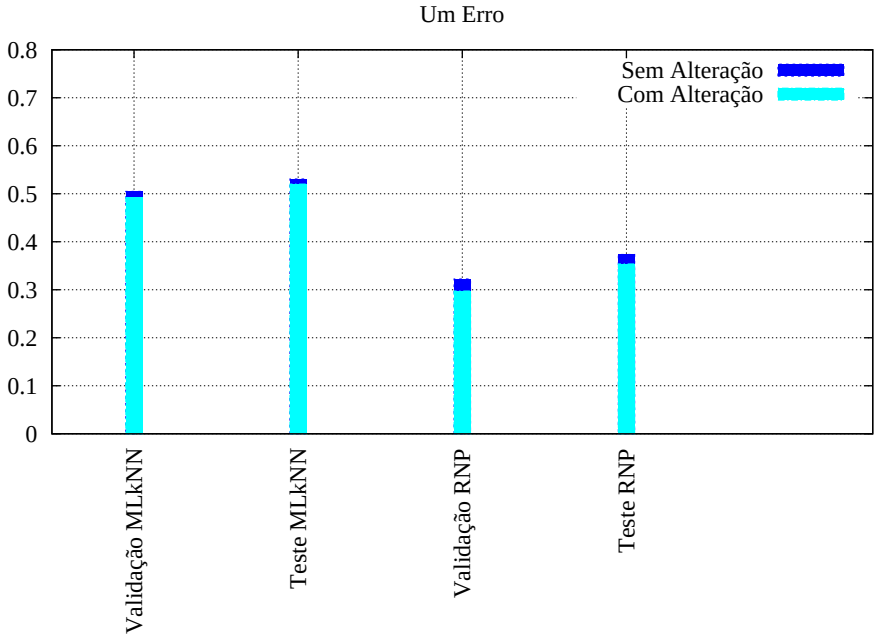


Figura 20: Resultado experimental da base de dados CNAE em termos do **um erro** com e sem alteração dos parâmetros do MLkNN e RNP.

5 Conclusões

Nesta Dissertação foi apresentada uma metologia para a classificação de atividades econômicas baseada numa versão modificada da Rede Neural Probabilística para problemas de classificação multi-rotulada. O desempenho desta Rede Neural foi inicialmente comparado a outros algoritmos, que foram projetados especialmente para este tipo de problema, através de várias bases de dados, e ela apresentou um desempenho ligeiramente superior a estes.

Com relação a classificação de atividades econômicas foram utilizadas a Rede Neural Probabilística e o MLkNN, que é uma versão modificada do tradicional classificador kNN, que apresentou bom desempenho para diversas bases de dados apresentadas na literatura. Ambos classificadores foram otimizados com a utilização de Algoritmo Genético, e os resultados obtidos mostraram que a Rede Neural alcançou um desempenho um tanto superior que o MLkNN para quase todas as métricas utilizadas e mostrou resultados comparáveis em uma das métricas (veja Figura 18 na Seção 4.3).

Neste trabalho foram otimizados, para ambos os classificadores citados acima, parâmetros para cada categoria, mas no entanto foi percebido que não foi alcançado uma otimização em nível desejado devido ao grande espaço de busca. Como trabalhos futuros, o uso de algoritmos de otimização, como Algoritmo Genético, podem alcançar melhores resultados se, antes de otimizar os parâmetros para cada categoria, fossem otimizados parâmetros que não seriam distintos entre as categorias. Este procedimento tornaria em parte o espaço de busca menos abrangente, além de permitir que o processo de otimização seja menos complexo. Outra abordagem para otimizar os resultados seria a disponibilização de um procedimento de adaptação de parâmetros que fosse permitido trabalhar em paralelo com um algoritmo de otimização.

Além disto, pelo problema constituir de uma classificação multi-rotulada a Rede Neural poderia apresentar um desempenho melhor se fosse considerada a relação que existe entre as categorias, ao invés de simplesmente considerar que as categorias são indepen-

dentes, como foi o presente caso.

Por último, a utilização de procedimentos que possam reduzir a dimensionalidade, selecionando as características mais relevantes para a representação das amostras, podem reduzir as características redundantes e aumentar a eficiência e eficácia da classificação realizada.

Referências

[1] BARRETO, J. M. Redes neurais - fundamentos e aplicações. **2^o Simposio Brasileiro de Automacao Inteligente**, 1995.

[2] HAYKIN, S. **Neural Networks - A Comprehensive Foundation**. Ninth edition. [S.l.]: Pearson Prentice Hall, 2005.

[3] RABUÑAL, J. R.; DORADO, J. **Arti cial Neural Networks in Real-Life Applications**. First edition. [S.l.]: Idea Group Publishing, 2006.

[4] LIU, W. et al. Optimal process design for minimum springback based on RBF network and evolutionary strategy. **Proceedings of the 6th World Congress on Intelligent Control and Automation (WCICA)**, Dalian, China, vol. 2, p. 7953 – 7957, Junho 2006.

[5] LIPPMANN, R. P. Review of neural networks for speech recognition. **Neural Computation**, MIT Press, Cambridge, MA, USA, vol. 1, n. 1, p. 1 – 38, 1990.

[6] ANTSAKLIS, P. J. Neural networks for control systems. **IEEE Transactions on Neural Networks**, vol. 1, n. 2, p. 242 – 244, Junho 1990.

[7] HOU, Z.-G. et al. A recurrent neural network for hierarchical control of interconnected dynamic systems. **IEEE Transactions on Neural Networks**, vol. 18, n. 2, p. 466 – 481, Março 2007.

[8] HAN, M.; CHENG, L.; MENG, H. Application of four-layer neural network on information extraction. **Proceedings of the International Joint Conference on Neural Networks**, vol. 3, p. 2146 – 2151, Julho 2003.

[9] CHAIYARATANA, N.; ZALZALA, A. M. S. Hybridisation of neural networks and genetic algorithms for time-optimal control. **Proceedings of the 1999 Congress on Evolutionary Computation (CEC)**, vol. 1, p. 389 – 396, Junho - Setembro 1999.

[10] BURKE, H. B.; ROSEN, D. B.; GOODMAN, P. H. Comparing artificial neural networks to other statistical methods for medical outcome prediction. **IEEE World Congress on Computational Intelligence**, vol. 4, p. 2213 – 2216, Junho 1994.

[11] LO, J. Y.; LAND, W. H.; MORRISON, C. T. Application of evolutionary programming and probabilistic neural networks to breast cancer diagnosis. **IEE International Joint Conference on Neural Networks**, vol. 5, p. 3712 – 3716, 1999.

[12] YAO, X. Evolving artificial neural networks. **Proceedings of the IEEE (PIEEE)**, vol. 87, n. 9, p. 1423 – 1447, Setembro 1999.

[13] WEIB, G. Neural networks and evolutionary computation. part I: Hybrid approaches in artificial intelligence. **IEEE World Congress on Computational Intelligence**, vol. 1, p. 268 – 272, Junho 1994.

[14] GAVOYIANNIS, A. E.; VOUMVOULAKIS, E. M.; HATZIARGYRIOU, N. D. On-line supervised learning for dynamic security classification using probabilistic neural networks. **IEEE Power Engineering Society General Meeting**, vol. 3, p. 2669 – 2675, Junho 2005.

[15] HOYA, T. On the capability of accommodating new classes within probabilistic neural networks. **IEEE Transactions on Neural Networks**, vol. 14, n. 2, p. 450 – 453, Março 2003.

[16] YANG, Y.; LIU, X. A re-examination of text categorization methods. **22nd Annual International SIGIR**, p. 42 – 49, Agosto 1999.

[17] CALVO, R.; CECCATTO, H. A. Intelligent document classification. **Intelligent Data Analysis**, p. 1 – 13, 2000.

[18] GREENWOOD, G. W. Training partially recurrent neural networks using evolutionary strategies. **IEEE Transactions on Speech and Audio Processing**, vol. 5, n. 2, p. 192 – 194, Março 1997.

[19] WIENER, E.; PEDERSEN, J. O.; WEIGEND, A. S. A neural network approach to topic spotting. **Proceedings of the 4th Annual Symposium on Document Analysis and Information Retrieval**, p. 317 – 332, 1995.

[20] OLIVEIRA, E. de; CIARELLI, P. M.; LIMA, F. O. The automation of the classification of economic activities from free text descriptions using an array architecture of probabilistic neural network. **VIII Simposio Brasileiro de Automacao Industrial (SBAI)**, Santa Catarina, Brasil, p. 5, Outubro 2007.

[21] CHEN, L. Pattern classification by assembling small neural networks. **Proceedings of International Joint Conference on Neural Networks**, p. 1947 – 1952, 2005.

[22] ELISSEEFF, A.; WESTON, J. A kernel method for multi-labelled classification. **Advances in Neural Information Processing Systems**, p. 681 – 687, 2002.

[23] BEKKERMAN, R.; EL-YANIV, R.; WINTER, Y. Distributional word clusters vs. words for text categorization. **Journal of Machine Learning Research**, p. 1183 – 1208, 2003.

[24] SEBASTIANI, F. Machine learning in automated text categorization. **ACM Computing Surveys**, vol. 34, n. 1, p. 1 – 47, 2002.

[25] ANAGNOSTOPOULOS, I. et al. Classifying web pages employing a probabilistic neural network. **IEEE Proceedings Software**, vol. 151, n. 3, p. 139 – 150, Junho 2004.

[26] FARKAS, J. Neural networks and document classification. **Canadian Conference on Electrical and Computer Engineering**, vol. 1, p. 1 – 4, Setembro 1993.

[27] LI, C. H.; PARK, S. C. Artificial neural network for document classification using latent semantic indexing. **IEEE International Symposium on Information Technology Convergence**, p. 17 – 21, 2007.

[28] SONG, H.-H.; KANG, S.-M.; LEE, S.-W. A new recurrent neural network architecture for pattern recognition. **IEEE Proceedings of the International Conference on Pattern Recognition**, p. 718, 1996.

[29] SONG, Y. H.; XUAN, Q. Y.; JOHNS, A. T. Comparison studies of five neural network based fault classifiers for complex transmission lines. **International Journal of Electric Power Systems Research**, vol. 43, n. 2, p. 125 – 132, 1997.

[30] DUPONT, E. M. et al. Online terrain classification for mobile robots. **ASME International Mechanical Engineering Congress and Exposition**, p. 1 – 7, Novembro 2005.

[31] SPECHT, D. Probabilistic neural networks. **Elsevier Science Ltd**, Oxford, UK, vol. 3, n. issue 1, p. 109 – 118, 1990.

[32] ZHANG, M.-L.; ZHOU, Z.-H. **MLkNN: A Lazy Learning Approach to Multi-Label Learning**. [S.l.], 2007. 25 p.

[33] ZHANG, M.-L.; ZHOU, Z.-H. A k-nearest neighbor based algorithm for multi-label classification. **IEEE { International Conference on Granular Computing**, vol. 2, p. 718 – 721, Julho 2005.

[34] EMPREENDEDOR, S. P. **Guia do Prefeito Empreendedor**. [S.l.], 2007. Disponível em: <http://www.biblioteca.sebrae.com.br/bds/BDS.nsf/D6A56CED2D24D458832572C800504609/$File/NT000351EE.pdf>.

[35] CNAE. **Classi cacao Nacional de Atividades Econômicas Fiscal**. 1.1. ed. Rio de Janeiro, RJ: IBGE – Intituto Brasileiro de Geografia e Estatística, 2003. Disponível em: <http://www.ibge.gov.br/concla>.

[36] DNRC. **Ranking das juntas comerciais segundo movimento de constituicao, alteracao e extincao e cancelamento de empresas**. [S.l.], 2007. Disponível em: <http://www.dnrc.gov.br/Estatisticas/ranking_2006.htm>.

[37] FUHR, N. et al. Probabilistic indexing and categorisation tool, intermediate prototype. **Computer Science 6, EuroSearch deliverable 4.2, LE4-8303**, p. 1 – 24, Setembro 1998. Disponível em: <http://www.is.informatik.uni-duisburg.de/bib/pdf/ir/Fuhr_etal:98b.pdf>.

[38] BUSSAB, W. de O.; MORETTIN, P. A. **Estat stica Basica**. 5º edição. ed. [S.l.]: Editora Saraiva, 2007.

[39] CNAE. **CNAE-Fiscal - Classi cacao Nacional de Atividades Econômicas-Fiscal**. [S.l.]. Disponível em: <http://200.189.113.39/sitecnae.nsf>.

[40] CNAE. **CNAE - Classi cacao Nacional de Atividades Econômicas**. Paraná, Brasil, 2005. Disponível em: <http://www.fazenda.pr.gov.br/subcomissaocnae/>.

[41] SILVA, A. B. de O.; CAMPOS, M. J. de O.; BRANDÃO, W. C. Proposta para um esquema de classificação das fontes de informação para negócio. **VI Enancib - 6º Encontro Nacional de Pesquisa em Ciência da Informacao**, p. 1 – 14, 2005. Disponível em: <dici.ibict.br/archive/00000438/01/proposta_esquema_classificação.pdf>.

[42] PORCARO, R. M. A informacão estatística oficial na sociedade da informação: uma (des)construção. **Revista Datagrama Zero**, Rio de Janeiro, vol. 2, n. 2, Abril 2001. Disponível em: <http://www.datagramazero.org.br/abr01/Art_04.htm>.

[43] FUJITA, M. S. L.; GONÇALVES, M. C.; RUBI, M. P. Política de indexação em sistemas de bibliotecas universitárias: Levantamento de subsídios para o treinamento temático do acervo bibliográfico da unesp. **Seminario Nacional de Bibliotecas Universitarias**, Salvador, n. 14, 2006.

[44] SCHMIDT, R. F. et al. **Neuro siologia**. 4º edição revista e ampliada. ed. Berlin – São Paulo: Springer-Verlag – Tradução da Editora da Universidade de São Paulo, 1979.

[45] SHEPHERD, G. M. **The Synaptic Organization of the Brain**. Fourth edition. New York: Oxford University Press Inc., 1998.

[46] MCCULLOCH, W.; PITTS, W. A logical calculus of the ideas immanent in nervous activity. **Bulletin of Mathematical Biophysics**, v. 5, p. 115 – 113, 1943.

[47] BRAGA, A. de P.; CARVALHO, A. P. de Leon F. de; LUDERMIR, T. B. **Fundamentos de Redes Neurais Arti ciais**. UFRJ, Rio de Janeiro, Brasil: 11º Escola de Computação, 1998.

[48] BISHOP, C. M. **Neural Networks for Pattern Recognition**. Second edition. New York, USA: Oxford University Press Inc., 2005.

[49] SPECHT, D. F. Probabilistic neural networks for classification, mapping, or associative memory. **IEEE International Conference on Neural Networks**, vol. 1, n. 24, p. 525 – 532, Julho 1988.

[50] PARZEN, E. On the estimation of a probability density function and mode. **Annals of Mathematical Statistics**, vol. 3, p. 1065 – 1076, 1962.

[51] NIKOLAEV, N. Y. **Probabilistic Neural Networks**. United Kingdom. Disponível em: <http://homepages.gold.ac.uk/nikolaev/311pnn.htm>.

[52] KARTHIKEYAN, B. et al. PNN and its adaptive version – an ingenious approach to PD pattern classification compared with BPA network. **Journal of Electrical Engineering**, vol. 57, n. 3, p. 138 – 145, 2006.

[53] SPECHT, D. F.; ROMSDAHL, H. Experience with adaptive probabilistic neural networks and adaptive general regression neural network. **IEEE International Conference on Neural Networks**, vol. 2, p. 1203 – 1208, 1994.

[54] DUDA, R. O.; HART, P. E.; STORK, D. G. **Pattern Classi cation**. Second edition. New York: Wiley-Interscience, 2001.

[55] GEORGIOU, V. et al. **Optimizing the Performance of Probabilistic Neural Networks in a Bionformatics Task**. Greece, 2004. Disponível em: <www.math.upatras.gr/ npav/papers/GPPAV_PNN.pdf>.

[56] KALATZIS, I. et al. Comparative evaluation of probabilistic neural network versus support vector machines classifiers in discriminating ERP signals of depressive patients from healthy controls. **Proceedings of the 3rd International Symposium on Image and Signal Processing and Analysis**, vol. 2, p. 981 – 985, Setembro 2003.

[57] HUANG, C.-J.; LIAO, W.-C. A comparative study of feature selection methods for probabilistic neural networks in cancer classification. **IEEE International Conference on Tools with Arti cial Intelligence**, p. 451 – 458, Novembro 2003.

[58] PATRA, P. K. et al. Probabilistic neural network for pattern classification. **IEEE Proceedings of the 2002 International Joint Conference on Neural Networks**, vol. 2, p. 1200 – 1205, 2002.

[59] FUNG, C. C. et al. Comparing the performance of different neural networks architectures for the prediction of mineral prospectivity. **IEEE Proceedings of 2005 International Conference on Machine Learning and Cybernetics**, vol. 1, n. 18 – 21, p. 394 – 398, Agosto 2005.

[60] JATMIKO, W. et al. Optimized probabilistic neural networks in recognizing fragrance mixtures using higher number of sensors. **IEEE Sensors**, p. 4, Novembro 2005.

[61] MAO, K. Z.; TAN, K. C.; SER, W. Probabilistic neural-network structure determination for pattern classification. **IEEE Transactions on Neural Networks**, vol. 11, p. 1009 – 1016, Julho 2000.

[62] LOFTSGAARDEN, D. O.; QUESENBERRY, C. P. A nonparametric estimate of a multivariate density function. **Annals of Mathematical Statistics**, vol. 36, p. 1049 – 1051, Junho 1965.

[63] CHATTERJI, G. P. B. N. A class of new KNN methods for low sample problems. **IEEE { Transactions on Systems Man and Cybernetics**, vol. 20, n. 3, p. 715 – 718, Maio / Junho 1990.

[64] HAO, X. et al. An effective method to improve kNN text classifier. **Eighth ACIS International Conference on Software Engineering, Arti cial Intelligence, Networking, and Parallel/Distributed Computing**, vol. 1, p. 379 – 384, Julho / Agosto 2007.

[65] BAEZA-YATES, R.; RIBEIRO-NETO, B. **Modern Information Retrieval**. First edition. New York: Addison-Wesley, 1998.

[66] CARDOSO-CACHOPO, A.; OLIVEIRA, A. L. An empirical comparasion of text categorization methods. **Proceedings of the 10th International Symposium on String Processing and Information Retrieval**, p. 183 – 196, 2003.

[67] YANG, Y. An evaluation of statistical approaches to text categorization. **Information Retrieval**, p. 69 – 90, Abril 1997.

[68] HWANG, W.-J.; WEN, K.-W. Fast kNN classification algorithm based on partial distance search. **Electronics Letters**, vol. 34, p. 2062 – 2063, Outubro 1998.

[69] COMITÉ, F. D.; GILLERON, R.; TOMMASI, M. Learning multi-label alternating decision trees from texts and data. **Lecture Notes in Computer Science**, Berlim, p. 35 – 49, 2003.

[70] SCHAPIRE, R. E.; SINGER, Y. Boostexter: A boosting-based system for text categorization. **Machine Learning**, p. 135 – 168, 2000.

[71] ELISSEEFF, A.; WESTON, J. **Kernel Methods for Multi-labelled Classi cation and Categorical Regression Problems**. [S.l.], 2001.

[72] AUSTIN, S. An introduction to genetic algorithims. **AI Expert**, vol. 5, n. 3, p. 48 – 53, Março 1990.

[73] DUKKIPATI, A.; MURTY, M. N. Selection by parts: "selection im two episodes" in evolutionary algorithms. **IEE Proceedings of the 2002 Congress on Evolutionary Computation**, vol. 1, p. 657 – 662, Maio 2002.

[74] PLAGIANAKOS, V. P.; MAGOULAS, G. D.; VRAHATIS, M. N. Supervised training using global search methods. **Advances in convex analysis and global optimisation**, vol. 54, p. 421 – 432, 2001.

[75] MICHALEWICZ, Z.; FOGEL, D. **How to Solve It: Modern Heuristics**. [S.l.]: Springer-Verlag, 2000.

[76] LAARHOVEN, P. V.; AARTS, E. **Simulated Annealing: Theory and Applications**. [S.l.]: Kluwer Academic Publishers, 1987.

[77] OMRAN, M. G. H. **Particle Swarm Optimization Methods for Pattern Recognition and Image Processing**. Tese (Doutorado) — University of Pretoria, Pretoria, Novembro 2004.

[78] MICHALEWICZ, Z. **Genetic Algorithms + Data Structures = Evolution Programs**. Third, revised and extended edition. USA: Springer-Verlag, 1996.

[79] BÄCK, T.; SCHWEFEL, H.-P. An overview of evolutionary algorithms for parameter optimization. **MIT Press**, Cambridge, MA, USA, vol. 1, n. 1, p. 1 – 23, 1993.

[80] RADWAN, E.; TAZAKI, E. Search for new learning rules for cellular neural networks using genetic programming. **IEEE - SICE Annual Conference in Sapporo, Hokkaido Institute of Tecnology**, Japan, vol. 1, p. 243 – 248, Agosto 2004.

[81] HIRSH, H. et al. Genetic programming. **IEEE Intelligent Systems Trends & Controversies**, p. 74 – 84, Junho 2000.

[82] XU, J.; ZHANG, J.; SONG, X. Evolutionary programming: The-state-of-the-art. **IEEE The Sixth World Congress on Intelligent Control and Automation**, vol. 1, p. 3296 – 3300, Junho 2006.

[83] WU, B. L.; YU, X. H. Enhanced evolutionary programming for function optimization. **IEEE International Conference on Evolutionary Computation Proceedings. IEEE World Congress on Computational Intelligence.**, p. 695 – 698, Maio 1998.

[84] BÄCK, T.; HOFFMEISTER, F.; SCHWEFEL, H.-P. A survey of evolution strategies. **In Proceedings of the Fourth International Conference on Genetic Algorithms and their Applications**, p. 2 – 9, 1991.

[85] KUNG, C. hsien et al. An adaptive power system load forecasting scheme using a genetic algorithm embedded neural network. **IEEE - Instrumentation and Measurement Technology Conference (IMTC)**, vol. 1, p. 308 – 311, Maio 1998.

[86] HE, J.; YAO, X. An analysis of evolutionary algorithms for finding approximation solutions to hard optimisation problems. **IEEE The 2003 Congress on Evolutionary Computation**, vol. 3, p. 2004 – 2010, Dezembro 2003.

[87] BENGIO, S.; BENGIO, Y.; CLOUTIER, J. Use of genetic programming for the search of a new learning rule for neural networks. **International Conference on Evolutionary Computation**, p. 324 – 327, 1994.

[88] HOLLAND, J. H. Outline for a logical theory of adaptive systems. **Journal of the Association for Computing Machinery**, vol. 3, p. 297 – 314, 1962.

[89] HOUCK, C. R.; JOINES, J. A.; KAY, M. G. **A Genetic Algorithm for Function Optimization: A Matlab Implementation**. [S.l.], 1995.

[90] GHOSHRAY, S.; YEN, K. K. More efficient genetic algorithm for solving optmization problems. **IEEE International Conference on Systems, Man and Cybernetics**, vol. 5, p. 4515 – 4520, Outubro 1995.

[91] CHAIYARATANA, N.; ZALZALA, A. M. S. Recent developments in evolutionary and genetic algorithms: Theory and applications. **IEEE Second International Conference On Genetic Algorithms in Engineering Systems: Innovations and Applications**, n. 2 – 4, p. 270 – 277, Setembro 1997.

[92] KINGDON, J.; DEKKER, L. Development needs for diverse genetic algorithm design. **IEEE Colloquium on Applications of Genetic Algorithms**, p. 1 – 11, Março 1994.

[93] LUCASINS, C. B.; KATEMAN, G. Application of genetic algorithms in chemometric. **Proceedings of the Third International Conference on Genetic Algorithms**, p. 170 – 176, 1989.

[94] DIAS, M. A. L. **Extracao Automatica de Palavras-Chave na L ngua Portuguesa Aplicada a Dissertacoes e Teses da Area das Engenharias**. Dissertação (Mestrado) — Universidade Estadual de Campinas (UNICAMP), Outubro 2004.

[95] PRISM. **The Prism Guide to Interpreting Statistical Results**. [S.l.]. Disponível em: <http://graphpad.com/articles/interpret/principles/p_values.htm>.

[96] MATLAB Central File Exchange - Student t Test for unpaired or paired samples. [S.l.]. Disponível em: <http://www.mathworks.com/matlabcentral/fileexchange/loadFile.do?objectId=12699&objectType=file>.

[97] ZHOU, Z.-H. et al. Combining regression estimators: GA-based selective neural network ensemble. **International Journal of Computational Intelligence and Applications**, vol. 1, n. 4, p. 341 – 356, Novembro 2001.

[98] ZHOU, Z.-H. et al. Genetic algorithm based selective neural network ensemble. **Proceedings of the 17th International Joint Conference on Arti cial Intelligence**, vol. 2, p. 797 – 802, 2001.

[99] ZHOU, Z.-H. et al. Selectively ensembling neural classifiers. **Proceedings of the International Joint Conference on Neural Networks**, p. 1411 – 1415, 2002.

Declarações

Como parte desse trabalho foram desenvolvidos e publicados os seguintes trabalhos:

1. Título: The Automation of the Classification of Economic Activities from Free Text Descriptions Using an Array Architecture of Probabilistic Neural Network. Autores: Elias Oliveira, Patrick Marques Ciarelli e Fabio O. Lima. Evento: VIII Simpósio Brasileiro de Automação Industrial (SBAI). Local: Florianópolis, Santa Catarina. Ano: 2007.

2. Título: Automated Free Text Classification of Economic Activities using VG-RAM Weightless Neural Networks. Autores: Alberto F. de Souza, Felipe Pedroni, Elias Oliveira, Patrick M. Ciarelli, Wallace F. Henrique e Lucas Veronese. Evento: 7º International Conference on Intelligent Systems Design and Applications (ISDA). Local: Rio de Janeiro, Rio de Janeiro. Ano: 2007.

3. Título: Intelligent Classification of Economic Activities from Free Text Descriptions. Autores: Elias Oliveira, Patrick Marques Ciarelli, Wallace F. Henrique, Lucas Veronese, Felipe Pedroni e Alberto F. de Souza. Evento: 5º Workshop em Tecnologia da Informação e da Linguagem Humana (TIL). Local: Rio de Janeiro, Rio de Janeiro. Ano: 2007.

4. Título: Uma Biblioteca Digital de Objetos Sociais de Empresas e a Classificação Automática Nacional de Atividades Econômicas. Autores: Patrick Marques Ciarelli, Wallace F. Henrique, Lucas Veronese, Rafael Zanoli e Elias Oliveira. Evento: Simpósio Internacional de Bibliotecas Digitais (SIBD) Local: São Paulo, São Paulo. Ano: 2007.

5. Título: Usando um Algoritmo Genético para Configuração de um Conjunto de Redes Neurais Probabilísticas (Resumo). Autores: Patrick Marques Ciarelli, Fabio O. Lima e Elias Oliveira. Evento: Sociedade Brasileira de Pesquisa Operacional (SOBRAPO). Local: Fortaleza, Ceará. Ano: 2007.