

**CREDIBILIDADE DE EXEMPLOS EM
CLASSIFICAÇÃO AUTOMÁTICA**

JOÃO RAFAEL DE MOURA PALOTTI

**CREDIBILIDADE DE EXEMPLOS EM
CLASSIFICAÇÃO AUTOMÁTICA**

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação do Instituto de Ciências Exatas da Universidade Federal de Minas Gerais como requisito parcial para a obtenção do grau de Mestre em Ciência da Computação.

ORIENTADOR: GISELE LOBO PAPPA

Belo Horizonte

Setembro de 2011

© 2011, João Rafael de Moura Palotti.
Todos os direitos reservados.

Palotti, João Rafael de Moura
D1234p Credibilidade de Exemplos em Classificação
Automática / João Rafael de Moura Palotti. — Belo
Horizonte, 2011
xx, 82 f. : il. ; 29cm

Dissertação (mestrado) — Universidade Federal de
Minas Gerais
Orientador: Gisele Lobo Pappa

1. Classificação automática. 2. Mineração de
Dados. 3. Programação Genética. 4. Credibilidade.
I. Título.

CDU 519.6*82.10



UNIVERSIDADE FEDERAL DE MINAS GERAIS
INSTITUTO DE CIÉNCIAS EXATAS
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÉNCIA DA COMPUTAÇÃO

FOLHA DE APROVAÇÃO

Credibilidade de exemplos em classificação automática

JOÃO RAFAEL DE MOURA PALOTTI

Dissertação defendida e aprovada pela banca examinadora constituída pelos Senhores:

Gisele Lobo Pappa
PROFA. GISELE LOBO PAPPA - Orientadora
Departamento de Ciéncia da Computação - UFMG

Marcos André Gonçalves
PROF. MARCOS ANDRÉ GOÑCALVES
Departamento de Ciéncia da Computação - UFMG

Adriano Alonso Veloso
PROF. ADRIANO ALONSO VELOSO
Departamento de Ciéncia da Computação - UFMG

Aurora Trinidad Ramirez Pozo
PROFA. AURORA TRINIDAD RAMIREZ POZO
Departamento de Informática - UFPR

Belo Horizonte, 23 de setembro de 2011.

Agradecimentos

Gostaria de agradecer algumas pessoas fundamentais para que este trabalho fosse realizado:

Muito obrigado Gisele, pela paciência, puxões de orelha e a excelente orientação.

Muito obrigado minha família, Cláudia, Adelmar e Pedro, pelo amor, conselhos e apoio nas minhas decisões.

Muito obrigado Giselle pelos nossos momentos juntos, seu amor e companheirismo que me alegraram demais durante o mestrado e me fizeram correr muito para acabar rápido e ir te ver!

Muito obrigado Thiago Salles pelas inúmeras dicas e horas dedicadas, sua ajuda foi fundamental para realização desse trabalho.

Muito obrigado ao pessoal do Speed, sem dúvida vocês fazem com que trabalhar seja ainda mais divertido!

Muito obrigado aos amigos de faculdade e ao pessoal da maratona pela companhia e amizade nos últimos anos!

Muito obrigado a todos os professores e funcionários do DCC que ao longo desses seis anos contribuíram para meu aprendizado e formação.

“I love deadlines. I like the whooshing sound they make as they fly by.”

(Douglas Adams)

Resumo

Organizar e recuperar grandes quantidades de informação tornaram-se tarefas de extrema importância, principalmente nas áreas de Mineração de Dados e Recuperação de Informação, responsáveis por estudar uma maneira de lidar com essa explosão de dados. Dentre as diversas tarefas estudadas por essas duas áreas destacamos a **Classificação Automática** de dados.

Nessa dissertação, tratamos o problema de classificar automaticamente a informação disponível. Em especial, esse trabalho foi desenvolvido em cima da ideia de que nem todos os exemplos de uma base de treinamento devem contribuir igualmente para a construção do modelo de classificação e, portanto, considerar que alguns exemplos são mais confiáveis que outros pode aumentar a eficácia do classificador. Para lidar com esse problema, propomos estimar e empregar **funções de credibilidade** capazes de capturar o quanto um classificador pode confiar em um exemplo ao gerar o modelo.

A credibilidade é considerada na literatura como dependente do contexto no qual está inserida, além de ser também dependente de quem a estima. Para tornar mais objetiva sua avaliação, recomenda-se que sejam definidos os fatores que influenciam no seu cálculo. Definimos que, do ponto de vista de um classificador, dois fatores são cruciais: as relações atributos/classe e relacionamentos entre exemplos. Relações atributos/classe podem ser facilmente extraídas utilizando um grande conjunto de métricas previamente propostas na literatura, principalmente para a tarefa de seleção de atributos. Relacionamentos entre exemplos podem ser criados a partir de uma característica presente na base. Por exemplo, no contexto de classificação de documentos, já foi mostrado que redes de citações e autorias (que relacionam dois documentos de acordo com seus autores ou artigos citados) provêm grande fonte de informação para classificação. Diversas métricas da literatura de redes complexas podem ser utilizadas para quantificar esses relacionamentos.

Baseados nesses dois fatores, selecionamos 30 métricas para explorar a credibilidade dos atributos e 16 para os relacionamentos. Elas foram inspiradas em métricas presentes na literatura que indicam a separação entre as classes e investigam as carac-

terísticas dos relacionamentos entre os exemplos. Porém, fica difícil dizer qual dessas métricas seria mais apropriada para estimar a credibilidade de um exemplo. Assim, por possuirmos um grande número de métricas para cada fator, após experimentos com métricas isoladas, criamos um algoritmo de **Programação Genética** para melhor explorar esse espaço de métricas, gerando funções de credibilidade capazes de melhorar a eficácia de classificadores se associadas a eles.

A programação genética é um algoritmo baseado nos princípios de evolução de Darwin, capaz de percorrer, de forma robusta e eficaz, o grande espaço de busca com que estamos trabalhando. As funções evoluídas foram então incorporadas a dois algoritmos de classificação: o *Naïve Bayes* e o KNN. Experimentos foram realizados com três tipos de bases de dados: bases de documentos, bases da UCI com atributos exclusivamente categóricos e uma grande base de assinaturas proteicas. Os resultados mostram ganhos consideráveis em todos os cenários, culminando em melhorias de até 17.51% na Macro F_1 da base *Ohsmed* e de 26.58% e 50.78% na Micro F_1 e Macro F_1 da base de assinaturas estruturais proteicas.

Palavras-chave: Classificação automática, Programação Genética, Credibilidade.

Abstract

Organization and recovery of large amounts of information became tasks of extreme importance, especially on the areas of Data Mining and Information Recovery, which are responsible for finding a way to deal with this data explosion. Among the topics studied in these two areas, there is the **Automatic Classification** of data.

In this thesis, we treat the problem of automatically classifying the available information. In particular, this work was developed on the consideration that not all examples in a training set contribute equally to the construction of a classification model, so, assuming that some examples are more trustworthy than others can increase the effectiveness of the classifier. To deal with this problem, we propose the use of **credibility functions** capable of capturing how much a classifier should trust an example while generating the model.

Credibility in the literature is considered as context dependent and also dependent on who is estimating it. To make its evaluation more objective, it is recommended that the factors used for its calculation are defined. We defined that, from the classifier's view, there are two crucial factors: the attribute/class relations and relationships among examples. The attribute/class relation can be easily extracted using lots of metrics already proposed in the literature, especially for the task of selecting the attributes. The relationships among the examples can be deduced from a feature that appear in the database. For example, in the context of document classification, it is shown that the networks of citations and authorship (which relate two documents based on its authors or citations) are a big source of information for the classification. Several metrics of complex networks can be used to quantify these relationships.

Given these two factors, we selected 30 and 16 metrics to explore the attributes' and relationships' credibility respectively. They were inspired in metrics that occur in the literature, and indicate the separation among the classes and investigate characteristics of the relationship between the examples. Nevertheless, it is hard to tell which of these metrics is more appropriate to estimate the credibility of an example. So, since there is a big number of metrics for each factor, after some experiments with

isolated metrics, we developed a **Genetic Programming** algorithm to better explore this search space, generating credibility functions capable of improving the effectiveness of classifiers associated with it.

Genetic programming is an algorithm based on Darwin's theory of evolution, capable of traversing the search space of functions in a robust and effective way. The evolved functions were then incorporated to two classification algorithms: Naive Bayes and KNN. Experiments have been run using three different kinds of databases: document databases, UCI databases of categorical attributes and a protein signature database. The results show considerable improvement of the classification in all cases. In particular, for the database *Oshmed*, Macro F_1 was improved by 17.51%, and for the protein signature database, Micro F_1 and Macro F_1 were improved by 26.58% and 50.78% respectively.

Keywords: Automatic classification, Genetic Programming, Credibility.

Lista de Figuras

3.1	O exemplo de teste triângulo está ligada ao grafo formado pela classe dos círculos e dos losangos, porém não apresenta ligações com os quadrados.	21
3.2	O algoritmo dos k vizinhos mais próximos.	23
3.3	Em (a) temos o algoritmo original dos K vizinhos mais próximos e em (b) temos um possível resultado da utilização da credibilidade conjuntamente ao KNN.	26
4.1	Fluxograma de um algoritmo de Programação Genética	30
4.2	Três possíveis funções de credibilidade de atributos.	33
4.3	Duas possíveis funções de credibilidade para relacionamentos.	33
4.4	A instância de teste triângulo está ligada ao grafo formado pela classe dos círculos e dos losangos, porém não apresenta ligações com os quadrados.	46
5.1	Distribuição dos exemplos nas classes das bases de documentos.	57
5.2	Distribuição dos exemplos nas classes das bases de atributos categóricos.	58
5.3	Distribuição dos exemplos nas classes na base de assinaturas estruturais proteicas.	58
5.4	Variação da <i>fitness</i> ao longo das gerações para a base da ACM-DL.	61
5.5	Modelo de Validação Cruzada com 5 partições.	62

Lista de Tabelas

4.1	Funções do PG, são usadas como vértices internos no PG	34
4.2	Explicação das principais variáveis utilizadas para definição das métricas de atributos textuais.	35
4.3	Explicação das principais expressões utilizadas para definição das métricas para relacionamentos.	45
4.4	Matriz de confusão usada para exemplificar as métricas de precisão e revo- cação.	53
5.1	Principais parâmetros utilizados no PG.	59
5.2	Experimentos mostrando a $\text{micro}F_1$ ao variar a função de <i>fitness</i>	60
5.3	Experimentos mostrando a $\text{macro}F_1$ ao variar a função de <i>fitness</i>	60
5.4	Avaliação da $\text{Micro}F_1$ com a aplicação de diversas métricas estudadas e o PG.	65
5.5	Avaliação da $\text{Macro}F_1$ com a aplicação de diversas métricas estudadas e o PG.	66
5.6	Funções de credibilidade geradas para cada uma das bases.	66
5.7	$\text{Micro}F_1$ obtida pelo <i>Naïve Bayes</i> quando usando a função de credibilidade F_{base} gerada uma dada base nas demais bases.	67
5.8	$\text{Macro}F_1$ obtida pelo <i>Naïve Bayes</i> quando usando a função de credibilidade F_{base} gerada uma dada base nas demais bases.	67
5.9	Resultados do uso de credibilidade na base da ACM-DL explorando a cre- dibilidade de termos, autoria e citação.	68
5.10	Resultados da $\text{Micro}F_1$ para as bases de atributos categóricos	69
5.11	Resultados da $\text{Macro}F_1$ para as bases de atributos categóricos	69
5.12	Resultados da $\text{Micro}F_1$ e $\text{Macro}F_1$ para a base de bioinformática.	70

Sumário

Agradecimentos	vii
Resumo	xi
Abstract	xiii
Lista de Figuras	xv
Lista de Tabelas	xvii
1 Introdução	1
1.1 Objetivos	4
1.2 Organização da Dissertação	5
2 Trabalhos Relacionados	7
2.1 Credibilidade	7
2.2 Credibilidade dos Atributos	9
2.3 Credibilidade dos Relacionamentos	11
2.4 Programação Genética	12
3 A Credibilidade na Classificação Automática	15
3.1 Classificador <i>Naïve Bayes</i>	16
3.2 Incorporando a Credibilidade ao <i>Naïve Bayes</i>	18
3.2.1 <i>Naïve Bayes</i> com Credibilidade Baseada nos Atributos.	18
3.2.2 <i>Naïve Bayes</i> com Credibilidade Baseada em Relacionamentos.	19
3.3 Classificador KNN.	22
3.4 Incorporando a Credibilidade ao KNN.	25
3.4.1 KNN com Credibilidade Baseada nos Atributos.	25
3.4.2 KNN com Credibilidade Baseada em Relacionamentos.	27

4 Modelando a Credibilidade com Programação Genética	29
4.1 Indivíduos	31
4.1.1 Credibilidade Baseada nos Atributos	33
4.1.2 Credibilidade baseada em Relacionamentos	44
4.2 Operadores Genéticos	50
4.3 <i>Fitness</i>	51
5 Experimentos	55
5.1 Bases de Dados	55
5.2 Configuração de Parâmetros	59
5.3 Metodologia Experimental	62
5.4 Credibilidade para Bases de Documentos	63
5.4.1 Credibilidade de atributos	63
5.4.2 Generalidade	64
5.4.3 Explorando Relacionamentos e Múltiplos Fatores	67
5.5 Credibilidade em Bases de Atributos Categóricos	68
5.6 Base de Bioinformática.	69
6 Conclusões e Trabalhos Futuros	71
Referências Bibliográficas	75

Capítulo 1

Introdução

É inegável que vivemos em uma época na qual temos uma capacidade nunca antes vista de gerar informação. A possibilidade de interagir com outras pessoas através da *Web* usando redes sociais com milhões de usuários (Thom-Santelli et al. [2011]), criar *blogs* com textos, vídeos e músicas (Baxter [2010]), extrair informações a respeito de todos os genes que formam organismos complexos como os nossos (Williams & Hayward [2001]), obter informações sobre as condições climáticas de um região com sensores que captam centenas de variáveis simultaneamente (Rotach et al. [2009]), são alguns entre os diversos exemplos de nossa capacidade de produzir informação. Entretanto, proporcionalmente ao aumento da disponibilidade desses dados, vemos também o aumento da dificuldade em analisar, compreender e classificar essas informações.

Organizar e recuperar grandes quantidades de informação tornaram-se tarefas de extrema importância, principalmente nas áreas de Mineração de Dados e Recuperação de Informação, responsáveis por estudar uma maneira de lidar com essa explosão de dados. Dentre as diversas tarefas estudadas por essas duas áreas destacamos a **Classificação Automática** de dados.

Nesta dissertação, tratamos o problema de classificar automaticamente a informação disponível. Substituir humanos por máquinas na tarefa de classificação não somente é útil, como necessário. Seria entediante e, muito provavelmente, impraticável para um humano classificar uma grande quantidade de dados.

De maneira simplificada, classificar consiste em estipular a qual classe, dentre as disponíveis, um certo exemplo pertence, partindo da premissa que dois exemplos estão em uma mesma classe se eles possuem um valor semântico próximo. Em geral, temos um conjunto de exemplos conhecidos, dos quais sabemos a que classe cada exemplo pertence, e um conjunto de exemplos com classe desconhecida, para os quais queremos prever a classe. O primeiro conjunto é chamado de treinamento, e o segundo de teste.

De forma geral, humanos e computadores conseguem construir um modelo de classificação a partir dos exemplos presentes no treinamento. Analisando os atributos, as semelhanças e as diferenças que fazem com que um exemplo pertença a uma classe ou a outra, um classificador, humano ou não, pode aprender, através de observações, uma maneira de classificar os exemplos de teste. Porém, existe uma diferença entre um humano e um algoritmo de classificação automática tradicional. Um humano é capaz de ponderar se um dado exemplo do conjunto de treinamento pode contribuir mais ou menos para a geração de um modelo de classificação. Isso significa que, em uma situação em que temos um exemplo difícil de classificar, dada a sua semelhança com exemplos existentes em mais de uma classe, o classificador humano pode escolher dar maior credibilidade para alguns exemplos de treino, confiando que determinados exemplos podem se assemelhar mais ao teste.

O estudo da capacidade humana de atribuir diferentes valores de credibilidade para as informações apresentadas impulsionou as pesquisas na área de credibilidade. Em geral, os pesquisadores estudam como alguns fatores (também chamados de dimensões) afetam na percepção humana de credibilidade. Eles tentam obter respostas para questões como: o que pode levar um humano a dizer que a informação A tem maior credibilidade que a informação B? Entretanto, o conceito de credibilidade nunca foi empregado do ponto de vista de um classificador automático. Dado que algumas informações são mais pertinentes que outras, buscamos nesse trabalho avaliar como explorar esse conhecimento para melhorar algoritmos de classificação automática.

Portanto, buscamos uma maneira de criar modelos de classificação que levem em conta o fato que os exemplos de treinamento não são todos igualmente significativos, assim como um humano faz. Logo, procuramos avaliar de forma automática a credibilidade dos exemplos de treinamento a fim de que aqueles exemplos com maior credibilidade possam ter maior relevância para classificar um exemplo de teste desconhecido.

Um fato importante observado na literatura é que a credibilidade não é uma propriedade fixa de um exemplo, podendo ser diferente dependendo de quem a estipula. Dessa forma, para se obter uma medida menos subjetiva, é necessário definir bem os fatores que influenciam na credibilidade de um exemplo. Neste trabalho, abordamos dois fatores que julgamos ser cruciais: (i) os atributos que compõem um exemplo e (ii) os relacionamentos mantidos entre os exemplos de treino e teste.

Para explorar a credibilidade de um exemplo usando os seus atributos, procuramos medir diretamente a relevância dos atributos na classificação. Assim, utilizamos métricas conhecidas de seleção de atributos para modelarmos a relação atributo-classe. Essa modelagem tem por objetivo prover indícios da separação entre as classes, cap-

turando assim, a importância e a confiança de um exemplo de treinamento para uma classe.

Podemos imaginar diversos exemplos práticos na tarefa de Classificação Automática de Documentos (CAD), que é especialização da classificação automática voltada para definir classes a documentos. Cada documento é composto por um conjunto de termos que são os seus atributos. Nesse caso, se um portal de notícias decidisse incorporar a credibilidade de atributos em seu classificador, provavelmente, ao avaliar a credibilidade baseada nos atributos, veríamos que o termo “Metallica” teria maior credibilidade que o termo “Anthrax” para prevermos que um documento é da classe Música. Isso acontece porque “Metallica” tradicionalmente se refere a uma banda de metal e seria um termo usado para definirmos exclusivamente a classe “Música”, enquanto “Anthrax” além de ser o nome de uma banda, também é uma doença bacteriana. Logo, ele não é tão confiável para distinguir entre “Música” e “Saúde”. Concluindo, analisando esses termos isoladamente, documentos de treino contendo o termo “Metallica” devem ter uma credibilidade maior que documentos contendo o termo “Anthrax” do ponto de vista de um classificador que pretende calcular se um documento de teste pertence a classe “Música”.

Por sua vez, para explorar a credibilidade de um exemplo usando os seus relacionamentos, temos que primeiro definir quais são esses relacionamentos, se eles existirem. Vários relacionamentos podem ser definidos dependendo do contexto que estamos abordando. Por exemplo, pessoas mantêm relacionamentos de amizade ou parentesco, livros e músicas têm relacionamentos de autorias, um país se relaciona com outro definindo estados como guerra ou paz, e assim por diante. Utilizamos a mesma linha de pensamento proposta para analisar a credibilidade de atributos, ou seja, se tivermos ao menos um relacionamento existente no problema que estamos tratando, explorá-lo tende a trazer benefícios, assim como reportado em Macskassy & Provost [2004].

Sendo assim, suponha que estivéssemos lidando com uma base de dados de livros, em que o objetivo é classificá-los de acordo com o tema, e modelássemos o relacionamento de autoria entre eles. Assim, utilizar um livro do treino que tenha pelo menos um autor em comum com o do teste poderia ser mais útil que utilizar um livro que não tem nenhum autor em comum. Indo mais além, um livro de treino que tivesse todos os autores em comum com o livro do teste poderia ter uma credibilidade ainda maior do ponto de vista do classificador. Como veremos com maiores detalhes ao longo dessa dissertação, decidimos por modelar os relacionamentos, quando existentes, com grafos e utilizar algumas métricas da área de Redes Complexas para extrair informações da relação relacionamento-classe, usada para definir a credibilidade do ponto de vista dos relacionamentos entre os exemplos de treinamento e o teste.

Tanto para a credibilidade definida pelos atributos quanto para a definida para os relacionamentos, temos uma quantidade elevada de métricas que podem ser usadas para estimá-la. Logo, surgem as perguntas:

- “Qual a melhor métrica?”
- “E se utilizássemos mais de uma métrica?”

O ponto importante é que a credibilidade é uma propriedade definida dependente do contexto que está inserida e, portanto, uma métrica ou uma combinação das métricas que define bem a credibilidade em um cenário pode não ser a melhor para outro. Logo, necessitamos de uma maneira de testar e avaliar as métricas disponíveis, assim como as suas combinações. Para tanto, empregamos o uso de Programação Genética (PG) (Koza [1992]).

PG é um método baseado no princípio da evolução de Darwin de que indivíduos mais bem adaptados tendem a sobreviver e gerarem indivíduos tão capazes ou mais aprimorados que os pais. Utilizamos essa técnica por possuir um mecanismo simples, porém eficaz, de representação de funções e por explorar de maneira inteligente o espaço de busca por uma função de credibilidade, a partir das métricas básicas que definem a credibilidade de atributos e relacionamentos. Além disso, PG já foi utilizada em vários contextos para selecionar/construir funções com sucesso (Golubski [2002], de Freitas et al. [2010]), dado seu poderoso mecanismo de busca global e tolerância a ruído (Bäck et al. [2000]).

Tendo definidos os métodos para estimar funções de credibilidade baseada em atributos e em relações, necessitamos incorporar essas funções nos classificadores, de maneira que os modelos de classificação criados levem em consideração a credibilidade dos exemplos. Nesse trabalho, avaliamos os classificadores *Naïve Bayes* e KNN em diversas bases de dados: desde simples bases da UCI (Newman et al. [1998]), passando por grandes bases de texto como a de artigos científicos da ACM e por um grande base de bioinformática (Pires et al. [2011]). Nossos resultados mostram significativos ganhos em praticamente todos os cenários, principalmente na métrica macro- F_1 , usada para capturar a dificuldade de classificar bases onde a distribuição dos exemplos é desbalanceada.

1.1 Objetivos

Destacamos os seguintes objetivos dessa dissertação:

- Definir a credibilidade de forma que ela possa ser utilizada em aplicações genéricas, e instanciá-la nos contextos de classificação automática e bioinformática.
- Incorporar a credibilidade a classificadores, para que esses criem modelos de classificação mais robustos, que levem em consideração o fato que os exemplos de treinamento não são todos igualmente significativos, e que o classificador não deveria confiar da mesma forma em todos eles.
- Testar e analisar o impacto da credibilidade nos classificadores automáticos tradicionais.

1.2 Organização da Dissertação

Esse trabalho é composto de outros cinco capítulos, apresentamos aqui, em alto nível. Buscamos, usando **Programação Genética**, representar a **credibilidade** dos exemplos através de dois fatores: seus **atributos** e seus **relacionamentos**. Esses quatro assuntos destacados são abordados na revisão da literatura apresentada no Capítulo 2. No Capítulo 3, explicamos os classificadores *Naïve Bayes* e KNN, e como incorporamos a credibilidade neles. Já o Capítulo 4 aborda como empregamos Programação Genética para evoluir funções de credibilidade. As bases usadas nos experimentos e os resultados obtidos são descritos no Capítulo 5. Por fim, concluímos e apontamos direções futuras no Capítulo 6.

Capítulo 2

Trabalhos Relacionados

Como previamente apontado, um dos principais objetivos dessa dissertação é definir o conceito de credibilidade no contexto de classificação automática, portanto reservamos a Seção 2.1 para mostrarmos como a credibilidade é vista e utilizada na literatura. Exploramos duas dimensões (ou fatores) para conseguirmos aproximar o valor da credibilidade de um exemplo presente no treinamento: os seus atributos e os seus relacionamentos. Discutimos sobre os atributos na Seção 2.2 e sobre os relacionamentos na Seção 2.3. Finalmente, a Seção 2.4 aborda a Programação Genética, que empregamos para combinar as diversas métricas de credibilidade usadas.

2.1 Credibilidade

Uma das primeiras pesquisas sobre a credibilidade data dos anos 50 (Hovland & Weiss [1951]). Naquele tempo, existia um maior foco para a credibilidade de uma fonte, por exemplo, uma pessoa que emitia uma mensagem no rádio. Observava-se a relação entre a credibilidade da fonte e os receptores da mensagem passada pela mesma. Era uma época bem próxima a Segunda Guerra Mundial e a ênfase estava no estudo do uso da *propaganda* como uma arma do governo dos Estados Unidos para ganhar o apoio das massas populares.

Na Ciência da Computação, o termo credibilidade apareceu pela primeira vez em Tseng & Fogg [1999], no qual foi estudado o que se entendia por credibilidade de computadores do ponto de vista dos humanos. Eles destacaram que, embora existissem outras dimensões¹ para o cálculo da credibilidade de um objeto, dois componentes são principais: a fidelidade (foi usado o termo inglês *trustworthiness*) e a perícia (do termo

¹Como já mostramos, “dimensões” e “fatores” são termos que aparecem com frequência em vários trabalhos relacionados a credibilidade e possuem o mesmo significado.

inglês *expertise*). Um ponto interessante dessa pesquisa é que são mostrados exemplos de como a credibilidade é tratada em diversos cenários em que ela é crucial, desde a interação de humanos com uma máquina de calcular de bolso até a informação provida por páginas na *Web*. Uma dessas situações, por exemplo, refere-se a um corretor ortográfico automático que teria sua credibilidade abalada se informasse que este texto não apresenta nenhum erro ortográfico e, posteriormente, algum erro fosse encontrado.

A credibilidade se tornou um assunto multidisciplinar (Rieh & Danielson [2007]) e alvo de muitas pesquisas relacionadas com a *Web* e com as fontes de informação presentes na mesma (Sundar [1999], Freeman & Spyridakis [2004], Flanagin & Metzger [2007], entre outros). Uma definição interessante de muitos trabalhos é que a credibilidade não é uma propriedade **exclusiva** da informação em si ou da fonte de informação, mas sim uma propriedade que é julgada pelo receptor daquela informação (Sundar [1999], Freeman & Spyridakis [2004]). Ou seja, uma mesma informação poderia ter uma credibilidade diferente dependendo de quem acessa aquela informação.

Mesmo sabendo que a credibilidade apresenta um aspecto inherentemente subjetivo, os pesquisadores procuram definir medidas objetivas para quantificá-la. Isso foi feito ao estender as dimensões previamente conhecidas, fidelidade e perícia, a fim de se obter mais métricas para mensurar a credibilidade da informação contida na *Web* (Flanagin & Metzger [2007]). Inspirados por outros trabalhos que já haviam buscado no *layout* de uma página uma opção para dimensões para credibilidade (Palmer et al. [2000], Fogg et al. [2001]), Flanagin & Metzger [2007] definiram a credibilidade de uma página como um valor composto de dimensões: o conteúdo, o *design* e os patrocinadores da página. Um resultado interessante foi mostrar como propagandas comerciais podem afetar negativamente a credibilidade da informação contida na página e que, de forma geral, páginas conhecidas de notícias têm maior credibilidade que páginas de comércio eletrônico e que essas, por sua vez, têm maior credibilidade que páginas pessoais.

Os fatores que afetam a credibilidade dependem fortemente do contexto da utilização e muitos trabalhos abordam domínios específicos. Destacamos os estudos que avaliam como usuários veem a informação contida na enciclopédia *online* Wikipédia (Chesney [2006], Lopes & Carriço [2008], Kubiszewski et al. [2011]) e em páginas de saúde e medicina (Lindberg & Humphreys [1998], Eastin [2001], Eysenbach & Köhler [2002], Rains & Karmikel [2009]). Por exemplo, em Eysenbach & Köhler [2002] identificou-se que, para que um usuário confie no conteúdo de uma página relacionada a saúde, ele analisa fatores como *emails* informados, as credenciais e as qualificações dos médicos encontradas na página.

O trabalho proposto aqui, em contraste com todos acima citados, considera a

credibilidade do ponto de vista de um classificador ao invés do usuário. Ao construir um modelo de classificação, o classificador, da mesma forma que um usuário dos sistemas citados acima, pode considerar um exemplo mais crível que outros. Procuramos construir uma função matemática que seja capaz de capturar em um valor real a credibilidade de um exemplo contido no treinamento, de forma que exemplos com baixa credibilidade sejam menos influentes que exemplos com alta credibilidade.

Como já mencionado, a credibilidade é uma característica definida por algumas dimensões que são relevantes ao problema. Para uma página *Web*, esses fatores compreendem entre várias outras coisas o *design* da página, seu patrocinador, além é claro, do conteúdo. No contexto de classificação, esses fatores têm que ser adaptados. Nesse trabalho, exploramos os dois fatores que consideramos os mais importantes: os atributos e os relacionamentos dos exemplos. Dessa forma, do mesmo modo que um usuário olha para o *design* e o conteúdo de uma página para ponderar sobre a credibilidade da informação provida naquela fonte, um classificador se baseia nos atributos e nos relacionamentos de um exemplo de treinamento para considerar a informação contida nele mais importante ou não na construção de um modelo de classificação.

2.2 Credibilidade dos Atributos

A primeira dimensão considerada para medir a credibilidade de um exemplo de treinamento é a de seus atributos. Como esse estudo iniciou-se na tarefa de classificação de documentos, uma linha de pesquisa com abordagem bem próxima a que propomos, é a *Ponderação Supervisionada de Termos* (STW - *Supervised Term Weighting*). Portanto, discutimos aqui alguns importantes trabalhos a respeito desse tema.

Um dos primeiros trabalhos a pesquisar como a ponderação de termos pode trazer significativas melhorias nas tarefas relacionadas a recuperação de informação textuais foi Salton & Buckley [1988]. Eles definiram que três componentes são importantes ao levar em consideração uma métrica de ponderação de termos: um fator referente a frequência dos termos, um fator referente a frequência dos documentos nas classes e uma normalização necessária para que termos muito populares não oprimam outros mais raros. Foram estudadas algumas métricas para cada componente e foi apontado que a frequência de um termo (TF) e o inverso da frequência dos documentos nos quais o termo aparece (IDF) formaram a melhor combinação possível para se ponderar um termo em um documento. Atualmente, essa métrica é amplamente utilizada também na classificação automática e comumente conhecida como TFIDF.

A métrica TFIDF e suas diversas variações foram posteriormente classificadas

como métricas de *Ponderação não supervisionada de termos* (do inglês, *Unsupervised Term Weighting* - UTW) por não levarem em consideração as classes às quais os exemplos de treinamento pertencem. Essa denominação foi usada pela primeira vez no trabalho de Debole & Sebastiani [2003], no qual foi definido também o termo *Ponderação Supervisionada de termos* (*Supervised Term Weighting* - STW). Debole & Sebastiani apontam a existência de três importantes fases às quais todos os classificadores são submetidos:

1. Seleção de atributos (termos, no caso de classificação de documentos);
2. Ponderação de atributos (novamente termos para classificadores textuais);
3. Aprendizado do classificador.

Os autores perceberam que, tradicionalmente, apenas as fases 1 e 3 levam em conta a relação atributo-classe, e que a fase 2 também poderia utilizar essa importante relação, ao invés de usar métricas como TFIDF. O nome *Supervised Term Weighting* vem exatamente da tentativa de realizar uma ponderação de termos supervisionada (com conhecimento da classe dos exemplos de treino) para classificação de documentos. Eles propõem que os pesos dos termos calculados na primeira fase (seleção) sejam ingredientes ativos na segunda fase (ponderação), ao invés de serem descartados como usualmente é feito.

Dessa forma, foram utilizadas as métricas χ^2 e ganho de informação para selecionar e ponderar os termos dos documentos. Por fim, são testadas métricas que buscam capturar relações locais entre termos e classes, representadas como funções $f(t_k, c_k)$, e métricas globais, nas quais temos somente pesos referentes aos termos, $f(t_k)$. Destacamos que, de acordo com a denominação de métricas locais e globais, poderíamos ter uma versão local $TFIDF(t_k, c_k)$ que considera a frequência de termos somente em uma determinada classe (ou somente os documentos de uma determinada classe no cálculo do *IDF*) e uma versão global $TFIDF(t_k)$ que é a mais comumente vista na literatura. Essa abordagem se popularizou rapidamente, dando origem a vários outros trabalhos como Lan et al. [2005], Batal & Hauskrecht [2009] e Liu et al. [2009].

Lan et al. [2005] realiza experimentos seguindo o mesmo padrão introduzido por Salton & Buckley [1988], onde são combinadas diferentes métricas dos três fatores previamente definidos: termos, documentos e normalização. Porém, dessa vez foram introduzidas algumas métricas que selecionavam e ponderavam os termos: χ^2 e *relevance frequency* (RF), sendo essa última criada no respectivo trabalho. Os experimentos realizados com o classificador SVM apontaram que a combinação de TF e RF foi a melhor entre as métricas combinadas, superando todas as combinações UTW.

Batal & Hauskrecht [2009] fizeram experimentos similares aos anteriores, mas utilizaram o KNN ao invés do SVM. Seus resultados mostram que o emprego de métricas STW provê significativas melhorias na *micro-F₁*, chegando a ter o KNN combinado com STW superando o SVM baseado em TFIDF em todos os testes.

Já em Liu et al. [2009], temos um estudo focado em como a ponderação de termos pode melhorar a precisão e revocação em problemas em que existe desbalanceamento no número de exemplos das classes mais e menos populares. Foram empregados os classificadores *Naïve Bayes* e SVM e avaliada a métrica *F₁* para todas as classes das coleções. Foi observado que classes menos populares têm um aumento substancial da métrica *F₁* ao utilizar métodos STW, o que resulta em um ganho substancial da métrica *macro-F₁*.

Por fim, além dos trabalhos relacionados a STW, destacamos o de Tang & Liu [2005]. Nele, não existe um estudo da ponderação de termos, mas sim de como podemos melhorar algoritmos de seleção de termos. Eles investigaram como diferentes métricas de seleção são sensíveis às características da base de dados explorada, sendo que algumas métricas criam um viés para o fato da existência de um certo termo em uma classe (baseadas em probabilidade como $P(t|c)$) enquanto outras tendem a se beneficiar da inexistência (baseadas em $P(\bar{t}|c)$). Os autores também estudam como algumas combinações de métricas de seleção podem se comportar melhor para bases de dados desbalanceadas.

Tendo como base o trabalho de Tang & Liu [2005], acreditamos que combinar as métricas de seleção pode resultar em uma solução mais robusta. Além disso, assim como feito pelos trabalhos relacionados a STW, acreditamos que a informação da classe dos exemplos de treinamento pode ser crucial e, portanto, deve ser empregado. Entretanto, diferentemente dos trabalhos apresentados aqui, não estamos interessados somente no problema de classificação de documentos, mas em classificação automática em geral. Além disso, procuramos resolver com Programação Genética uma das limitações do trabalho de Tang & Liu [2005]. Eles realizaram algumas combinações simples e específicas, baseadas em multiplicação de métricas de seleção, mas não mostraram se outros tipos de combinações ou envolvendo mais duas métricas poderia resultar em melhores soluções. No Capítulo 4, mostramos as várias métricas que empregamos e como podemos combiná-las através de PG.

2.3 Credibilidade dos Relacionamentos

Além dos atributos, exploramos também os relacionamentos existentes entre os exemplos para obtermos um cálculo da credibilidade mais aprimorado. Ao tomarmos uma característica existente entre duas entidades, podemos traçar um relacionamento entre as mesmas. Existem milhares de situações práticas nas quais podemos inferir algum relacionamento. Por exemplo, em uma empresa podemos traçar entre as pessoas o relacionamento de chefe/empregado, assim como dentro de uma universidade temos o relacionamento professor/aluno.

Enfim, estudar os relacionamentos entre entidades de um determinado sistema sempre foi uma tarefa que despertou o interesse de várias áreas do conhecimento (Onody & de Castro [2004], Shen & Wu [2005], Rubinov & Sporns [2010]) e a principal motivação para criação da área de Redes Complexas (Newman [2003]). Uma área que também utiliza relacionamentos para a tarefa de classificação é chamada classificação relacional.

Em suma, os algoritmos de classificação relacional se baseiam no conceito chamado homofilia (Blau [1977], McPherson et al. [2001]) que diz que exemplos similares tendem a se relacionar mais frequentemente e se comportar de maneira semelhante. Diversos trabalhos são inspirados nessa premissa. Por exemplo, em Macskassy & Provost [2004], temos descrito o método relacional de vizinhança, o qual avalia-se a classe de um dado exemplo levando em conta um certo número de vizinhos que ele possui. Mesmo sendo um método simples, ele se destaca com bons resultados. Existem várias métricas na literatura, modeladas para calcular se duas entidades em um relacionamento são semelhantes, por exemplo, a similaridade de Adamic e Adar (Adamic & Adar [2003]) ou a de Jaccard (Jaccard [1901]).

Entretanto, alguns trabalhos se dedicaram exclusivamente a certos relacionamentos específicos. Um relacionamento que é extremamente estudado na literatura por seus benefícios na área de Recuperação de Informação é o de citações. Métricas como o *hubs* e *authority* de Kleinberg (Kleinberg [1999]) ou o PageRank (Brin & Page [1998]) se tornaram populares e amplamente usadas em máquinas de busca por explorarem esse relacionamento para construção de *rankings* entre as entidades, considerando algumas mais confiáveis que outras.

Além das métricas citadas acima, diversas outras podem ser usadas para mensurar o quanto forte o relacionamento entre um exemplo de teste e os de treinamento é, auxiliando assim diretamente para capturarmos a credibilidade dos exemplos presentes no treino. Realizamos um estudo detalhado dessas e apresentamos na Seção 4.1.2. Destacamos que, novamente, temos o problema de apresentarmos um número muito

grande de métricas e querermos combiná-las a fim de alcançarmos relações que a princípio não são diretas. Para tanto, mais uma vez, recorremos ao uso da Programação Genética, pelos mesmos motivos já apontados.

2.4 Programação Genética

A Seleção Natural é um famoso processo idealizado por Charles Darwin no qual se baseia a teoria da evolução (Darwin [1859]). Ao contrário do que se acreditava na época, Darwin propôs a teoria de que os seres vivos passavam por uma contínua adaptação ao ambiente, e que aqueles que se adequavam melhor às adversidades enfrentadas eram capazes de sobreviver e procriar, gerando proles que seriam tão aptas à sobrevivência quanto foram os seus pais. Posteriormente, essa teoria foi amplamente aceita e, no campo da Computação, serviu como base para a criação da Computação Evolucionária (Goldberg [1989]).

Entre os algoritmos pertencentes ao campo da Computação Evolucionaria, destacamos a Programação Genética (PG), por ser um método fundamental para a realização dessa dissertação. PG consiste em uma técnica de aprendizado que utiliza o conceito de seleção natural para exploração do espaço de busca de soluções (Koza [1992]). Em suma, PG é um método para resolução de problemas que não requer que o usuário especifique previamente a forma ou estrutura da solução. Por exemplo, nesse trabalho, necessitamos de uma função matemática que seja capaz de representar a credibilidade proveniente de um exemplo do treinamento. Sabemos em alto nível o que queremos, e temos indícios de quais são as métricas que podemos utilizar para resolução desse problema. Entretanto, não sabemos qual é a forma mais correta para a função de credibilidade que captura a credibilidade dos exemplos. Detalhes de como um PG funciona e como o modelamos para a geração de funções de credibilidade são abordadas no Capítulo 4.

Temos em Koza [2010] uma análise de vários trabalhos com problemas resolvidos pela utilização de PG com resultados iguais ou superiores aos obtidos por outras abordagens. Embora muito usado para evolução de programas propriamente ditos, com comandos como “if”, “while”, entre outros (Spector et al. [1998], Hauptman & Sipper [2007] e Forrest et al. [2009]), PG também é usado diretamente em problemas de classificação (Cavaretta & Chellapilla [2009], Kishore et al. [2000], Freitas [2002]) e amplamente utilizado para geração de funções (Golubski [2002], de Freitas et al. [2010]). Por exemplo, em de Freitas et al. [2010], temos a utilização de PG para criação de funções capazes de identificar registros duplicados em bases de dados. Foram utiliza-

das diferentes métricas para avaliar o grau de similaridade entre dois registros, a fim de gerar uma função capaz de detectar duplicação.

Em nosso trabalho, a escolha da abordagem de evolução de funções de credibilidade com Programação Genética não foi feita devido apenas ao seu eficiente e robusto mecanismo de busca, mas também por ela ser flexível o suficiente para, de maneira fácil e elegante, representar as funções desejadas.

Capítulo 3

A Credibilidade na Classificação Automática

Como mostrado no Capítulo 2, a credibilidade é uma métrica bastante estudada na literatura, mas ainda não foi explorada sob o ponto de vista de um classificador. Como mencionado em Sundar [1999], a credibilidade não é uma propriedade exclusiva da informação em si, e sim julgada pelo receptor daquela informação. Isso quer dizer que a credibilidade de um exemplo, na tarefa de classificação, deve depender de como o classificador avalia aquele exemplo, sendo portanto, variável. Para avaliar um exemplo, contamos com o uso de duas dimensões: os seus atributos e as suas relações.

Independente da forma como é definida a credibilidade, temos que incorporá-la aos algoritmos de classificação. Isso quer dizer que, no contexto de classificação automática, os algoritmos devem ser alterados de forma a levar em consideração o valor da credibilidade dos exemplos utilizados. Nesse capítulo, explicamos e mostramos como incorporamos a credibilidade em dois algoritmos clássicos, o *Naïve Bayes* e o algoritmo dos K vizinhos mais próximos, KNN. O principal motivo para utilizarmos o *Naïve Bayes* é que esse apresenta um bom desempenho para classificar documentos (Salles et al. [2010]), que é o principal tipo de classificação que tratamos nesse trabalho. Embora o algoritmo *Support Vector Machine* (SVM) possa superar o *Naïve Bayes* em alguns cenários de classificação, o custo computacional de utilizar o SVM em um problema de classificação multi-classe é um obstáculo a ser levado em conta. Baseado no compromisso entre o custo computacional e os resultados obtidos, decidimos utilizar o algoritmo *Naïve Bayes*, detalhado na Seção 3.1. Queremos também mostrar que a credibilidade não está atrelada a apenas um classificador e, portanto, também realizamos modificações no KNN. Escolhemos o KNN, explicado na Seção 3.3, por sua simplicidade, apresentando apenas um parâmetro, seus bons resultados em bases de bi-

oinformática (Li et al. [2004], Yeang et al. [2001]), segunda aplicação considerada nesse trabalho, e por advogarmos que o KNN já utiliza uma credibilidade básica em sua própria estrutura. Finalmente, as Seções 3.2 e 3.4 abordam como podemos incorporar as funções de credibilidade nos algoritmos *Naïve Bayes* e KNN, respectivamente.

3.1 Classificador Naïve Bayes.

Nessa seção, explicamos o funcionamento do algoritmo *Naïve Bayes* original. É importante termos em mente a versão original do mesmo para que possamos compará-la à versão na qual a credibilidade é incorporada (ver Seção 3.2). Referências mais detalhadas do *Naïve Bayes* podem ser encontradas em Duda et al. [2001] e Manning et al. [2008].

De uma maneira sucinta, porém prática, podemos utilizar os seguintes passos para definir o classificador *Naïve Bayes*:

1. Cada um dos exemplos do conjunto de treinamento pode ser visto como uma tupla D -dimensional, $X = (x_1, x_2, x_3, \dots, x_D)$, onde x_i é o valor referente ao atributo A_i no exemplo X . É importante ressaltar que para um atributo A_i , podemos ter valores distintos de x_i nos vários exemplos de treino. Tipicamente, em um problema de classificação com atributos numéricos, x_i pode assumir qualquer valor real. Já em um problema de classificação categórico, x_i pode assumir uma faixa controlada de valores discretos. Em classificação de documentos, por exemplo, x_i pode ser qualquer valor inteiro natural, representando o número de vezes que temos um termo A_i em um documento X . Vale a pena lembrar também que podemos ter a coexistência de atributos numéricos e categóricos em uma mesma base de dados.
2. Suponha que existem M classes, c_1, c_2, \dots, c_M , formando o conjunto de possíveis classes \mathbb{C} . Dado um determinado exemplo X , o classificador Bayesiano prevê que X pertence à classe que tiver a maior probabilidade a *posteriori* $P(c_j|X)$. Ou seja, o classificador *Naïve Bayes* diz que X pertence a c_j , se e somente se:

$$P(c_j|X) > P(c_k|X) \quad \forall k, 1 \leq k \leq M, k \neq j, \quad (3.1)$$

onde $P(A|B)$ é um valor real entre 0.0 e 1.0 que define a probabilidade do evento A ser verdadeiro, dado que o evento B ocorreu. No caso, podemos interpretar a expressão $P(c_j|X)$ como a probabilidade da classe correta ser c_j dado o exemplo D -dimensional $X = (x_1, x_2, \dots, x_D)$.

3. Necessitamos, portanto, de uma forma de calcular a probabilidade a *posteriori* $P(c_j|X)$, que pode ser definida pelo teorema de Bayes como:

$$P(c_j|X) = \frac{P(X|c_j) \times P(c_j)}{P(X)} \quad (3.2)$$

4. Da Equação 3.2 temos que $P(c_j)$ pode ser obtida simplesmente calculando a proporção de exemplos da classe c_j que temos em nosso conjunto de treinamento. Além disso, a probabilidade $P(X)$ é uma constante independente da classe e, por isso, não precisamos calculá-la.
5. Obter $P(X|c_j)$ é uma tarefa extremamente cara computacionalmente. Porém podemos utilizar a premissa “ingênua” de que os valores dos atributos de um exemplo X são condicionalmente independentes um dos outros, dada uma certa classe. Logo,

$$P(X|c_j) = \prod_{i=1}^D P(x_i|c_j) \quad (3.3)$$

6. O valor de cada termo $P(x_i|c_j)$ da Equação 3.3 é usualmente calculado de forma diferente caso o atributo A_i seja categórico, textual ou numérico. A seguir mostramos as formas mais comuns vistas na literatura:

- Caso A_i seja categórico, então $P(x_i|c_j)$ é o número de exemplos no treinamento que pertencem a classe c_j , nos quais o valor de A_i é x_i , dividido pelo número de exemplos da classe c_j no treino:

$$P(x_i|c_j) = \frac{F_{x_i c_j}}{F_{c_j}}, \quad (3.4)$$

onde $F_{x_i c_j}$ é o número de vezes que temos o termo x_i nos exemplos de treino da classe c_j e F_{c_j} é o número de exemplos de treino da classe c_j .

- Caso A_i seja textual, temos uma versão ligeiramente diferente que pode ser expressa na seguinte fórmula:

$$P(t_i|c_j) = \frac{F_{t_i c_j}}{\sum_{k=1}^D F_{t_k c_j}}, \quad (3.5)$$

onde $F_{t_i c_j}$ é o número de vezes que temos o termo t_i nos exemplos de treino da classe c_j e D é o número de atributos existentes (tamanho do vocabulário

conhecido).

- Caso A_i seja um atributo numérico, tipicamente um valor real, então assumimos que o valor x_i do atributo A_i é dado por uma distribuição Gaussiana de média μ_i e desvio padrão σ_i e podemos usar a seguinte fórmula para calcular $P(x_i|c_j)$:

$$P(x_i|c_j) = g(x_i, \mu_{ic_j}, \sigma_{ic_j}) \quad (3.6)$$

$$g(x, \mu, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-\mu)^2}{2\sigma^2}} \quad (3.7)$$

onde μ_{ic_j} e σ_{ic_j} são a média e o desvio padrão dos valores de A_i nas tuplas de treinamento da classe c_j .

7. Finalmente, podemos juntar as Equações 3.1 e 3.3, definindo:

$$\text{Classe Atribuída} = \arg \max_{c_j \in \mathbb{C}} P(X|c_j) = \frac{F_{c_j}}{N} \cdot \prod_{i=1}^D P(x_i|c_j), \quad (3.8)$$

onde \mathbb{C} é o conjunto das possíveis classes, N é o número de exemplos no conjunto de treinamento, F_{c_j} é o número de exemplos da classe c_j em N e D é o número de atributos existentes.

3.2 Incorporando a Credibilidade ao Naïve Bayes.

Dada a formulação descrita na Seção 3.1, aqui explicamos como o *Naïve Bayes* pode ser modificado para incorporar o conceito de credibilidade. Primeiramente, modelamos a credibilidade de um exemplo inspirado unicamente em seus atributos (Seção 3.2.1), depois, somente nos relacionamentos existentes (Seção 3.2.2).

3.2.1 Naïve Bayes com Credibilidade Baseada nos Atributos.

O Naïve Bayes calcula a probabilidade de que um exemplo pertença a uma classe baseando-se diretamente nos atributos daquele exemplo, como mostrado na Equação 3.3. Procuramos estimar a credibilidade dos exemplos de treinamento explorando a influência de um termo na determinação da classe de um exemplo. Propomos a seguinte modificação na Equação 3.3:

$$P(X|c_j) = \prod_{i=1}^D (P(x_i|c_j) \cdot Cr_{atr}(x_i, c_j)), \quad (3.9)$$

onde o termo $Cr_{atr}(x_i, c_j)$ representa a credibilidade do atributo x_i na classe c_j .

Dessa forma estamos avaliando a credibilidade dos exemplos de treino para cada uma das classes disponíveis e assumimos que um atributo pode ser um bom discriminador para uma classe e ruim para outra.

Vemos esse comportamento quando usamos os termos “Metallica” e “Antrax” para determinarmos a credibilidade de um exemplo ao calcular $P(X|c_{Música})$, onde um exemplo que contém apenas o termo “Metallica” deve ter mais credibilidade que outro que contém apenas “Antrax”. Entretanto, para a avaliar $P(X|c_{Esporte})$, ambos os termos seriam impróprios e diminuiriam a credibilidade dos documentos em que eles aparecem. Ou seja, como mostrado na Equação 3.9, o que estamos dizendo é que a credibilidade de um dado exemplo do treinamento varia de acordo com a classe c_j que o Naïve Bayes está avaliando. Isto é bastante intuitivo se pensarmos novamente na classificação de documentos, onde um exemplo de treinamento da classe *Música* é um especialista em descobrir outros exemplos da mesma classe, mas não da classe *Esporte*. Logo, um documento de treino da classe *Música* deve ter alta credibilidade no momento que o classificador procura avaliar se o exemplo de teste é relacionado à *Música*.

Um interessante estudo de como a combinação de métricas pode ter bons resultados está em Tang & Liu [2005]. Os autores relatam como algumas métricas têm um viés para a existência de um atributo enquanto outras o têm para a ausência, e que combiná-las pode levar a bons resultados. Eles realizam combinações lineares simples para seleção de atributos, porém eles não as usaram para ponderação de atributos, como outros trabalhos fazem (Debole & Sebastiani [2003]).

Devido ao grande número de métricas, testar cada uma das suas possíveis combinações é uma tarefa combinatória muito cara computacionalmente, e, portanto, inviável. Por esse motivo, empregamos a *Programação Genética* (PG), um mecanismo capaz de combiná-las de forma elegante, formando funções de credibilidade mais robustas. Dedicamos o Capítulo 4 exclusivamente para abordamos em detalhes como usamos PG na geração de funções.

3.2.2 Naïve Bayes com Credibilidade Baseada em Relacionamentos.

Ao nos basearmos em atributos para calcular a credibilidade de um exemplo, o classificador ganha a noção de que se um atributo A_i tem valor x_i , então os exemplos do treinamento com essa característica podem ser melhor empregados para classificar o exemplo de teste. Nessa seção, queremos calcular o ganho do classificador ao explorar os relacionamentos existentes entre o exemplo de teste e os exemplos de treinamento.

Como vimos, o *Naïve Bayes* procura encontrar qual a classe mais provável para o exemplo de teste, calculando $P(X|c_i)$ para todas as classes. Definimos, ao lidar com a credibilidade de atributos, que alguns exemplos de treino devem ter maior credibilidade para uma classe do que para outra, pois podem ser especialistas em certa classe. Para usar a credibilidade baseada em relacionamento empregamos a mesma definição. Porém, ao invés de focarmos nos atributos, calculamos o quanto forte se relacionam os exemplos de certa classe. Logo, modificamos a Equação 3.3 para:

$$P(X|c_j) = \prod_{i=1}^D (P(x_i|c_j)) \cdot (\alpha + Cr_{rel}(X, c_j)), \quad (3.10)$$

onde Cr_{rel} representa a credibilidade dos relacionamentos e α é um fator de suavização para evitarmos que a falta de um relacionamento entre o exemplo X e uma classe c_j vá levar a obrigatoriamente a uma probabilidade nula de que X pertença a c_j .

A intuição no uso da Equação 3.10 é que se um exemplo de teste X está ligado fortemente aos exemplos de treino da classe c_j (não apenas a um, mas a todos), então aqueles exemplos devem ter maior credibilidade para a classificação.

É necessário, portanto, termos pelo menos um relacionamento definido entre os exemplos. Caso não seja possível, então a credibilidade baseada em relacionamentos não trará nenhum efeito. Entretanto, muitos problemas apresentam informações suficientes para que mais que um relacionamento seja traçado. Por exemplo, em classificação de documentos podemos ter redes de autoria e citação. No primeiro caso, criamos um elo entre dois documentos se eles têm um mesmo autor em comum e, no segundo, criamos um elo entre eles se um cita ou é citado pelo outro. Temos relacionamentos bastante parecidos em uma base de músicas ou vídeos, em que músicas (filmes) de mesmo gênero ou mesmo cantor (ator ou diretor) estariam relacionadas(os).

Para situações nas quais temos mais de um relacionamento, definimos que:

$$Cr_{rel}(X, c_j) = \prod_{i=1}^R Cr_i(X, c_j), \quad (3.11)$$

onde R é o número de relacionamentos existentes.

É oportuno destacar também que podemos modelar todos esses relacionamentos através de um conceito computacional chamado grafo, base para área de Teoria dos Grafos (Bondy & Murty [2008]).

Formalmente, um grafo $G = (V, E)$ é uma estrutura composta por um conjunto V de vértices e outro E de arestas. Os vértices representam os exemplos do conjunto de treinamento os quais estamos interessados, sejam documentos, músicas ou enzimas.

As arestas, por sua vez, cumprem o papel de relacionar os exemplos segundo algum critério previamente estipulado. Por exemplo, podemos modelar um grafo de autoria, definindo uma aresta $e(d_1, d_2)$ se d_1 e d_2 têm autores em comum. Podemos ir mais além e atribuir um valor inteiro k para essa aresta, significando que d_1 tem k autores em comum com o documento d_2 .

De forma geral, dado um relacionamento r entre os exemplos, é possível construir um grafo utilizando todo o conjunto de treinamento. Entretanto, estamos interessados em avaliar como as interações dentro de uma mesma classe podem nos ajudar a prever a qual classe o exemplo de teste pertence, e, portanto, decidimos por montar um grafo para cada possível classe. Ou seja, tomamos todos os exemplos de treinamento da classe c_i e construímos o grafo G_i , relativo ao relacionamento r . Assim podemos analisar com maior clareza a relação do teste com cada grafo de cada classe, evitando interferências de interações entre exemplos de treino de classes diferentes. Destacamos que é esperado que o exemplo de teste se conecte a mais de um grafo, representando suas vários ligações com classes distintas contidas no treino para um dado relacionamento r .

Na Figura 3.1, temos uma exemplificação dessa situação. Nela, o exemplo de teste é representado por um triângulo e contém arestas para todos os exemplos da classe *Losango*, uma aresta para um indivíduo da classe *Círculo* e nenhuma para nenhum exemplo da classe *Quadrado*. Esse cenário apresenta indícios para acreditarmos que o triângulo não pertença a classe *Quadrado*. Dependendo da função que utilizarmos para a credibilidade, a classe dos círculos ou dos losangos pode se tornar mais ou menos importante, porém temos certo que a credibilidade da classe *Quadrado*, baseando no relacionamento modelado, é muito baixa ou nula.

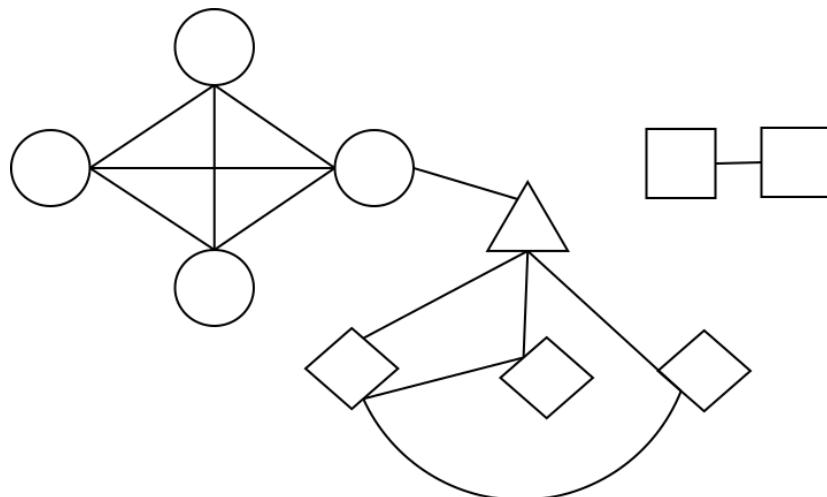


Figura 3.1: O exemplo de teste triângulo está ligada ao grafo formado pela classe dos círculos e dos losangos, porém não apresenta ligações com os quadrados.

Note que podemos, sem problema algum, juntar as Equações 3.9 e 3.10 dando origem a um classificador Bayesiano que leva em conta tanto a credibilidade dos atributos quanto a dos relacionamentos:

$$P(X|c_j) = \prod_{i=1}^D (P(x_i|c_j) \cdot Cr_{atr}(x_i, c_j)) \cdot (\alpha + Cr_{rel}(X, c_j)) \quad (3.12)$$

A credibilidade dos atributos é calculada a partir dos relacionamentos e existe mesmo sem um exemplo de treinamento, porém, a credibilidade dos relacionamentos somente existe quando temos um exemplo de teste específico. Isso faz com que essa seja uma técnica *lazy*, ou seja, somente utilizamos o conhecimento aprendido a partir dos relacionamentos quando estamos testando, sem a criação de um modelo previamente provido com o treinamento.

Pelo fato de modelarmos os relacionamentos existentes entre os exemplos por meio de grafos, decidimos utilizar as diversas métricas da área de Redes Complexas (Newman [2003]) a fim de explorarmos as propriedades dos grafos criados. Um exemplo simples de uma dessas métricas é contar o número de vizinhos de um vértice, $viz(v, c_j)$. Essa função retorna o número de conexões que o vértice v tem com seus vizinhos que são da classe c_j . Partimos do pressuposto que se um vértice for importante para uma determinada classe j , $viz(v, c_j)$, será um valor superior para aquela classe. Na Figura 3.1, a classe *Losango* seria a de maior credibilidade para classificarmos o triângulo, baseando nessa métrica. Outras várias métricas importantes podem ser listadas e combinadas, sendo assim, atribuímos toda a Seção 4.1.2 para maiores detalhes.

3.3 Classificador KNN.

O algoritmo dos K vizinhos mais próximos (KNN) é conhecido por ser um método de aprendizado baseado em analogias, ou seja, comparamos o teste com os exemplos contidos no treinamento a fim de conseguirmos encontrar aqueles com maior semelhança. Tipicamente, cada exemplo existente é uma tupla de D dimensões e representa um ponto em um espaço D -dimensional. Quando um novo exemplo de teste necessita ser classificado, o algoritmo KNN procura nesse espaço D dimensional pelos k exemplos do treinamento que estão mais perto do teste. Por fim, os k vizinhos mais próximos realizam uma votação para escolherem qual será a classe que o teste pertence.

Na Figura 3.2, mostramos um triângulo que representa o exemplo de teste, além de quadrados e losangos, representando os exemplos de treinamento pertencentes as classes *Quadrado* e *Losango*, respectivamente. As órbitas circulares existentes na fi-

gura são epicêntricas e servem apenas para fins didáticos. Por essa figura, vemos a importância do parâmetro k na decisão da classe que um exemplo pertence. Considerando que cada exemplo tem o mesmo peso na votação, caso $k = 1$, escolhemos a classe *Losango* para o exemplo de teste, entretanto, se $k = 5$, escolhemos a classe *Quadrado*. Para valores de k entre 2 e 4, como temos 3 vizinhos a uma mesma distância, necessitamos de algum método de desempate. Em nosso trabalho, ordenamos os vizinhos primeiramente pela distância até o teste e depois por um identificador único. Estamos condicionados a ordem dos identificadores como critério de desempate, o que não tem nenhuma semântica, mas é necessário definir.

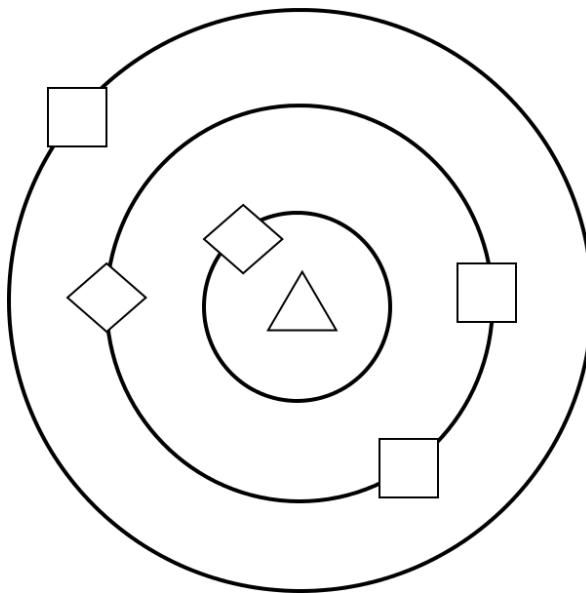


Figura 3.2: O algoritmo dos k vizinhos mais próximos.

Como vimos, uma parte fundamental do KNN é podermos calcular o quão próximo um exemplo de teste está dos de treinamento. Essa proximidade é tipicamente calculada por uma métrica de distância que pode variar de problema para problema. Utilizamos 3 métricas diferentes, dependendo se estamos classificando um atributo A_i que seja categórico, numérico ou textual. Para dois exemplos $X_1 = (x_{11}, x_{21}, x_{31}, \dots, x_{D1})$ e $X_2 = (x_{12}, x_{22}, x_{32}, \dots, x_{D2})$, nos quais x_{ij} representa o valor do atributo A_i para o exemplo j , temos que, caso A_i seja:

- Numérico, utilizamos a distância Euclidiana:

$$dist(X_1, X_2) = \sqrt{\sum_{i=1}^D (x_{i1} - x_{i2})^2} \quad (3.13)$$

Com o intuito de evitar que um atributo com grande escala de valores sobreponha um outro atributo de uma menor escala, empregamos a normalização *min-max* para todos os valores x_i do correspondente atributo A_i :

$$x'_i = \frac{x_i - \min_{A_i}}{\max_{A_i} - \min_{A_i}} \quad (3.14)$$

- Categórico, comparamos X_1 e X_2 e somamos uma unidade a distância entre os exemplos para cada x_i que tenha valores distintos entre X_1 e X_2 .

$$dist(X_1, X_2) = \sum_{1 < i < D \wedge x_{i1} \neq x_{i2}} 1.0 \quad (3.15)$$

- Textual, tomamos a distância dos cossenos entre os dois exemplos X_1 e X_2 e invertemos o seu sinal. Ou seja, sendo $\|X_j\| = \sqrt{(x_{1j} \cdot w_{1j})^2 + (x_{2j} \cdot w_{2j})^2 + (x_{3j} \cdot w_{3j})^2 + \dots + (x_{Dj} \cdot w_{Dj})^2}$ a norma do vetor representado por um exemplo X_j no espaço, x_{ij} como a frequência do peso de um termo A_i no documento j e w_{ij} representando o peso referente ao atributo A_i no documento j , temos que a semelhança entre dois exemplos X_1 e X_2 pode ser definida como:

$$cosSim(X_1, X_2) = \sum_{1 < i < D} \frac{x_{i1} \cdot w_{i1} \cdot x_{i2} \cdot w_{i2}}{\|X_1\| \cdot \|X_2\|} \quad (3.16)$$

Primeiro, cabe explicar que usamos a métrica IDF (*inverse document frequency*, Seção 4.1.1), como o peso w_{ij} do termo A_i no documento j . Segundo que, como dito, a métrica acima é chamada de semelhança dos cossenos por se basear no ângulo cosseno entre dois vetores no espaço. Como observado, estamos interessados na métrica inversa, logo:

$$dist(X_1, X_2) = -cosSim(X_1, X_2) \quad (3.17)$$

Note que após calcularmos a distância entre o exemplo de teste e os exemplos de treino, realizar uma votação entre os k vizinhos mais próximos é um processo muito simples. Basta contabilizar os votos de cada um dos k vizinhos e atribuir ao exemplo de teste a classe vencedora.

3.4 Incorporando a Credibilidade ao KNN.

Realizamos nessa seção um paralelo ao feito na discussão sobre o *Naïve Bayes* nas Seções 3.2.1 e 3.2.2. Portanto, iremos discutir como incorporar ao KNN a credibilidade baseada nos atributos dos exemplos na Seção 3.4.1 e aos relacionamentos na Seção 3.4.2.

3.4.1 KNN com Credibilidade Baseada nos Atributos.

Acoplamos a credibilidade ao algoritmo KNN de forma bem semelhante ao que fizemos com o *Naïve Bayes*. Novamente, buscamos uma maneira de quantificar o relacionamento entre um atributo e uma classe. Note que o KNN não realiza essa associação explicitamente, ou seja, como vimos na Seção 3.3, somente observamos a classe pertencente a um exemplo de treinamento quando já temos os k vizinhos mais próximos definidos, então realizando uma votação para saber qual a classe a ser escolhida.

O KNN é um algoritmo que, de uma maneira geral, compara dois exemplos, um do conjunto de teste e outro do de treino, e se baseia em algum cálculo respectivo a um mesmo atributo A_i de cada um dos exemplos para calcular a distância entre eles. Dessa vez, fica bem claro que alteramos a distância entre dois exemplos ao usar a credibilidade baseada nos atributos.

No algoritmo KNN, a fórmula da distância entre dois exemplos muda drasticamente dependendo do tipo dos atributos. Efetivamente, em nossos testes, empregamos a credibilidade para classificação textual e categórica. Embora acreditemos que seja perfeitamente possível utilizarmos o mesmo raciocínio para definirmos a credibilidade em atributos numéricos, deixamos essa tarefa como trabalho futuro. A seguir, nas Seções 3.4.1.1 e 3.4.1.2, apresentamos as modificações realizadas no algoritmo dos k vizinhos mais próximos para os atributos categóricos e textuais, respectivamente.

3.4.1.1 KNN Categórico.

Procuramos, ao desenvolver as modificações seguintes, ter em vista dois fatos importantes:

1. Se o exemplo de treinamento e o de teste têm os mesmos valores para todos os seus atributos, então a distância entre eles é zero.
2. Um exemplo de treino que apresenta diferença em T atributos do exemplo de teste, tende a ter uma distância menor que outro que apresente $T + 1$ atributos distintos.

A regra 1 é bastante intuitiva. A regra 2 diz que, ao levar em consideração a credibilidade do exemplo de treinamento, criamos uma mudança no espaço de atributos. Porém, queremos que essa mudança seja controlada. Ou seja, a hipótese básica do KNN de que exemplos parecidos (com grande número de atributos iguais) estão bem próximos em uma distribuição espacial deve ser respeitada.

Visualmente, o que queremos com a modificação realizada no algoritmo dos vizinhos mais próximos está expresso na Figura 3.3. Na Figura 3.3-(a) temos a situação já discutida anteriormente, composta de um exemplo do treino que difere em uma característica, três que diferem em duas e um outro que difere em três, sendo que esses são os cinco vizinhos mais próximos de nosso exemplo de teste. Por sua vez, na Figura 3.3-(b) temos os mesmos exemplos mostrados, porém com as distâncias entre o teste e o treino sendo comparadas utilizando a credibilidade de que o exemplo de teste pertence a cada uma das classes dos exemplo de treino. Pode-se observar que os exemplos da classe *Losango* foram mais afetados, o que pode significar que o exemplo de teste pertence a essa classe. Dessa vez, ao contrário do que temos na situação anterior, para valores de k entre 1 e 3, sabemos definir que o teste pertence a classe dos losangos, sem precisarmos utilizar uma métrica especial de desempate.

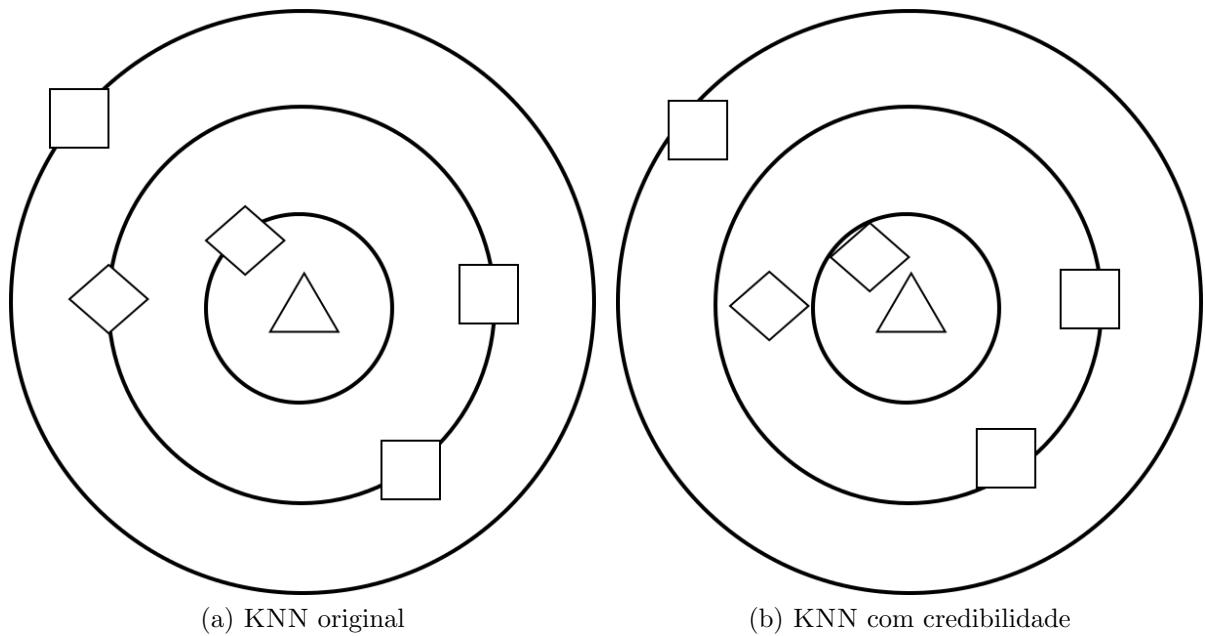


Figura 3.3: Em (a) temos o algoritmo original dos K vizinhos mais próximos e em (b) temos um possível resultado da utilização da credibilidade conjuntamente ao KNN.

Levando em consideração o que foi mostrado até esse ponto, sendo X_1 o exemplo de teste, X_2 o de treino e c_{X_2} a classe a qual X_2 pertence, modificamos a Equação 3.15

para:

$$dist(X_1, X_2) = \sum_{1 < i < D \wedge x_{i1} \neq x_{i2}} \frac{1.0}{1.0 + Cr_{atr}(x_{i1}, c_{X_2}))} \quad (3.18)$$

Adicionamos um termo relacionado a credibilidade na Equação 3.15, um fator inversamente proporcional ao valor calculado para a credibilidade. Note que como definimos a credibilidade como sendo sempre um valor positivo, a distância entre X_1 e X_2 será sempre menor ou igual a uma unidade, sendo que, quanto menor ela for, maior a credibilidade.

3.4.1.2 KNN Textual.

Assim como feito na incorporação da credibilidade nos atributos categóricos, utilizamos a informação da classe do exemplo de treino para aferir a credibilidade daquele exemplo em relação àquela classe. Sendo t_i o termo referente ao atributo A_i , x_i a frequência do mesmo e w_i o seu peso (como já explicado, usamos a métrica IDF), adaptamos a Equação 3.16, como:

$$dist(X_1, X_2) = \sum_{1 < i < D} \frac{Cr_{atr}(x_{i1}, c_{X_2}) \cdot x_{i1} \cdot w_{i1} \cdot x_{i2} \cdot w_{i2}}{\|X'_1\| \cdot \|X_2\|} \quad (3.19)$$

Repare que utilizamos $\|X'_1\|$ como a norma do vetor X_1 , levando em conta a credibilidade em relação à classe do exemplo de treino X_2 , logo temos que:

$$\|X'_1\| = \sqrt{(Cr_{atr}(x_{11}, c_{X_2}) \cdot x_{11} \cdot w_{11})^2 + \dots + (Cr_{atr}(x_{D1}, c_{X_2}) \cdot x_{D1} \cdot w_{D1})^2}.$$

Novamente, assim como discutido na Seção 3.2.1, existem diversas métricas que podemos utilizar para inferirmos a credibilidade de um elemento, a fim de melhorarmos a classificação do algoritmo KNN. Elas serão discutidas no Capítulo 3.

3.4.2 KNN com Credibilidade Baseada em Relacionamentos.

A mesma situação exposta com detalhes na Seção 3.2.2 retoma à cena. Modelamos a credibilidade existente nos relacionamentos entre um exemplo de teste e uma classe, baseando nos exemplos do treino. Diferentemente da modelagem da credibilidade para o conteúdo do algoritmo KNN, dessa vez podemos criar apenas um modelo que se adapta a qualquer tipo de atributo numérico, categórico ou textual. Logo, sendo R o número de relacionamentos modelados, X_1 o exemplo de teste, X_2 o exemplo de treino e α o fator somado a credibilidade dos relacionamentos para evitarmos valores nulos no denominador, temos:

$$dist(X_1, X_2) = \frac{dist(X_1, X_2)}{\alpha + Cr_{rel}(X_1, class_{X_2})} \quad (3.20)$$

$$Cr_{rel}(X, c_j) = \prod_{i=1}^R Cr_i(X, c_j) \quad (3.21)$$

Note que quanto maior a credibilidade do relacionamento do teste X_1 a uma classe c_i , menor será a distância entre o exemplo X_1 e os exemplos do treino que pertencem a classe c_i , exatamente como modelado para o *Naïve Bayes*.

Mais uma vez, podemos utilizar as várias propriedades dos grafos para calcularmos a credibilidade dos relacionamentos. Na Seção 4.1.2, definimos várias métricas que calculam essas propriedades.

Capítulo 4

Modelando a Credibilidade com Programação Genética

No Capítulo 3, além de mostrarmos os algoritmos de classificação *Naïve Bayes* e KNN, vimos como podemos incorporar a credibilidade aos mesmos. Discutimos também que existem várias métricas que podem ser utilizadas para mensurar a credibilidade de um exemplo, embora não detalhamos essas métricas.

Nessa dissertação, modelamos a credibilidade em duas abordagens diferentes, uma levando em consideração os atributos dos exemplos e outra utilizando os seus relacionamentos. Em ambas somos capazes de gerar uma grande gama de métricas que conseguem capturar a relação entre os exemplos e uma determinada classe, medindo a credibilidade que os exemplos de treino têm para cada classe. Entretanto, necessitamos de uma forma de selecionar e combinar essas métricas de uma maneira robusta. Para solucionar esse problema, recorremos ao uso de Programação Genética (PG) (Koza [1992]).

Segundo a Teoria da Evolução de Darwin, indivíduos mais adaptados ao ambiente em que se encontram têm uma chance maior de sobreviverem e se reproduzirem, passando seu material genético às gerações posteriores. Baseado nessa teoria, o método de Programação Genética é adequado para evoluir uma função de credibilidade, pois possui mecanismos de busca que o torna capaz de explorar bem o imenso espaço de soluções formado pelo grande número de métricas disponíveis (Fogel et al. [2000]). Além disso, PG é flexível o bastante para ser capaz de representar as funções que desejamos, sendo também tolerante a ruído.

Para entendermos melhor como é o funcionamento de um PG, utilizamos a Figura 4.1. Ela ilustra o comportamento do arcabouço de Programação Genética usado pelo pacote *gpc++* (Fraser & Weinbrenner [2011]), usado em todos os experimentos

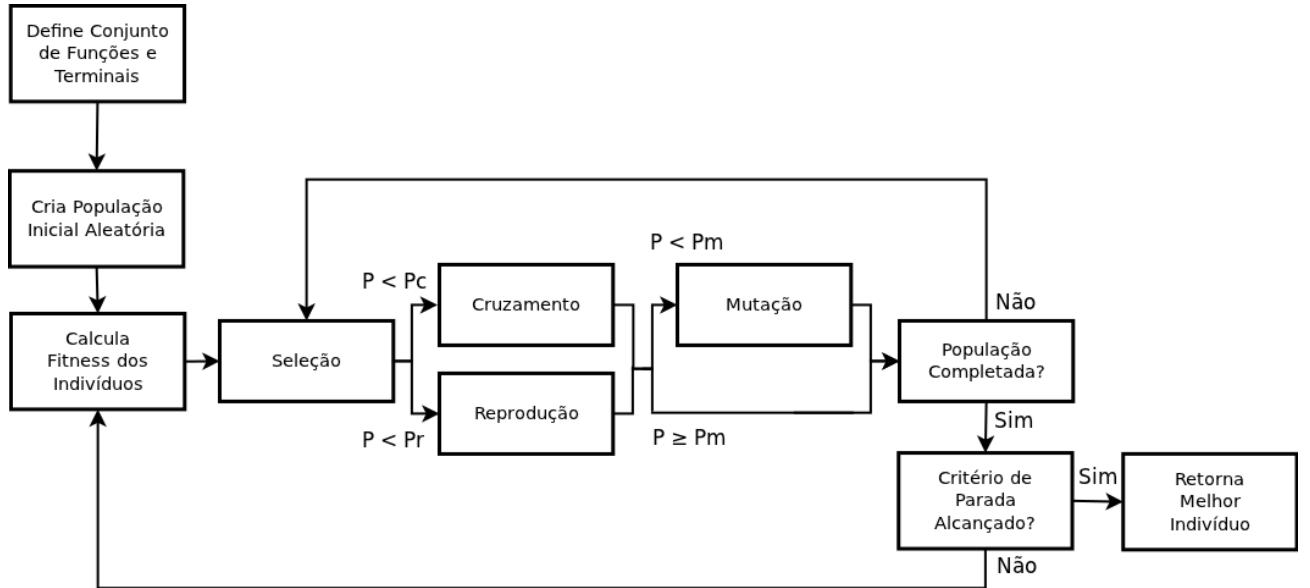


Figura 4.1: Fluxograma de um algoritmo de Programação Genética

dessa dissertação. Mesmo se tratando de um arcabouço com várias especificidades, ele pode ser facilmente adaptado para considerar as especificidades do problema aqui apresentado. Como podemos ver, o primeiro passo de qualquer PG é definir um conjunto de funções e terminais para serem usados pelo PG. É a partir desse conjunto de elementos que criamos nossos indivíduos. Esses, por sua vez, são soluções candidatas para o problema que abordamos.

Um conjunto de indivíduos é chamado de população e, como mostrado pelo segundo retângulo no diagrama da Figura 4.1, a população inicial é criada aleatoriamente, utilizando o método *Ramped-Half-and-half*, que garante que metade dos indivíduos sejam representados por árvores de tamanhos variados e a outra metade por árvores completas. Cada indivíduo da população é então avaliado por uma função de *fitness*, que calcula a habilidade do mesmo em resolver o problema apresentado, ou seja, o quanto bom aquele indivíduo é.

Após avaliarmos a *fitness* dos indivíduos, os melhores são selecionados e submetidos às operações genéticas de cruzamento, reprodução e mutação, com probabilidade P_c , P_r e P_m , respectivamente, definidas pelo usuário. Uma forma muito usada de selecionar os indivíduos é através de torneios, nos quais tomamos aleatoriamente um número pré-definido de indivíduos da população e selecionamos aquele com maior valor da função de *fitness*. Existem diversas outras variações do método de seleção (Koza [1992]), sendo que o importante é utilizar a função *fitness* dos indivíduos de forma a criarmos uma próxima geração mais bem preparada para resolução do problema.

Depois de serem selecionados, os indivíduos podem passar por reprodução ou

cruzamento, com probabilidades P_r e P_c , respectivamente. A operação genética de reprodução simplesmente transfere o indivíduo para próxima geração, enquanto o cruzamento exige a participação de dois indivíduos que são combinados a fim de obtermos uma prole mais adaptada na próxima geração. Após esse processo, o indivíduo resultante da reprodução ou cruzamento pode ser submetido à mutação com uma probabilidade P_m . Em geral, a mutação é uma pequena alteração em alguma parte específica do indivíduo, mudando uma função interna ou um terminal. Variações nas quais a operação de mutação ocorre em paralelo a de reprodução ou cruzamento também são comuns na literatura.

Esse processo de criação de indivíduos é repetido até atingirmos o tamanho limite para a população. Após chegarmos a esse limite, uma nova geração é iniciada. O processo então é repetido, geração a geração, até que um limite de gerações seja alcançado. Outras condições de parada do PG seriam chegar a uma variação arbitrariamente pequena de melhoria de uma geração para outra ou conseguir algum indivíduo com um valor de *fitness* pré-definido. Por fim, o PG retorna o melhor indivíduo evoluído.

Repare que todo o processo exposto na Figura 4.1 é *independente* da aplicação na qual o PG é utilizado. Entretanto, três importantes componentes devem ser instanciados dependendo do contexto no qual trabalhamos. Eles são:

1. A representação dos indivíduos, que inclui as funções do PG e terminais relevantes à aplicação, e são abordados na Seção 4.1,
2. Os operadores genéticos, descritos na Seção 4.2,
3. A função de *fitness* detalhada na Seção 4.3.

4.1 Indivíduos

Uma parte essencial da construção de um PG é definir a representação dos indivíduos que compõem a população. Para isso, temos que focar em três aspectos primordiais: o que um indivíduo representa, quais são suas funções e quais são os seus terminais.

A representação de um indivíduo é dependente das características do problema que estamos lidando. Portanto, o primeiro passo deve ser estudar a base de dados do problema para saber se podemos utilizar a credibilidade de atributos e/ou de relacionamentos e, se usarmos a credibilidade de relacionamentos, quantos e quais relacionamentos podem ser explorados. Por exemplo, em uma base de dados de documentos, os termos dos documentos seriam usados para criarmos uma função de credibilidade de atributos e os relacionamentos de autoria e/ou citação seriam usados para criarmos

funções de credibilidade de relacionamentos. Entretanto, se estivermos tentando resolver um problema em que não existam informações para inferirmos relacionamentos, então usaremos somente os atributos. O caso contrário também é possível.

Como inicialmente discutido no Capítulo 3, criamos uma função de credibilidade referente a cada dimensão que trabalhamos, ou seja, uma função para os atributos e uma para cada relacionamento utilizado. Dessa forma, um indivíduo é composto por um **conjunto** de funções de credibilidade que são usadas para melhorarmos os algoritmos de classificação empregados. As Figuras 4.2 e 4.3 mostram cinco exemplos de funções de credibilidade que podem ser geradas pelo PG. As três contidas na primeira figura ilustram funções de credibilidade baseadas em atributos e as demais são baseadas em relacionamentos entre os exemplos¹. Exemplificando, se o problema apresentar a possibilidade de explorarmos a credibilidade de atributos e de dois relacionamentos, então um possível indivíduo do PG poderia ser formado por uma das funções de credibilidade mostradas na Figura 4.2 e das duas funções da Figura 4.3, uma para cada relacionamento.

Como vemos nas Figuras 4.2 e 4.3, as funções de credibilidade apresentam a estrutura de árvore, comuns em trabalhos com PG (Koza [1992]). Árvores, em suma, são estruturas de dados semelhantes aos grafos, porém sem ciclos, ou seja, ao se percorrer as arestas a partir de um determinado vértice, não podemos voltar àquele vértice. Pedimos a atenção do leitor para o fato que em qualquer árvore existem dois tipos diferentes de vértices, os internos e os externos. Os vértices externos recebem o nome especial de folha e em nossas árvores representam as métricas de credibilidade usadas. Já os vértices internos são operadores usados para interconectar as folhas da árvore. Se observarmos a função 2 na Figura 4.2, os vértices internos são o “+” e o “%”, enquanto os externos são $AM(x,c)$, $P(x|c)$ e $IG(x,c)$, métricas usadas para estimar a credibilidade de atributos.

Na Tabela 4.1, mostramos a lista completa de funções do PG usadas. Como usual para um PG que evolui uma função matemática numérica, as funções PG se consistem de operações matemáticas conhecidas, como a multiplicação, divisão, soma, entre outros. Observe que modificamos as funções de subtração, divisão e logaritmo. Tanto a divisão por zero, quanto o logaritmo de números negativos não são matematicamente definidos. Além disso, não queremos ter uma função de credibilidade negativa, e por isso evitamos que a subtração e o logaritmo retornem números negativos. Portanto, tratamos todos esses casos explicitamente retornando zero.

Pelo fato de termos um grande número de terminais definidos, subdividimos essa

¹As cores são usadas apenas para auxiliar na explicação das operações genéticas de mutação e cruzamento na Seção 4.2

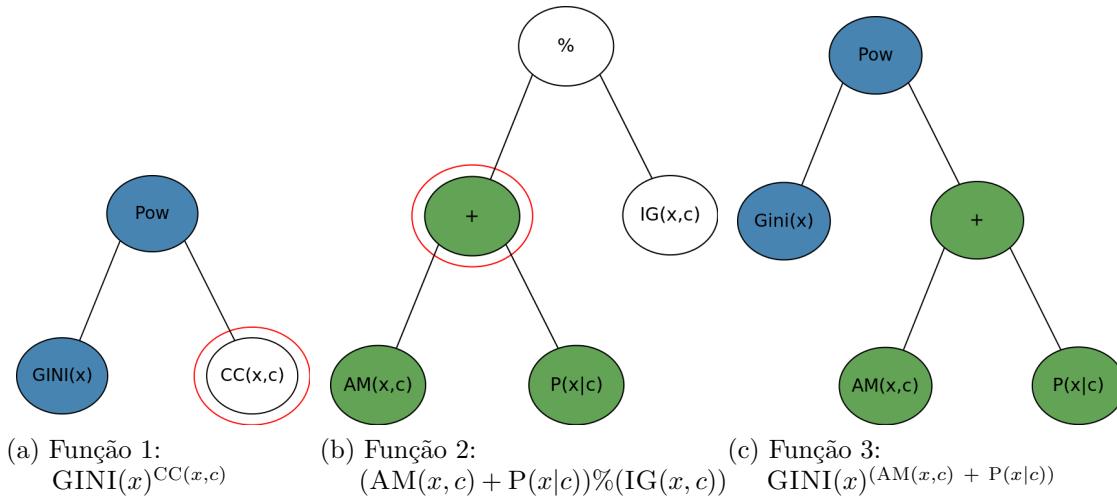


Figura 4.2: Três possíveis funções de credibilidade de atributos.

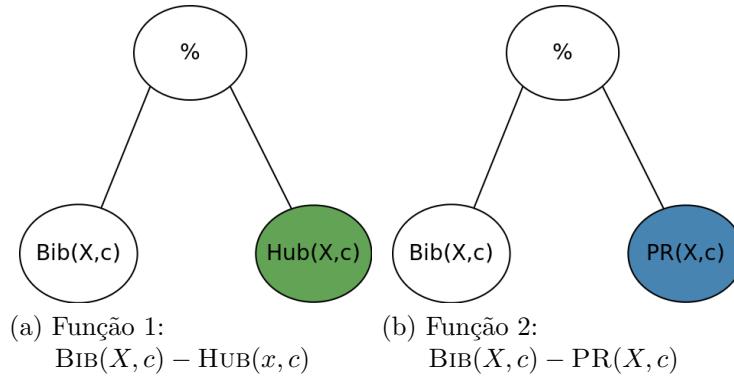


Figura 4.3: Duas possíveis funções de credibilidade para relacionamentos.

seção nas Seções 4.1.1 e 4.1.2 exclusivamente para descrevermos as métricas relacionadas ao atributos e aos relacionamentos entre os exemplos, respectivamente.

4.1.1 Credibilidade Baseada nos Atributos

Nessa seção apresentamos várias métricas utilizadas como terminais na construção dos indivíduos usados pela PG. Elas são o resultado de um extensivo estudo na literatura por métricas que possam ajudar o classificador a capturar a credibilidade de um dado exemplo de treinamento, ao tentar classificar o exemplo de teste em uma classe. Elas foram modeladas de maneira sutilmente diferente para os casos nos quais os atributos são textuais ou categóricos, sendo que a modelagem para atributos numéricos foi deixada como trabalho futuro.

Em suma, algumas métricas estão muito atreladas à classificação de documentos,

Tabela 4.1: Funções do PG, são usadas como vértices internos no PG

Função Interna	Explicação
$+(a, b)$	Soma a com b .
$-(a, b)$	Subtrai b de a , porém retorna 0 se b for maior que a .
$\times(a, b)$	Multiplica a com b .
$\%(a, b)$	Divide a por b , porém retorna 0 se b for 0.
$\text{Pow}(a, b)$	Eleva a a potência de b .
$\log(a, b)$	Logaritmo de a na base b , retorna 0 se a ou b forem menores que 1.

em especial aquelas que realizam cálculos baseados na frequência de termos e documentos. Entretanto, muitas são genéricas o suficiente para serem utilizadas em qualquer contexto de classificação baseada em atributos. Dessa forma, na Seção 4.1.1.1 mostramos as métricas que foram modeladas única e exclusivamente para serem utilizadas na tarefa de classificação de documentos e na Seção 4.1.1.2 temos as métricas que foram estendidas e estão sendo usadas também para a classificação categórica. Ao todo são quatorze métricas para uso exclusivo de classificadores de documentos e dezesseis que podem ser usadas para textos e categorias. Por fim, na Tabela 4.2 apresentamos a nomenclatura utilizada pelas métricas aqui listadas.

4.1.1.1 Métricas Modeladas Exclusivamente Para Atributos Textuais.

As métricas usadas para atributos textuais consistem em algumas variantes do TFIDF (Salton & Buckley [1988]), métrica mais difundida entre as apresentadas.

Frequência do Termo

A frequência de um termo (TF da expressão em inglês, *Term Frequency*) é simplesmente o número de vezes que um termo aparece na base de treinamento. Utilizamos o logaritmo desse valor como um fator normalizador, para evitar que termos muito frequentes dominem a função de credibilidade:

$$\text{TF}(t_i) = 1.0 + \log(F_{t_i}). \quad (4.1)$$

Temos que $\text{TF}(t_i)$ é a primeira e uma das muitas métricas que são chamadas de métricas globais. Essa denominação provém de trabalhos como o de Lan et al. [2005] ou Liu et al. [2009], nos quais métricas que não utilizam a informação da classe são vistas como globais por terem o mesmo valor para todas as possíveis classes. Em geral, métricas que calculam máximo são globais.

Tabela 4.2: Explicação das principais variáveis utilizadas para definição das métricas de atributos textuais.

Usadas na classificação de atributos textuais.	
Variável	Significado
D	Número de termos da base de dados.
DF_{t_i}	Número de documentos do conjunto do treino que contém o termo t_i .
$DF_{t_i c_j}$	Número de documentos do treino com o termo t_i e pertencentes a classe c_j .
CF_{t_i}	Número de classes em que o termo t_i ocorre.
F_{t_i}	Número de vezes que o termo t_i aparece nos documentos de treinamento.
$F_{t_i c_j}$	Número de vezes que o termo t_i aparece em documentos da classe c_j no treino.

Usadas na classificação de atributos categóricos.

Variável	Significado
D	Número de atributos da coleção.
$F_{x_i c_j}$	Número de exemplos de treino da classe c_j e com o atributo A_i valendo x_i .
F_{c_j}	Número de exemplos de treino da classe c_j .

Usadas em ambos os tipos de classificação.

Variável	Significado
N	Número de documentos na base de treinamento.
M	Número de classes da base de dados.

Frequência do Termo por Classe

A frequência de um termo em uma classe, TFCLASSE, segue o mesmo padrão utilizado pela métrica TF, porém dessa vez contamos apenas os termos que aparecem em uma dada classe c_j :

$$\text{TFCLASSE}(t_i, c_j) = 1.0 + \log(F_{t_i c_j}). \quad (4.2)$$

Frequência dos Documentos por Termo

A métrica DF, do inglês, *Document Frequency*, procura analisar a importância de um termo em relação ao número de documentos em que o mesmo aparece:

$$\text{DF}(t_i) = 1.0 + \log(DF_{t_i}). \quad (4.3)$$

Frequência de Documentos por Termo-Classe

A métrica DFCLASSE avalia o número de documentos que um termo aparece em relação a uma classe específica, tentando capturar a importância de um termo para uma classe em relação ao número de documentos onde aquele termo está presente:

$$\text{DFCLASSE}(t_i, c_j) = 1.0 + \log(DF_{t_i c_j}). \quad (4.4)$$

Inverso da Frequênci a de Documentos

O Inverso da Frequênci a de Documentos, IDF do inglês *Inverse Document Frequency*, é uma métrica que avalia a popularidade de um determinado termo em um conjunto de documentos. Temos que quanto mais popular é um termo ao longo dos documentos de uma coleção, pior é sua capacidade de discriminação, logo:

$$\text{IDF}(t_i) = \log\left(\frac{|N|}{DF_{t_i}}\right). \quad (4.5)$$

Inverso da Frequênci a de Documentos por Classe

O Inverso da Frequênci a de Documentos por classe, IDFCLASSE é uma versão do IDF em que selecionamos somente documentos de uma dada classe:

$$\text{IDFCLASSE}(t_i, c_j) = \log\left(\frac{DF_{c_j}}{DF_{t_i c_j}}\right). \quad (4.6)$$

Frequênci a do Termo Inverso da Frequênci a de Documentos

Uma das métricas de pesos para atributos em classificação de documentos mais populares na literatura é o TFIDF (do inglês, *Term Frequency Inversed Document Frequency* (Salton & Buckley [1988])). O TFIDF combina a frequênci a de um termo (*Term Frequency*) que assume que múltiplas aparições de um termo em um documento são mais importantes que aparições únicas com o inverso da frequênci a de um documento (*Inversed Document Frequency*) que diz que termos raros são de maior poder discriminativo que termos muito frequentes. Em síntese, a fórmula de TFIDF é:

$$\text{TFIDF}(t_i, c_j) = F_{t_i c_j} \cdot \log\left(\frac{|N|}{DF_{t_i}}\right). \quad (4.7)$$

Máxima Frequênci a do Termo Inverso da Frequênci a de Documentos

Calculamos o valor máximo de TFIDF para todas as classes e utilizamos esse valor como uma métrica discriminatória global que relacionará mais fortemente

os termos com as classes nas quais o TFIDF é máximo. Logo, usamos:

$$\text{MAXTFIDF}(t_i) = \text{TFIDF}(t_i, c_j) \mid \text{TFIDF}(t_i, c_j) > \text{TFIDF}(t_i, c_k), \forall c_k \in \mathbb{C} \quad (4.8)$$

Frequência do Termo Inverso da Frequência da Classe

O TFICF (do inglês, *Term Frequency Inversed Class Frequency*) é uma variação do TFIDF. Novamente, TF se refere a quanto importante é um termo em uma classe, pois trata-se de sua frequência. Por sua vez, ICF é usado para que termos que aparecem em poucas classes tenham maior importância. Como é possível observar, uma desvantagem do ICF é que um termo que aparecesse em todos os documentos de uma determinada classe e em um único documento de cada uma das outras teria um mesmo peso que um outro que fosse igualmente distribuído entre todas as classes. Como mostrado por How & Narayanan [2004], a fórmula para TFICF é:

$$\text{TFICF}(t_i, c_j) = N_{t_i c_j} \cdot \log\left(\frac{M}{CF_{t_i}}\right). \quad (4.9)$$

Máxima Frequência do Termo Inverso da Frequência da Classe

Calculamos o valor máximo de TFICF para todas as classes e utilizamos esse valor como uma métrica discriminatória global que relacionará mais fortemente os termos com as classes nas quais o TFICF é máximo. Logo, usamos:

$$\text{MAXTFICF}(t_i) = \text{TFICF}(t_i, c_j) \mid \text{TFICF}(t_i, c_j) > \text{TFICF}(t_i, c_k), \forall c_k \in \mathbb{C} \quad (4.10)$$

Category Term Description

Definido por How & Narayanan [2004], *Category Term Description* é uma métrica de seleção de atributos para classificação textual baseada em TFIDF e TFICF. How et al. propõe uma melhoria ao TDICF, tentando adicionar o fato que termos que aparecem em poucos documentos devem ter maior importância que os termos mais populares, pois tem maior poder discriminativo, logo:

$$\text{CDT}(t_i, c_j) = \text{TFCLASS}(t_i, c_j) \cdot \text{ICF}(t_i, c_j) \cdot \text{IDF}(t_i) \quad (4.11)$$

Máximo Category Term Description Calculamos o valor máximo de CTD para todas as classes e utilizamos esse valor como uma métrica discriminatória global que relacionará mais fortemente os termos com as classes nas quais o CTD é máximo. Logo, usamos:

$$\text{MAXCTD}(t_i) = \text{CTD}(t_i, c_j) \mid \text{CTD}(t_i, c_j) > \text{CTD}(t_i, c_k), \forall c_k \in \mathbb{C} \quad (4.12)$$

Dominância

Dominância é uma métrica originalmente proposta em Zaïane & Antonie [2002]. Utilizado exclusivamente em classificação textual, o método normaliza a frequência de um documento em uma classe por todas as classes existentes, logo:

$$\text{DOM}(t_i, c_j) = \frac{DF_{t_i c_j}}{\sum_{c_k \in \mathbb{C}} DF_{t_i c_k}} \quad (4.13)$$

Máxima Dominância

A máxima dominância é uma métrica extrapolada da Dominância. Aqui trabalhamos somente em estipular a dominância global de um termo e consideramos que ela é o valor máximo entre todas as possíveis classes. Logo,

$$\text{MAXDOM}(t_i) = \text{DOM}(t_i, c_j) \mid \text{DOM}(t_i, c_j) > \text{DOM}(t_i, c_k), \forall c_k \in \mathbb{C} \quad (4.14)$$

4.1.1.2 Métricas Modeladas Para Atributos Textuais e Categóricos.

Todas as métricas apresentadas nessa seção foram utilizadas para geração de funções de credibilidade que tratam tanto atributos textuais quanto categóricos. Elas são inspiradas em probabilidades que podem ser facilmente calculadas dos exemplos contidos no conjunto de treinamento. Destacamos que a probabilidade condicional $P(x_i|c_j)$ como a principal métrica, pois as demais, complexas ou não, são derivações dessa.

Na Tabela 4.2 está a referência quanto às variáveis usadas nessa seção.

Medida de Ambiguidade

A medida de ambiguidade (AM de *Ambiguity Measure*) foi definida por Mengle & Goharian [2008] e utilizada como um método de seleção de atributos. Ela atribui valores maiores para os atributos considerados menos ambíguos. Assim,

ela considera que um atributo não é ambíguo quando sua presença indica, com um alto grau de confiança, que o exemplo de teste pertence a uma classe específica. Podemos calcular $AM(x_i, c_j)$ como:

$$AM(x_i, c_j) = \frac{F_{x_i c_j}}{\sum_{c_k \in \mathbb{C}} F_{x_i c_k}}. \quad (4.15)$$

Como explicado, podemos usar essa métrica (e todas as demais dessa seção) com $F_{x_i c_k}$ significando o número de exemplos da classe c_k com o atributo A_i valendo x_i para problemas de classificação categórica ou a sua versão equivalente $F_{t_i c_k}$, que significa a frequência do termo t_i nos documentos da classe c_k para problemas de classificação de documentos.

Máxima Medida de Ambiguidade

Também sugerido por Mengle & Goharian [2008], o maior valor para a métrica AM dentre todas as classes pode ser utilizado como valor global discriminativo usando:

$$\text{MAXAM}(x_i) = AM(x_i, c_j) \mid AM(x_i, c_j) > AM(x_i, c_k), \forall c_k \in \mathbb{C}. \quad (4.16)$$

Probabilidade Condicional

A probabilidade condicional $P(x_i | c_j)$ advém do algoritmo *Naïve Bayes* como foi discutido na Seção 3.1. Temos dois modos de calcular $P(x_i | c_j)$, um para quando temos A_i categórico e outro para quando estamos realizando classificação textual. A ideia principal de ambos é a mesma, queremos saber a probabilidade de um atributo A_i ter o valor x_i (ou um termo t_i estar presente), dado que estamos analisando um exemplo pertencente a classe c_j .

Para classificação categórica, basta apenas contar quantas vezes A_i vale x_i para uma dada classe c_j :

$$P(x_i | c_j) = \frac{F_{x_i c_j}}{F_{c_j}} \quad (4.17)$$

Para classificação textual, contamos quantas vezes um termo t_i aparece em uma

classe em comparação a todos os termos possíveis:

$$P(t_i|c_j) = \frac{F_{t_i c_j}}{\sum_{k=1}^D F_{t_k c_j}} \quad (4.18)$$

Todas as demais métricas dessa seção estão diretamente ligadas a probabilidade condicional, e evitaremos diferenciar entre a classificação categórica e textual utilizando somente a notação $P(x_i|c_j)$, tanto para categorias quanto para textos.

Inverso da Probabilidade Condisional

Com o inverso da probabilidade condicional, calculamos a probabilidade de um atributo A_i não valer x_i para uma classe c_j . Podemos realizar esse cálculo com a seguinte fórmula:

$$P(\bar{x}_i|c_j) = 1.0 - P(x_i|c_j) \quad (4.19)$$

Índice de Gini Melhorado

O índice de Gini é uma métrica baseada na curva de Lorenz que mostra a função de distribuição acumulada de uma variável (Shang et al. [2007]). Esse índice é amplamente utilizado nas Ciências Econômicas como uma métrica para avaliação da distribuição de renda pela população de um certo país ou região. Infelizmente por essa métrica, o Brasil é um dos países mais desiguais do mundo (ver Central Intelligence Agency [2011]). Baseado na ideia de desigualdade, podemos pensar na distribuição de um atributo nas M classes possíveis. Um atributo que seja desigualmente distribuído é certamente um atributo com um maior poder de discriminação, e portanto, um atributo mais importante. O trabalho de Shang et al. [2007] criou um método de seleção de atributos para classificadores textuais baseando-se no Índice de Gini, chamado Índice de Gini Melhorado. Ao contrário da maioria das métricas expostas nessa seção, o Índice de Gini melhorado tem apenas um parâmetro: o valor do i -ésimo atributo, não levando em consideração nenhuma classe específica. Ele é considerado melhorado por algumas pequenas diferenças com o método tradicional de Gini, entre elas o fato de um maior valor se referir a um melhor atributo e não ao contrário, como é feito no método original. A fórmula sugerida por Shang et al. [2007] é dada por:

$$\text{GINI}(x_i) = \sum_{c_k \in \mathcal{C}} P(x_i|c_k)^2 \cdot P(c_k|x_i)^2 \quad (4.20)$$

Shang et al. destaca o fato da não utilização do fator $P(x_i)$, fazendo com que o Índice de Gini melhorado sofra menos influência de atributos frequentes, conseguindo capturar a capacidade de um atributo ser importante para distinguir uma classe, independente de qual classe. Destacamos que $P(c|x_i)$ é justamente a probabilidade que o algoritmo Bayesiano pretende calcular, logo aproximamos esse fator como:

$$P(c_j|x_i) = \frac{P(c_j \wedge x_i)}{P(x_i)} = \frac{\frac{F_{x_i c_j}}{\sum_{c \in \mathcal{C}} \sum_{k=1}^D F_{x_k c}}}{\frac{\sum_{c \in \mathcal{C}} F_{x_i c}}{\sum_{c \in \mathcal{C}} \sum_{k=1}^D F_{x_k c}}} = \frac{F_{x_i c_j}}{\sum_{c_k \in \mathcal{C}} F_{x_i c_k}}, \quad (4.21)$$

que é o mesmo valor definido por Mengle & Goharian [2008] para a métrica Medida da Ambiguidade mostrada anteriormente. Entretanto, Mengle et. al não explicita ou mostra nenhum cálculo de como foi feito para alcançar a fórmula da métrica AM.

Ganho de Informação

O Ganho de Informação (*Information Gain*, IG) mede a diminuição da entropia quando um atributo é usado ou não (Yang & Pedersen [1997]). A entropia é uma medida utilizada no campo da Ciência da Informação que tenta quantificar a desordem, a imprevisibilidade. Quanto maior a entropia, mais difícil é prever um resultado, portanto a métrica IG atribui valores mais elevados para os atributos que diminuam o valor da entropia, facilitando descobrir a qual classe um exemplo pertence. O Ganho da Informação pode ser calculado da seguinte forma:

$$IG(x_i, c_j) = \sum_{c \in \{c_j, \bar{c}_j\}} \sum_{x \in \{x_i, \bar{x}_i\}} P(x|c) \cdot \log_2 \frac{P(x|c)}{P(x) \cdot P(c)}. \quad (4.22)$$

Máximo Ganho de Informação

Assim como feito para outras métricas globais, definimos:

$$\text{MAXIG}(x_i) = \text{IG}(x_i, c_j) \mid \text{IG}(x_i, c_j) > \text{IG}(x_i, c_k), \forall c_k \in \mathcal{C}. \quad (4.23)$$

Cross Entropy

Cross Entropy (CE), assim como o Índice de Gini Melhorado, apresenta um atributo como parâmetro. Novamente, necessitamos aproximar o calculamos $P(c|x_i)$, pois esse já é o resultado do algoritmo *Naïve Bayes* e, portanto, não o teríamos

enquanto estamos calculando a credibilidade de um atributo em relação a uma classe. Assim como enunciado por Koller & Sahami [1997] e adaptado para seleção de atributos em Mladenić [1998], essa métrica pode mensurar a credibilidade de um atributo x_i pela seguinte fórmula:

$$\text{CE}(x_i) = P(x_i) \cdot \sum_{c \in \mathbb{C}} P(c|x_i) \cdot \log_2 \frac{P(c|x_i)}{P(c)} \quad (4.24)$$

CHI-Quadrado

O teste *CHI-quadrado* (ou χ^2) é utilizado no campo da Estatística para testar a independência entre duas variáveis aleatórias. Quando usado para seleção de atributos, tipicamente temos que as duas variáveis aleatórias são a ocorrência de um atributo A_i com valor x_i e a ocorrência de uma classe c_j , ver Zheng & Srihari [2003]. Logo,

$$\text{CHI}(x_i, c_j) = N \cdot \frac{[P(x_i|c_j) \cdot P(\bar{x}_i|\bar{c}_j) - P(x_i|\bar{c}_j) \cdot P(\bar{x}_i|c_j)]^2}{P(x_i) \cdot P(\bar{x}_i) \cdot P(c_j) \cdot P(\bar{c}_j)}. \quad (4.25)$$

Valores próximos de zero indicam a falta de relação entre x_i e c_j , enquanto valores próximos a um indicam tanto uma correlação positiva (quando ambas variáveis aumentam ou diminuem seus valores juntas), quanto uma correlação negativa (significando que uma variável aumenta seu valor enquanto a outra diminui).

Máximo CHI-Quadrado

Assim como feito para outras métricas globais, definimos:

$$\text{MAXCHI}(x_i) = \text{CHI}(x_i, c_j) \mid \text{CHI}(x_i, c_j) > \text{CHI}(x_i, c_k), \forall c_k \in \mathbb{C}. \quad (4.26)$$

Coeficiente de Correlação

O Coeficiente de Correlação (do inglês, *Correlation Coefficient*, CC) é uma métrica de seleção de atributos variante da métrica *CHI-Quadrada*. Definida por Ng et al. [1997], temos que $(CC)^2 = \chi^2$, logo:

$$\text{CC}(x_i, c_j) = \sqrt{N} \cdot \frac{P(x_i|c_j) \cdot P(\bar{x}_i|\bar{c}_j) - P(x_i|\bar{c}_j) \cdot P(\bar{x}_i|c_j)}{\sqrt{P(x_i) \cdot P(\bar{x}_i) \cdot P(c_j) \cdot P(\bar{c}_j)}}. \quad (4.27)$$

Os valores positivos para CC correspondem a pertinência do valor de um atributo a uma classe, enquanto valores negativos indicam a não pertinência. Quanto mais positivo (negativo) são os valores de CC, mais fortemente o atributo é relacio-

nado (não relacionado) a uma classe. Para fins de seleção de atributo, valores mais elevados de CC são os mais interessantes, pois mostram uma correlação positiva entre um atributo e uma classe. Em contraste com CC, χ^2 também considera importantes correlações negativas entre atributos e classes, o que acaba resultando que atributos que fortemente indicam a pertinência a uma classe são tão importantes quanto os que fortemente indicam a não pertinência.

Máximo Coeficiente de Correlação

Assim como já efetuado para outras métricas globais, definimos:

$$\text{MAXCC}(x_i) = \text{CC}(x_i, c_j) \mid \text{CC}(x_i, c_j) > \text{CC}(x_i, c_k), \forall c_k \in \mathbb{C}. \quad (4.28)$$

Coeficiente GSS

O coeficiente Galavotti–Sebastiani–Simi (GSS), introduzido por Galavotti et al. [2000], é bastante similar a χ^2 e pode ser definido como:

$$\text{GSS}(x_i, c_j) = P(x_i|c_j) \cdot P(\bar{x}_i|\bar{c}_j) - P(x_i|\bar{c}_j) \cdot P(\bar{x}_i|c_j). \quad (4.29)$$

Ele se mostra como uma forma bastante simplificada do χ^2 , levando em consideração somente parte do denominador e não utilizando o fator N , considerado dispensável pelos autores dessa métrica. Novamente temos que valores positivos correspondem à correlação de um atributo a uma categoria e, negativos, à falta de correlação.

Máximo Coeficiente GSS

Assim como já fizemos para outras métricas globais, definimos:

$$\text{MAXGSS}(x_i) = \text{GSS}(x_i, c_j) \mid \text{GSS}(x_i, c_j) > \text{GSS}(x_i, c_k), \forall c_k \in \mathbb{C}. \quad (4.30)$$

Odds Ratio

Proposta originalmente por van Rijsbergen [1979], a métrica *Odds Ratio* (OR), também é amplamente utilizada para seleção de atributos. A ideia básica é que a distribuição de atributos em exemplos relevantes é diferente da distribuição de atributos em exemplos não relevantes. Isso quer dizer que podemos definir dois eventos A e B e calculamos a probabilidade da ocorrência de A dividida pela probabilidade da não ocorrência de A e a comparamos com a probabilidade da

ocorrência de B dividida pela probabilidade da não ocorrência de B:

$$\text{OR}(A, B) = \frac{\frac{A}{1-A}}{\frac{B}{1-B}} = \frac{A \cdot (1 - B)}{B \cdot (1 - A)}. \quad (4.31)$$

Uma razão de chances de 1.0 indica que ocorrer A ou B é igualmente provável, uma razão maior do que um indica que ocorrer A é mais provável, enquanto que uma razão de chances menor do que 1 indica que o evento B tem uma probabilidade maior de ocorrer.

A razão de chances tem sido utilizada para selecionamento de atributos por Mladeníć [1998] fazendo com que A seja $P(x_i|c_j)$ e B seja $P(x_i|\bar{c}_j)$. Logo,

$$\text{OR}(x_i, c_j) = \frac{P(x_i|c_j) \cdot [1.0 - P(x_i|\bar{c}_j)]}{[1.0 - P(x_i|c_j)] \cdot P(x_i|\bar{c}_j)}. \quad (4.32)$$

Logo valores maiores que um OR indicam que uma maior chance de x_i estar relacionado com c_j , enquanto valores menores que um indicam que justamente o contrário.

Máximo Odss Ratio

Assim como feito para outras métricas globais, definimos:

$$\text{MAXOR}(x_i) = \text{OR}(x_i, c_j) \mid \text{OR}(x_i, c_j) > \text{OR}(x_i, c_k), \forall c_k \in \mathbb{C}. \quad (4.33)$$

4.1.2 Credibilidade baseada em Relacionamentos

Abordamos durante essa seção as métricas utilizadas como terminais dos indivíduos evoluídos para credibilidade baseada em relacionamentos, assim como descrito na Seção 4.1. Todas as métricas contidas aqui são amplamente utilizadas em Redes Complexas (Newman [2003]), pois são medidas que conseguem explorar bem as propriedades estruturais dos grafos modelados.

Lembramos que o primeiro passo é a construção dos grafos representando o relacionamento modelado. Como já dito, construímos um grafo para cada classe, de forma a isolar uma classe da outra. Logo depois, introduzimos o exemplo de teste, e criamos arestas dele para os exemplos de treinamento. Assim, de forma geral, calculamos para cada uma das classes o valor da métrica, $\text{MÉTRICA}(\text{teste}, \text{classe})$, que modela a credibilidade de determinada classe em relação ao exemplo de teste (estratégia *lazy*).

Várias métricas não relacionam diretamente o exemplo de teste a uma classe, mas a um exemplo de treinamento. Poderíamos usar diretamente essa relação com

uma função de credibilidade para o KNN, porém teríamos mais dificuldade com o *Naïve Bayes*. Para modelarmos todos os algoritmos de forma idêntica, optamos por considerar a credibilidade de todo um conjunto de treinamento da mesma classe. Assim, temos as mesmas métricas para o KNN e o *Naïve Bayes*, pois em nenhum momento o *Naïve Bayes* compara dois exemplos diretamente. Portanto, para as situações onde a métrica calcula algum valor entre um exemplo de teste e o do treino, agregamos esse cálculo para todos os exemplos de uma mesma classe. Logo,

$$\text{MÉTRICA}(\text{teste}, c_j) = \sum_{\text{treino} \in c_j} \text{MÉTRICA}(\text{teste}, \text{treino}), \quad (4.34)$$

resultando em somente um valor para a credibilidade dos exemplos de treino de uma classe em relação ao exemplo de teste. Por fim, mostramos a seguir as 16 métricas modeladas e exibimos na Tabela 4.3 as duas principais expressões utilizadas ao longo da descrição das métricas de credibilidade baseada em relacionamentos.

Tabela 4.3: Explicação das principais expressões utilizadas para definição das métricas para relacionamentos.

Expressão	Explicação
$adj(a)$	Conjunto de vértices conectados ao vértice a .
$dg(a)$	Número de vértices ligados ao vértice a .
$out(a)$	Número de vértices alcançáveis a partir do vértice a (grau de saída).

Tamanho da Vizinhança

Um vértice adjacente, ou seja, conectado por uma aresta é dito ser um vértice vizinho de primeira ordem. Na Figura 4.4, temos que os vértices 2, 3, 4 e 5 são vizinhos de primeira ordem do vértice 1. Seguindo esse raciocínio, o vértice 6 é vizinho de segunda ordem do vértice 1 e de terceira ordem do vértice 2 e 3. Simplesmente contar o número de vizinhos existentes, dada uma ordem de vizinhança, consiste na métrica mais simples que podemos utilizar para relacionar um vértice a um grafo. Um exemplo de teste com muitas ligações ao grafo de uma determinada classe tem mais chances de pertencer àquela classe do que outro exemplo que tem pouca ou nenhuma aresta que o conecta àquele grafo.

Repare que decidimos não levar em consideração a direção das arestas quando temos um grafo direcionado. Somente nos importamos com o número de conexões totais do teste ao grafo de uma determinada classe, independente de serem arestas de entrada ou de saída do vértice que representa o teste. Baseado nessa

teoria, criamos 3 métricas relacionadas a vizinhança, denominadas $VIZN(TESTE, CLASSE)$, onde N representa a ordem de vizinhança do teste em determinada classe.

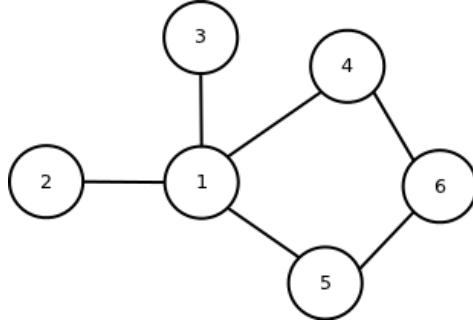


Figura 4.4: A instância de teste triângulo está ligada ao grafo formado pela classe dos círculos e dos losangos, porém não apresenta ligações com os quadrados.

Força

A métrica $FORÇA(TESTE, CLASSE)$ é uma variação do tamanho da vizinhança de primeira ordem. Quando temos um grafo no qual as arestas têm peso, essa métrica realiza a soma dos pesos das arestas vizinhas ao invés de contar o número das mesmas. O peso de uma aresta, como já dito, é uma forma de representarmos informações importantes em relação aos vértices ligados pela aresta. Em um grafo em que os vértices representam cidades e as arestas representam ligações entre essas cidades, o peso de uma aresta pode significar a distância entre duas cidades, por exemplo. Dessa forma, pode ser mais significativo saber o quanto longe (ou perto) uma cidade está de suas vizinhas do que simplesmente saber o número de vizinhas que a mesma possui. Destacamos que quando um grafo não possui pesos nas arestas, atribuímos um peso unitário para todas as arestas, logo a $FORÇA$ de um vértice seria igual ao grau de primeira ordem dele.

Proximidade

De maneira intuitiva, dizemos que dois objetos estão próximos se eles estão a uma distância arbitrariamente pequena um do outro. Muitas vezes, entretanto, não é fácil estipular o que vem a ser uma distância pequena.

Em teoria dos grafos, medimos a distância entre os vértices utilizando o algoritmo conhecido como *caminho mínimo*, que caso não leve em consideração o peso das arestas, conta o número mínimo de vértices que necessitamos atravessar para ligar conectar vértices escolhidos em um grafo. A $PROXIMIDADE(TESTE, CLASSE)$ (do inglês, *Closeness*) é uma métrica que estima o quanto próximo um vértice v está em relação a todo o grafo usando a média dos caminhos mínimos de v para todos

os outros vértices alcançáveis a partir de v (Beauchamp [1965]). Dessa forma, podemos medir a importância de um vértice calculando quanto tempo em média é gasto para uma informação se espalhar a partir de um vértice v para todo o resto do grafo. Por fim, são considerados vértices mais “próximos” aqueles que minimizam esse tempo.

Centralidade

A métrica de centralidade (em inglês, *Betweenness*) CENTRALIDADE(TESTE, CLASSE) se baseia no fato que um vértice é importante em um grafo se ele está no percurso de outros vértices, sendo obrigatoriamente muito visitado. Sendo assim, um vértice tem maior credibilidade se for usado para conectar os vários vértices em um grafo, pertencendo a vários caminhos mínimos. A métrica CENTRALIDADE calcula a importância de um vértice contando quantas vezes aquele vértice participa do caminho mínimo entre quaisquer dois outros vértices do grafo em questão (Sabidussi [1966]). Obviamente, um vértice central que está no caminho mínimo de vários outros tem mais acesso a informação que circula pelo grafo que um vértice periférico com poucas ligações.

Centralidade do Autovetor.

A centralidade do Autovetor (EIGEN(teste, classe), do inglês *Eigenvector Centrality*) é uma medida de centralidade que avalia a importância de um vértice em todo o grafo. Ela atribui valores aos vértices baseados nas conexões que os mesmos têm, sendo que um vértice ganhará uma credibilidade maior se estiver conectado a vértices de maior credibilidade. Formalmente, dado que x_i é o valor atribuído ao vértice i e que podemos montar a matriz de adjacência A com os N vértices do grafo, na qual $A_{ij} = 1$ se existe uma aresta entre i e j e $A_{ij} = 0$, caso contrário, temos:

$$x_i = \frac{1}{\lambda} \cdot \sum_{j=1}^N A_{ij} \cdot x_j. \quad (4.35)$$

Que pode ser reescrita utilizando vetores como:

$$X = \frac{1}{\lambda} AX \iff AX = \lambda X, \quad (4.36)$$

onde X é o *autovetor* formado pelos valores de x_i com $0 \leq i \leq N$ e associado ao *autovalor* λ . Podem existir muitos valores para λ para os quais a Equação 4.36 possui solução. Entretanto, se utilizarmos todos os valores do *autovetor* como

positivos, teremos o único e maior valor possível para o *autovalor* (Newman [2010]).

Hub e Autoridade de Kleinberg

As métricas conhecidas como HUB(TESTE, CLASSE) e AUTORIDADE(TESTE, CLASSE) de Kleinberg são provenientes do trabalho de Kleinberg [1999]. Elas também podem ser apresentadas em conjunto pelo nome de Algoritmo *Hyperlink-Induced Topic Search* (HITS) e por serem predecessoras do *PageRank*.

Em suma, a ideia aqui modelada se baseia no fato de que na *Web*, algumas páginas são conhecidas como *hubs* por não serem especialistas em nenhum assunto específico, mas possuírem ligações para vários outras páginas que são especialistas em seus respectivos assuntos, sendo, portanto, *autoridades* no assunto tratado. Logo, o que temos é que um bom *hub* é representado por uma página (vértice, no grafo que a *Web* representa) que aponta para várias autoridades e uma boa autoridade é aquela apontada por vários *hubs*. Uma página com poucas ligações e com poucas referências não é nem um bom *hub* e nem uma boa autoridade.

PageRank

O algoritmo *PageRank* tem seu nome proveniente do seu criador Lawrence Page (Brin & Page [1998]). Assim como o algoritmo *HITS*, Seção 4.1.2, o *PageRank* se baseia na ideia de ordenar os vértices de um grafo baseado-se nas relações entre eles, sendo que quando mais “popular” um vértice é, maior é o seu *PageRank*. Em resumo, temos que

$$PR(teste, classe) = \sum_{v \in Adj(a)} \frac{PR(v, classe)}{out(v)}. \quad (4.37)$$

Burt’s Constraint

O termo capital social é um conceito sociológico abrangente, mas em geral pode-se dizer que está relacionado a relações sociais e consiste na expectativa de benefícios derivados de relações e cooperações entre indivíduos de um grupo e entre os vários grupos existentes na sociedade modelada. Ronald Burt é um sociólogo que estudou alguns benefícios que indivíduos podem ter no mercado de trabalho proveniente das relações sociais que mantêm (Burt [1992]). Investigando as estruturas de rede do capital social, focando em indivíduos chaves em distintas organizações, ele chegou a conclusão que quem tem boas e diversificadas relações sociais consegue, entre outras coisas, melhores cargos, promoções e salários.

Outro conceito importante são os buracos estruturais, que podem ser definidos como a falta de acesso entre sub-grafos distintos que compõe um mesmo grafo. Burt propôs que vértices que conseguem preencher esses buracos estruturais, unindo vários sub-grafos, são mais importantes que aqueles vértices que estão unidos somente a um mesmo grafo ou os que estão isolados. Esse conceito é usado diretamente para calcular o *Burt's Constraint* de um vértice. Logo, BURT(teste, classe) é maior se o vértice de teste é capaz de se conectar a mais indivíduos que pertencem a partes não conectadas entre si do grafo modelado pela classe em questão.

Bibliographic Coupling

Bibliographic Coupling é uma métrica introduzida em Kessler [1963] que calcula, para dois trabalhos científicos, o número de referências em comum que ambos possuem. A ideia dessa métrica é que se dois trabalhos apresentam muitas referências em comum, então provavelmente eles abordam o mesmo assunto. Em geral, podemos definir que:

$$\text{BIB}(a, b) = |\text{Adj}(a) \cap \text{Adj}(b)|. \quad (4.38)$$

Como estamos interessados em calcular a similaridade de um vértice (exemplo de teste) com uma classe, necessitamos apenas de um valor que defina o quanto aquele vértice se assemelha aos demais. Logo, calculamos o valor do *bibliographic coupling* do exemplo de teste com todos os vértices que ele se conecta:

$$\text{BIBCOUP}(\text{teste}, c_j) = \sum_{v' \in c_j} \text{Bib}(\text{teste}, v'). \quad (4.39)$$

Co-Citação

A métrica co-citação mede, para dois vértices, o número de outros vértices que citam ambos (Small [1973]). Assim como a métrica *Bibliographic Coupling*, procuramos atribuir um valor único para um exemplo de teste e, portanto, calculamos o somatório da co-citação entre o teste e todos os vértices presentes no grafo.

Similaridade de Jaccard

A similaridade de Jaccard, ver Jaccard [1901], é uma métrica muito antiga que pode ser definida como o tamanho da interseção de dois conjuntos divididos pela união dos mesmos. Podemos definir a similaridade de Jaccard matematicamente

como:

$$\text{JAC}(a, b) = \frac{|\text{Adj}(a) \cap \text{Adj}(b)|}{|\text{Adj}(a) \cup \text{Adj}(b)|}, \quad (4.40)$$

e da mesma forma que já realizado anteriormente, calculamos o somatório dessa métrica entre o exemplo de teste e todos os vértices do grafo.

Similaridade de Dice

O coeficiente de similaridade de Dice de dois vértices é duas vezes o número de vizinhos em comum dividido pela soma de graus dos dois vértices, ver Dice [1945]. Matematicamente temos:

$$\text{DICE}(A, B) = \frac{2 \cdot |\text{adj}(a) \cap \text{adj}(b)|}{|a| + |b|}. \quad (4.41)$$

Novamente, calculamos a similaridade de Dice entre o exemplo de teste e todos os vértices do grafo, a fim de termos um único valor que represente a credibilidade do ponto de vista dessa métrica.

Similaridade de Adamic e Adar

As similaridades de Jaccard e Dice se baseiam no princípio que todos os vértices adjacentes ao vértice que analisamos são igualmente importantes. Entretanto, esse não é sempre o caso. Inspirados em uma ideia similar ao TDIDF, Adamic e Adar propuseram atribuir pesos aos vértices de maneira que um vértice com menos conexões possa ter maior poder discriminativo, ver Adamic & Adar [2003]. Dessa forma, a similaridade de Adamic e Adar entre dois vértices é o número de vizinhos que ambos têm em comum, balanceados pelo inverso do logaritmo de seus graus. Ou seja,

$$\text{AD\&AD}(A, B) = \sum_{v \in (\text{adj}(a) \cap \text{adj}(b))} \frac{1}{\log(\text{dg}(v))}. \quad (4.42)$$

Como já dito, por fim necessitamos calcular o valor de $\text{AD\&AD}(\text{TESTE}, \text{CLASSE})$ realizando o somatório da similaridade entre o teste e todos os vértices do grafo da classe em questão.

4.2 Operadores Genéticos

Em nosso trabalho, utilizamos três operadores genéticos na geração dos indivíduos: cruzamento, reprodução e mutação. Antes de discutirmos cada um desses, lembramos o

leitor que cada um dos indivíduos pode representar mais de uma função de credibilidade, uma para cada relacionamento e uma para os atributos. Sem perda de generalidade, a seguir vamos relatar como seriam aplicadas as operações genéticas em um problema em que cada indivíduo representasse apenas uma função de credibilidade, para facilitar o entendimento do leitor.

Como mostrado na Figura 4.1, os indivíduos podem ser submetidos primeiramente às operações de cruzamento ou reprodução. Os indivíduos usados nessas operações são selecionados por meio de um torneio, em que escolhemos aleatoriamente T indivíduos da população atual (parâmetro configurado pelo usuário) e dizemos que aquele com maior *fitness* é o ganhador do torneio.

A operação de reprodução é a mais simples, e insere o indivíduo ganhador do torneio na próxima geração sem realizar nenhuma modificação na sua função de credibilidade, exceto quando ele é selecionado para sofrer mutação, como veremos abaixo. Já na operação de cruzamento, realizarmos dois torneios, selecionando dois indivíduos. Depois disso, escolhemos aleatoriamente um ponto na função de credibilidade de cada um dos dois indivíduos selecionados e geramos dois novos indivíduos contendo funções com partes de ambos os pais. A Figura 4.2 ilustra esse processo para as funções de credibilidade de atributos. Os indivíduos 1 e 2 são selecionados utilizando dois torneios distintos, e em suas funções são escolhidos dois pontos para que ocorra a operação de cruzamento genético. No indivíduo 1, o ponto de troca escolhido foi a métrica $CC(x,c)$ e no indivíduo 2, a função “+”, ambos em destaque na Figura 4.2. Por fim, trocamos a métrica $CC(x,c)$ pela subárvore do vértice selecionado no indivíduo 2, gerando o indivíduo 3. Note que também é gerado um indivíduo 4 (não mostrado na figura) representando a função de credibilidade $CRED(x,c) = CC(x,c) \% IG(x,c)$.

Finalmente, a prole resultante da reprodução ou cruzamento, pode ser submetida a operação de mutação. Utilizamos a mutação de ponto, na qual o indivíduo tem uma probabilidade P_m de ter um ponto selecionado para a substituição de um terminal ou função por outro aleatório. Na Figura 4.3, vemos uma mutação ocorrendo no indivíduo 1, gerando o indivíduo 2. Note que a métrica HUB foi substituída pela métrica *PageRank* (PR).

Quando aplicamos qualquer uma dessas operações, aplicamos para cada uma das funções de credibilidade separadamente, ou seja, se estivermos aplicando uma mutação, modificaremos uma por uma das funções de credibilidade em separado, sem que elas tenham qualquer intervenção uma na outra.

4.3 Fitness

Necessitamos de um modo de avaliar os indivíduos da população a fim de sabermos quais são aqueles mais aptos a sobreviverem para a próxima geração, ou seja, os que melhor estimam a credibilidade de um exemplo. Para tanto, utilizamos a chamada função de *fitness*.

Em nosso caso, estamos criando funções de credibilidade que serão usadas para que um classificador possa criar modelos de classificação mais aprimorados. Dessa forma, nossa função de *fitness* necessita estar atrelada a uma maneira de avaliar um classificador automático. Na literatura, uma métrica muito utilizada para avaliação do desempenho de classificadores é a F_1 e, por isso, decidimos utilizá-la.

Antes de falarmos sobre a F_1 , vamos descrever o funcionamento da função de *fitness*, mostrada no Algoritmo 1, que leva em consideração funções evoluídas tanto para atributos quanto para relacionamentos.

Algorithm 1 Calula Fitness.

Função CALCULAFITNESS(*individuo*)

Credibilidade dos atributos:

Se Utilizando Credibilidade Baseada em Atributos then

Para Cada $x \in \mathbb{A}$ **Faça**

Para Cada $c \in \mathbb{C}$ **Faça**

$$f_a(x, c) \leftarrow eval(individuo_{attrs}, x, c)$$

Credibilidade dos relacionamentos:

Se Utilizando Credibilidade Baseada em Relacionamentos then

Para Cada $r \in \mathbb{R}$ **Faça**

Para Cada $e \in \mathbb{E}$ **Faça**

Para Cada $c \in \mathbb{C}$ **Faça**

$$f_r(r, e, c) \leftarrow eval(r, individuo_{rel}, e, c)$$

Avaliação da Fitness:

$$\text{fitness} \leftarrow F_1(\text{CLASSIFIER}(\mathbb{T}, \mathbb{E}, \mathbb{C}, f_a, f_r))$$

return fitness

No Algoritmo 1, vemos que existem duas partes relativas a cada uma das credibilidades e, ao final, o teste do classificador ciente da credibilidade. Na primeira parte, testamos se o problema de classificação tratado permitir a utilização da credibilidade baseada em atributos. Em caso positivo, formamos o mapeamento $f_a(x, a)$. Ele é o resultado de todas as combinações de atributos e classes possíveis em um número real

avaliado pela função *eval*. O parâmetro *individuo attrs* usado na função eval é a função de credibilidade baseada em atributos evoluída pelo indivíduo.

Na segunda parte, temos que o mesmo processo é efetuado para a credibilidade dos relacionamentos. Porém, temos um laço de repetição a mais, relativo ao fato que podem existir mais de um relacionamento sendo explorado simultaneamente. Como foi observado no Capítulo 3, aplicamos a credibilidade dos relacionamentos diretamente ao exemplo de teste, verificando quanto de credibilidade os exemplos de treinamento de cada classe têm. Portanto, o laço referente aos atributos foi trocado por um que se refere aos exemplos de teste, formando o mapa $f_r(r, e, c)$.

Utilizamos um exemplo prático para facilitar o entendimento das duas primeiras partes do cálculo da *fitness*. Como veremos, usamos em nossos experimentos a base de dados de documentos da ACM (Capítulo 5 para mais detalhes), que apresenta a possibilidade de empregarmos a credibilidade dos atributos e de dois relacionamentos: autoria e citação. Assim, um indivíduo em nosso PG seria composto de três funções de credibilidade, uma para os atributos e duas para os relacionamentos. No cálculo da *fitness*, o mapa $f_a(x, a)$ seria obtido pela avaliação de todas as combinações de atributos e classes à função de credibilidade de atributos, que poderia ser qualquer um dos indivíduos da Figura 4.2. Depois, obteríamos os mapas $f_r(citação, e, c)$ e $f_r(autoria, e, c)$ aplicando as funções de credibilidade de citação e autoria, respectivamente.

Finalmente, o último passo do Algoritmo 1 é a utilização um classificador com o conceito de credibilidade incorporado, como visto nas Seções 3.2 e 3.4, para o cálculo da métrica F_1 . O classificador recebe o conjunto \mathbb{T} de exemplos de treinamento, o conjunto \mathbb{E} de exemplos de teste, o conjunto \mathbb{C} de classes e os valores mapeados f_a e f_r de credibilidade de atributos e relacionamentos, respectivamente, e atribui para cada exemplo de \mathbb{E} uma possível classe de \mathbb{C} . Assim, baseado nos resultados do classificador, calculamos a F_1 .

Para explicar a métrica F_1 , utilizamos a Tabela 4.4. Nela, temos um cenário simplificado no qual duas classes são possíveis para um exemplo de teste, + e -, e as quatro situações podem ser geradas, VP, FP, FN ou VN. Dessa forma, VP é a situação na qual o exemplo de teste pertence a classe + e é classificado corretamente (verdadeiro positivo), FP ocorre quando o exemplo é da classe - e é classificado como + (falso positivo), FN ocorre nas vezes quando o exemplo pertence a classe + e classificado como - (falso negativo) e, finalmente, VN é quando classificamos o exemplo como - e realmente pertence a - (verdadeiro negativo).

A partir dos conceitos de VP, FP, FN e VN, podemos definir duas importantes métricas comumente utilizadas na literatura, precisão e revocação. A precisão P é

Tabela 4.4: Matriz de confusão usada para exemplificar as métricas de precisão e revocação.

	Pertence a classe +	Pertence a classe -
Classificado como +	VP	FP
Classificado como -	FN	VN

definida como:

$$P = \frac{VP}{(VP + FN)} = \frac{\# \text{de exemplos da classe c corretamente classificados como classe c}}{\# \text{ total de exemplos classificados como classe c}}, \quad (4.43)$$

e a revocação R como sendo:

$$R = \frac{VP}{(VP + FP)} = \frac{\# \text{ de exemplos da classes c corretamente classificados como classe c}}{\# \text{ de exemplos existentes na classe c}}. \quad (4.44)$$

Dessa forma, a precisão calcula o quanto um classificador acerta em uma determinada classe e a revocação mede o quanto o classificador é bom em achar os exemplos pertencentes àquela classe. Ambas métricas são bastante importantes e a média harmônica delas é utilizada para formar a medida chamada F_1 :

$$F_1 = \frac{2 \cdot P \cdot R}{(P + R)}. \quad (4.45)$$

Existem ainda duas formas derivadas da F_1 , nomeadas *micro- F_1* e *macro- F_1* . A primeira, *micro- F_1* , leva em consideração a precisão e a revocação do classificador como um todo. Portanto, o componente VP da Tabela 4.4 usado na *micro- F_1* é representado pelo número de exemplos corretamente classificados, independente de qual classe eles pertencem. Por sua vez, a *macro- F_1* realiza a média da F_1 calculada individualmente para cada uma das classes.

Dada a forma como são enunciadas, a *macro- F_1* e *micro- F_1* tendem a se diferenciar se a base de dados tem classes desbalanceadas. Em geral, uma exemplo pertencente a uma classe pouco popular é mais difícil de ser classificado que um outro pertencente a uma classe muito popular. Portanto, se estivermos analisando uma base de dados desbalanceada, a *macro- F_1* tenderá a ter um valor menor que a *micro- F_1* , pois a primeira é prejudicada pelas classes mais raras.

Por ser uma métrica amplamente mais utilizada na literatura, optamos por utilizar a *micro- F_1* como função de *fitness*. Porém sempre medimos e reportamos a *macro- F_1* . Mostramos os resultados dos vários experimentos com *micro* e *macro- F_1* , além de

uma combinação de ambas no Capítulo 5.

Capítulo 5

Experimentos

Nesse capítulo relatamos os diversos experimentos efetuados para testar a eficácia dos métodos para estimar a credibilidade de exemplos. Decidimos por realizar a divisão desse capítulo em seis partes. Na primeira, a Seção 5.1, mostramos as várias bases de dados que usamos em nossos experimentos. Depois, mostramos na Seção 5.2 os diversos parâmetros utilizados pelo PG, incluindo sua função de *fitness*. Já a Seção 5.3 aborda a metodologia usada para realização dos testes. Nas Seções 5.4 e 5.5, abordamos a classificação com a utilização de funções credibilidade em atributos textuais e categóricos, respectivamente. Os atributos textuais provêm de bases de documentos muito utilizadas na literatura, em especial a base da ACM-DL, que contém também informações usadas para credibilidade de relacionamentos. Por sua vez, os atributos categóricos vêm de bases do UCI. Finalmente, na Seção 5.6, apresentamos os resultados provenientes de utilizar a classificação de atributos numéricos e as funções de credibilidade de relacionamentos em uma base de assinaturas estruturais proteicas.

5.1 Bases de Dados

Nosso trabalho se divide em três tipos de bases de dados: as bases de documentos, as bases do UCI (Newman et al. [1998]) e uma base de bioinformática.

Iniciamos descrevendo as quatro bases de documentos: ACM-DL, *Reuters*, *Ohsumed* e *20-newsgroup*. Todas elas foram pré-processadas, com remoção de *stop words* e *stemming*, assim como foi atribuída somente uma única classe para todos os documentos que originalmente poderiam pertencer a mais de uma.

A base de documentos digitais da ACM, chamada ACM-DL (*Association for Computing Machinery Digital Library*), é um rico acervo de artigos acadêmicos da área da Ciência da Computação. Utilizamos somente um subconjunto da base, formado por

56.450 termos encontrados em 24.897 artigos divididos em 11 classes. A base da ACM é a única que apresenta informações sobre os autores e citações contidas nos seus documentos. Podemos criar dois grafos de relacionamentos com essas informações: um para os autores e outro para as citações. Ao todo o grafo de autoria tem 16.005 vértices (documentos) e 72.645 arestas, sendo que cada aresta representa o número de autores em comum dois documentos possuem. Já o grafo de citações tem 31.482 vértices e 95.812 arestas, onde os vértices são documentos e as arestas são direcionadas e significam que um documento cita o outro. Observe que o número de documentos no grafo de citações é maior que o número de artigos existentes no subconjunto da ACM-DL. Isso acontece porque estamos usando um subconjunto dos documentos da ACM-DL e ele contém informações sobre artigos que estão fora desse subconjunto. Para ser mais exato, apenas 5.305 artigos do grafo de citação estão presentes na base que usamos, enquanto os outros 26.176 não estão.

A base *Reuters*, por sua vez, contém 8.184 documentos divididos em 8 classes, e 24.986 termos distintos. Os documentos são provenientes da agência de notícias com o mesmo nome da base. Já a base *Ohsuemed* apresenta 18.302 documentos médicos divididos em 23 classes e 45.991 termos. Finalmente, a base *20-newsgroup* (*20ng*) contém 18.827 mensagens de texto com 110.502 termos únicos, enviadas para grupos de notícia de diversos assuntos como ciência, religião, entre outros, totalizando 20 classes.

A Figura 5.1 mostra a distribuição exemplos/classe das bases citadas acima. Todos os pontos mostrados na Figura 5.1 são referentes à quantidade de exemplos de cada classe, ordenados de maneira crescente, da classe de menor popularidade para a de maior. Verificamos que a base *20-newsgroup* é a que apresenta a distribuição mais equilibrada de exemplos por classe, enquanto a *Ohsuemed* apresenta a pior, com 17 das 23 classes contendo menos que 1.000 exemplos por classe e com duas classes contendo mais de 2.500.

Já a Figura 5.2, mostra o perfil de quatro bases do repositório de bases para aprendizado de máquina da Universidade da Califórnia em Irvine (UCI). Todas as bases são compostas por poucos atributos, todos categóricos, e poucas classes. A base *Cars* contém 1.728 exemplos com 6 atributos cada, apresentando características importantes para decidir a condição de um carro usado entre não aceitável, aceitável, bom e muito bom. A base *chess* utiliza 36 atributos e 3.196 instâncias para decidir se a partir de alguns movimentos finais do jogo de xadrez, o jogador que joga com as peças brancas pode ganhar ou não. *Nursery* é uma base formada por candidaturas para as escolas de enfermaria de Liubliana, Eslovênia. Ela é composta de 12.960 exemplos com 8 atributos que descrevem aspectos de um(a) candidato(a) para a escola de enfermaria. Cada exemplo pode ser classificado em cinco classes que vão desde não recomendado até for-

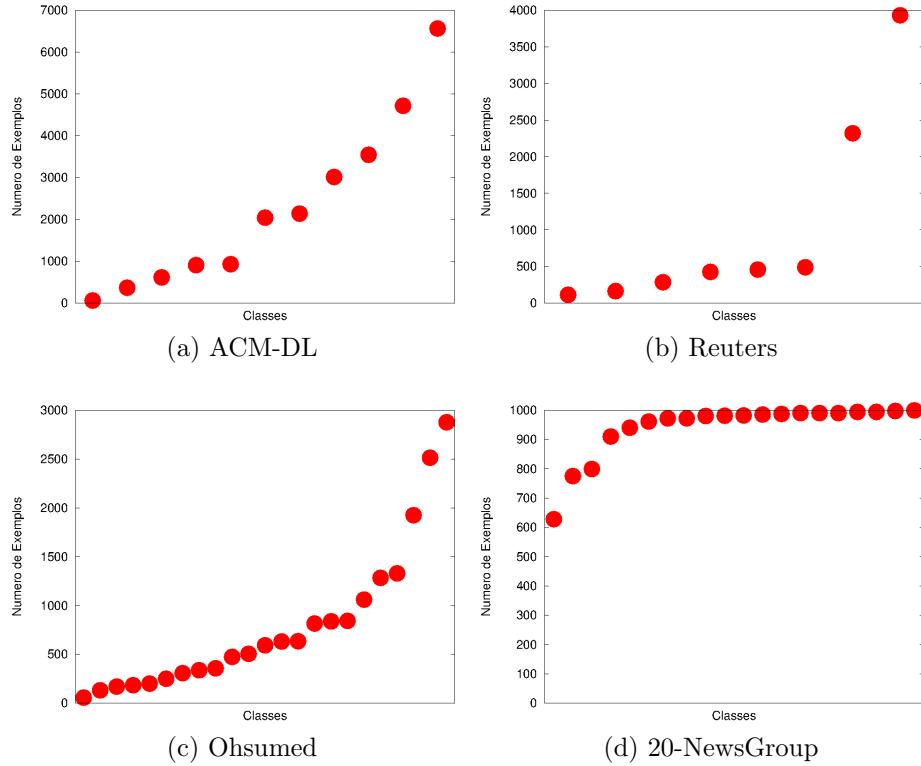


Figura 5.1: Distribuição dos exemplos nas classes das bases de documentos.

temente recomendado, com a classe recomendado apresentando apenas dois exemplos. Por último, a base *tictactoe* mostra as 958 combinações possíveis das 9 casas do jogo da velha, sendo as classes possíveis a vitória do jogador X ou não.

Finalmente, utilizamos uma base de assinaturas estruturais proteicas, geradas pelo método CSM (Pires et al. [2011]) a partir do repositório de domínios proteicos ASTRAL (Brenner et al. [2000]). Ela é utilizada para a tarefa de classificação estrutural de proteínas e usa o nível de família da classificação SCOP (Murzin et al. [1995]), que classifica proteínas nos níveis hierárquicos de classe, enovelamento, super família e família, sendo família o nível mais específico e muitas vezes o mais difícil de classificar. Assim como feito em Pires et al. [2011], foi utilizado o método de decomposição por valor singular (SVD) (Alter et al. [2000]) para reduzir a dimensionalidade e ruídos da base para apenas 15 atributos, tornando a execução dos algoritmos de classificação mais rápida sem grande degradação dos resultados. Na Figura 5.3 vemos como as 110.799 proteínas são distribuídas nas 4.193 classes existentes, sendo que todas as classes têm ao menos dez exemplos. Nesse domínio exploramos a credibilidade de relacionamentos, onde o relacionamento corresponde a similaridade entre duas sequências proteicas. Essa similaridade é gerada utilizando o método BLAST (Altschul et al. [1990]). Dessa

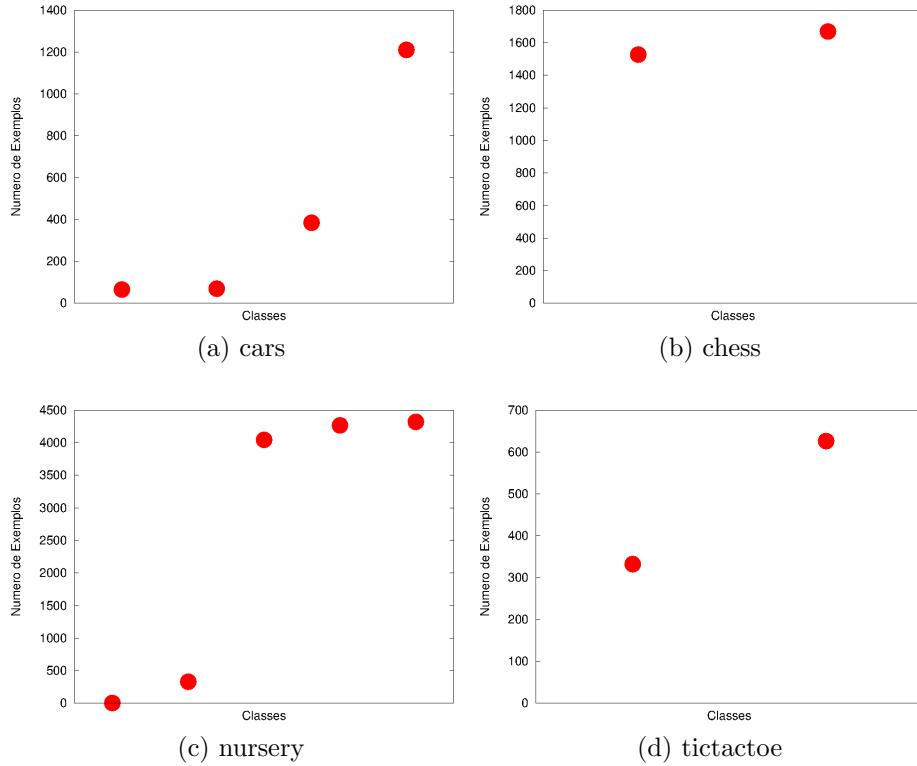


Figura 5.2: Distribuição dos exemplos nas classes das bases de atributos categóricos.

forma, definimos uma relação entre todos os pares de estruturas presentes na base. Com o intuito de utilizar somente as informações mais relevantes e diminuir o tamanho do grafo gerado, estipulamos um limite inferior de corte de 40% de similaridade. Ainda assim restaram 11.461.022 ligações entre os exemplos da base.

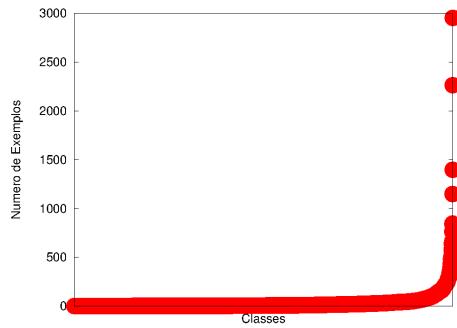


Figura 5.3: Distribuição dos exemplos nas classes na base de assinaturas estruturais proteicas.

5.2 Configuração de Parâmetros

A configuração de parâmetros em um algoritmo de Programação Genética é um dos muitos desafios encontrados nesse trabalho. Procuramos utilizar uma combinação que seja boa o suficiente para todos os testes aqui exibidos. Isso quer dizer que poderíamos obter resultados ainda melhores se ajustássemos os parâmetros focando em cada teste, porém iríamos ter dezenas de tabelas de configurações, o que definitivamente não é desejado. Portanto, efetuamos vários testes prévios em todas as bases estudadas para, finalmente, chegarmos aos parâmetros mostrados na Tabela 5.1. Destacamos o uso do programa de visualização chamado Galapagos (Brunoro et al. [2011]), que nesse ano ganhou o prêmio de melhor ferramenta para visualização de algoritmos evolucionários da conferência ACM-Gecco 2011 (*Genetics and Evolutionary Computation Conference*).

Tabela 5.1: Principais parâmetros utilizados no PG.

Parâmetro	Valor
Tamanho da População	100
Número de Gerações	100
Tipo de Seleção	Torneio
Tamanho do Torneio	2
Probabilidade de Reprodução	10%
Probabilidade de Cruzamento	90%
Probabilidade de Mutação	10%
Tamanho Máximo da Árvore	6
Tamanho Inicial Máximo	4

Como mostrado no Capítulo 4, o parâmetro “tamanho da população” controla quantos indivíduos teremos em cada uma das gerações e o “número de gerações” define até quando o PG evoluirá. Como já falamos também, utilizamos a seleção por torneios formados por apenas dois indivíduos, evitando assim a convergência prematura do algoritmo. Mostramos também os diversos valores de probabilidade para a criação da população da próxima geração.

Além disso, três importantes configurações aparecem nas últimas linhas da tabela. Na antepenúltima linha, exibimos que estamos usando a técnica chamada elitismo, na qual o melhor indivíduo de cada geração é automaticamente reproduzido na próxima geração. Finalmente, na última linha, temos o parâmetro utilizado pelo método de inicialização do PG. Ele força que metade da população tenha um tamanho inicial **igual** ao tamanho inicial máximo, ou seja, quatro, e que a outra metade tenha um tamanho inicial de **no máximo** o tamanho inicial máximo.

Além dos parâmetros convencionais apresentamos também resultados de experimentos preliminares que determinam a *fitness* do algoritmo. Como apontado na Seção 4.3, a *fitness* desempenha importante papel em um algoritmo de Programação Genética, pois define quem são os melhores indivíduos, sendo um importante meio para compará-los. Usualmente, os trabalhos de classificação presentes na literatura reportam a Micro e Macro- F_1 , pelo papel que apresentam em tentar balancear a taxa de acerto com uma boa cobertura, medindo a capacidade do classificador prever corretamente indivíduos em todas as classes.

Em nossos trabalhos passados (Palotti et al. [2010, 2011]) e nos experimentos aqui presentes, utilizamos a Micro- F_1 como função de *fitness*, pois é mais comum encontrar trabalhos na literatura reportando a Micro- F_1 do que a Macro- F_1 . Entretanto, é interessante investigar quais são os resultados obtidos por alterar a função de *fitness*, substituindo a Micro- F_1 pela (i) Macro- F_1 e (ii) pela soma das duas.

Nas Tabelas 5.2 e 5.3, mostramos os resultados da Micro- F_1 e Macro- F_1 , respectivamente, ao aplicar as três funções de *fitness* propostas. A utilização da *fitness* = Micro- F_1 serve como linha de base e informarmos os ganhos nas duas últimas colunas relativos a ela. Assim como em várias das tabelas presentes nesse capítulo, usamos 3 símbolos: \blacktriangle , \blacktriangledown , \bullet , para dizer, respectivamente, que temos uma comparação significativamente melhor, pior ou impossível de se afirmar de acordo com um teste de hipóteses (*test-t*) com nível de confiança de 99%.

Tabela 5.2: Experimentos mostrando a micro- F_1 ao variar a função de *fitness*.

Bases	Função de Fitness		
	Micro- F_1	Macro- F_1	Micro- F_1 + Macro- F_1
ACM	74.33 ± 0.72	72.96 ± 0.98 (-1.84 \blacktriangledown)	73.99 ± 0.80 (-0.45 \bullet)
20ng	89.06 ± 0.15	87.92 ± 1.59 (-0.01 \bullet)	87.65 ± 2.12 (-0.32 \bullet)
Ohsuemed	69.34 ± 0.55	68.83 ± 1.47 (-0.73 \bullet)	69.76 ± 1.19 (0.60 \bullet)
Reuters	94.60 ± 0.44	93.96 ± 0.75 (-0.67 \bullet)	94.59 ± 0.50 (-0.01 \bullet)

Tabela 5.3: Experimentos mostrando a macro- F_1 ao variar a função de *fitness*.

Bases	Função de Fitness		
	Micro- F_1	Macro- F_1	Micro- F_1 + Macro- F_1
ACM	59.72 ± 1.26	60.03 ± 1.45 (0.52 \bullet)	60.20 ± 1.54 (0.81 \bullet)
20ng	88.69 ± 0.22	86.46 ± 3.80 (-1.06 \bullet)	87.11 ± 2.47 (-0.32 \bullet)
Ohsuemed	63.56 ± 0.89	63.62 ± 1.88 (0.10 \bullet)	64.38 ± 1.91 (1.30 \bullet)
Reuters	89.33 ± 0.90	88.04 ± 0.73 (-1.44 \blacktriangledown)	89.08 ± 0.83 (-0.28 \bullet)

Comparando a *fitness* = Micro- F_1 com *fitness* = Macro- F_1 , percebemos que usar a

$fitness = \text{Macro}F_1$ obtém resultados um pouco piores (porém, somente na base ACM-DL a piora foi estatisticamente significativa) para a $\text{Micro}F_1$ e não consistentes para $\text{Macro}F_1$, apresentando uma piora significativa apenas para base *Reuters*. Esperávamos obter um melhor resultado para a $\text{Macro}F_1$ ao usá-la como função de $fitness$, mas não foi o que aconteceu. Já a utilização de uma função um pouco mais complexa, como $fitness = \text{Micro}F_1 + \text{Macro}F_1$, mostra alguns ganhos e perdas, mas nada que possa ser considerado estatisticamente significativo.

Optamos por continuar usando a $fitness = \text{Micro}F_1$ por obter resultados um pouco melhores que a $fitness = \text{Macro}F_1$ e ser mais simples que a $fitness = \text{Micro}F_1 + \text{Macro}F_1$. A última decisão foi tomada à luz do princípio da Navalha de Occam (Blumer et al. [1987]), que diz que entre um sistema mais complexo e um mais simples que obtém os mesmos resultados, devemos usar o mais simples.

Por fim, utilizando os parâmetros acima configurados, a $\text{Micro}F_1$ como função de $fitness$ e a base de testes da ACM-DL, obtemos uma curva de *Fitness* ao longo das gerações como a mostrada na Figura 5.4. Essa curva tem a mesma aparência em todas as outras bases. Em geral, verificamos que o algoritmo encontra indivíduos bons rapidamente e, ao longo das gerações, eles vão sendo refinados. Verificamos que o indivíduo da pior $fitness$ está muito aquém dos demais e que o comportamento da $fitness$ média de todos os indivíduos é relativamente estável e apresenta um aspecto convergente ao passar das gerações.

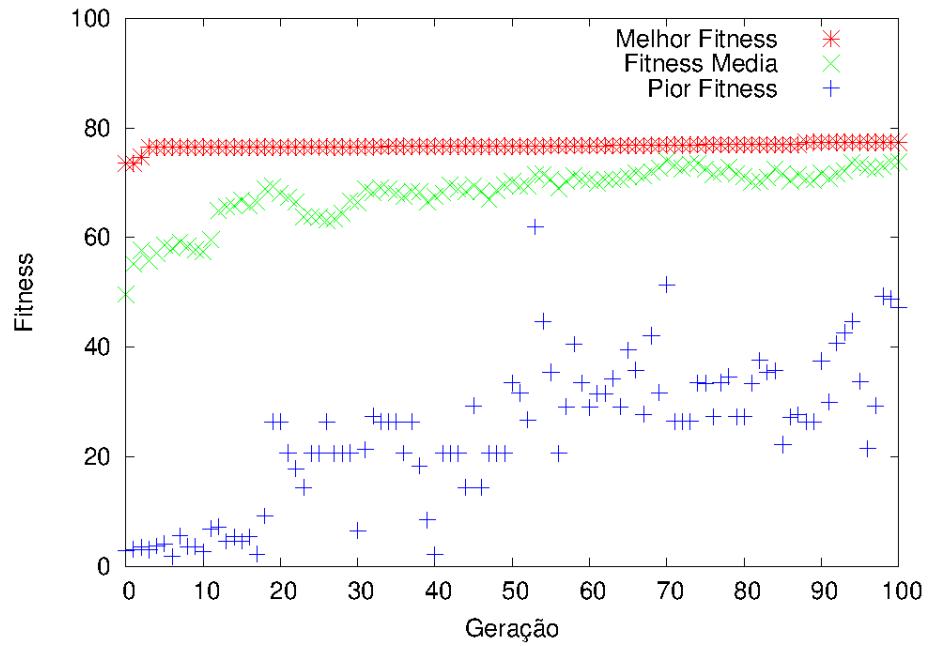


Figura 5.4: Variação da *fitness* ao longo das gerações para a base da ACM-DL.

5.3 Metodologia Experimental

Para todos os experimentos, empregamos a técnica de Validação Cruzada (Refaeilzadeh et al. [2009]) com 5 partições, com exceção dos testes na base de bioinformática, onde usamos 10 partições. A validação cruzada é bem utilizada e aceita na literatura pelo seu poder se avaliação da generalização de um modelo. Ela consiste em dividir a base de dados em k partições de tamanho igual, onde utilizamos $k - 2$ para treinar um modelo, uma para a validação e uma para o teste final. Combinamos a validação cruzada com nosso algoritmo de Programação Genética da seguinte forma: usamos o conjunto de treino para poder gerar o modelo de classificação, a validação para testar a *fitness* dos indivíduos e reservamos o teste final para ser usado somente depois que já chegamos a um indivíduo evoluído.

Além disso, uma validação cruzada de k partições dá origem a k experimentos diferentes, onde cada um possui um conjunto diferente de treino, validação e testes, resultado da rotação das partições. A Figura 5.5 ilustra bem o processo de rotação para 5 partições, onde a partir de uma divisão inicial da base por 5 partições, podemos gerar os 5 experimentos mostrados.

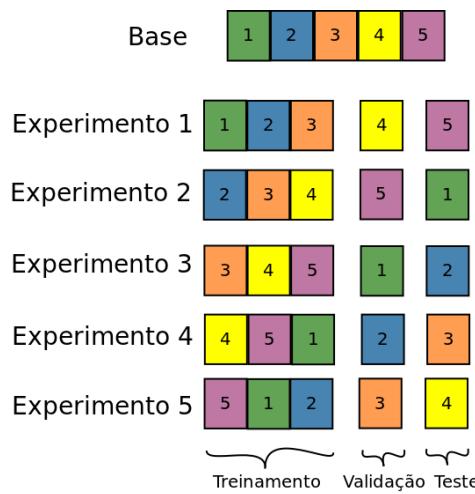


Figura 5.5: Modelo de Validação Cruzada com 5 partições.

Como dito, pelo fato da base de bioinformática ser muito grande, tivemos que mudar a forma como os experimentos estavam sendo executados. Logo, ela foi dividida em 10 partições, sendo que usamos uma partição para treino, uma para validação e as outras 8 para teste. Dessa forma, evoluímos as funções de credibilidade em apenas dois décimos da base e utilizamos a função evoluída nos outros oito décimos.

Por fim, destacamos que todos os nossos resultados apresentados são analisados através do *test-t* com nível de confiança de 99%.

5.4 Credibilidade para Bases de Documentos

Nessa seção realizamos diferentes testes com as bases de documentos. Iniciamos por mostrar na Seção 5.4.1 como o uso de Programação Genética trouxe benefícios para que conseguíssemos obter funções de credibilidade mais eficazes. Esses resultados que são válidos também para os demais experimentos mostrados nesse trabalho. Já na Seção 5.4.2 estudamos os efeitos de aplicarmos uma função de credibilidade evoluída para uma determinada base em outras bases. Por último, na Seção 5.4.3 trabalhamos com as redes de citação e autoria somadas aos atributos textuais, disponíveis somente para a base ACM-DL.

5.4.1 Credibilidade de atributos

Na literatura, encontramos muitos trabalhos, conforme abordamos na Seção 2.2, que empregam várias das métricas de credibilidade de atributos utilizadas aqui, porém sem nenhuma combinação mais elaborada das mesmas. Assim como previamente mencionamos na Seção 3.2.1 essas métricas sozinhas já poderiam ser funções de credibilidade de atributos, caso não usássemos o PG. Porém, não saberíamos dizer exatamente qual é a melhor métrica para usar em uma dada situação específica, e nem qual combinação, dentre as milhares existentes, é a melhor.

Com o intuito de mostrarmos que a combinação das métricas durante e evolução do PG é benéfica para a criação de uma função de credibilidade mais robusta e evitarmos a escolha de uma linha de base sem nenhuma informação adicional sobre as métricas, construímos as Tabelas 5.4 e 5.5. Nelas, visualizamos a aplicação das diversas métricas de credibilidade nas quatro bases de texto que estudamos. Mostramos a linha de base na primeira linha, que trata do algoritmo *Naïve Bayes* sem modificação, o SVM usado com o pacote *SVMLight* (Joachims [1999]) na segunda linha, e as demais métricas nas outras linhas. Entre parêntesis, temos a diferença da aplicação de uma métrica (ou PG) para a linha de base. Vale ressaltar que utilizamos a função de *kernel* linear para o SVM, que, como mostrado em Salles et al. [2010], é a melhor para quando temos uma alta dimensionalidade e tendência que as classes sejam linearmente separáveis, como acontece em bases de texto.

Vemos que usar o PG fornece uma solução bem mais robusta que usar qualquer um das outras métricas em separado. O resultado do PG da Micro F_1 da base ACM-DL foi o único que não trouxe resultados significativamente melhores, porém também não foram piores, como todas as outras métricas. Observe que, nesse caso, o PG utilizou como terminais as trinta métricas para evolução dos atributos.

Os resultados com o SVM confirmam o que foi exibido em Salles et al. [2010], no qual o SVM apresenta resultados semelhantes aos do algoritmo *Naïve Bayes*. É interessante destacar que nos cenários onde o SVM foi superior ao *Naïve Bayes*, a utilização do PG fez com que essa diferença fosse nitidamente diminuída. Por exemplo, na Macro- F_1 da base *Reuters*, onde os resultados do SVM não são estatisticamente superiores aos do PG.

Em geral, as métricas apresentaram resultados positivos em alguns cenários, mas negativos em outros, como a métrica IDF ou GSS. Mesmo as métricas que apresentaram resultados negativos em todos os experimentos, como MAXTFIDF, MAXCTD, foram mantidas como terminais para o algoritmo do PG por duas razões: (i) não temos certeza se existe um cenário no qual elas possam trazer benefícios, como usar uma outra base ou mesmo usar o inverso da métrica ($\frac{1}{MaxTFIDF}$), (ii) o PG naturalmente vai eliminar métricas que não trazem benefício ao longo das gerações.

Por fim, realizamos experimentos semelhantes para funções de credibilidade de relacionamentos na base da ACM-DL, obtendo o mesmo tipo de resultados que mostramos aqui, onde a função evoluída pelo GP é sempre melhor ou igual à aplicação das métricas em separado.

5.4.2 Generalidade

Na Seção 5.4.1, analisamos os resultados obtidos ao aplicar todas as métricas de credibilidade de atributos em separado e através do uso de PG. Como mencionamos na Seção 5.3, usamos validação cruzada com 5 partições para obter os resultados mostrados na Tabelas 5.4 e 5.5. Logo, para cada uma das bases, tivemos cinco funções evoluídas pelo uso do PG, uma para cada rotação, como mostrado na Figura 5.5. Cada uma dessas funções obteve um resultado em seu experimento e exibimos, ao final, os resultados médios das cinco funções e desvio padrão. Decidimos por escolher a função que obteve o melhor resultado em seu experimento como representante da base de dados e mostramos essas funções na Tabela 5.6. Assim, por exemplo, a F_{ACM-DL} mostrada foi a função escolhida para a base ACM-DL por ter obtido maiores ganhos que as outras quatro funções evoluídas.

Resolvemos então aplicar as melhores funções evoluídas para cada base nas demais bases (e nos outros experimentos de uma mesma base também). Assim podemos testar o quanto uma função de credibilidade evoluída para uma determinada base é geral e pode ser usada em outro contexto. Nas Tabelas 5.7 e 5.8, observamos os resultados do estudo da generalização das funções. Se analisarmos as tabelas na linha diagonal central, veremos a aplicação das funções da Tabela 5.6 em toda a base, não mais

Tabela 5.4: Avaliação da Micro F_1 com a aplicação de diversas métricas estudadas e o PG.

Métrica	ACM-DL	20ng	Ohsumed	Reuters
Linha de Base	73.63 ± 2.02	84.94 ± 0.58	66.56 ± 0.66	93.13 ± 0.29
SVM	71.02 ± 0.77	80.03 ± 0.33	70.39 ± 1.08	96.19 ± 0.33
PG	74.33 ± 0.72 (0.95 ●)	89.06 ± 0.15 (4.85 ▲)	69.34 ± 0.55 (4.19 ▲)	94.60 ± 0.44 (1.57 ▲)
TF(t)	66.16 ± 5.91 (-10.14 ▼)	85.44 ± 0.69 (0.68 ●)	64.73 ± 3.16 (-2.74 ●)	86.32 ± 9.91 (-7.31 ●)
TFClasse(t,c)	66.93 ± 0.85 (-9.09 ▼)	86.74 ± 0.39 (2.21 ▲)	67.06 ± 0.55 (0.76 ●)	92.80 ± 0.75 (-0.35 ●)
DF(t)	66.81 ± 0.64 (-9.26 ▼)	87.53 ± 0.28 (3.15 ▲)	67.05 ± 0.71 (0.75 ●)	92.79 ± 0.77 (-0.37 ●)
DFClasse(t,c)	66.99 ± 0.82 (-9.02 ▼)	87.80 ± 0.31 (3.47 ▲)	67.07 ± 0.55 (0.78 ●)	92.80 ± 0.75 (-0.35 ●)
IDF(t)	69.60 ± 0.79 (-5.47 ▼)	86.64 ± 0.43 (2.10 ▲)	60.43 ± 0.72 (-9.21 ▼)	94.29 ± 0.35 (1.25 ▲)
IDFClaße(t,c)	67.07 ± 0.77 (-8.91 ▼)	21.44 ± 32.33 (-74.73 ▼)	67.28 ± 0.42 (1.09 ▲)	92.74 ± 0.46 (-0.42 ●)
TFIDF(t,c)	72.84 ± 1.05 (-1.06 ▼)	85.04 ± 0.30 (0.21 ●)	66.38 ± 0.70 (-0.27 ●)	93.77 ± 0.36 (0.68 ▲)
MaxTFIDF(t)	65.40 ± 0.51 (-11.18 ▼)	81.25 ± 1.97 (-4.26 ▼)	62.56 ± 0.49 (-6.00 ▼)	87.60 ± 1.07 (-5.94 ●)
TFICF(t,c)	58.75 ± 1.83 (-20.20 ▼)	84.99 ± 0.31 (0.16 ●)	60.43 ± 0.72 (-9.21 ▼)	72.41 ± 1.73 (-22.25 ▼)
MaxTFICF(t)	23.81 ± 1.29 (-67.66 ▼)	80.23 ± 1.62 (-5.45 ▼)	34.37 ± 0.40 (-48.35 ▼)	24.28 ± 1.02 (-73.93 ●)
CTD(t,c)	58.75 ± 1.83 (-20.20 ▼)	84.99 ± 0.31 (0.16 ●)	60.43 ± 0.72 (-9.21 ▼)	72.41 ± 1.73 (-22.25 ▼)
MaxCTD(t)	37.69 ± 2.08 (-48.81 ▼)	80.14 ± 0.45 (-5.56 ▼)	48.34 ± 0.80 (-27.36 ▼)	61.78 ± 1.59 (-33.67 ▼)
DOM(t,c)	73.38 ± 0.97 (-0.33 ▼)	84.65 ± 0.37 (-0.24 ●)	64.15 ± 0.59 (-3.62 ▼)	93.43 ± 0.38 (0.31 ●)
MaxDom(t)	72.25 ± 0.70 (-1.87 ▼)	85.66 ± 0.31 (0.94 ▲)	67.99 ± 0.58 (2.15 ▲)	92.73 ± 0.64 (-0.43 ●)
AM(t,c)	73.13 ± 1.05 (-0.68 ▼)	81.98 ± 0.48 (-1.96 ▼)	64.21 ± 0.60 (-3.52 ▼)	93.43 ± 0.38 (0.31 ●)
MaxAM(t)	72.17 ± 0.67 (-1.97 ▼)	85.36 ± 0.18 (0.59 ●)	67.97 ± 0.61 (2.12 ▲)	92.73 ± 0.64 (-0.43 ●)
P(t c)	72.96 ± 1.06 (-0.90 ▼)	84.93 ± 1.32 (0.08 ●)	63.33 ± 1.09 (-4.84 ▼)	93.32 ± 0.38 (0.20 ●)
P(̄t c)	63.43 ± 0.90 (-13.85 ▼)	86.22 ± 0.29 (1.60 ▲)	62.51 ± 0.79 (-6.08 ▼)	92.84 ± 0.43 (-0.32 ●)
GINI(t)	71.35 ± 0.77 (-3.10 ▼)	84.81 ± 0.15 (-0.06 ●)	68.82 ± 0.47 (3.41 ▲)	92.45 ± 0.60 (-0.73 ●)
IG(t,c)	71.31 ± 0.65 (-3.15 ▼)	86.85 ± 0.34 (2.34 ▲)	68.18 ± 0.58 (2.45 ▲)	94.04 ± 0.46 (0.97 ▲)
MaxIG(t)	68.49 ± 0.60 (-6.98 ▼)	87.62 ± 0.34 (3.25 ▲)	67.74 ± 0.66 (1.77 ▲)	93.61 ± 0.70 (0.51 ●)
CE(t)	44.65 ± 0.34 (-39.35 ▼)	54.16 ± 1.48 (-36.17 ▼)	31.86 ± 0.13 (-52.13 ▼)	76.72 ± 0.44 (-17.67 ▼)
CHI(t,c)	69.86 ± 0.74 (-5.12 ▼)	87.23 ± 0.39 (2.79 ▲)	67.99 ± 0.54 (2.16 ▲)	93.76 ± 0.54 (0.67 ▲)
MaxCHI(t)	68.41 ± 0.58 (-7.09 ▼)	87.55 ± 0.29 (3.17 ▲)	67.35 ± 0.64 (1.19 ▲)	93.22 ± 0.81 (0.09 ●)
CC(t,c)	69.69 ± 0.70 (-5.35 ▼)	72.19 ± 7.43 (-14.93 ▼)	67.96 ± 0.54 (2.11 ▲)	93.58 ± 0.55 (0.66 ●)
MaxCC(t)	1.49 ± 0.05 (-97.98 ▼)	5.29 ± 0.02 (-93.77 ▼)	7.26 ± 0.01 (-89.10 ▼)	5.22 ± 0.05 (-94.40 ▼)
GSS(t,c)	69.65 ± 0.71 (-5.40 ▼)	72.19 ± 7.43 (-14.93 ▼)	67.95 ± 0.56 (2.10 ▲)	93.77 ± 0.54 (0.68 ▲)
MaxGSS(t)	1.48 ± 0.06 (-97.99 ▼)	5.29 ± 0.02 (-93.77 ▼)	7.26 ± 0.01 (-89.10 ▼)	5.19 ± 0.04 (-94.42 ▼)
OR(t,c)	68.54 ± 0.95 (-6.91 ▼)	84.77 ± 0.19 (-0.11 ●)	65.59 ± 0.84 (-1.45 ●)	92.42 ± 0.59 (-0.76 ▼)
MaxOR(t)	48.15 ± 0.43 (-34.60 ▼)	65.20 ± 0.71 (-23.17 ▼)	34.75 ± 0.14 (-47.79 ▼)	79.99 ± 0.27 (-14.12 ▼)

em apenas um dos cinco experimentos da validação cruzada. Observe que todos os resultados são tão bons ou melhores que os reportados anteriormente na Seção 5.4.1. Em especial, esse foi o único resultado que apresentou ganhos significativos para base da ACM-DL.

Ao observarmos as Tabelas 5.7 e 5.8 coluna a coluna, vemos o comportamento de cada uma das funções em cada uma das bases. É nítido que nenhuma das funções evoluídas nas outras bases foi boa para a base da ACM-DL, e nem a função evoluída pela ACM-DL obteve resultados bons para as outras bases. Acreditamos que isso é devido ao fato da proximidade existente entre as classes que pertencem à base da ACM-DL, i. e., enquanto as outras bases apresentam grandes diferenças semânticas entre as diversas classes, a base da ACM-DL apresenta classes muito mais semelhantes e, portanto, mais difíceis de serem separadas. Por isso, existe uma dificuldade maior em

Tabela 5.5: Avaliação da Macro F_1 com a aplicação de diversas métricas estudadas e o PG.

Métricas	ACM-DL	20ng	Ohsumed	Reuters
Linha de Base	57.26 ± 2.08	83.68 ± 0.82	54.76 ± 1.27	81.96 ± 1.44
SVM	59.44 ± 0.44	79.83 ± 0.33	64.72 ± 1.51	91.92 ± 1.21
PG	59.72 ± 1.26 (4.30 ▲)	88.69 ± 0.22 (5.99 ▲)	63.56 ± 0.89 (16.06 ▲)	89.33 ± 0.90 (8.99 ▲)
TF(t)	54.89 ± 3.33 (-4.14 ●)	83.12 ± 2.08 (-0.59 ●)	58.48 ± 2.76 (6.80 ▲)	79.64 ± 9.43 (-2.83 ●)
TFClasse(t,c)	55.57 ± 0.72 (-2.95 ▼)	85.66 ± 0.55 (2.44 ▲)	61.66 ± 1.36 (12.59 ▲)	87.56 ± 1.55 (6.84 ▲)
DF(t)	55.83 ± 0.57 (-2.50 ▼)	86.85 ± 0.39 (3.86 ▲)	61.43 ± 1.49 (12.18 ▲)	87.24 ± 0.98 (6.44 ▲)
DFClasse(t,c)	55.60 ± 0.70 (-2.89 ▼)	87.22 ± 0.38 (4.30 ▲)	61.71 ± 1.41 (12.69 ▲)	87.56 ± 1.55 (6.84 ▲)
IDF(t)	56.84 ± 0.80 (-0.73 ▼)	85.58 ± 0.57 (2.35 ▲)	57.07 ± 0.52 (4.22 ▲)	89.01 ± 0.60 (8.61 ▲)
IDFClasse(t,c)	54.88 ± 0.61 (-4.15 ▼)	17.43 ± 33.83 (-79.16 ▼)	61.50 ± 1.36 (12.31 ▲)	85.94 ± 0.85 (4.86 ▲)
TFIDF(t,c)	58.33 ± 1.08 (1.87 ▲)	83.85 ± 0.40 (0.28 ●)	56.37 ± 1.29 (2.94 ▲)	85.25 ± 0.64 (4.01 ▲)
MaxTFIDF(t)	54.33 ± 0.41 (-5.12 ▼)	80.24 ± 1.51 (-4.05 ▼)	59.62 ± 1.14 (8.87 ▲)	82.65 ± 1.74 (0.84 ●)
TFICF(t,c)	50.63 ± 0.97 (-11.58 ▼)	80.60 ± 1.52 (-3.61 ▼)	57.07 ± 0.52 (4.22 ▲)	66.44 ± 1.50 (-18.93 ▼)
MaxTFICF(t)	28.94 ± 0.93 (-49.46 ▼)	76.26 ± 0.18 (-8.81 ▼)	38.85 ± 0.88 (-29.05 ▼)	32.38 ± 1.54 (-60.49 ●)
CTD(t,c)	50.63 ± 0.97 (-11.58 ▼)	80.60 ± 1.52 (-3.61 ▼)	57.07 ± 0.52 (4.22 ▲)	66.44 ± 1.50 (-18.93 ▼)
MaxCTD(t)	38.29 ± 1.43 (-33.13 ▼)	77.38 ± 1.13 (-7.47 ▼)	48.40 ± 0.94 (-11.62 ▼)	54.09 ± 0.96 (-34.01 ▼)
DOM(t,c)	58.34 ± 1.62 (1.89 ●)	83.21 ± 0.55 (-0.49 ▼)	51.77 ± 1.00 (-5.47 ▼)	85.28 ± 0.87 (4.06 ▲)
MaxDom(t)	58.74 ± 1.27 (2.58 ▲)	84.31 ± 0.52 (0.82 ●)	62.60 ± 0.83 (14.32 ▲)	87.07 ± 0.95 (6.23 ▲)
AM(t,c)	58.02 ± 1.73 (1.33 ●)	83.78 ± 0.27 (-1.27 ▼)	52.21 ± 1.12 (-4.65 ▼)	85.28 ± 0.87 (4.06 ▲)
MaxAM(t)	58.64 ± 1.20 (2.41 ▲)	83.68 ± 0.27 (0.07 ●)	62.93 ± 1.03 (14.91 ▲)	87.07 ± 0.95 (6.23 ▲)
P(t,c)	57.51 ± 1.54 (0.44 ●)	82.62 ± 1.55 (-1.20 ●)	54.20 ± 2.53 (-1.02 ●)	84.92 ± 0.81 (3.61 ▲)
P(\bar{t})c	52.25 ± 0.90 (-8.75 ▼)	83.42 ± 1.83 (-0.24 ●)	56.36 ± 1.39 (2.92 ▲)	87.42 ± 0.40 (6.66 ▲)
GINI(t)	58.11 ± 1.00 (1.49 ▲)	83.10 ± 0.25 (-0.62 ●)	63.29 ± 1.25 (15.58 ▲)	86.58 ± 0.95 (5.63 ▲)
IG(t,c)	58.06 ± 1.04 (1.41 ▲)	85.89 ± 0.43 (2.71 ▲)	62.91 ± 1.44 (14.89 ▲)	88.88 ± 0.93 (8.44 ▲)
MaxIG(t)	56.42 ± 0.63 (-1.47 ●)	87.07 ± 0.34 (4.12 ▲)	62.24 ± 1.53 (13.66 ▲)	88.05 ± 1.01 (7.43 ▲)
CE(t)	22.94 ± 0.41 (-59.93 ▼)	50.23 ± 1.40 (-39.93 ▼)	8.78 ± 0.11 (-83.96 ▼)	31.55 ± 1.48 (-61.73 ▼)
CHI(t,c)	56.81 ± 1.00 (-0.78 ●)	86.54 ± 0.46 (3.49 ▲)	62.59 ± 1.41 (14.30 ▲)	88.31 ± 0.95 (7.75 ▲)
MaxCHI(t)	56.22 ± 0.64 (-1.8 ▼)	87.01 ± 0.34 (4.05 ▲)	61.67 ± 1.50 (12.61 ▲)	87.43 ± 1.22 (6.68 ▲)
CC(t,c)	56.93 ± 0.90 (-0.57 ●)	74.52 ± 5.89 (-10.89 ▼)	62.57 ± 1.37 (14.25 ▲)	88.56 ± 0.02 (7.31 ▲)
MaxCC(t)	0.27 ± 0.01 (-99.52 ▼)	0.53 ± 0.02 (-99.36 ▼)	0.59 ± 0.00 (-98.93 ▼)	1.32 ± 0.18 (-98.39 ▼)
GSS(t,c)	56.90 ± 0.91 (-0.62 ●)	74.52 ± 5.89 (-10.89 ▼)	62.57 ± 1.39 (14.25 ▲)	88.43 ± 0.91 (7.90 ▲)
MaxGSS(t)	0.26 ± 0.01 (-99.54 ▼)	0.53 ± 0.02 (-99.36 ▼)	0.59 ± 0.00 (-98.93 ▼)	1.23 ± 0.01 (-98.49 ▼)
OR(t,c)	56.80 ± 1.22 (-0.80 ●)	83.20 ± 0.39 (-0.51 ●)	60.67 ± 1.55 (10.79 ▲)	86.67 ± 0.56 (5.75 ▲)
MaxOR(t)	28.15 ± 0.74 (-50.83 ▼)	62.09 ± 0.51 (-25.75 ▼)	12.21 ± 0.28 (-77.70 ▼)	40.94 ± 1.06 (-50.04 ▼)

Tabela 5.6: Funções de credibilidade geradas para cada uma das bases.

Base	Função de Credibilidade
ACM-DL	$F_{ACM-DL} = DOM^{(GSS)^{(CE+TF)}}$
20ng	$F_{20ng} = DF + MaxAM + \left(\frac{CHI}{TFIDF^{(MaxTFIDF)}}\right)$
Ohsumed	$F_{Ohsumed} = \left(\frac{AM}{MaxIG \times CC^{(sumDF)}}\right)^{(TFICF)^{(TFICF)}}$
Reuters	$F_{Reuters} = IG^{(MaxIG \times GINI)}$

encontrar uma função que capture melhor esse fato e, assim, a base da ACM-DL não generaliza bem. Verificamos também, através das tabelas, que todas funções tiveram êxito na melhora da Macro F_1 das bases *Ohsumed* e *Reuters*, mas o mesmo não ocorreu na Micro F_1 . Finalmente, destacamos que a única função que obteve resultados bons nas demais bases (exceto na Micro F_1 da ACM) foi a função evoluída para a base *20ng*.

Sendo assim, concluímos que usar as funções evoluídas pelo PG se comportam muito bem na própria base, mas não podemos dizer o mesmo para as demais bases.

Por fim, uma análise interessante pode ser feita usando as funções mostradas na Tabela 5.6 e verificando os resultados obtidos por utilizar cada uma delas individualmente como feito nas Tabelas 5.4 e 5.5. Por exemplo para a base ACM-DL, as métricas DOM, GSS, CE e TF foram combinadas usando as operações de potência e soma resultando em um ganho estatisticamente significativo de 1.91% para a $\text{Micro}F_1$, enquanto essas métricas isoladas obtiveram perdas significativas em relação à linha de base, com um máximo de -39.35% da métrica CE.

Tabela 5.7: $\text{Micro}F_1$ obtida pelo *Naïve Bayes* quando usando a função de credibilidade F_{base} gerada uma dada base nas demais bases.

	ACM-DL	20ng	Ohsmed	Reuters
Linha de Base	73.63 ± 0.90	84.86 ± 0.54	66.56 ± 0.66	93.13 ± 0.29
F_{ACM-DL}	75.04 ± 0.88 (1.91 \blacktriangle)	84.45 ± 0.35 (-0.49 \blacktriangledown)	66.06 ± 0.59 (-0.75 \bullet)	92.68 ± 0.45 (-0.48 \blacktriangledown)
F_{20ng}	66.69 ± 0.65 (-9.42 \blacktriangledown)	88.97 ± 0.17 (4.84 \blacktriangle)	67.94 ± 0.62 (2.08 \blacktriangle)	94.01 ± 0.53 (0.94 \blacktriangle)
F_{Ohsmed}	70.09 ± 0.82 (-4.81 \blacktriangledown)	85.60 ± 0.27 (0.87 \blacktriangle)	69.54 ± 0.69 (4.49 \blacktriangle)	93.45 ± 0.57 (0.34 \bullet)
$F_{Reuters}$	66.55 ± 0.89 (-9.61 \blacktriangledown)	87.04 ± 0.41 (2.57 \blacktriangle)	63.32 ± 0.82 (-4.86 \blacktriangledown)	94.87 ± 0.25 (1.86 \blacktriangle)

Tabela 5.8: Macro F_1 obtida pelo *Naïve Bayes* quando usando a função de credibilidade F_{base} gerada uma dada base nas demais bases.

	ACM-DL	20ng	Ohsmed	Reuters
Linha de Base	57.26 ± 0.93	83.62 ± 0.74	54.76 ± 1.27	81.96 ± 1.44
F_{ACM-DL}	60.13 ± 1.44 (5.02 \blacktriangle)	82.88 ± 0.51 (-0.89 \blacktriangledown)	56.41 ± 0.92 (3.00 \blacktriangle)	85.49 ± 0.94 (4.31 \blacktriangle)
F_{20ng}	55.83 ± 0.66 (-2.50 \blacktriangledown)	88.63 ± 0.18 (5.99 \blacktriangle)	62.12 ± 1.41 (13.43 \blacktriangle)	88.64 ± 1.03 (8.16 \blacktriangle)
F_{Ohsmed}	57.59 ± 0.56 (0.58 \bullet)	84.26 ± 0.49 (0.77 \blacktriangle)	64.35 ± 1.17 (17.51 \blacktriangle)	87.41 ± 0.64 (6.65 \blacktriangle)
$F_{Reuters}$	54.84 ± 1.44 (-4.23 \blacktriangledown)	83.15 ± 2.04 (-0.56 \bullet)	57.97 ± 1.59 (5.85 \blacktriangle)	89.45 ± 0.74 (9.14 \blacktriangle)

5.4.3 Explorando Relacionamentos e Múltiplos Fatores

Nessa seção avaliamos como o uso de mais de um fator pode tornar o cálculo da credibilidade mais preciso. Somente a base da ACM apresenta informações disponíveis sobre os autores e as citações contidas em seus documentos. Portanto, essa foi a única base que pudemos realizar experimentos com mais de um fator.

Na Tabela 5.9, mostramos a aplicação dos fatores isoladamente e a combinação dos mesmos. A primeira coluna, termos, se refere a credibilidade dos atributos e as segunda e terceira colunas, à credibilidade dos relacionamentos. Vemos que simplesmente usar a rede de autoria ou citações provê benefícios imediatos que não foram alcançados ao explorarmos os atributos.

Esse fato evidencia o poder da classificação relacional, e que podemos obter ganhos acentuados ao explorar a credibilidade dos relacionamentos. Com única exceção da $\text{Micro}F_1$ ao aplicar os três fatores, observamos que quanto mais fatores envolvemos na credibilidade, melhores são os resultados. Por exemplo, ao usar somente a citação, temos ganhos de 3.4% na $\text{Micro}F_1$ e 5.43% na $\text{Macro}F_1$, ao combinar a citação com os termos, os ganhos vão para 4.18% e 7.96%. Acreditamos que não obtemos um resultado ainda mais expressivo para a combinação dos três fatores pelo fato que o número de métricas é muito elevado (30 dos atributos e 16 para cada um dos relacionamentos) e os parâmetros usados no PG não foram otimizados para esse caso.

Tabela 5.9: Resultados do uso de credibilidade na base da ACM-DL explorando a credibilidade de termos, autoria e citação.

Fatores			$\text{Micro}F_1$	$\text{Macro}F_1$
Termos	Autoria	Citação		
Linha de Base			73.63 ± 0.91	57.26 ± 0.93
X			74.33 ± 0.72 (0.95 ●)	59.72 ± 1.26 (4.30 ▲)
	X		76.19 ± 0.82 (3.48 ▲)	60.37 ± 0.70 (5.43 ▲)
		X	75.58 ± 0.85 (2.65 ▲)	59.00 ± 0.90 (3.04 ▲)
X	X		76.70 ± 1.08 (4.18 ▲)	61.82 ± 1.38 (7.96 ▲)
X		X	75.63 ± 1.03 (2.72 ▲)	60.69 ± 1.30 (5.98 ▲)
	X	X	77.29 ± 0.79 (4.98 ▲)	61.41 ± 0.65 (7.25 ▲)
X	X	X	77.01 ± 0.98 (4.60 ▲)	62.24 ± 1.01 (8.70 ▲)

5.5 Credibilidade em Bases de Atributos Categóricos

Na Seção 2.2, vimos que vários trabalhos lidam com o uso de diversas das métricas que utilizamos para gerar funções de credibilidade de atributos. Um ponto comum a todos é que se preocupam somente com classificação de documentos, ou seja, os atributos são sempre textuais. Expandimos essa abordagem ao lidar também com atributos categóricos.

Nas Tabelas 5.10 e 5.11, mostramos os resultados de evoluir funções de credibilidade para as bases usando tanto o algoritmo de classificação *Naïve Bayes* quanto o KNN. Dessa vez, mostramos os resultados com o KNN pelo fato de sua linha de base ser muito superior à do *Naïve Bayes* (excepto para a $\text{Micro}F_1$ da base *Nursery*). A título de informação, utilizamos $K = 6, 9, 10, 8$ para as bases *Cars*, *Tictac*, *Nursery* e *Chess*, respectivamente. Esses valores foram definidos em experimentos preliminares.

Destacamos os ganhos mais interessantes na $\text{Macro}F_1$ da base *Cars*. Esse fato evidencia que os exemplos de maior dificuldade, por serem de classes menos popula-

res, foram corretamente classificados. Isso alterou muito pouco a $\text{Micro}F_1$, pois, em quantidade, os exemplos difíceis são poucos, mas alterou muito a $\text{Macro}F_1$, pois existiram aumentos substanciais na métrica F_1 das classes com menos exemplos. O mesmo ocorre na base *Nursery* quando usamos o *Naïve Bayes*. Já as bases *TicTacToe* e *Chess* apresentaram ganhos pequenos, inferiores a 2%.

Uma análise das funções evoluídas em todas as bases (não mostramos as funções pelo fato delas serem 40 distintas (5 por base, 4 bases, 2 algoritmos)), mostra que as métricas AM, CHI, GINI e IG são as mais frequentes, sendo que ao menos uma delas aparece em todas as funções. Por outro lado, as métricas que calculam máximos, MAXAM, MAXCC, entre outras, não apareceram. Isso pode mostrar que métricas globais são menos efetivas para métrica de credibilidade de atributos categóricos.

Tabela 5.10: Resultados da $\text{Micro}F_1$ para as bases de atributos categóricos

Bases	Naïve Bayes		KNN	
	Linha de Base	Micro F_1	Linha de Base	Micro F_1
<i>Cars</i>	86.32 ± 1.85	$87.64 \pm 2.53 (1.52 \Delta)$	91.84 ± 1.95	$92.43 \pm 1.68 (0.64 \bullet)$
<i>TicTacToe</i>	70.63 ± 2.10	$71.74 \pm 2.02 (1.58 \Delta)$	96.45 ± 0.85	$97.42 \pm 0.94 (1.00 \bullet)$
<i>Nursery</i>	90.29 ± 0.56	$91.21 \pm 0.92 (1.01 \bullet)$	95.84 ± 0.73	$96.00 \pm 0.69 (0.17 \bullet)$
<i>Chess</i>	87.85 ± 0.69	$88.04 \pm 0.75 (0.21 \bullet)$	94.77 ± 0.41	$96.00 \pm 0.60 (1.29 \Delta)$

Tabela 5.11: Resultados da $\text{Macro}F_1$ para as bases de atributos categóricos

Bases	Naïve Bayes		KNN	
	Linha de Base	Micro F_1	Linha de Base	Micro F_1
<i>Cars</i>	67.74 ± 3.08	$76.85 \pm 5.65 (13.44 \Delta)$	75.36 ± 3.44	$81.77 \pm 3.90 (8.51 \Delta)$
<i>TicTacToe</i>	64.48 ± 2.12	$64.10 \pm 1.96 (-0.59 \bullet)$	95.99 ± 0.98	$97.12 \pm 1.10 (1.18 \bullet)$
<i>Nursery</i>	66.19 ± 7.41	$77.19 \pm 12.80 (16.62 \Delta)$	80.83 ± 9.06	$84.46 \pm 9.22 (4.50 \Delta)$
<i>Chess</i>	87.81 ± 0.68	$88.03 \pm 0.75 (0.25 \bullet)$	94.75 ± 0.40	$95.98 \pm 0.61 (1.30 \Delta)$

5.6 Base de Bioinformática.

Realizamos experimentos também com a base de assinaturas estruturais proteicas, geradas pelo método CSM. Ela é composta de quinze atributos numéricos, resultantes do uso do SVD para diminuir sua dimensionalidade. Nessa base, exploramos os relacionamentos das proteínas baseado em uma rede de similaridade. Essa rede mede o quanto duas proteínas da base são similares através da distância das sequencias das estruturas proteicas, utilizando o método BLAST.

Lembramos que, como dito na Seção 5.3 por se tratar de uma base muito grande, utilizamos uma validação cruzada de 10 partições, com uma partição sendo treino, uma servindo de validação e oito como teste. Na Tabela 5.12, mostramos três experimentos que realizamos, um por linha. A primeira linha exibe a utilização do KNN com $k = 1$, que foi o melhor resultado obtido para essa base, em comparação a outros valores para k e ao *Naïve Bayes*. A próxima linha exibe o uso do KNN baseado somente na rede de similaridades gerada, sem usar os atributos, servido de segunda linha de base para o uso do PG. Vemos que a $\text{Micro}F_1$ obtém grande melhora, em detrimento da $\text{Macro}F_1$. Por fim, mostramos os resultados ao evoluir com PG uma função de credibilidade para o relacionamento criado com a rede de similaridades. Destacamos que, mesmo com a utilização de apenas uma pequena parte da base para treinamento, conseguimos ganhos de 26.58% e 50.78%, respectivamente ao uso do KNN que considera apenas os atributos. Em relação ao uso do KNN que considera apenas os relacionamentos, o PG obteve ganhos de 7.74% e 74.18%, para $\text{Micro}F_1$ e $\text{Macro}F_1$, respectivamente.

Finalmente, uma análise das dez funções evoluídas pelo PG revela que a métrica FORÇA foi a mais utilizada, sendo ela sozinha a função de credibilidade de cinco dessas funções

Tabela 5.12: Resultados da $\text{Micro}F_1$ e $\text{Macro}F_1$ para a base de bioinformática.

Algoritmo	$\text{Micro}F_1$	$\text{Macro}F_1$
Linha de Base	70.23 ± 1.402	50.28 ± 1.44
Vizinhos Peso Aresta	82.59 ± 0.26 (17.49 \blacktriangle)	43.53 ± 0.47 (-13.43 \blacktriangledown)
PG	88.98 ± 4.87 (26.58 \blacktriangle)	75.81 ± 4.51 (50.78 \blacktriangle)

Capítulo 6

Conclusões e Trabalhos Futuros

Este trabalho propôs um método baseado no conceito de credibilidade para melhorar classificadores automáticos. Sabemos que nem sempre os exemplos de treinamento devem contribuir igualmente para um modelo de classificação e, portanto, estimar a credibilidade de um conjunto de exemplos e considerá-la na construção do modelo pode aumentar sua eficácia. Para avaliar o quanto um classificador pode confiar em um exemplo de treinamento, propomos a utilização do que chamamos de **função de credibilidade**.

A credibilidade é vista na literatura como uma característica dependente do contexto e de quem a avalia. Ou seja, um mesmo objeto pode ser confiável para um observador e não para outro. Uma maneira de ter uma medida mais objetiva da credibilidade é definindo fatores que influenciam na mesma. Aqui focamos nos atributos e nos relacionamentos que os exemplos mantêm, dois importantes fatores que podem exprimir bem a credibilidade de um exemplo na tarefa de classificação. Usamos de métricas que provêm indícios de separações entre as classes para avaliarmos a credibilidade baseada em atributos e métricas de Redes Complexas para extrair a credibilidade dos relacionamentos. Ao total, trinta métricas de atributos e dezenas de métricas de relacionamentos foram modeladas.

Devido ao grande número de métricas, combiná-las a fim de capturar relações entre elas se tornou uma tarefa muito complexa. Para resolver esse impasse, utilizamos a Programação Genética. Ela, com seu mecanismo de busca baseado no princípio evolutivo de Darwin, nos fornecem uma solução robusta, elegante e eficaz de criar uma função de credibilidade adaptada para o uso em um determinado contexto.

De posse de uma função de credibilidade, um importante passo é incorporar essa função nos algoritmos de classificação. Utilizamos nesse trabalho o *Naïve Bayes* e o KNN. Uma direção futura é estender a credibilidade para os demais classificadores

existentes na literatura, tais como o SVM.

Na última parte dessa dissertação, realizamos diversos experimentos com bases textuais, categóricas e de bioinformática. Em nossos experimentos preliminares, mostramos os bons resultados da utilização de PG em relação à aplicação das métricas em separado. Pudemos verificamos, dessa forma, o poder de adaptação do PG. Em um segundo grupo de experimentos com as bases de documentos, investigamos o poder de generalização das funções de credibilidade. Concluímos que as funções obtêm resultados expressivamente melhores quando aplicadas na própria base que foram evoluídas. Entretanto, os resultados não são tão bons ao aplicarmos em outras bases, tendo algumas perdas, em especial na base da ACM-DL. Destacamos a melhoria de **16.06%** na Macro F_1 da base *Ohsumed* ao evoluir funções de credibilidades para os experimentos com validação cruzada de cinco partições, e de **17.51%** ao utilizar a melhor dessas funções de credibilidade para classificar toda a base.

Os últimos experimentos com bases textuais foram feitos com a base da ACM em foco, pelo fato dela ser a única com a presença de redes de autoria e citação. Verificamos que utilizar mais fatores para definir a credibilidade de um exemplo é uma técnica benéfica, culminando em ganhos de **4.60%** e **8.70%** da Micro F_1 e Macro F_1 , respectivamente.

Realizamos experimentos também com bases de atributos categóricos, onde os resultados da métrica Macro F_1 novamente foram de maior destaque. Enquanto que para as *TicTacToe* e *Chess*, o uso da credibilidade não surtiu tanto efeito, para as base *Cars* e *Nursery*, obtivemos ganhos na Macro F_1 de **13.44%** e **16.62%**, respectivamente.

Finalmente, os experimentos com a base de bioinformática mostram que utilizar o conhecimento dos relacionamentos obteve resultados expressivos, com **26.58%** e **50.78%** de ganhos na Micro F_1 e Macro F_1 , respectivamente, em relação ao KNN sem credibilidade.

Como trabalhos futuros, além de ampliar o número de classificadores utilizados, planejamos explorar melhor as relações entre os fatores que impactam na credibilidade de um exemplo. Por exemplo, nas situações nas quais modelamos mais de um relacionamento, como feito para a base ACM-DL, apenas multiplicamos os valores de credibilidade de cada relacionamento (ver Equação 3.11). Acreditamos que explorar outros tipos de combinações pode ser benéfico.

Além disso, gostaríamos de investigar a relação da credibilidade dos exemplos com o passar do tempo. Nesse caso, gostaríamos de poder responder questões como: (i) o que aconteceria se tentássemos utilizar a credibilidade em tempo real? (ii) As funções evoluídas em um dado instante ainda seriam boas em outro instante? Outra direção de trabalhos é buscar soluções para diminuir a quantidade de treinamento, ainda

assim obtendo funções de credibilidade representativas para as bases nas quais elas são evoluídas. Isso poderia ser feito retirando aqueles exemplos de menor credibilidade ao longo das gerações, fazendo com que o treinamento do PG diminua ao evoluir.

Referências Bibliográficas

- Adamic, L. & Adar, E. (2003). Friends and neighbors on the Web. *Social Networks*, 25(3):211--230.
- Alter, O.; Brown, P. O. & Botstein, D. (2000). Singular value decomposition for genome-wide expression data processing and modeling. *Proceedings of the National Academy of Sciences of the United States of America*, 97(18):10101--10106.
- Altschul, S.; Gish, W.; Miller, W.; Myers, E. & Lipman, D. (1990). Basic local alignment search tool. *Journal of molecular biology*, 215(3):403--410.
- Bäck, T.; Eiben, A. E. & van der Vaart, N. A. L. (2000). An empirical study on gas “without parameters”. Em *PPSN*, pp. 315–324.
- Batal, I. & Hauskrecht, M. (2009). Boosting knn text classification accuracy by using supervised term weighting schemes. Em *Proceedings of the 18th ACM Conference on Information and Knowledge Management*, New York, NY, USA. ACM.
- Baxter, R. (2010). Exponential growth using the internet and your web site. *Facial Plastic Surgery*, 26(1):39--44.
- Beauchamp, M. A. (1965). An improved index of centrality. *Systems Research and Behavioral Science*, 10:161--163.
- Blau, P. (1977). *Inequality and heterogeneity: a primitive theory of social structure*. Free Press.
- Blumer, A.; Ehrenfeucht, A.; Haussler, D. & Warmuth, M. K. (1987). Occam’s razor. *Inf. Process. Lett.*, 24:377--380.
- Bondy, J. & Murty, U. (2008). *Graph Theory*. Springer Publishing Company, Incorporated, 1st edição.

- Brenner, S.; Koehl, P. & Levitt, M. (2000). The astral compendium for protein structure and sequence analysis. *Nucleic Acids Research*, 28(1):254.
- Brin, S. & Page, L. (1998). The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30(1-7):107--117.
- Brunoro, G.; Pappa, G. L.; Palotti, J. & Minardi, R. (2011). Galapagos: A simple and cross-plataform graphical tool to visualize evolution on genetic programming algorithms. <http://brunoro.github.com/galapagos/>.
- Burt, R. S. (1992). *Structural holes: The social structure of competition*. Harvard University Press, Cambridge, MA.
- Cavaretta, M. J. & Chellapilla, K. (2009). Data mining using genetic programming: The implications of parsimony on generalization error. Em *Proceedings of the Congress on Evolutionary Computation*, volume 2. IEEE Press.
- Central Intelligence Agency (2011). The world factbooks - distribution of family income - gini index. <https://www.cia.gov/library/publications/the-world-factbook/rankorder/2172rank.html>.
- Chesney, T. (2006). An empirical examination of Wikipedia's credibility. 11(11).
- Darwin, C. (1859). *The Origin Of Species*.
- de Freitas, J.; Pappa, G. L.; da Silva, A. S.; Gonçalves, M. A.; de Moura, E. S.; Veloso, A.; Laender, A. H. F. & de Carvalho, M. G. (2010). Active learning genetic programming for record deduplication. Em *IEEE Congress on Evolutionary Computation*, pp. 1-8.
- Debole, F. & Sebastiani, F. (2003). Supervised term weighting for automated text categorization. Em *Proceedings of the ACM symposium on Applied computing*, SAC '03, pp. 784--788, New York, NY, USA. ACM.
- Dice, L. R. (1945). Measures of the Amount of Ecologic Association Between Species. *Ecology*, 26(3):297--302.
- Duda, R. O.; Hart, P. E. & Stork, D. G. (2001). *Pattern Classification (2nd Edition)*. Wiley-Interscience, 2 edição.
- Eastin, M. S. (2001). Credibility Assessments of Online Health Information: The Effects of Source Expertise and Knowledge of Content. *Journal of Computer-Mediated Communication*, 6(4):0.

- Eysenbach, G. & Köhler, C. (2002). How do consumers search for and appraise health information on the world wide web? Qualitative study using focus groups, usability tests, and in-depth interviews. *BMJ*, 324(7337):573--577.
- Flanagin, A. J. & Metzger, M. J. (2007). The role of site features, user attributes, and information verification behaviors on the perceived credibility of web-based information. *New Media Society*, 9(2).
- Fogel, D.; B "ack, T. & Michalewicz, Z. (2000). *Evolutionary Computation: Basic algorithms and operators*. Evolutionary Computation. Institute of Physics Publishing.
- Fogg, B. J.; Marshall, J.; Laraki, O.; Osipovich, A.; Varma, C.; Fang, N.; Paul, J.; Rangnekar, A.; Shon, J.; Swani, P. & Treinen, M. (2001). What makes web sites credible?: a report on a large quantitative study. Em *Proceedings of the SIGCHI conference on Human factors in computing systems*, CHI '01, pp. 61--68, New York, NY, USA. ACM.
- Forrest, S.; Nguyen, T.; Weimer, W. & Le Goues, C. (2009). A genetic programming approach to automated software repair. Em *GECCO '09: Proceedings of the 11th Annual conference on Genetic and evolutionary computation*, pp. 947--954, New York, NY, USA. ACM.
- Fraser, A. & Weinbrenner, T. (2011). Gpc++ - genetic programming c++ class library. <http://www.cs.ucl.ac.uk/staff/W.Langdon/ftp/weinbrenner/gp.html>.
- Freeman, K. & Spyridakis, J. H. (2004). An Examination of Factors That Affect the Credibility of Online Health Information. *Technical Communication*.
- Freitas, A. A. (2002). *Data Mining and Knowledge Discovery with Evolutionary Algorithms*. Springer-Verlag.
- Galavotti, L.; Sebastiani, F. & Simi, M. (2000). Experiments on the use of feature selection and negative evidence in automated text categorization. Em *ECDL*, pp. 59--68.
- Goldberg, D. E. (1989). Genetic Algorithms in Search, Optimization, and Machine Learning.
- Golubski, W. (2002). New results on fuzzy regression by using genetic programming. Em Foster, J.; Lutton, E.; Miller, J.; Ryan, C. & Tettamanzi, A., editores, *Genetic*

- Programming*, volume 2278 of *Lecture Notes in Computer Science*, pp. 5–95. Springer Berlin / Heidelberg.
- Hauptman, A. & Sipper, M. (2007). Evolution of an efficient search algorithm for the mate-in-n problem in chess. Em *Proceedings of the 10th European conference on Genetic programming*, EuroGP'07, pp. 78--89, Berlin, Heidelberg. Springer-Verlag.
- Hovland, C. I. & Weiss, W. (1951). The Influence of Source Credibility on Communication Effectiveness. *Public Opin Q*, 15(4):635--650.
- How, B. C. & Narayanan, K. (2004). An empirical study of feature selection for text categorization based on term weightage. Em *WI '04: Proceedings of the 2004 IEEE/WIC/ACM International Conference on Web Intelligence*, Washington, DC, USA. IEEE Computer Society.
- Jaccard, P. (1901). Étude comparative de la distribution florale dans une portion des Alpes et des Jura. *Bulletin de la Société Vaudoise des Sciences Naturelles*, 37:547--579.
- Joachims, T. (1999). Making large-scale support vector machine learning practical. pp. 169--184.
- Kessler, M. M. (1963). An experimental study of bibliographic coupling between technical papers. *IEEE Transaction on Information Theory*.
- Kishore, J. K.; Patnaik, L. M.; Mani, V. & Agrawal, V. K. (2000). Application of genetic programming for multiclass pattern classification. *IEEE Transactions on Evolutionary Computation*, 4(3):242--258.
- Kleinberg, J. M. (1999). Authoritative sources in a hyperlinked environment. *J. ACM*, 46:604--632.
- Koller, D. & Sahami, M. (1997). Hierarchically classifying documents using very few words. Em *Proceedings of the 14th International Conference on Machine Learning (ML), Nashville, Tennessee, July 1997*, pp. 170--178.
- Koza, J. (2010). Human-competitive results produced by genetic programming. *Genetic Programming and Evolvable Machines*, 11(3):251--284.
- Koza, J. R. (1992). *Genetic Programming: On the Programming of Computers by Means of Natural Selection (Complex Adaptive Systems)*. The MIT Press, Cambridge, MA, USA.

- Kubiszewski, I.; Noordewier, T. & Costanza, R. (2011). Perceived credibility of Internet encyclopedias. *Computers & Education*, 56(3):659--667.
- Lan, M.; Sung, S.-Y.; Low, H.-B. & Tan, C.-L. (2005). A Comparative Study on Term Weighting Schemes for Text Categorization. *Proceedings of International Joint Conference on Neural Networks '05*, pp. 546--551.
- Li, T.; Zhang, C. & Ogihara, M. (2004). A comparative study of feature selection and multiclass classification methods for tissue classification based on gene expression. *Bioinformatics*, 20:2429--2437.
- Lindberg, D. A. & Humphreys, B. L. (1998). Medicine and health on the Internet: the good, the bad, and the ugly. *JAMA : the journal of the American Medical Association*, 280(15):1303--1304.
- Liu, Y.; Loh, H. & Sun, A. (2009). Imbalanced text classification: A term weighting approach. *Expert Systems with Applications*, 36(1):690--701.
- Lopes, R. & Carriço, L. (2008). On the credibility of wikipedia: an accessibility perspective. Em *WICOW '08: Proceeding of the 2nd ACM workshop on Information credibility on the web*, pp. 27--34, New York, NY, USA. ACM.
- Macskassy, S. A. & Provost, F. (2004). Simple models and classification in networked data. Em *In CeDER Working Paper 03-04. Stern School of Business*.
- Manning, C. D.; Raghavan, P. & Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA.
- McPherson, M.; Smith-Lovin, L. & Cook, J. M. (2001). Birds of a feather: Homophily in social networks. *Annual Review of Sociology*, 27(1):415--444.
- Mengle, S. S. R. & Goharian, N. (2008). Using ambiguity measure feature selection algorithm for support vector machine classifier. Em *Proceedings of the 2008 ACM symposium on Applied computing*, SAC '08, pp. 916--920, New York, NY, USA. ACM.
- Mladenić, D. (1998). Feature subset selection in text-learning. Em *Proc. of the 10th European Conference on Machine Learning ECML98*.
- Murzin, A. G.; Brenner, S. E.; Hubbard, T. & Chothia, C. (1995). SCOP: a structural classification of proteins database for the investigation of sequences and structures. *Journal of molecular biology*, 247(4):536--540.

- Newman, D.; Hettich, S.; Blake, C. & Merz, C. (1998). Uci repository of machine learning databases.
- Newman, M. (2010). *Networks: An Introduction*. Oxford University Press, Inc., New York, NY, USA.
- Newman, M. E. J. (2003). The structure and function of complex networks. *SIAM REVIEW*, 45:167--256.
- Ng, H. T.; Goh, W. B. & Low, K. L. (1997). Feature selection, perceptron learning, and a usability case study for text categorization. Em *Proceedings of the 20th annual international ACM SIGIR conference on Research and development in information retrieval*, SIGIR '97, pp. 67--73, New York, NY, USA. ACM.
- Onody, R. N. & de Castro, P. A. (2004). Complex network study of brazilian soccer players. *Phys. Rev. E*, 70(3):037103.
- Palmer, J. W.; Bailey, J. P. & Faraj, S. (2000). The role of intermediaries in the development of trust on the www: The use and prominence of trusted third parties and privacy statements. *J. Computer-Mediated Communication*, 5(3).
- Palotti, J.; Salles, T.; Pappa, G. L.; Arcanjo, F.; Gonçalves, M. A. & Meira, W. (2010). Estimating the credibility of examples in automatic document classification. *Journal of Information and Data Management*, 1(3):439--454.
- Palotti, J.; Salles, T.; Pappa, G. L.; Goncalves, M. A. & Meira, W. (2011). Assessing documents' credibility with genetic programming. Em Smith, A. E., editor, *Proceedings of the 2011 IEEE Congress on Evolutionary Computation*, pp. 200-207, New Orleans, USA. IEEE Computational Intelligence Society, IEEE Press.
- Pires, D. E. V.; Melo-Minardi, R. C.; Santos, M. A.; H, d. C.; Santoro, M. M. & W, M. (2011). Cutoff Scanning Matrix (CSM): structural classification and function prediction by protein inter-residue distance patterns. *BMC Bioinformatics*. Accepted for publication.
- Rains, S. A. & Karmikel, C. D. (2009). Health information-seeking and perceptions of website credibility: Examining Web-use orientation, message characteristics, and structural features of websites. *Computers in Human Behavior*, 25(2):544--553.
- Refaeilzadeh, P.; Tang, L. & Liu, H. (2009). Cross-validation. Em *Encyclopedia of Database Systems*, pp. 532–538.

- Rieh, S. Y. & Danielson, D. R. (2007). Credibility: A multidisciplinary framework. *Ann. Rev. Info. Sci. Tech.*, 41(1):307–364.
- Rotach, M.; Abrosetti, P.; Ament, F.; Appenzeller, C.; Arpagaus, M.; H.-S., B.; A., B.; Bouttier, F.; Buzzi, A.; Corazzo, M. DaVolio, S.; Denhard, M.; Dorninger, M.; Fontannaz, L.; Frick, J.; Fundel, F.; Germann, U.; Gorgas, T.; Hegg, C.; Hering, A.; Keil, C.; Liniger, M.; Marsigli, C.; R., M.-C.; Montaini, A.; Mylne, K.; Ranzi, R.; Richard, E.; Rossa, A.; Santos-Munoz, A.; Schär, C.; Seity, Y.; Studinger, M.; Stoll, M.; Volkert, H.; Walser, A.; Wang, Y.; Wehrhahn, J.; Wulfmeyer, V. & Zappa, M. (2009). Map d-phase: Real-time demonstration of weather forecast quality in the alpine region. *Bull. Amer. Meteorol. Soc.*, 90(9):1321–1336.
- Rubinov, M. & Sporns, O. (2010). Complex network measures of brain connectivity: Uses and interpretations. *NeuroImage*, 52(3):1059 – 1069. Computational Models of the Brain.
- Sabidussi, G. (1966). The centrality index of a graph. *Psychometrika*, 31(4):581–603.
- Salles, T.; Rocha, L.; Pappa, G. L.; Mourão, F.; Gonçalves, M. A. & Jr., W. M. (2010). Temporally-aware algorithms for document classification. Em *Proceedings of the International ACM SIGIR Conference on Research & Development of Information Retrieval*, pp. 307–314, Genebra, Switzerland.
- Salton, G. & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Inf. Process. Manage.*, 24(5):513–523.
- Shang, W.; Huang, H.; Zhu, H.; Lin, Y.; Qu, Y. & Wang, Z. (2007). A novel feature selection algorithm for text categorization. *Expert Systems with Applications*, 33(1).
- Shen, K. & Wu, L. (2005). Folksonomy as a complex network. *CoRR*, abs/cs/0509072.
- Small, H. (1973). Co-citation in the scientific literature: A new measure of the relationship between two documents. *J. Am. Soc. Inf. Sci.*, 24(4):265–269.
- Spector, L.; Barnum, H. & Bernstein, H. J. (1998). Genetic programming for quantum computers. Em *In Genetic Programming*, pp. 365–374. Morgan Kauffman.
- Sundar, S. S. (1999). Exploring receivers' criteria for perception of print and online news. *Journalism and Mass Communication Quarterly*, (76):373–386.
- Tang, L. & Liu, H. (2005). Bias analysis in text classification for highly skewed data. Em *ICDM '05: Proceedings of the Fifth IEEE International Conference on Data Mining*, Washington, USA. IEEE Computer Society.

- Thom-Santelli, J.; Millen, D. R. & Gergle, D. (2011). Organizational acculturation and social networking. Em *Proceedings of the ACM 2011 conference on Computer supported cooperative work*, CSCW '11, pp. 313–316, New York, NY, USA. ACM.
- Tseng, S. & Fogg, B. J. (1999). Credibility and computing technology. *Communications of the ACM*, 42(5).
- van Rijsbergen, C. J. (1979). *Information Retrieval*. Butterworths, London, 2 edição.
- Williams, S. J. & Hayward, N. K. (2001). The impact of the human genome project on medical genetics. *Trends Mol Med*, 7(5):229–31.
- Yang, Y. & Pedersen, J. O. (1997). A comparative study on feature selection in text categorization. Em *ICML '97: Proceedings of the Fourteenth International Conference on Machine Learning*, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.
- Yeang, C. H.; Ramaswamy, S.; Tamayo, P.; Mukherjee, S.; Rifkin, R. M.; Angelo, M.; Reich, M.; Lander, E.; Mesirov, J. & Golub, T. (2001). Molecular classification of multiple tumor types. *Bioinformatics*, 17 Suppl 1.
- Zaijane, O. R. & Antonie, M.-L. (2002). Classifying text documents by associating terms with text categories. Em *Proceedings of the 13th Australasian database conference - Volume 5*, ADC '02, pp. 215–222, Darlinghurst, Australia, Australia. Australian Computer Society, Inc.
- Zheng, Z. & Srihari, R. (2003). Optimally Combining Positive and Negative Features for Text Categorization. Em *Workshop for Learning from Imbalanced Datasets II, Proceedings of the ICML*, Washington, DC.