# USO DE CONTEXTOS TEMPORAIS PARA CLASSIFICAÇÃO DE DOCUMENTOS

LEONARDO CHAVES DUTRA DA ROCHA

# USO DE CONTEXTOS TEMPORAIS PARA CLASSIFICAÇÃO DE DOCUMENTOS

Tese 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 Doutor em Ciência da Computação.

ORIENTADOR: WAGNER MEIRA JR.

Belo Horizonte

06 de fevereiro de 2009

Rocha, Leonardo Chaves Dutra da
R762u Uso de Contextos Temporais para Classificação de Documentos / Leonardo Chaves Dutra da Rocha. — Belo Horizonte, 2009
xviii, 128 f. : il. ; 29cm

Tese (doutorado) — Universidade Federal de Minas Gerais
Orientador: Wagner Meira Jr.

1. Classificação de Documentos. 2. Evolução Temporal. 3. Algoritmos. I. Título.

CDU 519.6*73(043)

UNIVERSIDADE FEDERAL DE MINAS GERAIS
INSTITUTO DE CIÊNCIAS EXATAS
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

# FOLHA DE APROVAÇÃO

Uso de contextos temporais para classificação de documentos

## LEONARDO CHAVES DUTRA DA ROCHA

Tese defendida e aprovada pela banca examinadora constituída pelos Senhores:

PROF. WAGNER MEIRA JÚNIOR - Orientador
Departamento de Ciência da Computação - UFMG

PROF. MARCOS ANDRÉ GONÇALVES - Co-orientador
Departamento de Ciência da Computação - UFMG

PROF. ANDRÉ CARLOS PONCE DE L. F. DE CARVALHO
Instituto de Ciências Matemática e de Computação - USP

PROFA. BIANCA ZADROZNY
Centro Tecnológico - UFF

PROF. ALBERTO HENRIQUE FRADE LAENDER
Departamento de Ciência da Computação

PROF. NIVIO ZIVIANI
Departamento de Ciência da Computação - UFMG

*Ao meu filho Diogo, amor de minha vida.*

*À minha amada mãe Conceição.*

*À minha querida esposa Elisa.*

*Aos meus amigos e irmãos Leandro e Luciano.*

# Agradecimentos

Primeiramente agradeço a Deus por mas essa oportunidade em minha vida. Agradeço a minha mãe querida, no verdadeiro sentido da palavra MÃE, por me ensinar os verdadeiros valores da vida como honestidade, caráter, humildade e perseverança. Agradeço pelo seu apoio incondicional, por ser o pai do Diogo na minha ausência (e que ausência nesses últimos anos...), por ser meu pai quando precisei, minha amiga nos momentos de dificuldade. Mãe, você me ensinou o quanto é importante o conhecimento e hoje cá estou em busca dele, cada vez mais. Muito obrigado! Te amo muito!!!

Agradeço aos meus dois pais: papai e vovô Chaves, que, mesmo não presente mais entre nós, de uma forma inexplicável, me sussurravam força e garra para lutar e não desistir do meu sonho. Agradeço aos meus irmãos Leandro e Luciano. Vocês dois são meus melhores amigos, companheiros. Mesmo sem entender muito bem o que eu estava fazendo, estavam sempre do meu lado! Amo demais vocês dois, e sempre será assim. Muito obrigado ao meu irmão emprestado João que em vários momentos difíceis dessa trajetótia tinha sempre uma palavra de conforto para me dar.

Agradeço a minha amada Elisa pelo companheirismo incondicional, procurando sempre entender minha ausência em casa, me apoiando em minhas decisões, por mais difíceis que a mesmas fossem, revezando comigo nas longas madrugadas dos primeiros meses de vida do nosso filho. Amor, sabemos o quanto foi e é difícil, passamos por tudo juntos, continuamos lutando juntos e sempre estaremos juntos...TE AMO!! Agradeço ao meu sogro José Braga e a a minha sogra Inez Helena, pelo amor e carinho com que vocês cuidaram e cuidam do nosso Diogo. Por me acolherem na família de vocês, não como o

marido da Elisa mas como um filho.

Agradeço ao meu filho e amigo Diogo. Diogo, hoje, com 4 anos você ainda não entende o quanto você foi, é e sempre será importante em minha vida. Graças a você meu filho, aprendi a ter um pouco mais de paciência, aprendi o quanto é bom amar de forma incondicional. Graças a você, me sinto muito importante porque te vendo assim, tão pequenino e tão inocente, vejo o quanto você precisa de mim. Se hoje você não entende, espero que um dia isso aconteça, mas independente de qualquer coisa, sempre te amarei.

Agradeço ao meu mestre, orientador e amigo Prof. Wagner Meira. Meira, trabalhamos juntos desde de 2000, quando após uma aula de AEDS3 você me convidou para uma iniciação científica. Desse dia para cá foram muitos trabalhos, muitos artigos publicados, muitos artigos recusados, mas ao mesmo tempo muita amizade e conversa. Seu entusiasmo, paciência e dedicação em ensinar é algo contagiante. Hoje, após quase 10 anos trabalhando juntos, tenho por você uma profunda admiração e respeito. Muito obrigado mesmo Meira!! Agradeço também ao professor Marcos que na ausência do Meira, era quem me ajudava, era a quem eu recorria com minhas dúvidas.

Devo grande parte desse trabalho ao meu amigo e companheiro de trabalho Fernando. São mais de 4 anos de trabalho juntos, e mais do que isso, convivendo como amigos. Saiba que tenho por você uma grande admiração. Como costumava dizer, eramos orientador e orientado um do outro, ora um sendo orientador ora o outro, nas longas discussões sobre nossos trabalhos. Tenho você como um amigo, de verdade! Muito obrigado amigão!

Ao pessoal da secretaria (Túlia, Maristela, Renata, Gilmara, Cida, Sheila, Cláudia, Stella, Sônia, Emília, Ludmila, Lizete, Orlando, Alexandre e Geraldo) meu muito obrigado pelo apoio e amizade durante todos esses anos. Vocês são pessoas maravilhosas!

Agradeço ao pessoal do speed, aos desmaiados, ao pessoal de Lafaiete pelos momentos de descontração. Não só de trabalho vive o homem, e vocês contribuíram e muito nessa caminhada, me proporcionando momentos de alegria e descontração para esquecer um pouco o trabalho árduo que está por trás dessa tese.

# Resumo

Devido à crescente quantidade de informação que vem sendo armazenada e acessada por meio da Web, Classificação Automática de Documentos (CAD) tem se tornado um importante tópico de pesquisa. CAD normalmente segue uma estratégia de aprendizado em que primeiro se constrói um modelo de classificação utilizando documentos pré-classificados e então aplica-se esse modelo para classificar os demais documentos. Um dos maiores desafios em CAD é que as características dos documentos e das classes às quais eles pertencem mudam ao longo do tempo, uma vez que novos documentos são criados, novas informações surgem, novos termos também são introduzidos e, conseqüentemente, as definições das classes podem mudar. Apesar da potencial redução de qualidade dos modelos de classificação associado com as mudanças relacionadas ao tempo, a maioria das técnicas atuais de CAD não consideram a evolução temporal das coleções de documentos. Assim, conforme veremos nesse trabalho, um importante desafio é construir modelos de classificação que sejam capazes de lidar com essa evolução temporal.

As duas principais hipóteses desta tese são: (1) a evolução temporal das coleções de documentos afeta significativamente o desempenho dos classificadores automáticos de texto e (2) as dimensões que compõem essa evolução temporal podem ser exploradas de forma a construir melhores classificadores, mais eficientes e mais efetivos. Dessa forma, o objetivo dessa tese é caracterizar, quantificar e qualificar, o impacto da evolução temporal das coleções de documentos sobre os classificadores automáticos de documentos, identificando as dimensões que compõem esse impacto. Além disso, utilizando o conhecimento adquirido com essa caracterização, o objetivo é propor alternativas para que os

problemas causados pela evolução temporal das coleções nos classificadores automáticos de documentos sejam minimizados.

As principais contribuições desta tese são: (1) demonstração da existência e quantificação do impacto da evolução temporal em classificação automática de documentos, (2) identificação das dimensões que compõem a evolução temporal, (3) qualificação e quantificação de cada uma das dimensões identificadas, (4) criação de um modelo de seleção de contextos do conjunto de treinamento que minimize os efeitos temporais e (5) a instanciação e validação do modelo utilizando diferentes coleções de documentos e algoritmos de classificação.

# Abstract

Due to the increasing amount of information being stored and accessible through the Web, Automatic Document Classification (ADC) has become an important research topic. ADC usually employs a supervised learning strategy, where we first build a classification model using pre-classified documents and then use it to classify unseen documents. One major challenge for ADC in many scenarios is that the characteristics of the documents and the classes to which they belong may change over time, since new documents are created, new information rise, new terms also are introduced and, consequently, the class definitions may change. Despite the potential quality reduction in the classification models associated with temporal-related changes, most of the current techniques for ADC are applied without taking into account the temporal evolution of the collection of documents. As we will see in this work, an important challenge in building classifiers is to deal with this temporal evolution.

The two main hypotheses of this dissertation are: (1) the temporal evolution of the document collections significantly affects the performance of automatic document classifiers, and (2) the dimensions that compose this temporal evolution can be taken into account in order to build better classifiers, which are more efficient and effective. Thus, the goal of this thesis is to characterize, quantitatively and qualitatively, the impact of the temporal evolution of document collections on automatic document classifiers, identifying the dimensions that compose this impact. Moreover, based on this characterization, the goal is to propose alternate strategies that can be used to minimize the challenges associated with the temporal evolution of the collections on automatic document classifiers.

The main expected contributions of this thesis are: (1) demonstration and quantification of the temporal evolution and how this affects automatic document classifiers, (2) identification of the effect dimensions that compose the temporal evolution, (3) quantification and qualification of each dimension identified, (4) design of a model to select contexts of the training set that minimize the temporal effects, and (5) the validation of this model using different document collections and classification algorithms.

# Lista de Figuras

# Lista de Tabelas

# Sumário

# Capítulo 1

# Introdução

A Internet tem vivenciado uma notória expansão e popularização nos últimos anos. Conseqüentemente, esse crescimento tem provocado um aumento significativo da quantidade de informação que é armazenada e acessada por meio da Web. Essa informação é freqüentemente organizada em forma de documentos textuais e tem sido o alvo principal de máquinas de busca e outras ferramentas que executam tarefas como busca e filtragem. Uma estratégia comum para lidar com essa grande quantidade de informação é associá-la a categorias semanticamente significativas em um processo conhecido como Classificação Automática de Documentos (CAD) (Borko & Bernick, 1963). CAD tem se tornado um importante tópico de pesquisa em Recuperação de Informação (Koll, 1981; Baeza-Yates & Ribeiro-Neto, 1999) e Mineração de Textos (Dörre et al., 1999; Tan, 1999). Esse processo de organização de documentos em categorias pode apoiar e melhorar diversas outras tarefas como etiquetagem automática de tópicos (i.e., assinalamento automático de rótulos a documentos), criação de diretório de tópicos, identificação de estilo de escrita em documentos, criação de bibliotecas digitais (Fahmi, 2004), melhora na precisão de pesquisas na Web e auxílio na interação entre usuários e máquinas de busca (Terveen et al., 1999). Outros cenários comuns de aplicações que podem se beneficiar incluem filtragem de *spam*, detecção de conteúdo adulto e plágio.

CAD usualmente segue uma estratégia de aprendizagem supervisionada (Sebastiani,

2002), em que primeiramente um modelo de classificação é construído utilizando os documentos pré-classificados e posteriormente esse modelo é utilizado para classificar os demais. O processo de construção de modelos de classificação significa encontrar o conjunto de características (e.g., termos) que melhor identifica uma classe de documentos. Um dos maiores desafios em CAD é que as características dos documentos e das classes às quais eles pertencem mudam ao longo do tempo, como conseqüência do surgimento de novas informações, introdução de novos termos, criação de novas áreas e a divisão de áreas maiores em subáreas menores e mais especializadas. Assim, o conjunto de características que distingue uma determinada categoria de documentos evolui ao longo do tempo, e esse conjunto, que antes era útil para separar as classes, à medida que evolui pode distorcer as diferenças entre elas.

Recentemente, pudemos observar um interessante exemplo do impacto da evolução de classes. Além de ser o Deus do Inferno na mitologia romana, Plutão também era considerado um planeta até meados de 2006. Até essa data, documentos que possuíam o termo Plutão em seu vocabulário tinham uma probabilidade alta de serem assinalados à classe astrofísica, devido à grande quantidade de referências que mencionavam Plutão como um planeta. A partir dessa data, uma vez que Plutão não é mais considerado um planeta, ocorreu uma redução significativa no número de documentos que se referem a ele nesse contexto. Entretanto, no contexto de mitologia romana, o número de referências ao termo Plutão não sofreu uma variação significativa. Dessa forma, modelos de classificação construídos após essa mudança apresentaram um significativo aumento na probabilidade de que um documento que tenha o termo seja assinalado à classe mitologia romana. No entanto, a criação de tais modelos é possível somente quando os aspectos temporais dos documentos são levados em consideração, o que não é o caso da maioria das estratégias de construção de classificadores.

Alonso et al. (2007) defendem que o tempo é uma importante dimensão da informação em qualquer cenário e que pode ser muito útil na área de recuperação de informação. A despeito da potencial redução na qualidade dos modelos de classificação associada às

mudanças temporais, a maioria dos algoritmos atuais para CAD, como “vizinho mais próximo” (kNN - k nearest-neighbor) (Lam & Ho, 1998), abordagens probabilísticas bayesianas (Naïve Bayes) (McCallum & Nigam, 1998), modelos baseados em espaço vetorial (SVM - *suport vector machines*) (Joachims, 1998a) e modelos de regras de classificação (Friedman et al., 1996; Veloso et al., 2006) são aplicados sem considerar a evolução temporal das coleções de documentos. Apesar de a evolução temporal ser considerada um fator relevante no contexto de CAD (Lanquillon & Renz, 1999; Lewis, 1995; Liebscher, 2004), existem algumas questões relacionadas que ainda não estão claras e o objetivo desse trabalho é respondê-las. São elas:

1. Como a evolução temporal influencia no desempenho dos classificadores?
2. Quais são as características relacionadas ao tempo que afetam a qualidade dos classificadores?
3. Como tais características podem ser exploradas ou isoladas para melhorar a precisão dos classificadores?

## 1.1 Hipóteses da Tese

Dessa forma, baseando-se nas três questões que precisam ser respondidas, são duas as hipóteses desta tese:

1. A evolução temporal das coleções de documentos afeta significativamente o desempenho dos classificadores automáticos de texto (relacionada às questões 1 e 2).
2. As características que compõem essa evolução temporal podem ser exploradas de forma a construir melhores classificadores, mais eficientes e mais efetivos (relacionada à questão 3).

## 1.2 Descrição do Trabalho

Existem alguns trabalhos em CAD que consideram a evolução temporal das coleções, mas a maioria deles foca apenas no desafio de tratar o desbalanceamento nas freqüências das classes e na mudanças dessas freqüências ao longo do tempo, que aqui mencionamos como distribuição de classes. Em nosso trabalho vamos além, consideramos outros dois fatores que estão relacionados à evolução temporal. O segundo fator está associado ao relacionamento entre termos e classes e como esse relacionamento muda ao longo do tempo, como conseqüência do surgimento e desaparecimento de termos e da variação do poder discriminativo dos mesmos entre as classes ao longo do tempo. O terceiro fator é como a similaridade entre as classes, como uma função dos termos que ocorrem em seus documentos, varia ao longo do tempo. Por exemplo, duas classes podem ser muito similares em um determinado momento e posteriormente serem completamente distintas. Com o objetivo de responder às duas primeiras questões que estão relacionadas à primeira hipótese dessa tese, entendendo e caracterizando esses três fatores, avaliamos a evolução temporal e seus efeitos em duas coleções de documentos distintas: uma da biblioteca digital da ACM (ACM-DL) contendo documentos da área de Ciência da Computação, e outra da MedLine, uma biblioteca digital contendo documentos relacionados à Medicina. Propomos e aplicamos uma nova metodologia capaz de caracterizar o impacto da evolução temporal das coleções de documentos no desempenho dos classificadores. Nossa metodologia permite analisar separadamente cada um dos fatores, apontando as causas que resultam na alteração do desempenho dos classificadores.

Após entender como a evolução temporal afeta os algoritmos de CAD e quais efeitos compõem essa evolução, o próximo passo é responder à terceira questão previamente apresentada, relacionada à segunda hipótese dessa tese, e propor algoritmos que explorem os efeitos temporais com o objetivo de construir modelos de classificação mais precisos. Para tanto, esses algoritmos devem ser capazes tanto de classificar documentos de acordo com o contexto dos mesmos, ou seja, de acordo com o seu momento de criação,

quanto de considerar apenas as características que se apresentaram estáveis nesse contexto. Uma possível estratégia é criar modelos de classificação que sejam capazes de lidar com os efeitos temporais mencionados. Uma estratégia diferente, que é proposta e avaliada nesse trabalho, é selecionar uma porção do conjunto de treinamento (documentos pré-classificados) que minimize esses efeitos, a qual chamamos de seleção de contextos temporais. Assim, propomos um algoritmo genérico para seleção de contextos temporais e derivamos duas heurísticas que adequadamente instanciam esse algoritmo genérico. A principal diferença entre as duas heurísticas é referente aos contextos temporais gerados por elas serem simétricos ou não em relação aos termos que compõem os documentos de treino e teste. Adaptamos os conceitos de assimetria e simetria apresentados por Faith (1983), no qual a "assimetria em dados binários surge quando um dos dois estados (e.g., estado 1) é interpretado como sendo mais informativo que o outro estado". Em nosso caso, um contexto é simétrico quando o mesmo não faz distinção entre os termos que ocorrem em documentos de teste e aqueles que não ocorrem, caso contrário ele é considerado assimétrico. Mais especificamente, a heurística *GreedyChronos* considera apenas as características presentes nos documentos de teste (termos de teste) para selecionar os contextos temporais. Por outro lado, a heurística *WindowChronos* gera contextos simétricos, uma vez que ela considera todas as características (termos de teste e não teste).

Com o objetivo de demonstrar a aplicabilidade do algoritmo genérico e a efetividade dessas duas heurísticas, aplicamos as mesmas nas duas coleções de dados anteriormente mencionadas (ACM-DL e MedLine) em conjunto com diversos algoritmos de classificação automática de documentos tais como kNN, Naïve Bayes, Rocchio e SVM, além de caracterizar detalhadamente os resultados alcançados. Por meio dessa caracterização mostramos que a heurística *GreedyChronos* funciona muito bem com algoritmos que utilizam apenas os termos presentes nos documentos de teste (as quais são naturalmente assimétricas), mas não funciona apropriadamente com algoritmos que consideram todos os termos (i.e., "simétricas")[1]. Entretanto, utilizando os contextos temporais selecionados

[1]De agora em diante, sempre que nos referirmos à simetria e assimetria consideramos nossas definições.

pela heurística *WindowChronos*, alcançamos melhorias significativas em todos os cenários avaliados. Por fim, analisando em conjunto todos os resultados e caracterizações discutidos ao longo desse trabalho, apresentamos algumas conclusões sobre os potenciais e as limitações do uso de contextos temporais por cada um dos algoritmos avaliados, correlacionando também com as características das coleções.

## 1.3 Contribuições

As contribuições alcançadas neste trabalho são:

- Demonstração e quantificação da existência do impacto da evolução temporal em CAD.
- Identificação de quais as dimensões que compõem o impacto da evolução temporal.
- Qualificação e quantificação de cada uma das dimensões identificadas.
- Criação de um modelo de seleção de contextos temporais que considere a evolução e que seja capaz de minimizar os efeitos causados por ela.
- Instanciação e validação do modelo utilizando diferentes coleções de documentos e algoritmos de classificação.
- Criação de um repositório contendo os artefatos produzidos nesse trabalho, bem como as coleções de documentos utilizadas. [2]

Em suma, esse trabalho proverá os seguintes benefícios:

- Algoritmos de CAD mais robustas quanto à evolução temporal.
- Melhorias em outras tarefas que dependem de CAD como criação de bibliotecas digitais, recuperação de informação, filtragem de spam etc.

[2] www.chronos.speed.dcc.ufmg.br

## 1.4 Organização da Tese

Está tese está organizada em mais 6 capítulos. O Capítulo 2 apresenta um resumo dos principais trabalhos correlatos. O Capítulo 3 caracteriza a evolução temporal e os efeitos que a compõem em duas coleções de documentos distintas. Além disso, nesse capítulo demonstramos como esses efeitos podem degenerar a qualidade dos algoritmos de classificação automática de documentos. O Capítulo 4 apresentamos um definição formal do problema de seleção de contextos temporais e a proposta de um algoritmo genérico para esse problema. O Capítulo 5 descreve as duas heurísticas baseadas no algoritmo genérico proposto: *GreedyChronos* e *WindowChronos*, O Capítulo 6 avalia e caracteriza ambas as heurísticas utilizando diversos algoritmos de classificação automática de documentos. Finalmente, o Capítulo 7 resume nossas principais contribuições e apresenta as conclusões, além dos trabalhos futuros.

# Capítulo 2

# Trabalhos Relacionados

Apesar de classificação automática de documentos ser um assunto amplamente estudado, a análise dos aspectos temporais relacionados à evolução das coleções de documentos e como essa evolução afeta os algoritmos nessa tarefa ainda são recentes - vêm sendo estudados apenas nas últimas duas décadas. Fizemos um amplo levantamento dos principais esforços desse assunto e neste capítulo apresentamos os mesmos separados em quatro seções.

Primeiramente, na Seção 2.1 apresentamos alguns trabalhos relacionados a diversas áreas da computação que também têm estudado mudanças que ocorrem ao longo do tempo. Por exemplo, mudanças em redes sociais, mudanças no comportamento de usuários de comércio eletrônico, entre outros. Além disso, nessa seção apresentamos alguns trabalhos de classificação que abordam os aspectos temporais, porém de forma bastante superficial se comparado ao trabalho apresentado nesta tese.

Na Seções 2.2 e 2.3 apresentamos dois conjuntos de trabalhos muito correlatos: classificação adaptativa de documentos e mudanças de Conceitos, respectivamente. Ambos os conjuntos de trabalhos consideram a existência de mudanças ao longo do tempo nas coleções e que essas mudanças podem afetar a qualidade da tarefa de classificação, conforme consideramos neste trabalho. Enquanto o primeiro conjunto de trabalhos procura contornar essas mudanças por meio de técnicas que visam encontrar associações

entre termos, o segundo procura contornar esses problemas mantendo como conjunto de treinamento apenas uma porção mais recente dos documentos. Apesar desses trabalhos considerarem a existência dos efeitos temporais nas coleções, em nenhum deles esses efeitos são detalhadamente estudados e compreendidos como em nosso trabalho. Além disso, esses trabalhos estão voltados para fluxo de documentos (*streams*), em que apenas documentos novos precisam ser classificados, diferentemente de nossa proposta, que explorando os contextos originais dos documentos, pode ser aplicada para se classificar tanto novos documentos como documentos antigos ainda não classificados.

Por fim, na Seção 2.5 apresentamos alguns trabalhos relacionados à filtragem adaptativa de informação, uma classificação binária (relevantes ou não) de acordo com os interesses dos usuários. Nesse conjunto de trabalhos, a evolução temporal está associada às mudanças nos interesses dos usuários ao longo do tempo. Assim, esses trabalhos procuram alterar o modelo de classificação de acordo com as mudanças de interesse dos usuários e o desafio, portanto, é detectar essas mudanças. Apesar de não ser o objetivo desta tese, nossas estratégias podem ser facilmente adaptadas para tratar esse problema.

## 2.1 Evolução Temporal em Aplicações Web

As mudanças que ocorrem ao longo do tempo têm sido alvo de estudos em diversas áreas da computação ligadas à Web, como recuperação de informação, redes sociais, comércio eletrônico, etc. A evolução da Web, por exemplo, foi estudada recentemente (Brewington & Cybenko, 2000; Cho & Garcia-Molina, 2000; Fetterly et al., 2004). Brewington & Cybenko (2000) descrevem alguns resultados de análises estatísticas da freqüência e a natureza das modificações que acontecem nas páginas da Web ao longo do tempo. Utilizando modelos empíricos e uma nova métrica analítica denominada *up-to-dateness*, os autores estimam a taxa com que as máquinas de busca devem reindexar para se manterem atualizadas, de acordo com a taxa de mudanças estimada para as coleções. Outro trabalho que também aborda a evolução da Web é apresentado

por Cho & Garcia-Molina (2000). Nesse trabalho, foram coletadas cerca de 720.000 páginas da Web ao longo de quatro meses, organizado-as em uma base diária, contando as páginas que tiveram seu *checksum MD5* alterado nesse período. Eles descobriram que 40% de todas as páginas da Web do conjunto coletado mudaram dentro de uma semana e 23% desse conjunto, que pertenciam aos domínios *.com*, mudaram diariamente. Fetterly et al. (2004) apresentam uma extensão desse trabalho em que esse conjunto foi ampliado para 150.836.209 páginas da Web e outras informações além do *checksum MD5* foram analisadas, como, por exemplo, status HTTP da página, tamanho da página, vetor de palavras das páginas etc. Foi constatado que o grau de mudanças nas páginas variam de acordo com o seu domínio e que páginas maiores mudam mais freqüentemente que páginas menores.

Esses estudos de evolução da Web motivaram outros esforços que investigam como os aspectos temporais podem ser utilizados para melhorar a qualidade de alguns processos específicos em recuperação de informação (Jones & Diaz, 2007; Berberich et al., 2004; Yu et al., 2004). Jones & Diaz (2007) mostram que levar em consideração os aspectos temporais na geração de resultados de consultas em recuperação de informação é muito importante e pode melhorar significativamente a qualidade dos resultados. Analisando a linha de tempo do resultado das consultas, é possível caracterizar o quão um tópico é temporalmente dependente e quão relevante são os resultados em função do tempo. Eles concluem que meta-características associadas a consultas podem ser combinadas com técnicas de recuperação de informação para melhorar o entendimento e o tratamento de busca de texto em documentos datados. Berberich et al. (2004) propõem um algoritmo, chamado de *T-Rank*, o qual estende o *PageRank* (Brinkmeier, 2006) para melhorar a ordenação por meio da proximidade temporal entre as páginas Web. Yu et al. (2004) adaptam o algoritmo *PageRank* para ponderar cada citação de acordo com a data da citação. Além desses trabalhos, existem outros esforços que exploram a evidência temporal em recuperação de informação na Web. Um resumo bastante completo desses esforços é apresentado por Nunes (2007).

Recentemente, o estudo das mudanças temporais em redes sociais vêm ganhando

importância (Falkowski et al., 2006a; Lauw et al., 2005). Uma rede social consiste de pessoas que interagem umas com as outras por meio de comunidades *online*, compartilhando informações. O estudo dessas redes e da relação temporal existente entre seus membros e os interesses dos mesmos é extremamente útil em tarefas como investigações criminais, agentes de recomendação de compra, etc. Falkowski et al. (2006a) apresentam duas abordagens para analisar a evolução de dois tipos de comunidade *online*, uma abordagem que permite avaliar a evolução em comunidades com taxa de adesão baixa e outra para detecção de comunidades em ambientes de alta variabilidade de entrada e saída de membros. Para ambos os métodos, os autores apresentam resultados experimentais para dados reais de comunidades de estudantes. Lauw et al. (2005) propõem um modelo de criação de redes sociais baseado em informações espaço-temporais dos indivíduos, como, por exemplo, origem geográfica dos acessos ao longo do tempo, atividades da Internet acessadas ao longo do tempo, entre outras. Em outras áreas da computação ligadas à Web, como comércio eletrônico e classificação de e-mails, também encontramos alguns trabalhos que consideram propriedades temporais (Kiritchenko et al., 2004; Tseng et al., 2006). Kiritchenko et al. (2004) apresentam uma nova e eficiente solução de classificação de e-mails que utiliza características das informações temporais dos e-mails combinadas com as características tradicionais como assunto, corpo da mensagem, etc. Tseng et al. (2006) apresentam um método de construção de modelos de previsão de comportamento de usuários de serviços de comércio que utiliza informações temporais. Esse modelo é utilizado na personalização dos serviços.

No processo de classificação de documentos existem alguns trabalhos que abordam aspectos temporais (Lanquillon & Renz, 1999; Lewis, 1995; Liebscher, 2004), no entanto de uma forma bastante superficial se comparado com a caracterização apresentada nesta tese e também com os demais trabalhos apresentados nas seções seguintes. Lanquillon & Renz (1999) baseiam o trabalho na idéia de garantir a precisão dos classificadores. Eles argumentam que basicamente existem duas estratégias para se adaptar um classificador a mudanças. O classificador (modelo de classificação) pode ser com-

pletamente recriado de acordo com o conjunto de treino corrente ou o classificador pode ser atualizado baseado somente em alguns exemplos. O trabalho ainda discute as dificuldades associadas a cada uma dessas técnicas e, devido a esse fato, o trabalho se restringe somente em detectar mudanças nos textos, por meio de métricas de precisão, não lidando com a fase de adaptação. Lewis (1995) apresenta sistemas autônomos de classificação de textos, em que ele sugere monitorar o comportamento de uma coleção com o objetivo de se adaptar a mudanças, se necessário. Liebscher (2004) apresenta uma caracterização simplificada da distribuição dos termos ao longo do tempo (o segundo aspecto temporal mencionado neste trabalho). Assim como em nosso trabalho, o autor assume que os termos podem perder importância em uma classe e ganhar em outra ao longo do tempo. Além disso, baseado nessa caracterização, o autor propõe um novo algoritmo de classificação para contornar os problemas relacionados à distribuição dos termos ao longo do tempo. Entretanto, os outros dois aspectos temporais (distribuição de classes e similaridade entre classes) não são considerados. Apesar de algumas premissas desses esforços serem similares à adotada nesta tese, nenhum deles trata do problema de construção de classificadores que sejam efetivamente robustos em relação à evolução temporal.

## 2.2 Classificação Adaptativa de Documentos

Uma linha de pesquisa em que os aspectos temporais estão sendo estudados mais detalhadamente é classificação adaptativa de documentos (Cohen & Singer, 1999). Essa linha abrange um conjunto de técnicas que visam contornar os problemas relacionados aos aspectos temporais e a outros aspectos com o objetivo de melhorar a efetividade e a precisão dos classificadores de documentos por meio da adaptação incremental e eficiente desses classificadores (Liu & Lu, 2002). Classificação adaptativa de documentos traz uma variedade de desafios para mineração de textos que serão discutidas a seguir.

O primeiro desafio é o conceito de contexto adotado por esses trabalhos e como esse conceito pode ser explorado para se alcançar melhores modelos de classificação. Um

contexto é uma partição de dados semanticamente significativa. Seleção de contextos pode ser adotada para se reduzir a complexidade associada à geração de um modelo de classificação. Trabalhos anteriores nessa linha identificaram duas formas de geração de contexto: termos vizinhos que estão próximos a uma palavra-chave (Lawrence & Giles, 1998) e termos que indicam o escopo e a semântica dos documentos (Caldwell et al., 2000). A maioria dos estudos focam no primeiro tipo de geração de contexto. Entretanto, determinar termos semanticamente significativos para cada classe tem uma grande aplicabilidade. Esses termos podem ser usados para definir categorias de contextos. Ao contrário de termos vizinhos, que são extraídos com base na proximidade desses com um termo simples, contexto de termos são determinados através da análise de múltiplos documentos de múltiplas categorias.

Entretanto, tanto o conteúdo quanto o vocabulário das coleções podem evoluir à medida que novos documentos são adicionados. Assim, o segundo desafio está relacionado à criação de modelos de classificação de forma incremental. Como uma coleção evolui, o contexto que caracteriza cada uma das classes também evolui. Obviamente, como o vocabulário pode evoluir, nenhum contexto pode ser presumido. Assumindo um contexto e incrementando esse contexto de forma a incorporar os termos que surgem com os novos documentos, termos inapropriados podem introduzir ineficiência e erros no classificador de documentos. Assim, a construção de um contexto "ótimo", caso ele exista, consiste de uma série de processos de ajuste (remoção e adição de informações), que precisam ser refeitos à medida que novos documentos sejam inseridos. A repetição desses processos de ajuste na construção dos modelos, a cada novo documento inserido, é claramente impraticável em termos computacionais. Por exemplo, Kim et al. (2004) argumentam que em domínios onde a informação muda continuamente, por exemplo, a Web, um processo incremental de construção da base de conhecimento (modelo de classificação) é essencial. Para tanto, eles propõem um método baseado em *Multiple Classification Ripple Down Rules* (MCRDR) que utiliza regras de classificação onde, a partir de conjunto inicial de regras, novas regras de exceção são inseridas para captar as mudanças ao longo do tempo.

O terceiro desafio é a eficiência dos classificadores de texto. O uso de classificadores é, na maioria dos casos, mais freqüente que sua construção. A eficiência computacional desses classificadores é indispensável no processo de gerência da informação e do conhecimento. Dessa forma, o fator chave para melhorar a precisão dos classificadores é a adaptação desses de forma incremental, no entanto essa adaptação precisa ser computacionalmente eficiente. Novamente, Kim et al. (2004) também consideram esse desafio, demonstrando que o método por eles proposto é computacionalmente eficiente, além de eficaz em termos de precisão de classificação.

Apesar de existirem esses diversos esforços para identificar novos contextos e para lidar com os desafios previamente mencionados, nenhum desses esforços está concentrado na classificação dos documentos de acordo com seu contexto temporal original, isto é, classificar um documento de acordo com o período no tempo em que ele pertença. Além disso, nenhum desses trabalhos explora a relação existente entre esses contextos e o tempo propriamente dito. Esses estudos focam apenas em identificar o surgimento de novos contextos, sem relacioná-los com seu tempo cronológico.

## 2.3 Mudanças de Conceitos

Outro conjunto relevante de esforços com interesse na questão temporal está relacionado a mudanças de conceitos ou de tópicos (*concept drift*) (Tsymbal, 2004). Uma vez que os conceitos e interesses se modificam ao longo do tempo e que essas mudanças podem tornar inconsistentes modelos de classificação construídos com dados antigos, o problema de *concept drift* consiste em identificar essas mudanças mantendo o modelo de classificação consistente com os conceitos e com assuntos atuais. Existem diversas propostas de várias técnicas que visam lidar com esse problema, sendo muito importantes em cenários como detecção de fraude e detecção de invasão de rede, entre outros.

Um subconjunto de trabalhos relacionados à *concept drift* concentra-se no uso de *boosting*, ou seja, na combinação de vários modelos de classificação gerados a par-

tir de distintos algoritmos de classificação (Kolter & Maloof, 2003; Wang et al., 2003; Folino et al., 2007; Scholz & Klinkenberg, 2007). Kolter & Maloof (2003) apresentam uma técnica em que cada novo documento é classificado de acordo com a classe mais votada entre os modelos gerados. O modelo de classificação de cada algoritmo é reconstruído de acordo com o desempenho dos mesmos, ou seja, quando o modelo de um determinado algoritmo passa a classificar de forma divergente da maioria dos modelos, esse modelo é reconstruído. Wang et al. (2003) propõem um arcabouço para mineração de mudanças de conceitos em que vários classificadores podem ser acoplados. A ponderação dos modelos de classificação é feita de forma criteriosa, baseada na precisão esperada de cada modelo. Folino et al. (2007) apresentam uma outra técnica em que vários modelos de classificação são gerados a partir de diferentes partes do conjunto de treinamento. Posteriormente, esses modelos são combinados para gerar um modelo único para classificação (utilizando programação genética). Scholz & Klinkenberg (2007) apresentam um método que adapta-se naturalmente às mudanças de conceito e permite quantificar as tendências baseado nos modelos dos classificadores que o compõem. Eles ainda demonstram empiricamente que o método supera algoritmos baseados em aprendizado de máquina que não consideram mudanças de conceito.

Outro subconjunto de trabalhos concentra-se na utilização de uma janela contendo a parte mais representativa do conjunto de treinamento (Schlimmer & Granger, 1986; Klinkenberg & Joachims, 2000; Klinkenberg, 2004; Widmer & Kubat, 1996; Forman, 2006; Lazarescu et al., 2004). Schlimmer & Granger (1986) argumentam que, em domínios de classificação de fluxo de documentos, podem existir mudanças de conceitos e falsas mudanças de conceitos, que eles denominam de ruído. Com base nisso, os autores apresentam um método capaz de identificar as mudanças de conceitos e diferenciá-las dos ruídos. Klinkenberg & Joachims (2000) apresentam um novo método para reconhecer mudanças de conceito e lidar com tais mudanças utilizando *Suport Vector Machines* (SVMs) (Joachims, 1998a). Esse método mantém uma janela contendo os documentos de treinamento mais recentes e a idéia é ajustar automaticamente o tamanho dessa janela

em função da generalização do erro estimado. Klinkenberg (2004) apresenta várias técnicas que também utilizam *Suport Vector Machines*. Nessas técnicas, uma janela temporal adaptativa é mantida selecionando exemplos representativos do conjunto de treinamento ou ponderando os exemplos do treinamento de acordo com sua representatividade. A idéia aqui também é ajustar o tamanho da janela, selecionando ou ponderando os documentos de treinamento de forma que o erro estimado seja minimizado. Widmer & Kubat (1996) descrevem um conjunto de algoritmos que são capazes de reagir, detectar mudanças de conceitos, tomando vantagem de situações em que um determinado conceito ressurge. A idéia por trás dos algoritmos que esse trabalho apresenta está em manter uma janela com exemplos verdadeiros do conceito atual e armazenar conceitos antigos para reutilizá-los quando necessário. Forman (2006) apresenta o *Daily Classification Task* (DCT), um arcabouço que lida com a classificação de documentos com o objetivo de contornar os aspectos temporais. Nesse trabalho, o tempo é discretizado em períodos, (e.g., dias). Para cada período discretizado, um amostra randômica de tamanho limitado de documentos já rotulados é fornecida como conjunto de treinamento. Uma métrica de desempenho, como precisão de classificação ou *F-measure*, é calculada para cada dia do conjunto de dados de teste, e a média é reportada para todos os dias. Lazarescu et al. (2004) apresentam uma abordagem alternativa para lidar com *concept drift*. Diferentemente das demais abordagens, o método apresentado utiliza três janelas de diferentes tamanhos contendo documentos de treinamento. Através de uma combinação dessas janelas o método consegue identificar as mudanças, adaptando-se progressivamente às mesmas.

Na maioria desses trabalhos, os autores consideram a existência dos efeitos temporais sem entender exatamente quais são esses efeitos. Eles discutem que apesar do fato que algumas classes apresentam explosões de popularidade em alguns períodos, ou seja, determinados assuntos se tornam mais importantes em determinados períodos, a verdadeira identidade de uma classe não se modifica. Em nosso trabalho, por outro lado, demonstramos através de uma exaustiva análise empírica que existem outros fatores além da variação da popularidade das classes e que utilizando esses aspectos temporais podemos

melhorar os classificadores.

## 2.4 Filtragem Adaptativa de Informação

Uma outra linha de pesquisa em que existe a preocupação pela questão temporal é a filtragem adaptativa de informação (Riordan & Sorensen, 2002; Hanani et al., 2001). Filtragem de informação é descrita como um problema de classificação binária em que os documentos são classificados como relevantes ou não, de acordo com o interesse do usuário. De acordo com Dumais et al. (1998), a construção de modelos de classificação em domínios estáticos é suficientemente bem controlada. Diversas aplicações assumem que a distribuição dos dados de treino e a distribuição de novos dados são bastante similares. Essa afirmação pode ser verdadeira para aplicações que duram um curto período de tempo, mas se torna inapropriada para aplicações com longos períodos de duração. Nesse caso, é inevitável adaptar os classificadores para novas situações com o objetivo de manter a qualidade da classificação.

Nesse contexto, Klinkenberg fez algumas investigações interessantes relacionadas à filtragem adaptativa (Klinkenberg & Renz, 1998). Nesse trabalho ele explora métodos para reconhecer mudanças de conceito e para determinar janelas de interesse dos usuários nos dados de treinamento, cujo o tamanho pode ser fixo ou automaticamente adaptado para captar as mudanças de conceito correntes. Entretanto, a efetividade dessa abordagem varia de acordo com as mudanças de interesse dos usuários ao longo do tempo. Um outro trabalho relevante é apresentado por Allan (1996). Nesse trabalho, é verificado a relevância incremental das respostas dos usuários no processo de filtragem de informação e também avaliaram as mudanças de interesse dos usuários. Allan et al. (1998) também avaliou aspectos dinâmicos associados à detecção de tópicos e temas. Yang & Kisiel (2003) desenvolveram um nova técnica chamada de *Margin-based Local Regression*, a qual, automaticamente, ao longo do tempo, delimita a fronteira entre documentos relevantes e não relevantes, baseado na importância das respostas anteriores, ordenadas cronologicamente,

do sistema de recuperação de informação, otimizando, conseqüentemente, a precisão do classificador ao longo do tempo.

Outro conjunto de trabalhos é apresentada por Arampatzis et al. (Arampatzis et al., 2000b,a; Arampatzis & van der Weide, 2001). Arampatzis et al. (2000b) investigam a utilização das mudanças temporais na tarefa de filtragem de informação. Nesse trabalho é apresentado um método de seleção de termos em filtragem de informação chamado *Term Occurrence Uniformity* (TOU), que se baseia na hipótese de que os termos que ocorrem uniformemente no tempo são mais importantes que os demais, apresentando resultados bastante promissores. Arampatzis et al. (2000a) também investigaram a importância de um documento de treino ao longo do tempo. Baseado nessa investigação, eles propõem uma técnica de filtragem de documentos incrementalmente adaptativa. Por fim, Arampatzis & van der Weide (2001) propõem uma outra técnica de filtragem adaptativa em que a evolução dos interesses dos usuários é considerada. Além disso, esse trabalho apresenta um resumo dos outros trabalhos do autor relacionados a filtragem adaptativa, além de outros trabalhos que tratam marginalmente de *concept drift*, apresentada anteriormente, por se tratarem de linhas de pesquisa muito correlatas.

Nesse conjunto de trabalhos, o grande desafio é detectar as mudanças de interesse dos usuários e adaptar o modelo de classificação a essas mudanças. Apesar de o trabalho apresentado nesta tese estar relacionado a mudanças temporais nas coleções de documentos e como isso afeta os classificadores, a mesmo pode ser também utilizado para entender as mudanças que ocorrem nos interesses dos usuários. Além disso, as heurísticas apresentadas nesta tese também podem ser adaptadas para classificar documentos de acordo com os interesses dos usuários ao longo do tempo.

## 2.5 Sumário

Neste capítulo apresentamos alguns dos trabalhos relacionados com esta tese. Inicialmente listamos esforços em outras linhas de pesquisa ligadas à Web, como recupe-

ração de informação, redes sociais, comércio eletrônico, e classificação de e-mail, em que as mudanças que ocorrem ao longo do tempo também têm sido investigadas. Em seguida apresentamos um conjunto de estudos relacionados à classificação adaptativa de documentos e seus desafios. O objetivo desses trabalhos é a adaptação incremental dos modelos de classificação de acordo com as mudanças que ocorrem com as coleções ao longo do tempo. Posteriormente, apresentamos também alguns trabalhos de identificação de mudanças de conceitos (*concept drift*) em classificação de documentos que também visam adaptar os modelos de classificação às mudanças que ocorrem nas coleções. Por fim, avaliamos alguns trabalhos relacionados a filtragem adaptativa de informação, que consistem em classificar documentos como relevantes ou não, de acordo com os interesses dos usuários, e essa classificação também precisar se adaptar as mudanças de interesse dos usuários.

Apesar de vários trabalhos na literatura considerarem que a evolução temporal é um fator relevante no contexto de CAD, nenhum deles comprova a existência dessa evolução. Nesta tese, por meio de uma exaustiva análise empírica, além de demonstrar a existência dessa evolução, identificamos quais os fatores que a compõem. Baseado nesses estudos, propomos duas heurísticas que visam classificar os documentos de acordo com seus contextos temporais originais, ou seja, de acordo com o período no tempo em que os mesmo pertençam. Conforme veremos nos capítulos a seguir, essas técnicas podem ser facilmente adaptadas para serem utilizadas em qualquer uma das linhas de pesquisa acima descritas.

# Capítulo 3

# Efeitos Temporais em Classificação de Documentos

Neste capítulo apresentamos uma detalhada caracterização da evolução temporal de duas coleções de documentos distintas (ACM-DL e MedLine). Além disso, apresentamos quais os efeitos que compõem essa evolução temporal. Por fim, demonstramos como esses efeitos podem degenerar a qualidade dos algoritmos de classificação automática de documentos. Todos as análises e resultados apresentados neste capítulo foram publicados na *ACM International Conference on Web Search and Data Mining* (Rocha et al., 2008a).

Classificação automática de documentos usualmente segue um estratégia de aprendizado supervisionado em que, primeiramente, é construído o modelo de classificação utilizando documentos pré-classificados e, posteriormente, esse modelo é utilizado para classificar os demais documentos. Construir modelos de classificação de documentos usualmente significa encontrar o conjunto de termos discriminativos que melhor identifica as classes de documentos. Durante esse processo, vários desafios, que não são necessariamente relacionados ao tempo, devem ser tratados. Alguns desses desafios são discutidos a seguir.

O primeiro desafio é lidar com a distribuição das classes, isto é, algumas classes são muito mais freqüentes que outras, tornando complicada a tarefa de gerar modelos de

classificação que não sejam tendenciosos. O segundo desafio está relacionado ao poder discriminativo dos termos para uma classe específica, o qual chamaremos de distribuição de termos. O poder discriminativo dos termos está relacionado à qualidade com que esses termos podem prever uma determinada classe. Nesse caso, podemos perceber que alguns termos são mais discriminativos para uma dada classe do que para outras. O terceiro desafio é a sobreposição entre as classes em relação aos termos que estão associados a elas, ou seja, aparecem em mais de uma classe, o qual chamamos de similaridade entre classes. Quanto maior é a sobreposição, mais difícil se torna a geração de um bom modelo que as distingue. É importante notar que a intensidade de cada um dos desafios está negativamente correlacionada com a precisão e a generalidade do modelo.

Conforme veremos a seguir, a evolução temporal pode agravar bastante os desafios listados acima, no sentido que os modelos baseados em toda a coleção de treinamento podem não funcionar muito bem, e considerar a evolução dessa coleção é chave para se obter uma classificação eficaz. Mostramos também que a evolução temporal pode ser uma tendência clara e definida ou uma variação aparentemente randômica, e isso aumenta a dificuldade em se construir classificadores precisos, afetando a tarefa de classificação significativamente. A seguir revisitamos cada um dos desafios mencionados sob o ponto de vista da evolução temporal.

O impacto da evolução temporal na distribuição das classes é a variação da mesma ao longo do tempo, que deve ser quantificada para evitar modelos tendenciosos enquanto tratamos os outros desafios. É importante notar que classes podem surgir ou mesmo desaparecer como uma conseqüência de divisão ou junção, respectivamente, de classes existentes. Por exemplo, as sub-classes *Information Retrieval* e *Artificial Intelligence* na biblioteca digital da ACM (*Association for Computing Machinery*)[1] considerando o esquema de classificação de 1964, pertencem à mesma classe: *Applications*. No novo esquema de classificação da ACM-DL, cada uma delas pertence a classes diferentes, *Information Systems* e *Computing Methodologies*, respectivamente. Em suma, o efeito da evolução

[1] http://portal.acm.org/dl.cfm

temporal na distribuição das classes deve ser qualificado e quantificado, para que assim se possa considerá-los na construção do modelo.

Sob a perspectiva da evolução temporal, o segundo desafio está relacionado com a evolução da distribuição dos termos ao longo do tempo. É importante quantificar e qualificar quando os termos surgem, desaparecem, migram entre as classes ou simplesmente perdem ou ganham poder discriminativo em determinadas classes. Com o objetivo de ilustrar esse desafio podemos retomar o exemplo apresentado no Capítulo 1 relacionado ao termo Plutão. Naquele caso, o termo Plutão perde poder discriminativo na classe astrofísica e ganha na classe mitologia. Intuitivamente, podemos afirmar que a evolução acontece normalmente de forma gradual, onde os períodos temporais mais próximos, a distribuição dos termos são mais similares.

O terceiro desafio aborda a evolução temporal da similaridade entre as classes, sem considerar as razões por trás da intensidade ou grau de similaridade. Por exemplo, vamos supor uma coleção de documentos em que os mesmos estejam organizados em diversas classes, dentre elas as classes Crime e Biologia. Algum tempo atrás as classes Crime e Biologia não eram muito similares, mas recentemente elas têm se tornado similares, uma vez que a análise de DNA vem sendo bastante utilizada em investigações criminais. A similaridade entre classes quantifica quão diferentes ou relacionadas duas classes são. Diferenciamos dois tipos de similaridade entre classes: global e local, que surgem quando consideramos a evolução temporal. A similaridade global considera toda a coleção, enquanto que a similaridade local considera apenas a uma porção da coleção. Note que lidar com a similaridade local pode ser muito difícil, não só pela variação da mesma, mas também pela granulação variável que podemos assumir para ela. Por exemplo, duas classes podem ser muito similares considerando um período de 10 anos, mas podem se tornar diferentes se considerarmos um intervalo de dois anos. É importante perceber que os últimos dois desafios utilizam as diferenças de ocorrência de termos nas classes, isto é, enquanto a distribuição de termos considera as diferenças em relação à mesma classe (características intra-classe), a similaridade entre classe considera as diferenças entre classes

distintas (inter-classe).

Existe um compromisso claro, relacionado à evolução temporal, em encontrar a granulação adequada em termos do intervalo de tempo a ser considerado quando construímos os modelos de classificação. A questão desse compromisso está associada ao efeito amostral no contexto da evolução temporal das coleções. O efeito amostral surge como uma conseqüência da estratégia popular de construção de modelos baseados em uma amostra de documentos da coleção e do fato que essa amostra pode não possuir características suficientes para identificar ou discriminar adequadamente as classes da coleção. Por exemplo, se observamos a coleção como um todo, a mesma pode parecer balanceada em termos de documentos por classe, mas se verificarmos essa distribuição em períodos específicos, o efeito amostral pode surgir. A ocorrência do efeito amostral sugere que devemos aumentar o tamanho da amostra, enquanto que a evolução temporal induz o inverso, uma vez que períodos longos podem resultar em conflitos de evidências.

Determinar a granulação de tempo adequada, ou seja, o período de tempo em que as características sejam estáveis e ao mesmo tempo sejam suficientes em termos de quantidade, é uma tarefa extremamente difícil. Assim, no restante deste capítulo apresentamos uma série de experimentos realizados que visam entender esse compromisso, identificando cada um dos efeitos que fazem parte dele.

## 3.1 Caracterização dos Efeitos Amostrais e Temporais

Nesta seção descrevemos a análise dos dois fatores que contribuem para o compromisso previamente discutido: o efeito amostral e a evolução temporal. Primeiro, discutimos brevemente o efeito amostral a fim de reconhecer a sua existência e sua influência em CAD. Além disso, essa análise é relevante porque mostra que é necessário isolar o efeito amostral nos experimentos para estudar a evolução temporal. Assim, após estudar o efeito amostral, caracterizamos e apresentamos uma metodologia para qualificar e quantificar a

evolução temporal, a qual pode influenciar no desempenho de classificadores automáticos. Com esse propósito, analisamos cada um dos desafios previamente apresentados sob a perspectiva da evolução temporal, os quais chamamos de efeitos temporais.

### 3.1.1 Ambiente Experimental

Para demonstrar a existência da evolução temporal, seus efeitos e o quanto ela pode influenciar no desempenho de classificadores automáticos, realizamos uma série de experimentos utilizando um classificador estado-da-arte, o Support Vector Machine (SVM) (Joachims, 1998a, 2006), em duas coleções de documentos distintas. A primeira coleção de documentos é um conjunto de aproximadamente 30.000 documentos da biblioteca digital da ACM contendo artigos publicados entre os anos de 1980 e 2002. Nessa coleção, são os próprios autores que assinalam seus documentos a uma das classes do esquema de classificação da ACM. Utilizamos apenas o primeiro nível dessa taxonomia, com 11 categorias (Millieux, C. Applications, Methodologies, Information, Mathematics, Theory, Data, Software, CSO, Hardware e GL ), as quais não sofreram alterações no período de tempo considerado. Cada documento foi assinalado a uma única classe. A segunda coleção de documentos é a MedLine, a qual consiste de 4.112.069 documentos, com artigos publicados entre 1970 e 2006, classificados em 7 classes distintas (Toxicology, SpaceLife, History, CM, Cancer, Bioethics, Aids). Todos os documentos estão assinalados a apenas uma classe também.

A métrica de precisão utilizada para quantificar como o desempenho do classificador é afetado pela evolução temporal é a acurácia (*accuracy*) (Lewis, 1995), a qual é calculada como a razão entre o número total de documentos corretamente classificados e o número total de documentos classificados.

### 3.1.2 Efeito Amostral

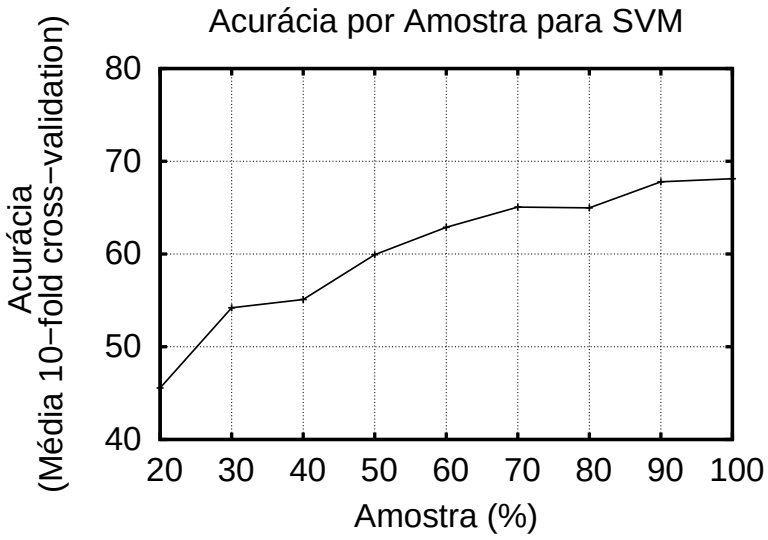
Apesar de o efeito amostral ser um fenômeno bem estudado (Olken & Rotem, 1995; Chawla, 2003), por razão de completeza realizamos um conjunto de experimentos para

ilustrar o impacto desse efeito em classificadores. Especificamente, no nosso caso, precisamos isolá-lo do efeito temporal. Dessa forma, construímos um conjunto de documentos $C$ com documentos pertencentes a apenas um ano, mais especificamente 1999 para ACM-DL e 1970 para a MedLine.

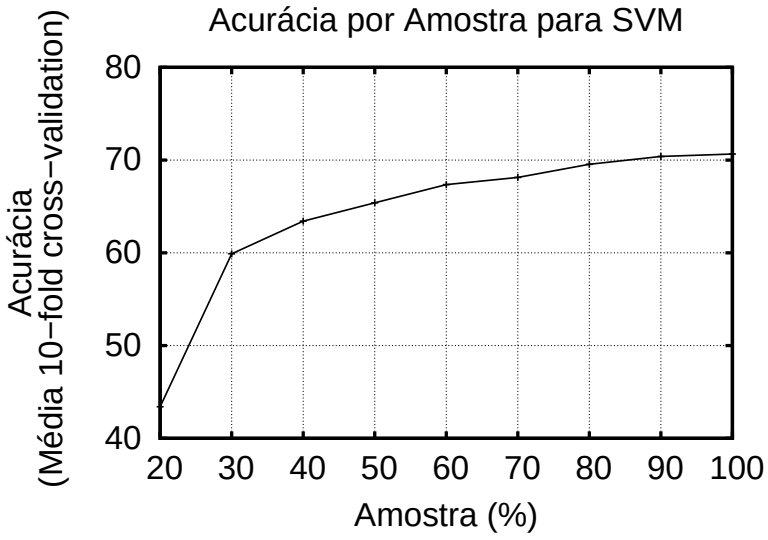
A partir desse novo conjunto $C$, geramos um subconjunto $S$, cujo tamanho representa $X\%$ de $C$, para ser utilizado no processo de classificação. Então, dividimos esse subconjunto $S$ em dez partes, cada parte contendo o mesmo número de documentos. Utilizando essa estratégia de particionamento, realizamos uma validação cruzada de 10 partes (*10-fold cross-validation*) (Brieman & Spector, 1992), variando o valor de $X$ de 20% até 100%, uma vez que todos os documentos pertencem ao mesmo ano, para analisar como a acurácia é afetada pelo tamanho da amostra no processo de classificação. A Figura 3.1 mostra os resultados desses experimentos. Como podemos observar, à medida que aumentamos o tamanho da amostra, o desempenho do classificador também melhora, como esperado.

Em ambas as coleções, o efeito amostral também é influenciado pela variabilidade do número de documentos ao longo dos anos. Na coleção da ACM-DL, o número médio de documentos por ano, em um intervalo de 1980 a 2002, é 1.086,5, com um desvio padrão de 550,8 documentos. O ano com o menor número de documentos é 1980 com apenas 441 documentos, enquanto que o ano de 1999 é um dos anos com o maior número de documentos: 2.728. Na coleção da MedLine, o número médio de documentos por ano é 100.789 com um desvio padrão de 51.649,3 documentos. O menor número de documentos por ano ocorreu em 1970: 34.755 documentos. O número máximo de documentos ocorreu em 2005: 208.878 documentos.

Esses resultados demonstram que é impossível analisar a evolução temporal sem isolar o efeito amostral, que desempenha um papel importante em CAD. Além disso, quando construímos um modelo de classificação, o efeito amostral deve ser levado em consideração.




(a) Coleção ACM-DL




(b) Coleção MedLine

**Figura 3.1.** Demonstração Efeito Amostral

### 3.1.3 Efeitos Temporais

Esta seção está dividida em duas partes. Na primeira parte apresentamos o conjunto de experimentos que foram realizados para demonstrar a existência da evolução temporal e sua influência em CAD. Então, analisamos separadamente cada um dos desafios anteriormente mencionados sob a perspectiva da evolução temporal, ou seja, os efeitos temporais. É importante notar que, para cada efeito temporal, isolamos os demais efeitos

que também podem influenciar na tarefa de classificação.

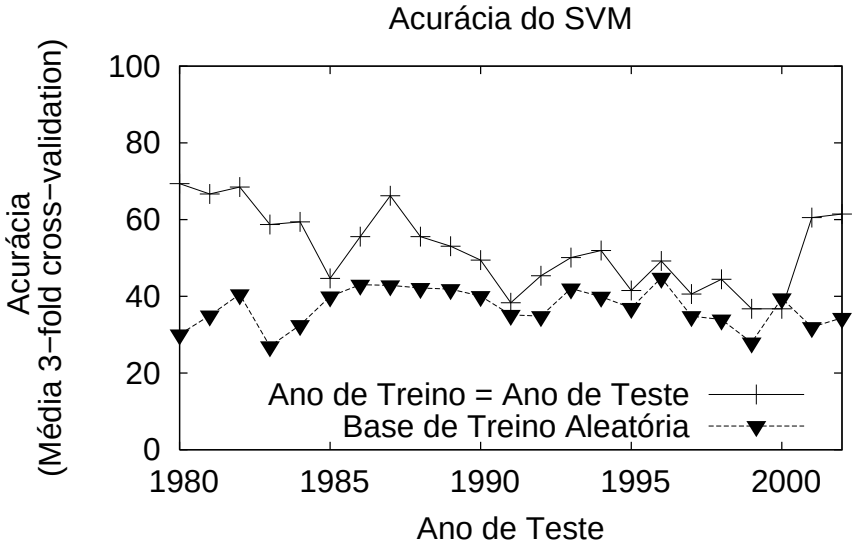
#### 3.1.3.1 Quantificação da Evolução Temporal

Para demonstrar e quantificar o efeito da evolução temporal, realizamos dois conjuntos de experimentos. No primeiro deles, dividimos a coleção original por ano, selecionando aleatoriamente o mesmo número de documentos em cada ano. Utilizamos o menor número de documentos encontrados em um dado ano para cada base, i.e., 441 para ACM-DL e 34.755 para MedLine. Utilizando o mesmo número de documentos para todos os anos, nossa intenção foi isolar os efeitos da evolução temporal do efeito amostral a fim de que pudéssemos analisá-los isoladamente. Feito isso, treinamos o classificador utilizando uma estratégia de validação cruzada de 3 partes (*3-fold cross-validation*) (Brieman & Spector, 1992) para cada ano.

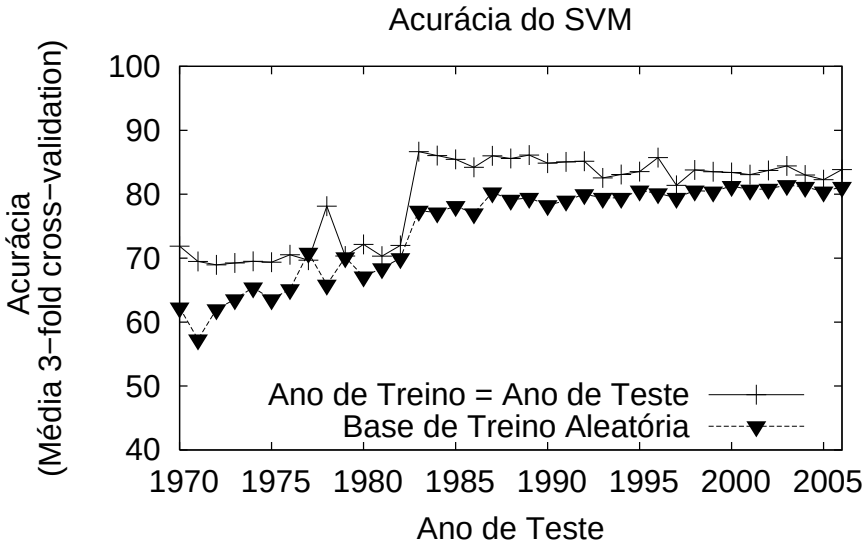
Com o objetivo de contrastar com esse resultado, selecionamos, aleatoriamente, de toda a coleção de documentos, o mesmo número de documentos (441 e 34.755) em ambas coleções para formar um outro conjunto de documentos. Então, dividimos aleatoriamente esse novo conjunto em três partes para aplicar uma validação cruzada de três partes (*3-fold cross-validation*), em que criamos um modelo de classificação para cada uma das possíveis combinações usando duas das três partes existentes. É importante destacar que, enquanto no primeiro experimento o conjunto de treinamento era composto apenas de documentos de cada ano específico, no segundo experimento existem documentos dos vários anos dos conjuntos de treinamento. Posteriormente, testamos cada modelo construído no conjunto de documentos pertencentes a um ano específico $A_i$ criados no primeiro experimento. Variamos $i$ para cobrir todos os anos de cada coleção e calculamos o desempenho médio dos três modelos obtidos com a validação cruzada para cada ano.

A Figura 3.2 mostra a comparação entre os resultados obtidos com esses experimentos. Podemos notar que, para a maioria dos anos em ambas as coleções, a acurácia da tarefa de classificação que considera apenas os documentos do conjunto de treinamento do mesmo ano dos documentos a serem testados foi melhor que a tarefa de classificação

que considera documentos escolhidos aleatoriamente. Note que essa última é a estratégia normalmente utilizada na classificação de documentos.




(a) Coleção ACM-DL




(b) Coleção MedLine

**Figura 3.2.** Demonstração do Efeito Temporal

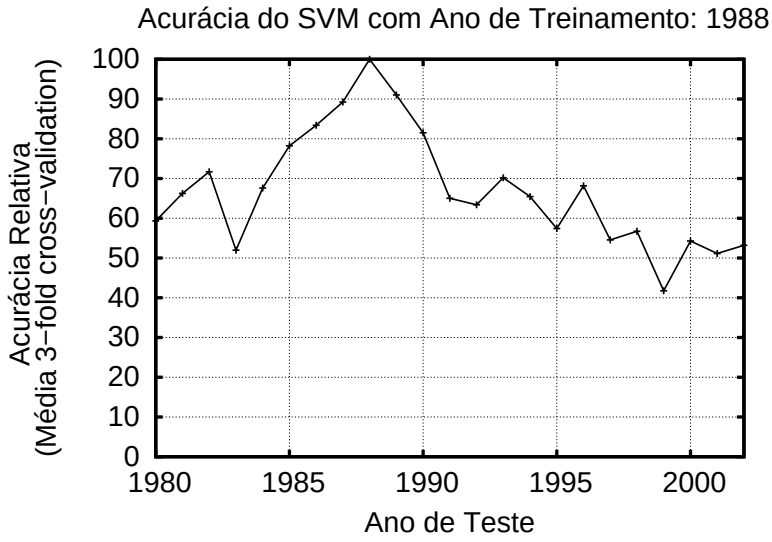
Também caracterizamos o efeito temporal usando uma estratégia diferente. Nesse segundo conjunto de experimentos, conforme feito anteriormente, dividimos a coleção

original pelos anos dos documentos e usamos o mesmo número de documentos por ano. Novamente, dividimos aleatoriamente as bases de cada ano $A_i$ em três partes de mesmo tamanho e usamos duas das três partes como conjunto de treinamento para gerar o modelo de classificação. Esse modelo foi então avaliado sobre os documentos das coleções dos demais anos $A_j$ em que $j \neq i$. Além disso, o modelo foi testado com a porção restante de $A_i$, a qual não foi utilizada na geração do modelo. Repetimos esse processo para cada uma das três combinações possíveis utilizando um processo de validação cruzada de três partes. Após a avaliação dos três modelos gerados a partir de $A_i$, calculamos o desempenho médio dos modelos usados para cada ano testado. Então, ordenamos os valores de acurácia e o maior valor obtido foi transformado em 100 e os demais valores foram relativizados em relação a esse valor.

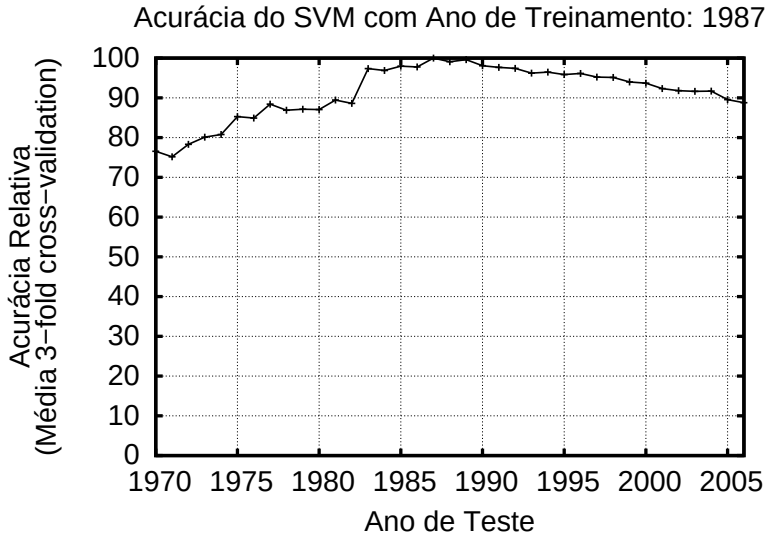
A Figura 3.3 mostra os resultados utilizando os anos de 1988 e 1987 como anos de treinamento para as coleções ACM-DL e MedLine, respectivamente. Podemos observar que existe um pico na acurácia no ano em que as coleções foram treinadas ou nos anos temporalmente próximos. Existe uma visível degradação no desempenho quando o ano dos documentos de teste torna-se mais distantes do ano de treinamento.

A Figura 3.4 sumariza os experimentos descritos acima para ambas as coleções, ACM-DL e MedLine. Para cada ano de treinamento, obtivemos os resultados dos teste após a validação cruzada e ordenamos esses resultados como anteriormente. Então computamos para cada distância temporal (relativa ao ano do conjunto de treinamento) o desempenho médio, o qual corresponde aos pontos realçados nos gráficos. Apresentamos também, para cada desempenho médio, o intervalo determinado pelo desvio padrão. Por exemplo, a distância temporal positiva 10 no eixo $x$ significa que os documentos de teste utilizados são 10 anos mais novos que os documentos de treinamento. Como podemos observar, na maioria dos casos, o melhor cenário encontrado é quando os documentos de treinamento e de teste pertencem ao mesmo ano (intervalo zero).

Os experimentos descritos apresentam uma forte evidência da existência do efeito temporal como um fator que contribui para a degradação do desempenho dos classifica-




(a) Coleção ACM-DL



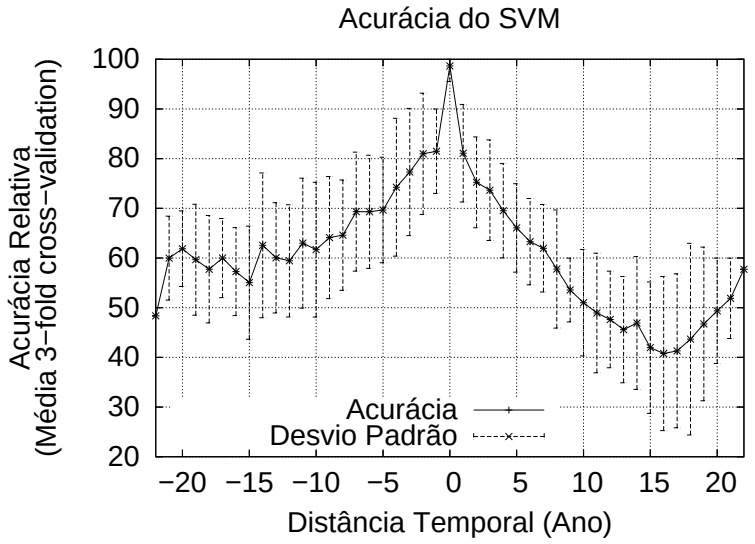

(b) Coleção MedLine

**Figura 3.3.** Localidade Temporal em Classificação - Exemplos

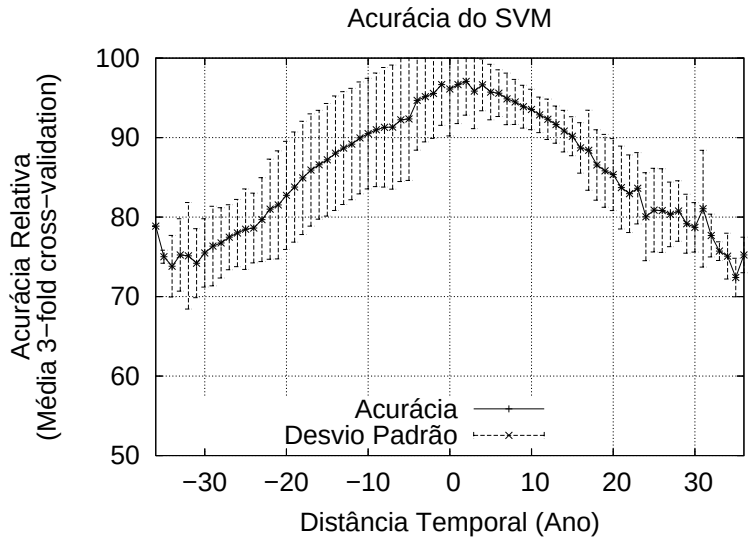
dores automáticos de documentos. A seguir, apresentamos e analisamos os três desafios que podem contribuir para essa degradação, como mencionado anteriormente, chamados: distribuição de classes, distribuição de termos e similaridade entre classes.

#### 3.1.3.2 Distribuição de Classes

Nesta seção analisamos mais detalhadamente o efeito da evolução temporal sob a distribuição de classes. Na Figura 3.5, onde geramos, para cada ano, a distribuição de



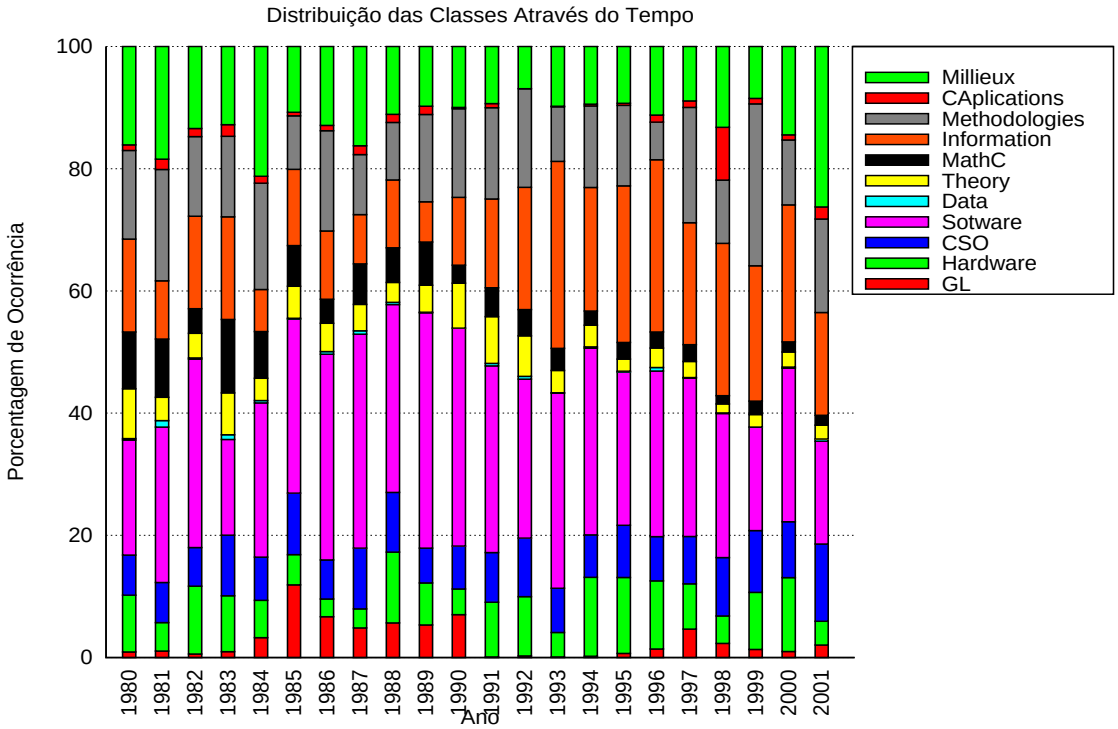

(a) Coleção ACM-DL



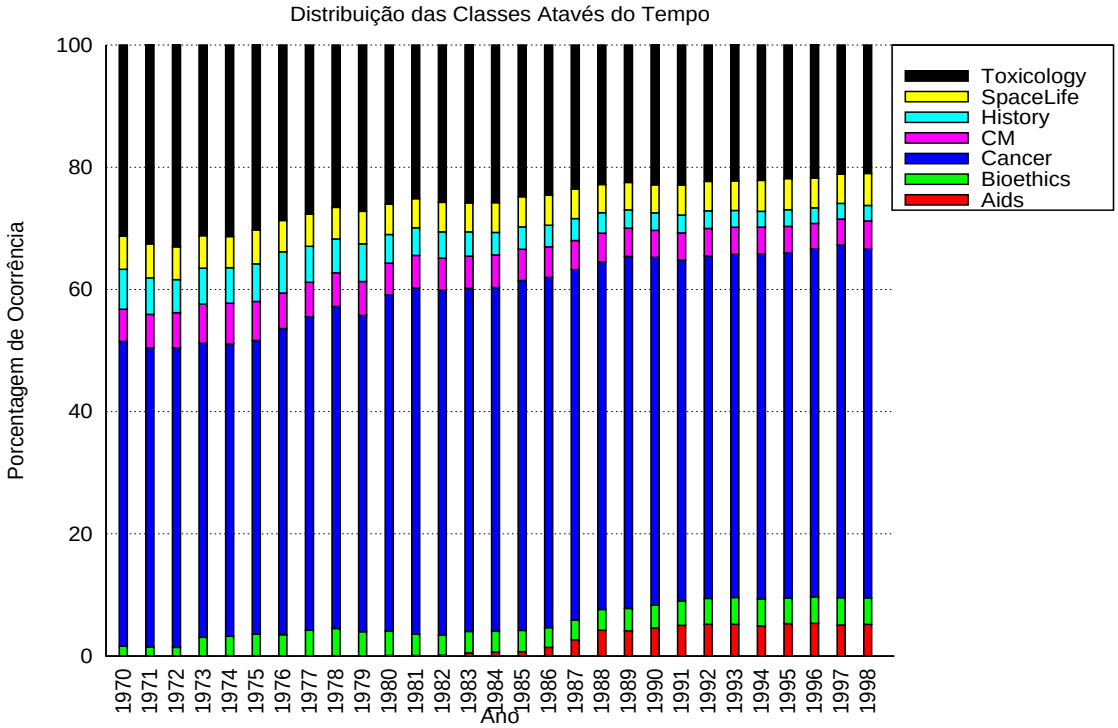

(b) Coleção MedLine

**Figura 3.4.** Localidade Temporal em Classificação - Sumário

probabilidade de classes, ilustramos a variação em termos da representatividade das classes ao longo do tempo para as coleções ACM-DL e MedLine. Como podemos observar, a oscilação da ocorrência das classes ao longo dos anos é um fenômeno freqüente. Por exemplo, na coleção MedLine existem documentos na classe *Aids* datados de 1963, mas a classe se torna significativa somente após 1985. Outras classes, como por exemplo a classe *Mathematics* (Math) da ACM-DL, torna-se menos freqüente com o passar do tempo.




(a) Coleção ACM-DL




(b) Coleção MedLine

**Figura 3.5.** Variação da Ocorrência das Classes Através do Tempo

Para mostrar o efeito dessa oscilação (i.e., distribuição de classes) isoladamente, realizamos o seguinte experimento. Selecionamos aleatoriamente o mesmo número de

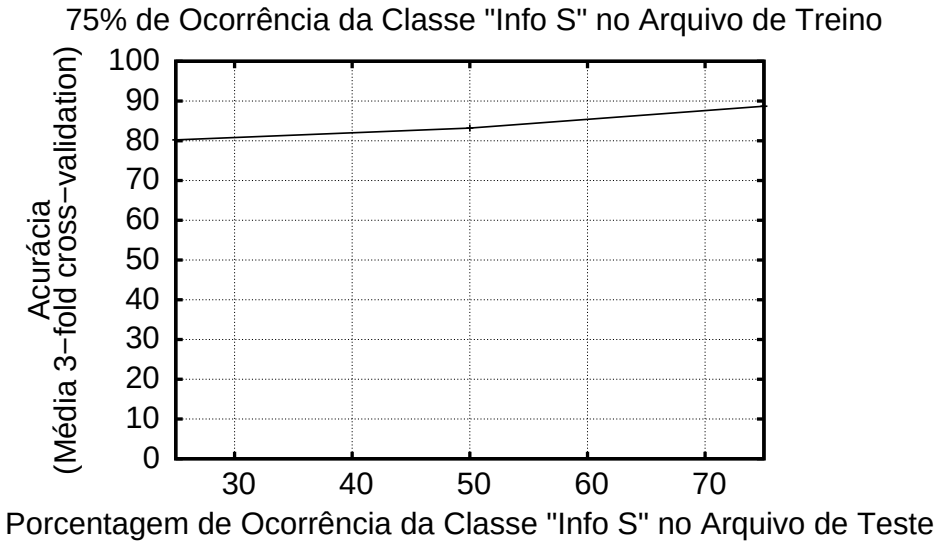
documentos de duas classes diferentes ($C_k$ e $C_l$), em um determinado ano, para cada coleção, para criar outras sub-coleções. Utilizando essas sub-coleções, geramos subconjuntos $T_X$ de mesmo tamanho. Esses subconjuntos foram criados utilizando o seguinte procedimento: $X\%$ dos documentos de $T_X$ são aleatoriamente selecionados da classe $C_k$ e o restante dos documentos de $T_X$ são aleatoriamente selecionados da outra classe ($C_l$). Variamos o valor de $X$ entre 25, 50 e 75. Então, cada um dos subconjuntos $T_X$ foi aleatoriamente dividido em três partes diferentes. Para cada combinação possível entre duas dessas três partes, geramos um modelo de classificação utilizando o SVM. Cada modelo criado foi testado com os outros subconjuntos $T_X$, além de ser testado com a parte restante do mesmo subconjunto $T_X$ do qual ele foi criado.

Realizamos esse experimento para ambas as coleções e os gráficos da Figura 3.6 mostram como a acurácia aumenta à medida que a distribuição de classes do treinamento se aproxima da distribuição de classes do conjunto de teste. Dessa forma, fica claro que a distribuição dos documentos pelas classes influencia no desempenho dos classificadores.

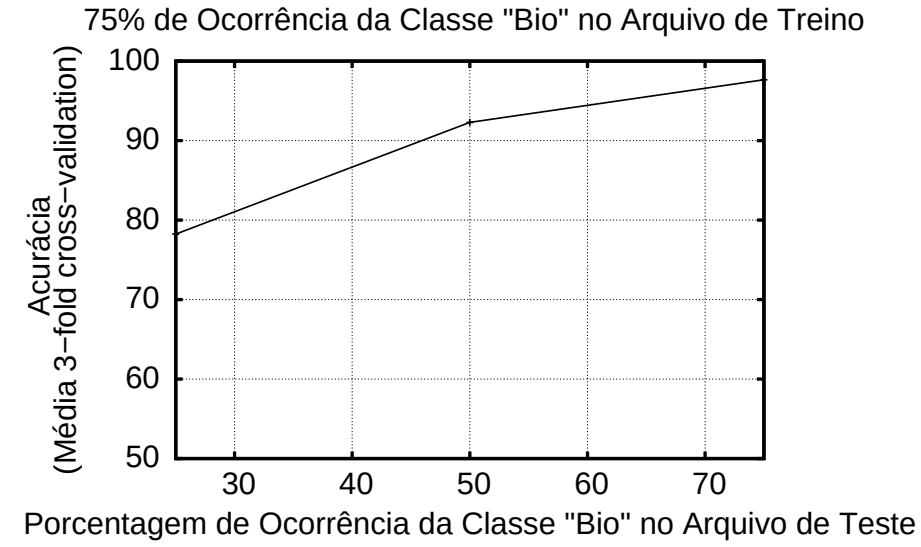
Esses resultados nos mostram que precisamos ser muito cuidadosos ao criar de modelos classificação, uma vez que podemos gerar um modelo tendencioso que pode não ser preciso para o conjunto de dados a ser testado. Quando consideramos que a freqüência das classes muda constantemente ao longo do tempo, isso se torna um problema ainda maior e que precisa ser levado em consideração. Assim, esse efeito temporal é muito importante no processo de construção de modelos de classificação.

#### 3.1.3.3 Distribuição de Termos

Nesta seção analisamos o segundo efeito temporal: a evolução da distribuição dos termos. Este desafio ocorre devido à variação dos termos importantes de uma determinada classe ao longo do tempo e está associado às características intra-classe. Termos podem aparecer, desaparecer ou migrar entre classes e tornam-se mais ou menos discriminativos para uma classe em períodos distintos. O efeito da evolução temporal da distribuição dos termos deve ser quantificada e qualificada para que seja possível considerá-la no modelo




(a) Coleção ACM-DL




(b) Coleção MedLine

**Figura 3.6.** Distribuição de Classes ao Longo do Tempo

de classificação.

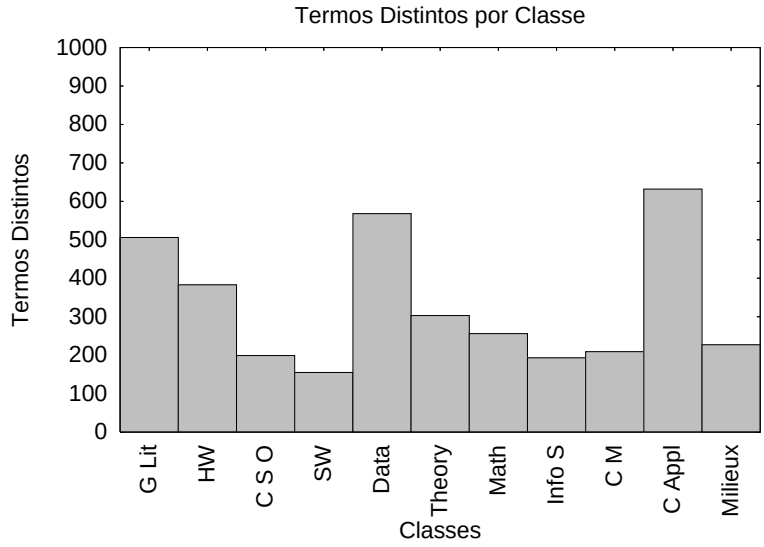
Com o objetivo de demonstrar esse efeito, criamos um vocabulário $V_{k,i}$ contendo os $t$ termos de maior ganho de informação (*info-gain*) (Forman, 2003) em uma classe $C_k$ em um dado ano $A_i$. O *info-gain* mede o quanto um determinado termo é capaz de separar os exemplos de treinamento de acordo com a classificação dos mesmos. Assim, esses termos de maior *info-gain* podem ser considerados os mais discriminativos da classe, de forma isolada, para um dado ano.

A seguir, representamos esse vocabulário em um espaço vetorial onde cada posição no vetor representa um termo contendo o peso do termo baseado em sua freqüência na classe $C_k$ no ano $A_i$. Feito isso, computamos a união de todos os vetores da mesma classe $C_k$ para todos os anos. Dessa forma, seria possível avaliar o quão constante é o vocabulário de uma classe ao longo dos anos. Em teoria, para uma dada classe $C_k$, se os vocabulários para cada ano $V_{k,i}$ ($k$ é classe e $i$ o ano) forem constantes, ou seja, não variarem ao longo do tempo, a união desses vocabulários para a classe $C_k$ deverá conter exatamente $t$ termos. Por outro lado, se todos os vocabulários $V_{k,i}$ forem completamente diferentes uns dos outros, o tamanho da união desses vetores deverá ser $t$ vezes o número total de anos. Para a coleção ACM-DL utilizamos $t = 50$ e para a coleção MedLine usamos $t = 100$, por se tratar de uma coleção maior. A união dos vetores para cada classe em ambas as coleções é apresentado na Figura 3.7. Como podemos observar, a variação dos termos é diferente entre as classes, o que significa que existem algumas classes que são mais dinâmicas que outras em ambas coleções.

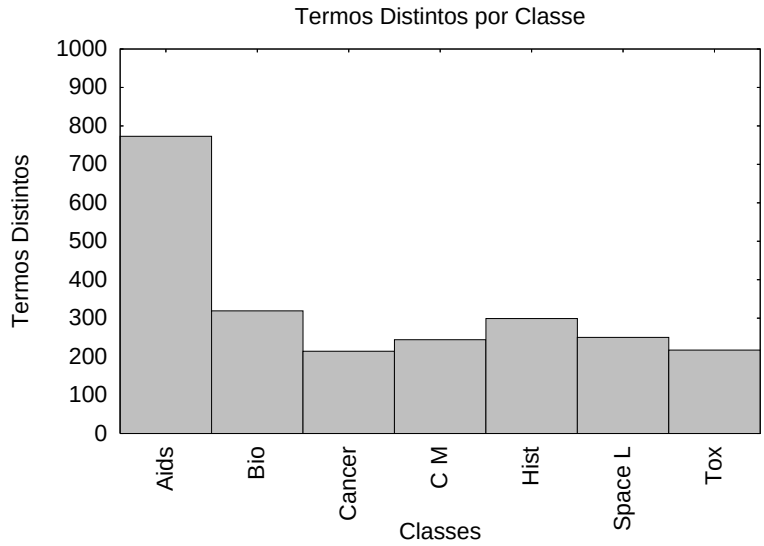
Para melhor caracterizar o efeito da evolução temporal sob a distribuição de termos, executamos o seguinte experimento: dividimos as coleções por ano e, para cada ano, separamos os documentos de acordo com suas respectivas classes. Para cada classe de um certo ano, criamos o vocabulário $V_{k,i}$, composto pelos $t$ termos de maior *info-gain* (Forman, 2003) da classe $C_k$ no ano $A_i$. Então, representamos esse vocabulário como um vetor e verificamos a similaridade do cosseno (Salton & McGill, 1983) entre os vetores de vocabulário $V_{k,i}$ e $V_{k,j}$ da mesma classe sob todos os anos da coleção.

A Figura 3.8 mostra que os vocabulários dos anos próximos tendem a ser mais similares em ambas as coleções[2]. Quanto maior a distância no tempo entre dois documentos de uma certa classe, menor é a probabilidade que os mesmos não tenham termos em comum (esses resultados foram empiricamente recorrentes). A partir desses gráficos, podemos observar que mesmo quando uma certa classe continua a existir (a classe não desaparece, não é dividida e não é agregada a outras classes), seu assunto, suas carac-

[2]Os gráficos da Figura 3.8 não contêm os valores de similaridade do cosseno para o ano de treinamento, uma vez que esse valor será sempre igual a 1



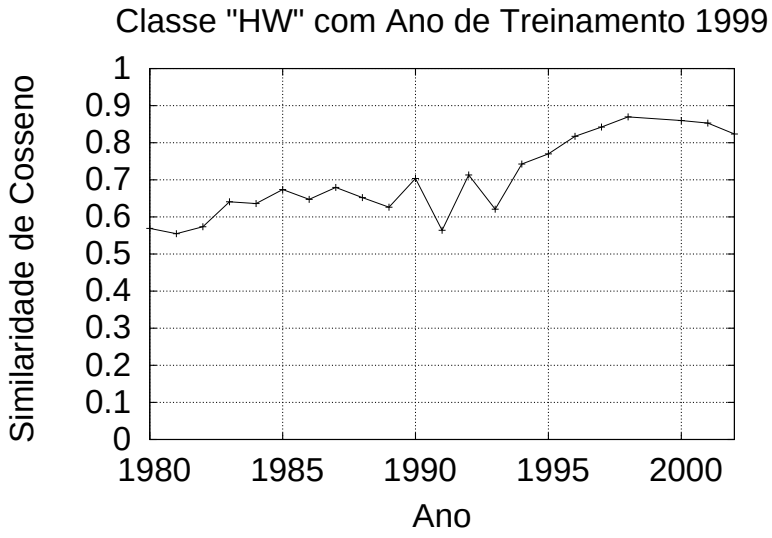

(a) Coleção ACM-DL



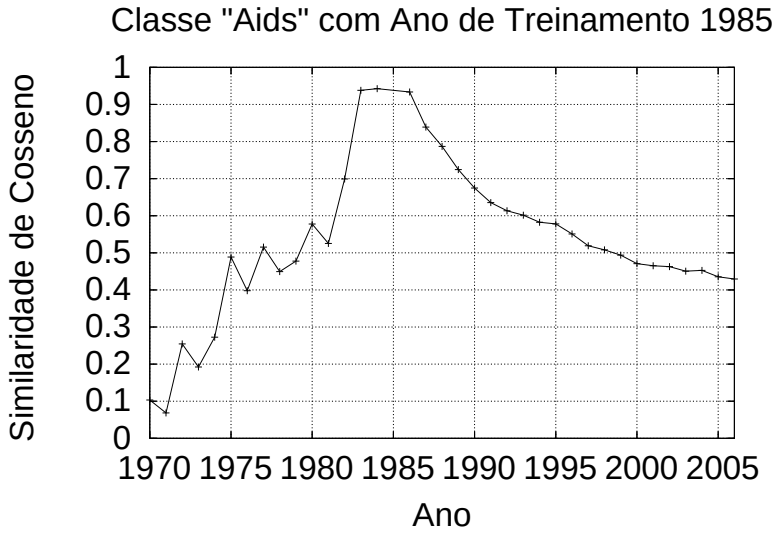

(b) Coleção MedLine

**Figura 3.7.** Análise do Movimento do Termos em cada Classe

terísticas principais podem mudar ao longo do tempo. Por exemplo, apesar de a classe *Artificial Inteligence* estar presente no esquema de classificação desde a criação da ACM-DL, provavelmente em alguns períodos no tempo o assunto mais estudado nessa área tenha sido *neural networks* enquanto que em outros tenha sido *first-order logic*. É importante notar que as classes $Hardware$ (“HW") na coleção ACM-DL e a classe $Aids$ na MedLine possuem algumas diferenças em seus comportamentos. Enquanto a curva da classe $Hardware$ apresenta um declínio mais suave, a curva da classe $Aids$ apresenta um




(a) Coleção ACM-DL



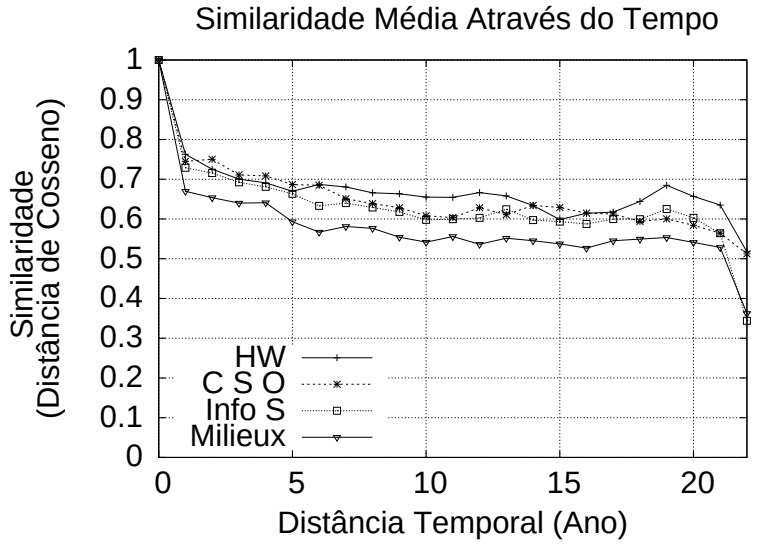

(b) Coleção MedLine

**Figura 3.8.** Distribuição dos Termos ao Longo do Tempo

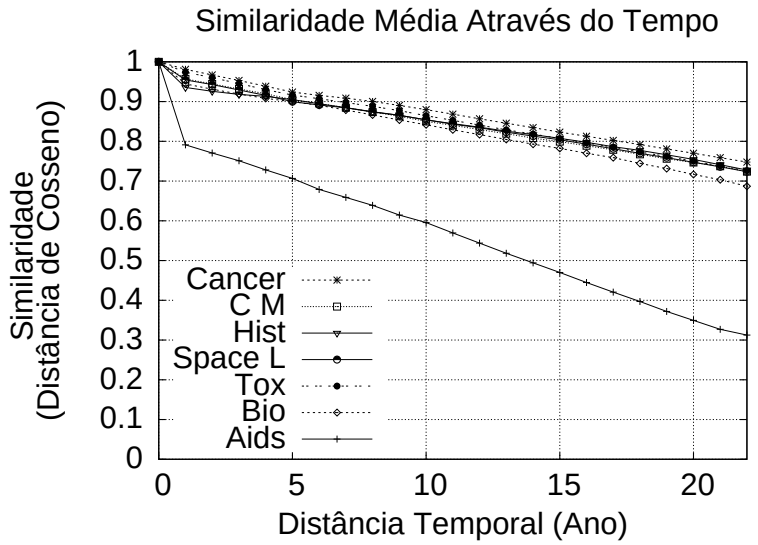
declínio bastante acentuado. Isso mostra que o vocabulário da classe $Hardware$ muda mais suavemente ao longo do tempo enquanto o vocabulário da classe $Aids$ é bem mais dinâmico. Assim, o tempo teve um impacto maior sobre a classificação dos documentos da classe $Aids$ do que sobre os documentos da classe $Hardware$.

A Figura 3.9 mostra a média da similaridade do cosseno das classes quando variamos a distância temporal entre os vetores de vocabulário de cada ano das classes em ambas as coleções. Distância zero significa que comparamos um vocabulário consigo

mesmo, o que obviamente corresponde à similaridade máxima. Podemos observar que, quanto maior a distância temporal entre os vocabulários de uma classe, menos similares eles são, o que demonstra a evolução dos vocabulários de cada classe ao longo dos anos. Um exemplo interessante é a classe $Aids$ da coleção MedLine. Diferentemente das outras classes, a similaridade entre os vetores de vocabulário dessa classe decresce muito rápido, mostrando que a classe $Aids$ é muito dinâmica.




(a) Coleção ACM-DL




(b) Coleção MedLine

**Figura 3.9.** Média da Distribuição dos Termos

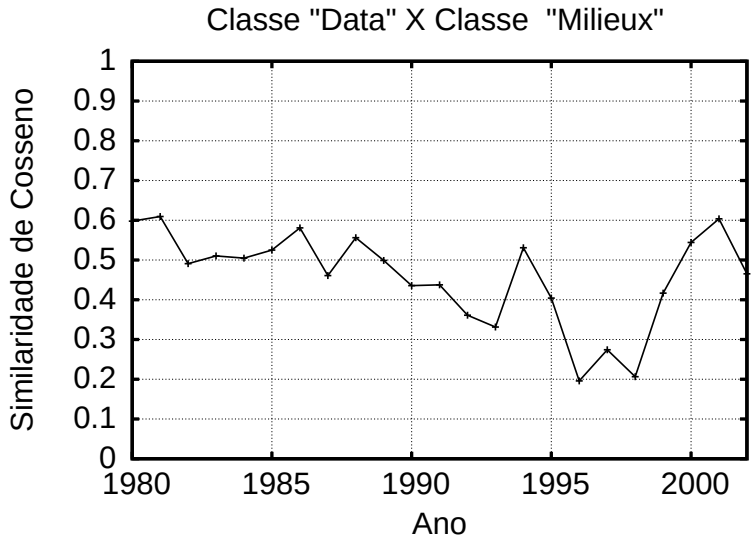
Conforme demonstramos, o vocabulário das classes evolui. Isso torna óbvio que o

modelo de classificação gerado considerando documentos de um certo período no tempo pode ser menos eficaz quando testado utilizando documentos de outro período no tempo, uma vez que o vocabulário pode evoluir no sentido que as premissas construídas podem não ser mais verdadeiras, isto é, os termos discriminativos podem não ser mais os mesmos, tornando esse efeito temporal bastante importante de ser considerado.

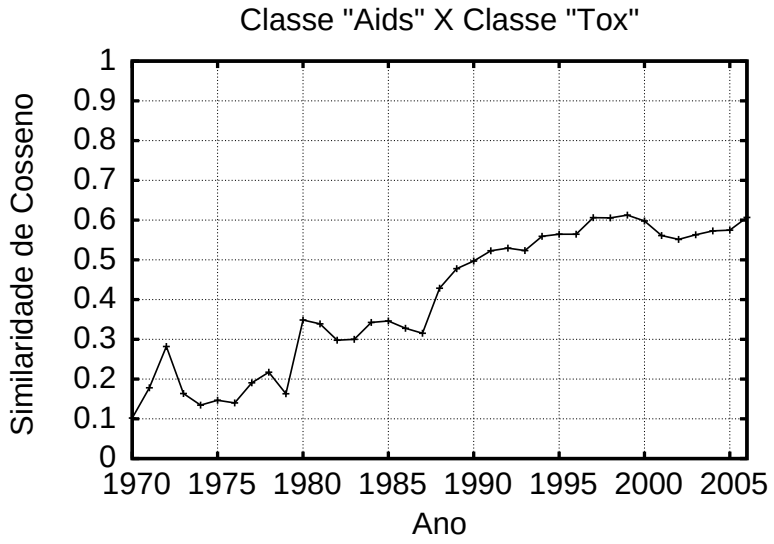
#### 3.1.3.4 Similaridade entre Classes

Finalmente, nesta seção analisamos o efeito da evolução temporal sobre o último desafio: a similaridade entre as classes. Esse efeito temporal é bem interessante devido à migração e à variação da freqüência dos termos nos vocabulários das classes ao longo dos anos. É importante notar que esse efeito está relacionado às características inter-classe.

Para analisar esse efeito temporal mais detalhadamente, realizamos o seguinte experimento. Como fizemos nos outros experimentos previamente apresentados, dividimos a coleção original pelos anos dos documentos e selecionamos aleatoriamente o mesmo número de documentos por classe em cada ano. Também como feito anteriormente, criamos um vocabulário $V_{k,i}$. Para simplificar a análise do experimento, consideramos apenas duas classes e calculamos a similaridade do cosseno entre os vetores que representam os vocabulários de certo ano $A_i$. Variando $A_i$ por todos os anos das coleções, obtivemos o gráfico apresentado na Figura 3.10. Como podemos observar, os vocabulários dessas duas classes são mais ou menos similares em diferentes períodos do tempo. Por exemplo, as classes $Data$ e $Millieux$ da coleção ACM-DL são menos similares entre si no período de 1996 a 1998. Isso indica que, se fôssemos classificar os documentos desse intervalo utilizando como conjunto de treinamento apenas os documentos desse período também, o desempenho do classificador seria melhor que se utilizássemos documentos escolhidos aleatoriamente de vários períodos. Isso aconteceria uma vez que, se considerássemos um vocabulário de treinamento para construir o modelo de classificação para essas duas classes que fosse proveniente de um período diferente do período em questão, as duas classes tenderiam a ser mais similares do que elas realmente são nesse período de tempo. Ob-




(a) Coleção ACM-DL




(b) Coleção MedLine

**Figura 3.10.** Similaridade do Cosseno Através do Tempo

servações similares são válidas para as classes $Aids$ e $Toxicology$ da coleção MedLine. A Figura 3.10 mostra que essas classes estão se tornando mais similares entre si desde 1970. Conseqüentemente, a classificação de ambas as classes é mais fácil de ser feita para documentos dos anos 70 e 80 do que para documentos de 2000.

As Tabelas 3.1 e 3.2 mostram a variação ao longo do tempo da similaridade entre cada par de classes para as coleções MedLine e ACM-DL, respectivamente. O valor de cada entrada na tabela representa o desvio padrão da média de similaridade entre as classes

correspondentes em todos os anos. Como podemos observar, para alguns pares de classes a variação da similaridade é muito alta. Por exemplo, o desvio padrão para similaridade média entre as classes *Complementary Medicine* (CM) e a classe *History* (Hist) é 21%, muito alto. Isto significa que essas duas classes foram muito similares em alguns períodos e pouco relacionadas em outros momentos. Conseqüentemente, a dificuldade em separar essas duas classes também varia consideravelmente ao longo do tempo. Outra observação interessante é que a classe *Cancer* da MedLine difere das outras classes uma vez que a variabilidade de sua similaridade com as demais classes ao longo dos anos é baixa. Isso indica que o tempo afeta menos essa classe do que afeta as outras classes da MedLine.

| | Lit | HW | C S O | SW | Data | Theory | Math | Info S | C M | C App | Millieux |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Lit | 0 | 0,14 | 0,12 | 0,12 | 0,12 | 0,29 | 0,14 | 0,13 | 0.14 | 0.12 | 0.29 |
| HW | - | 0 | 0,08 | 0,13 | 0,11 | 0,12 | 0,11 | 0,12 | 0,10 | 0,10 | 0,13 |
| C S O | - | - | 0 | 0,10 | 0,09 | 0,10 | 0,08 | 0,07 | 0,08 | 0,10 | 0,13 |
| SW | - | - | - | 0 | 0,09 | 0,06 | 0,09 | 0,10 | 0,11 | 0,12 | 0,13 |
| Data | - | - | - | - | 0 | 0,05 | 0,08 | 0,09 | 0,10 | 0,13 | 0,13 |
| Theory | - | - | - | - | - | 0 | 0,14 | 0,13 | 0,07 | 0,06 | 0,29 |
| Math | - | - | - | - | - | - | 0 | 0,13 | 0,10 | 0,09 | 0,15 |
| Info S | - | - | - | - | - | - | - | 0 | 0,10 | 0,08 | 0,15 |
| C M | - | - | - | - | - | - | - | - | 0 | 0,11 | 0,13 |
| C App | - | - | - | - | - | - | - | - | - | 0 | 0,12 |
| Millieux | - | - | - | - | - | - | - | - | - | - | 0 |

**Tabela 3.1.** Matriz de Similaridade - ACM-DL

| | Aids | Bio | Cancer | C M | Hist | Space L | Tox |
|---|---|---|---|---|---|---|---|
| Aids | 0 | 0,19 | 0,16 | 0,18 | 0,19 | 0,18 | 0,19 |
| Bio | - | 0 | 0,04 | 0,20 | 0,17 | 0,19 | 0,12 |
| Cancer | - | - | 0 | 0,04 | 0,03 | 0,04 | 0,05 |
| C M | - | - | - | 0 | 0,21 | 0,08 | 0,05 |
| Hist | - | - | - | - | 0 | 0,20 | 0,11 |
| SpaceL | - | - | - | - | - | 0 | 0,05 |
| Tox | - | - | - | - | - | - | 0 |

**Tabela 3.2.** Matriz de Similaridade - MedLine

Por esses resultados observamos que as definições das classes evoluem ao longo do tempo, ou seja, os assuntos principais de cada classe mudam no decorrer do tempo. Essas mudanças fazem com que as mesmas ora sejam muito similares ora sejam bastante

diferentes. Essa variação na similaridade entre classes é muito importante e deve ser considerada no processo de construção de modelos de classificação.

## 3.2 Sumário

Em suma, baseado na caracterização apresentada acima, foi possível prover evidências que as classes e os termos variam ao longo do tempo. Foi possível perceber também que a similaridade entre as classes também varia. Além disso, demonstramos que essas variações possuem uma grande influência no desempenho dos classificadores.

Observamos também pelos resultados apresentados neste capítulo que existe um grande desafio em lidar com os efeitos amostral e temporal simultaneamente. Esses efeitos indicam estratégias opostas em termos de seleção do conjunto de treinamento. Aumentando o tamanho do conjunto de treinamento podemos introduzir documentos que estão fora do contexto temporal que podem resultar em conflitos de evidências, o que pode reduzir a precisão do classificador. Por outro lado, reduzindo o tamanho do conjunto de treinamento e utilizando apenas documentos muito próximos temporalmente dos documentos de teste, corremos o risco de que a quantidade de informação não seja suficiente para distinguir entre as classes e, conseqüentemente, insuficiente também para construir um modelo de classificação mais preciso.

# Capítulo 4

# Seleção de Contextos Temporais

Conforme vimos no Capítulo 3, existe um compromisso claro entre o efeito amostral e o efeito temporal no processo de seleção do conjunto de treinamento que será utilizado na construção de um modelo de classificação automática de documentos. Enquanto o primeiro efeito sugere utilizar a maior quantidade possível de documentos no conjunto de treinamento (abordagens tradicionais utilizam todos documentos já classificados), o segundo sugere utilizar uma amostra reduzida que seja temporalmente próxima dos documentos a serem classificados. A boa qualidade de um modelo de classificação depende, portanto, da otimização desse compromisso, por meio da seleção do conjunto de treinamento com a maior quantidade possível de documentos já classificados e que estejam temporalmente próximos dos documentos de teste. Denominamos esse problema de otimização do compromisso entre efeito amostral e efeito temporal na seleção do conjunto de treinamento como *seleção de contextos temporais*.

Neste capítulo, primeiramente apresentamos uma formalização para o problema de seleção de contextos temporais. Em seguida, demonstramos o quanto os algoritmos de classificação podem se beneficiar se o conjunto de treinamento for bem escolhido. Essa demonstração é feita por meio da aplicação de uma heurística exaustiva de seleção de contextos temporais, obtendo ganhos significativos de precisão de classificação. Apesar de a heurística utilizada ser apropriada para uma avaliação analítica, pode ser impraticá-

vel em cenários reais. Esse fato nos motiva, assim, a propor soluções eficientes para o problema de seleção de contextos temporais. Assim, baseando-se na caracterização apresentada no capítulo anterior, identificamos três requisitos fundamentais que precisam ser considerados ao selecionar um contexto temporal: *Ponto de Referência, Estabilidade das Características* e *Redução da Incerteza*. Respeitando esses requisitos, propomos um algoritmo genérico que pode ser utilizado como modelo nas soluções que visam selecionar um bom conjunto de treinamento.

## 4.1 Definição do Problema

Primeiramente, antes de formalizarmos o problema de seleção de contextos temporais, definimos o problema de classificação automática de documentos. Seja $D = \{d_1, d_2, \ldots, d_K\}$ o conjunto de documentos treinamento, $C = \{c_1, c_2, \ldots, c_L\}$ o conjunto de categorias (classes) que ocorrem na coleção de documentos, $T = \{t_1, t_2, \ldots, t_N\}$ o conjunto de termos associados com o conjunto de treinamento da coleção e $M = \{m_1, m_2, \ldots, m_P\}$ o conjunto de momentos no espaço temporal (datas) em que os documentos de treino e de teste são criados. Um documento de treino $d_i$ é definido pelo tripla $< x, m_p, c_l >$, onde $x \subseteq T$, $m_p \in M$ e $c_l \in C$. Dado um conjunto $S = \{s_1, s_2, \ldots, s_V\}$ que compreende os documentos que queremos classificar e $E = \{e_1, e_2, \ldots, e_Z\}$ o conjunto de termos associados a $S$, a tarefa de CAD consiste em determinar a classe $c_i$ associada ao documento de teste $s_i$, em que $s_i = \{y, m_p\}$, $y \subseteq E$ e $m_p \in M$.

CAD usualmente segue uma estratégia de aprendizado supervisionada, em que primeiro precisamos selecionar um contexto $X \subseteq D$ que é utilizado para construir o modelo de classificação. Freqüentemente $X$ é igual a $D$ ou uma amostra aleatória de $D$. Então, usamos o modelo construído para classificar os demais documentos que ainda não foram classificados ($S$). Dessa forma, o contexto $X$ é composto por documentos que foram criados em diferentes momentos no tempo ($M$). Como previamente discutido no capítulo anterior, o efeito amostral sugere que devemos maximizar o tamanho de $X$, en-

quanto que os efeitos temporais sugerem o contrário, uma vez que utilizar longos períodos pode gerar conflitos de evidência, os quais podem dificultar a tarefa de criação de um bom classificador. Assim, o problema de seleção de contextos temporais é determinar uma porção do conjunto de treinamento (um contexto) que otimize o compromisso entre o efeito amostral e os efeitos temporais. Note que o número de possíveis contextos cresce exponencialmente com o número de documentos, o que demanda heurísticas que restrinjam a busca pelo melhor contexto eficientemente.

É importante distinguir o problema de seleção de contextos do problema de seleção de características (*feature selection*) (Molina & Nebot, 2002; Rogati & Yang, 2002; Mladenic & Grobelnik, 1998), o qual é usado para reduzir o número de características (i.e., termos) que serão utilizadas para construir o modelo de classificação, normalmente para fins de eficácia, no entanto mantendo a precisão e a robustez. Apesar de ser uma conseqüência, o principal objetivo da seleção de contextos temporais não é apenas reduzir o tamanho da amostra de treinamento, mas selecionar a maior porção do conjunto de treinamento em que os efeitos temporais (distribuição de classes, distribuição de termos e similaridade entre classes) e, conseqüentemente, a incerteza que o classificador precisa lidar, sejam minimizados.

Para ilustrar o problema de seleção de contextos, considere o conjunto de treinamento apresentado na Tabela 4.1. Este conjunto consiste de documentos que contêm o termo Plutão. Conforme apresentado no Capítulo 1, além de ser o Deus do Inferno na mitologia Romana, Plutão também era considerado um planeta até meados de 2006. Assim, como observamos na Tabela 4.1, antes de 2005, nosso conjunto de treinamento consiste apenas de documentos da classe Astrofísica. Depois de 2005, nosso conjunto contém apenas documentos da classe Mitologia, enquanto que no ano de 2005 existe um documento de cada classe. Se utilizássemos todo o conjunto de treino para construir um modelo de classificação, estaríamos dependentes de outras características (i.e., outros termos) para determinar a classe correta para um documento, uma vez que a palavra Plutão está associada a ambas as classes. Entretanto, se considerássemos a informação temporal para

delimitar os contextos em que Plutão tivesse apenas um significado, a mesma seria muito útil no modelo de classificação.

| **Palavra: Plutão** | | | |
|---|---|---|---|
| **Classe: Astrofísica** | | **Classe: Mitologia** | |
| Doc_Id | Ano | Doc_Id | Ano |
| $d_1$ | 1998 | $d_6$ | 2005 |
| $d_2$ | 1999 | $d_7$ | 2006 |
| $d_3$ | 2000 | $d_8$ | 2006 |
| $d_4$ | 2001 | $d_9$ | 2007 |
| $d_5$ | 2005 | $d_{10}$ | 2007 |

**Tabela 4.1.** Exemplo de Conjunto de Treinamento

O exemplo acima descrito é uma ilustração simplificada dos problemas que um algoritmo de classificação deve contornar para a geração de modelos de classificação mais eficientes, uma vez que muitas outras características podem apresentar mudanças semelhantes ao longo do tempo. Na próxima seção apresentamos o potencial da seleção de contextos temporais no auxílio aos classificadores para minimizar esses problemas.

## 4.2 Exploração de Seleção de Contextos Temporais

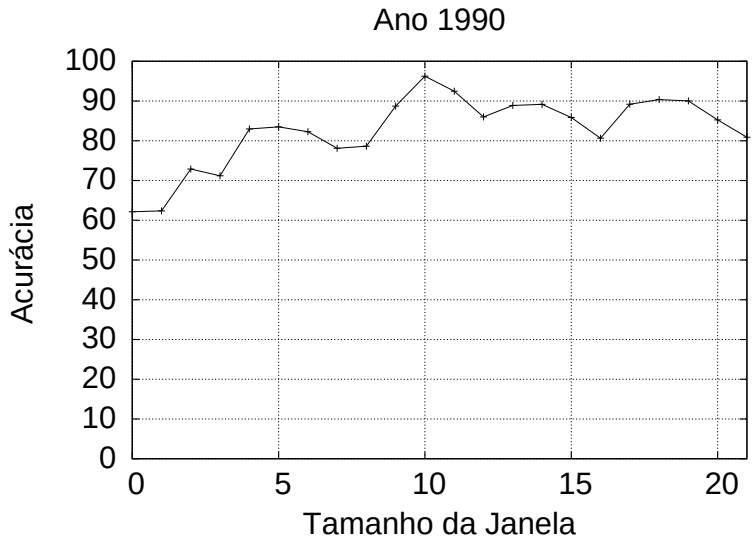
Nesta seção demonstramos que a seleção de um conjunto de treinamento apropriado pode levar a melhorias significativas no processo de classificação automática de documentos. Essa demonstração foi feita por meio da aplicação de uma heurística exaustiva de seleção de contextos temporais, que apesar de ser apropriada para uma avaliação analítica, pode ser impraticável em cenários reais. Nesta seção, nosso objetivo não é propor nenhum método computacionalmente eficiente para seleção de contextos temporais mas mostrar que existe uma lacuna para o desenvolvimento de tais métodos como uma maneira de melhorar o desempenho dos classificadores.

Melhorias devem ser obtidas considerando tanto o efeito amostral quanto os efeitos

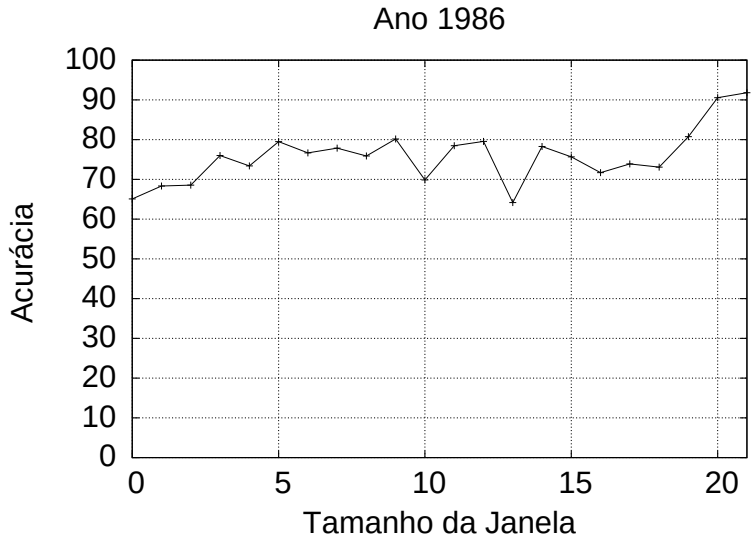
temporais. O desafio, assim, é lidar com esses dois efeitos simultaneamente uma vez que ambos demandam estratégias diferentes. Simplesmente aumentar o tamanho do conjunto de treinamento sem nenhum critério pode não prover melhorias no desempenho, uma vez que isso implica em aumentar a quantidade de documentos cujas datas de criação não sejam temporalmente próximas das datas de criação dos documentos de teste o que, como demonstrado no capítulo anterior, pode introduzir ruído. Por outro lado, considerar apenas os documentos próximos temporalmente dos documentos de teste pode reduzir muito o conjunto de treinamento e torná-lo escasso de informação.

Nossa estratégia para otimizar esse compromisso e conseqüentemente o desempenho do classificador é utilizar uma seleção de documentos para treinamento sensível ao tempo. Ou seja, selecionamos como conjunto de treinamento, para cada documento a ser testado, apenas documentos que estejam temporalmente próximos dele. A proximidade é definida por uma janela que pode crescer simetricamente para ambas as direções, passado e futuro, partindo do ano do documento de teste. Em nossa estratégia, para também considerarmos o efeito amostral, variamos o tamanho dessa janela de 0 (ou seja, o ano $A_i$ de criação do documento de teste) até $N$, em que $N$ representa o número de anos antes e depois de $A_i$. O valor de $N$ é definido como aquele que maximiza o desempenho do classificador para os documentos que pertencem ao ano $A_i$. A Figura 4.1 mostra a acurácia do SVM conforme variamos o tamanho da janela para os documentos de teste dos anos 1986 e 1990 (escolhidos aleatoriamente) na coleção ACM-DL. A Figura 4.2 mostra a acurácia do classificador conforme variamos o tamanho da janela para os documentos de teste dos anos 1971 e 1974 (escolhidos aleatoriamente) na coleção MedLine.

Como podemos observar pelos exemplos, não existe um tamanho ótimo de janela para toda a coleção, em ambas as coleções, mas um tamanho ótimo de janela pode ser encontrado para cada ano específico. Por exemplo, na coleção ACM-DL, para os documentos de 1990, o tamanho ótimo para a janela é 10 anos. Para os documentos de 1986, entretanto, o tamanho da janela é 20 anos. Apesar de o mecanismo utilizado nesses experimentos ser apropriado para uma análise de como podemos melhorar os classificadores




(a) Ano 1990

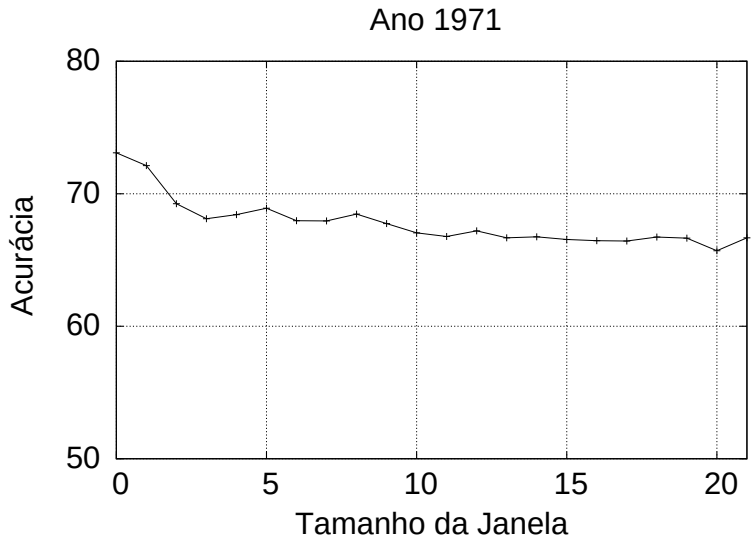



(b) Ano 1986

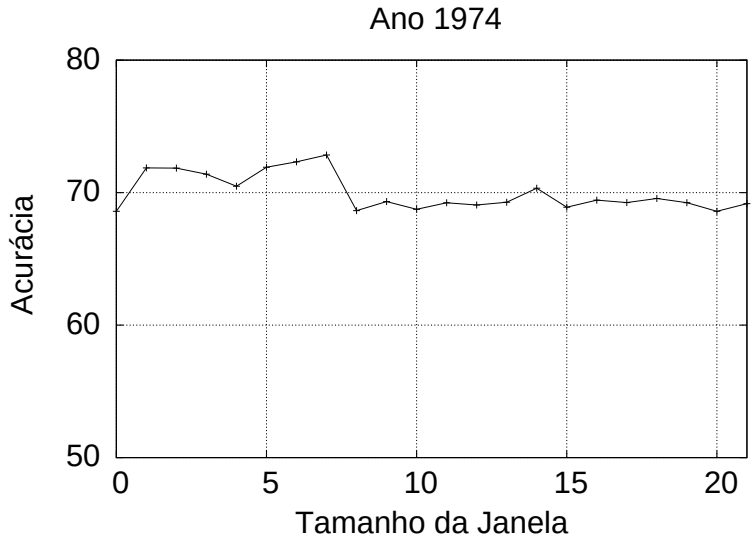
**Figura 4.1.** Janela Deslizante de Classificação - Coleção ACM-DL

utilizando seleção de contextos temporais, ele não é viável em cenários reais. Entretanto, como será explicado a seguir, a precisão do processo de classificação quando consideramos a janela temporal é maior do que quando consideramos todo o conjunto de treinamento. Além disso, mostramos que um classificador que considera os aspectos temporais além de alcançar uma melhor precisão, pode também melhorar sua eficácia, uma vez que o conjunto de treinamento é bem menor.

Primeiramente, mostramos que melhorias podem ser obtidas utilizando um tamanho



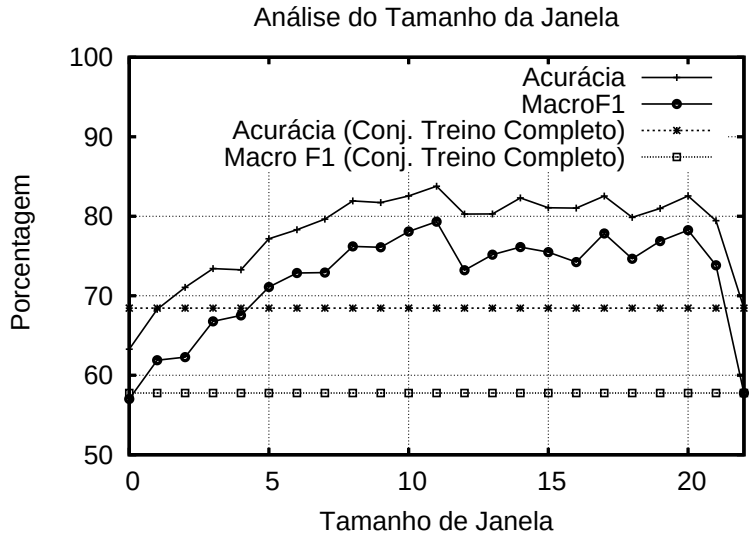

(a) Ano 1971



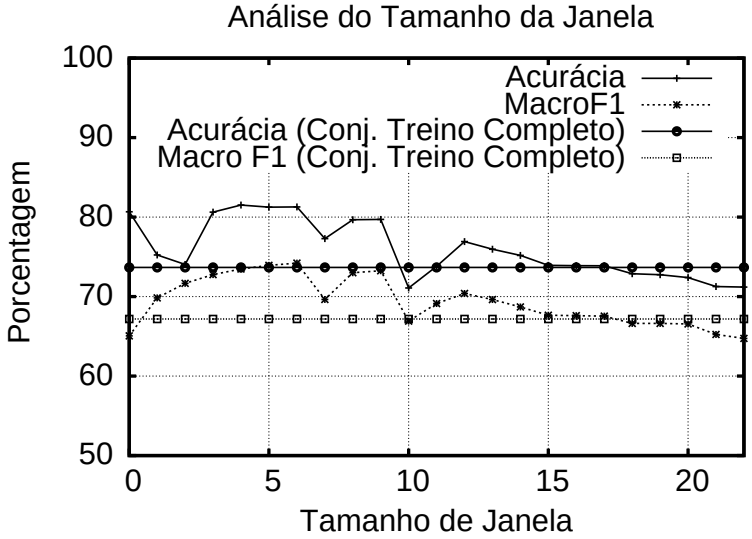

(b) Ano 1974

**Figura 4.2.** Janela Deslizante de Classificação - Coleção MedLine

médio de janela, i.e., uma janela comum para todos os anos. A Figura 4.3 mostra os resultados utilizando um tamanho de janela global para todos os documentos de teste. Mesmo quando não consideramos uma janela ótima para cada ano, o uso de uma janela ótima global ainda é melhor que o uso de todo o conjunto de treinamento. O gráfico da Figura 4.4 mostra o tamanho do conjunto de treinamento pelo tamanho da janela utilizada. Podemos notar que usando 33% de todo o conjunto de treinamento da coleção ACM-DL (tamanho de janela igual a 5) alcançamos um desempenho, em termos de acurácia, tão



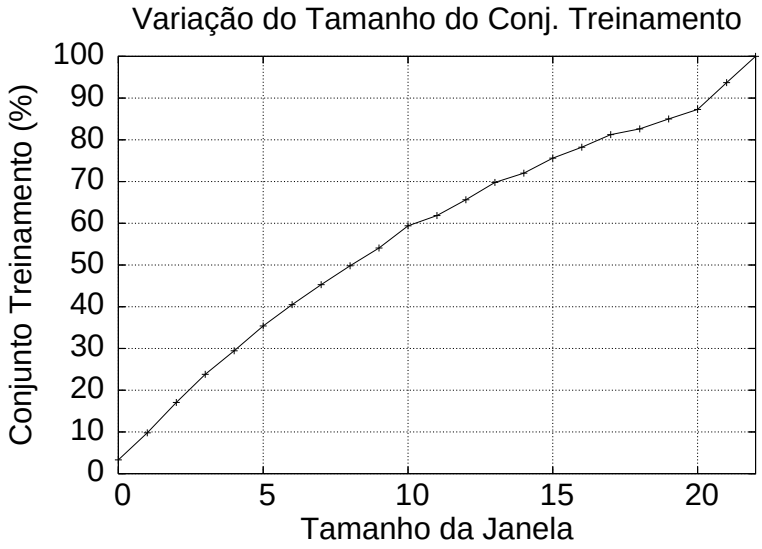

(a) Coleção ACM-DL



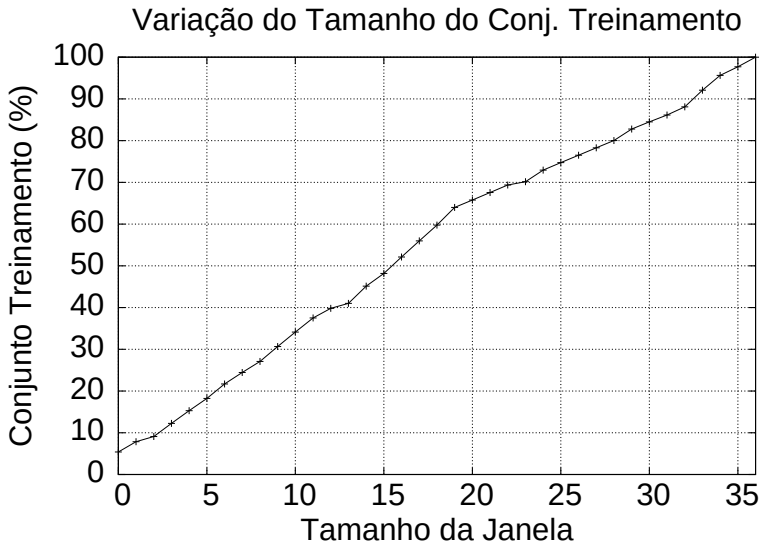

(b) Coleção MedLine

**Figura 4.3.** Desempenho por Tamanho Médio de Janela

bom quanto utilizando todo o conjunto de treinamento. Para a coleção da ACM-DL, o tamanho ótimo da janela global é 11 anos, o que corresponde a 61% de todo o conjunto de treinamento. Analisando a coleção MedLine, podemos notar que utilizando apenas 14% (tamanho de janela igual a 3) do conjunto de treinamento é suficiente para alcançar um desempenho tão bom quanto utilizando toda a coleção. Para essa coleção, o melhor resultado foi encontrado utilizando um tamanho ótimo de janela global igual a 4, o que representa 22% de todo o conjunto de treinamento.




(a) Coleção ACM-DL

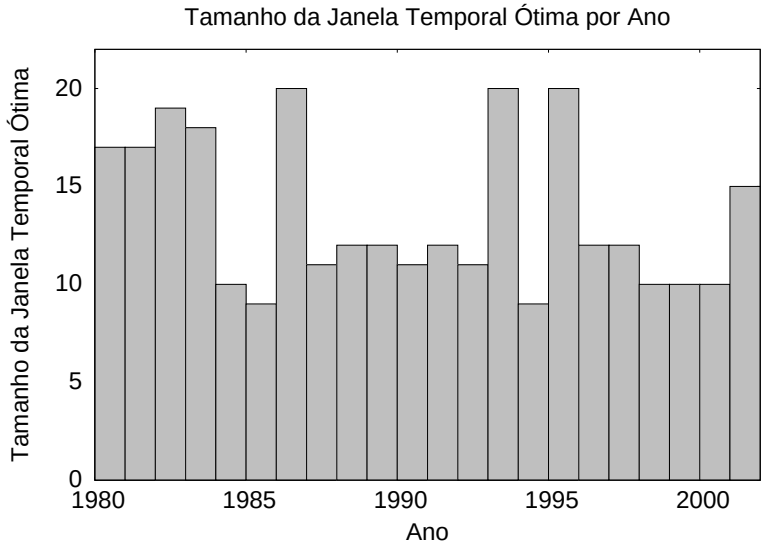



(b) Coleção MedLine

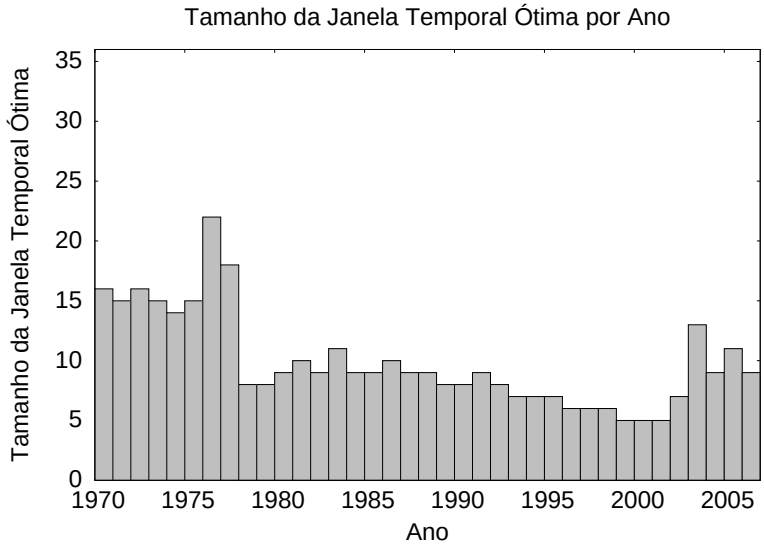
**Figura 4.4.** Tamanho da Coleção em função do Tamanho da Janela

Um extensão natural do procedimento apresentado é tentar explorar uma janela temporal ótima para cada ano. A Figura 4.5 mostra que o tamanho da janela temporal ótima varia significativamente para cada ano. A Figura 4.6 mostra o maior valor de acurácia para cada ano em ambas as coleções. Analisando esses gráficos, é possível perceber que o desempenho médio do classificador é maior do que quando utilizamos um tamanho de janela temporal global para todos os anos, em que conseguimos um desempenho de 83.7% e 81.5% para as coleções ACM-DL e MedLine, respectivamente, conforme mostrado na

Figura 4.3. Por outro lado, quando variamos o tamanho da janela temporal para cada ano, conseguimos valores superiores para ambas as coleções. Medindo todos os documentos corretamente classificados em toda a coleção, alcançamos uma precisão de 89.76% para ACM-DL e 87.57% para MedLine.



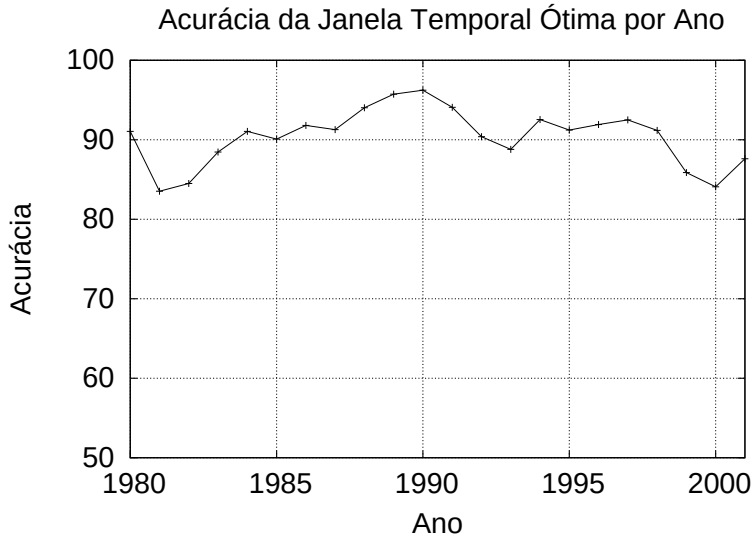

(a) Coleção ACM-DL



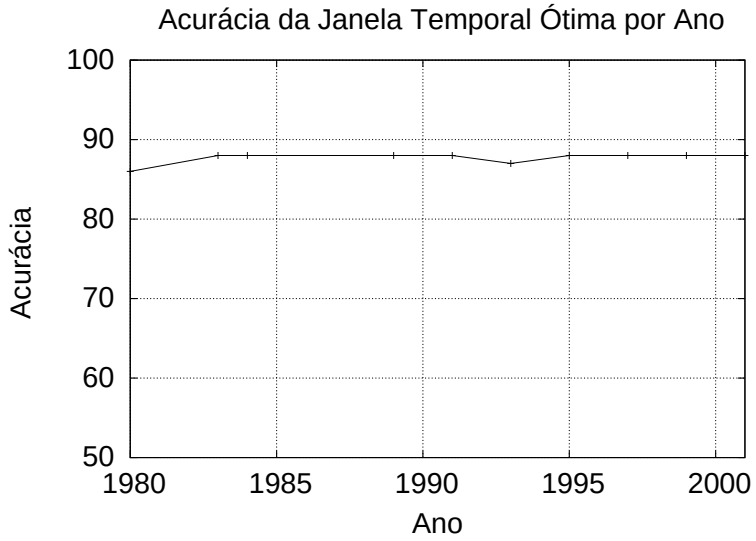

(b) Coleção MedLine

**Figura 4.5.** Análise da Janela Temporal Ótima por Ano

Esses resultados são ainda melhores quando os comparamos com os resultados encontrados utilizando todo o conjunto de treinamento no processo de construção do modelo de classificação, em que não consideramos os aspectos temporais. Nesse caso, tivemos




(a) Coleção ACM-DL




(b) Coleção MedLine

**Figura 4.6.** Accuracy da Janela Temporal Ótima por Ano

um desempenho de 68.44% para ACM-DL e 73.67% para MedLine, o que corresponde a um ganho de 31% na acurácia para ACM-DL no método que considera uma janela temporal ótima para cada ano e um ganho aproximado de 19% para MedLine. Ganhos em tempo de execução também podem ser obtidos se considerarmos que, aplicando uma janela temporal ótima para cada ano, o tamanho aproximado do conjunto de treinamento é 69% e 25% da ACM-DL e MedLine, respectivamente. Uma vez que o tempo de execução do SVM está diretamente relacionado com o tamanho do conjunto de treinamento, essa

redução pode fazer com que o tempo de execução do SVM diminua significativamente.

Em suma, nossa análise empírica mostra que a exploração adequada dos contextos temporais pode levar a melhorias significativas no processo de classificação. Apesar dessa heurística exaustiva ser apropriada em uma avaliação analítica, como no contexto dessa seção, a mesma pode ser impraticável em cenários reais. Dessa forma, na próxima seção apresentamos um algoritmo genérico que pode ser utilizado como modelo nas soluções que visam selecionar um conjunto de treinamento que considere o efeito amostral e os efeitos temporais.

## 4.3 O Algoritmo Chronos

Nesta seção, apresentamos um algoritmo genérico para seleção de contextos temporais que pode ser instanciado por propostas de soluções para o problema de seleção de contextos (previamente denominado de $X$) no conjunto de treinamento, em que os aspectos temporais sejam minimizados.

Durante a caracterização dos efeitos temporais presentes nas coleções de documentos apresentada no Capítulo 3, observamos que quanto mais próximo temporalmente os documentos de treino estavam dos documentos de teste, menor era a ação desses efeitos sobre a qualidade do modelo de classificação e à medida que essa distância aumentava, maior era ação dos mesmos. Dessa forma, identificamos três requisitos fundamentais que precisam ser considerados ao selecionar um contexto:

1. **Ponto de Referência:** como observado nos experimentos do Capítulo 3, o melhor cenário é encontrado quando os documentos de teste e treino pertencem ao mesmo momento no tempo. Dessa forma, o contexto a ser usado precisa ser temporalmente coerente com uma referência, em nosso caso com o documento a ser classificado, para capturar as características temporais do conjunto de treinamento (distribuição de classes, distribuição de termos e similaridade entre classes) associadas com o momento em que o documento de teste foi criado.

2. **Estabilidade das Características:** Como observado no capítulo anterior, à medida que a distância temporal entre os documentos de treino e teste aumenta, as características dos documentos tendem a variar. Assim, o contexto temporal precisa ser estável em relação às características temporais do conjunto de treinamento (distribuição de classes, distribuição de termos e similaridade entre classes) observadas no momento em que os documentos de teste foram criados, evitando alterações bruscas na definição das classes e de seus termos. Este requisito restringe a distância temporal entre os documentos do contexto selecionado.

3. **Redução da Incerteza:** Conforme observamos no Capítulo 3, os termos e a definição das classes evoluem ao longo do tempo e essa evolução tende a tornar o conjunto de treinamento $D$ muito confuso, degradando o desempenho dos classificadores. Assim, o contexto temporal $X$ precisa reduzir essa confusão e aumentar o ganho de informação dos termos que o compõem, ou seja, a incerteza inerente ao contexto selecionado precisa ser reduzida se comparada com a incerteza do conjunto de treinamento completo.

**Algoritmo 1** Chronos

```
1: function CHRONOS(D, S, δ)
2:     min ← ∞
3:     X ← ∅
4:     state ← GetReference(D, S)
5:     option ← Enumerate(D)
6:     repeat
7:         if (StabilityFunction(option, state) < δ) then
8:             uncertainty ← TemporalUncertainty(option)
9:             if (uncertainty < min) then
10:                X ← option
11:                min ← uncertainty
12:            end if
13:        end if
14:        option ← Enumerate(D)
15:    until (option = ∅)
16:    return(X)
17: end function
```

Baseado nesses requisitos, projetamos um algoritmo (Algoritmo 1), chamado de **Algoritmo Chronos** [1], que determina contextos enquanto reduz os efeitos temporais. O algoritmo Chronos recebe como entrada duas coleções, $D$ (conjunto dos documentos de treinamento) e $S$ (conjunto de documentos de teste provenientes do mesmo momento temporal), e um fator de estabilidade $\delta$. O primeiro requisito é abordado na linha 4. A função **GetReference** $(D, S)$ captura as características temporais (distribuição de classes, distribuição de termos e similaridade entre classes) do conjunto de treinamento $D$ no momento em que os documentos de teste $S$ foram criados e armazena na varável $state$. A função **Enumerate** $(D)$, na linha 5, lista, um por vez, os possíveis contextos temporais que serão analisados pelo algoritmo. Na linha 7, o segundo requisito é abordado. A função **StabilityFunction** $(option, state)$ avalia o quanto as características do contexto em questão ($option$) diferem das características do conjunto de treinamento $D$ no momento em que os documentos do conjunto de teste $S$ foram criados. Essa diferença precisa ser menor que o fator $\delta$. Finalmente, o terceiro requisito é abordado na linha 8. A função **TemporalUncertainty**$(option)$ calcula o grau de incerteza inerente ao contexto em questão ($option$). Então, dentre os possíveis contextos avaliados que são estáveis, o algoritmo seleciona aquele com menor grau de incerteza.

## 4.4 Estratégias de Implementação

Uma possível estratégia é realizar a seleção do contexto temporal apenas uma vez para todo o conjunto de teste e utilizar essa seleção para construir o modelo de classificação e classificar todos os documentos de teste. Para ilustrar essa solução, podemos retomar o exemplo apresentado na Tabela 4.1. Considerando um conjunto de teste $S$ composto pelos documentos $s_1, s_2, s_3 e s_4$ de 1999, 2006, 2001 e 2005, respectivamente, selecionar uma porção do conjunto de treino que considere os três requisitos apresentados na Seção 4.3 não é possível, uma vez que o nosso conjunto de teste é composto de

[1] Na mitologia Grega, Chronos é a personificação do tempo.

documentos de diferentes momentos (como normalmente acontece em cenários reais de classificação). Como podemos contextualizar temporalmente no conjunto de treinamento utilizando as datas de criação dos documentos de teste se cada documento de teste foi criado em um momento diferente? A função **GetReference** do Algoritmo 1 irá capturar características temporais de diferentes momentos e conseqüentemente a incerteza inerente ao conjunto de treinamento não será reduzida, o que reforça a necessidade de uma solução sob-demanda. O problema dessa estratégia também pode ser confirmado pelos experimentos apresentados na Seção 4.2. Esses experimentos mostram claramente que não existe um tamanho ótimo para a janela temporal que possa ser utilizado para todos os documentos de teste. Esse algoritmo foi publicado na *ACM International Conference on Information and Knowledge Management* (Rocha et al., 2008b).

Um outra estratégia é realizar uma busca exaustiva pelo melhor contexto temporal para cada documento de teste ou conjunto de documentos de teste do mesmo momento no espaço temporal. Para fazer isso, é necessário que a função **Enumerate**$(D)$ liste todas as possibilidades de contextos temporais do conjunto de treinamento $D$. Então, para cada documento de teste $s_Z$ (ou conjunto de documentos de teste), a estratégia escolheria o melhor contexto temporal possível, de acordo com suas características temporais. Essa solução é claramente uma solução sob-demanda (*Lazy*) (Yang, 1994; Veloso et al., 2006). Algoritmos sob-demanda também são baseados na premissa de que não existe um modelo de classificação universal e normalmente selecionam o conjunto de treinamento de acordo com as características (i.e., termos) dos documentos de teste como parte do processo de classificação.

Dessa forma, toda solução *Lazy*, para ser viável, precisa ser computacionalmente eficiente. Considerando que existem $2^D$ possíveis contextos temporais onde, conforme já mencionado, $D$ é o número total de documentos de treinamento, uma solução por busca exaustiva não é computacionalmente viável, um vez que a mesma requer a avaliação de cada um dos $2^D$ possíveis contextos para encontrar o melhor contexto para cada documento de teste ou conjunto de documentos. Retomando o exemplo da Tabela 4.1, a fun-

ção **Enumerate**$(D)$ listará $2^{10}$ contextos possíveis e avaliar cada um deles é aceitável. Entretanto, quando lidamos com uma coleção real como a ACM-DL que contém 30.000 documentos, avaliar $2^{30.000}$ possíveis contextos para cada documento de teste é impossível. Uma estratégia para se reduzir o custo computacional é empregar heurísticas que enderecem os requisitos previamente mencionados enquanto mantenham baixo o custo computacional, obviamente obtendo soluções sub-ótimas. Dessa forma, nos próximos capítulos apresentamos e avaliamos duas propostas de heurística baseadas no Algoritmo 1.

## 4.5 Sumário

Considerando a existência do compromisso existente entre o efeito amostral e os efeitos temporais, ambos caracterizados no Capítulo 3, neste capítulo formalizamos o problema de seleção de contextos temporais, cujo principal objetivo é otimizar esse compromisso, selecionando a maior porção do conjunto de treinamento em que os efeitos temporais (distribuição de classes, distribuição de termos e similaridade entre classes), e, conseqüentemente, a incerteza que o classificador precisa lidar ao gerar o modelo de classificação, sejam minimizados. Formalizado o problema, apresentamos uma análise empírica que mostra que a exploração adequada dos contextos temporais no processo de escolha do conjunto de treinamento pode levar a melhorias significativas no processo de classificação. Apesar de a heurística exaustiva apresentada ser apropriada em uma avaliação analítica, certamente a mesma pode ser impraticável em cenários reais. Entretanto, isso mostra que existe uma lacuna para o desenvolvimento de heurísticas eficientes de seleção de contextos temporais como uma maneira de melhorar o desempenho dos classificadores.

Durante a caracterização apresentada no capítulo anterior, observamos que quanto mais próximo temporalmente os documentos de treino estavam dos documentos de teste, menor era a ação desses efeitos sobre a qualidade do modelo de classificação. Baseado

nessa observação, levantamos três requisitos fundamentais que precisam ser considerados ao selecionar um contexto temporal: (1) Ponto de Referência; (2) Estabilidade das Características e (3) Redução da Incerteza. Respeitando esses requisitos, projetamos um algoritmo genérico que pode ser utilizado como modelo nas soluções que visam selecionar um bom conjunto de treinamento.

Ainda neste capítulo apresentamos uma discussão breve de algumas possíveis soluções de implementação para o Algoritmo *Chronos*. Observamos que uma solução que gere um contexto temporal único para todo o conjunto de treinamento não seria eficiente, uma vez que o conjunto de teste pode ser formado por documentos oriundos de diferentes momentos e capturar as características desses momentos de uma só vez poderia manter a confusão inerente ao conjunto de treinamento. Logo, precisamos de uma solução sob-demanda que seja aplicada a cada documento de teste ou conjunto de documentos de teste do mesmo momento no espaço temporal. Adotando uma solução de busca exaustiva, para cada documento de teste (ou conjunto de documentos) seria necessário avaliar $2^D$ possibilidades, o que não é computacionalmente viável. Assim, concluímos que a melhor maneira de se construir uma solução de contextos temporais seria por meio de heurísticas, discutidas a seguir.

# Capítulo 5

# Heurísticas GreedyChronos e WindowChronos

Neste capítulo apresentamos duas heurísticas para seleção de contextos temporais implementadas a partir do algoritmo genérico *Chronos*: *GreedyChronos* e *WindowChronos*. A estratégia de ambas as heurísticas é selecionar um contexto temporal do conjunto de treinamento para cada documento de teste *GreedyChronos* ou conjunto de documentos de teste *WindowChronos*, baseado nas características desse documento (i.e., termos), em que os três efeitos temporais (distribuição de classes, distribuição de termos e similaridade entre classes), anteriormente mencionados, sejam minimizados.

## 5.1 GreedyChronos

O *GreedyChronos* é executado para cada documento de teste, capturando as características associadas a cada documento (mais especificamente, seus termos e suas datas de criação) separadamente, um por vez, naturalmente endereçando o requisito Ponto de Referência. Em seguida, definimos uma janela temporal para cada termo do documento de teste, a qual cresce para ambas as direções, passado e futuro, partindo da data de criação do documento de teste. O tamanho da janela de cada termo é determinado pelo

período durante o qual o termo permaneceu "estável". A estabilidade de um termo é medida pelo seu grau de exclusividade a uma determinada classe por um período determinando. Esse grau de exclusividade é quantificado pela métrica denominada Predominância ($Dominance$) (Zaiane & Antonie, 2002). Formalmente, seja $T = \{t_1, t_2, t_3, \ldots, t_M\}$ o o conjunto de termos associados com a coleção, $C = \{c_1, c_2, \ldots, c_K\}$ o conjunto de classes que ocorrem na coleção e $df(t_i, c_j)$ o número de documentos associados à classe $c_j$ que contém $t_i$, definimos a Predominância do termo $t_i$ na classe $c_j$ da seguinte forma:

$$Predominancia(t_i, c_j) = \frac{df(t_i, c_j)}{\sum_{l=1}^{K} df(t_i, c_l)} \tag{5.1}$$

Em nossa abordagem, a janela temporal de cada termo é contígua por duas razões. Primeiro por uma questão computacional, uma vez que para determinar uma janela temporal que não seja contígua o espaço de busca é de $2^D$, enquanto que o espaço de busca para se encontrar uma janela contígua é $D^2$. Segundo porque o relacionamento entre termos e classes tende a mudar à medida que a distância temporal entre o documento de teste e os documentos de treino (contexto temporal) aumenta, sugerindo assim que a janela seja contígua. Em suma, a Predominância quantifica tanto a estabilidade quanto o grau de incerteza associado com a janela temporal, uma vez que quanto mais forte é o relacionamento entre um termo e uma classe, menor é o grau de incerteza desse termo. Assim, conforme mostramos acima, os outros dois requisitos identificados no capítulo anterior (Estabilidade das Características e Redução da Incerteza) são tratados simultaneamente em nossa heurística.

Após determinar uma janela temporal para cada termo, o último passo é selecionar os documentos que irão compor o contexto temporal e serão utilizados como conjunto de treinamento para classificar o documento de teste. Para cada termo de teste, selecionamos os documentos que possuem o termo e cuja a data de criação pertence à sua janela temporal. Finalmente, fazemos a união dos documentos selecionados para cada termo do documento de teste e essa união será o contexto temporal, o conjunto de treinamento do documento de teste. Como podemos observar, os contextos temporais selecionados pela

| Descrição | | # Ocorrências | | | Predominância | |
|---|---|---|---|---|---|---|
| Termo | Ano | Classe A | Classe B | Classe C | Valor | Classe |
| $t_1$ | 1982 | 4 | 4 | 2 | 40% | A/B |
| | **1983** | **5** | **3** | **2** | **50%** | **A** |
| | **1984** | **5** | **2** | **3** | **50%** | **A** |
| | **1985** | **6** | **4** | **0** | **60%** | **A** |
| | 1986 | 4 | 4 | 2 | 40% | A/B |
| | 1987 | 3 | 3 | 3 | 33% | A/B/C |
| | 1988 | 3 | 3 | 3 | 33% | A/B/C |
| $t_2$ | 1982 | 3 | 3 | 3 | 33% | A/B/C |
| | 1983 | 4 | 4 | 2 | 40% | A/B |
| | **1984** | **2** | **5** | **3** | **50%** | **B** |
| | **1985** | **2** | **6** | **2** | **60%** | **B** |
| | **1986** | **3** | **5** | **2** | **50%** | **B** |
| | 1987 | 3 | 3 | 3 | 33% | A/B/C |
| | 1988 | 3 | 3 | 4 | 40% | C |
| $t_3$ | 1982 | 4 | 4 | 2 | 40% | A/B |
| | 1983 | 4 | 3 | 3 | 40% | A |
| | **1984** | **5** | **3** | **2** | **50%** | **A** |
| | **1985** | **6** | **2** | **2** | **60%** | **A** |
| | **1986** | **6** | **2** | **2** | **60%** | **A** |
| | **1987** | **5** | **3** | **2** | **50%** | **A** |
| | 1988 | 3 | 3 | 3 | 33% | A/B/C |

**Tabela 5.1.** Ocorrências de Termos entre Classes ao Longo dos Anos

*GreedyChronos* são assimétricos.

Para mostrar o funcionamento da heurística *GreedyChronos*, vamos considerar que queremos classificar o documento de teste $s_1$ do ano de 1985 composto pelos termos $t_1, t_2$ e $t_3$. Como vimos anteriormente, o primeiro passo é utilizar a Predominância para determinar a janela temporal, partindo de 1985, em que cada um dos termos se manteve "estável". A Tabela 5.1 ilustra as ocorrências de cada termo de teste do documento $s_1$ entre as classes da coleção, no conjunto de treinamento, ao longo dos anos. Por exemplo, temos que o termo $t_1$, em 1985, ocorreu em seis documentos da classe A, quatro documentos da classe B e em nenhum documento da classe C, dessa forma temos que esse termo tem uma Predominância de 60% para a classe A em 1985. Adotando uma Predominância mínima de 50% para determinar a estabilidade de um termo, temos que o ano de 1985 faz

parte da janela temporal de $t_1$. Esse processo deve ser repetido para os anos adjacentes a 1985 (que é o ano do documento de teste) onde encontraremos que a janela temporal de $t_1$ corresponde ao período de 1983 a 1986 (linhas destacadas na Tabela 5.1). Todo o processo deve ser feito para os demais termos de $s_1$, onde encontraremos que as janelas temporais 1984 a 1986 e 1984 a 1987 para os termos $t_2$ e $t_3$, respectivamente. Dessa forma, o contexto temporal que será utilizado para classificar o documento de teste $s_1$ será composto pelos documentos do conjunto de treinamento que contenham o termo $t_1$ e que pertençam aos anos 1983 a 1986, mais os que contenham o termo $t_2$ e que pertençam aos anos de 1984 a 1986 e mais os que contenham o termo $t_3$ e que pertençam aos anos de 1984 a 1987.

Dessa forma, podemos notar que nossa heurística pode ser facilmente adaptada para situações em que temos apenas informação do passado (Schlimmer & Granger, 1986; Forman, 2006), como no caso de estratégias de *Concept Drift* e Classificação Adaptativa de Documentos.

## 5.2 WindowChronos

Diferentemente da *GreedyChronos*, que é executada para cada documento de teste, a *WindowChronos* visa encontrar o melhor contexto temporal para o conjunto de documentos de teste de cada momento distinto $m_p$ no espaço temporal (e.g., um determinado ano $A_i$). Essa estratégia aborda o requisito de Ponto de Referência uma vez que todos os documentos de $A_i$ possuem o mesmo ponto de referência. Nessa heurística, cada contexto temporal é determinado por uma seleção de documentos do conjunto de treinamento sensível ao tempo, ou seja, seleciona-se como conjunto de treinamento, para o conjunto de documentos de teste de cada ano, todos os documentos que estejam temporalmente próximos aos documentos de teste. Essa proximidade é definida por uma janela temporal que pode crescer em ambas as direções, passado e futuro, baseado no ano dos conjuntos de documentos de teste.

Para determinar quais os anos irão compor a janela temporal de cada conjunto de documentos de teste, a estabilidade das características desses anos, relacionadas às características do ano base $A_i$, precisam ser calculadas. Assim, primeiro os documentos precisam ser separados de acordo com suas datas de criação (ambos os conjuntos de treino e de teste). Para cada ano base $A_i$, obtemos uma lista com todos os termos que ocorrem nesse ano. Para cada termo, partindo do ano base $A_i$, calculamos o período durante o qual ele se manteve estável, onde a estabilidade de um termo é calculada, assim como na *GreedyChronos*, pela métrica Predominância apresentada na Equação 5.1. Ao final, cada termo possui uma lista com os anos durante os quais ele se manteve estável. Em seguida, calculamos a porcentagem de ocorrência desse anos nas janelas temporais de todos os termos, diferente da *GreedyChronos* que calcula apenas para os termo que ocorrem no documento de teste. Esse anos são ordenados de acordo com essa porcentagem (criação de um *ranking*) e os $N$ anos com maior porcentagem de ocorrência serão os anos que irão compor a janela temporal do conjunto de documentos de teste de $A_i$. Dessa forma, essa heurística considera todos os termos que ocorrem em cada ano, não somente os termos de teste como na *GreedyChronos*. Além disso, utilizando essa estratégia, endereçamos o requisito de Estabilidade das Características, uma vez que a estabilidade de todos os termos é avaliada a fim de determinar os anos de $N$.

Para ilustrar o funcionamento da heurística *WindowChronos*, considere um coleção formada por documentos de diferentes anos. Tomando, por exemplo, o ano de 1995 , temos que nesse ano aparece o seguinte conjunto de termos $(a_1, a_2, \ldots, a_{10})$. Conforme descrito acima, o primeiro passo da *WindowChronos* é calcular, partindo do ano base 1995, a janela temporal de cada um dos termos, que corresponde ao período durante o qual eles se mantiveram estáveis (assumindo um valor de Predominância adequado). A Tabela 5.2 apresenta uma possível configuração para as janelas dos termos. Por exemplo, enquanto a janela temporal do termo $a_3$ é de 1993 até 1998, a janela temporal do termo $a_8$ é vazia, o que significa que o termo $a_8$ não se mostrou estável o suficiente segundo o valor de Predominância assumido. Em seguida, baseado nessa configuração, apresentamos na

Tabela 5.3 o resultado da segunda fase da heurística *WindowChronos*, que corresponde ao cálculo das porcentagens de ocorrências dos anos nas janelas temporais dos termos. Por exemplo, enquanto que o ano $1997$ ocorre em 70% dos termos do ano base $1995$, o ano $1990$ ocorre em apenas 10%. Como descrito acima, os $N$ anos com maior porcentagem de ocorrência serão os anos que irão compor a janela temporal do conjunto de documentos de teste de $1995$.

| **Ano Base: 1995** | | | |
|---|---|---|---|
| Term_Id | Janela | Term_Id | Janela |
| $a_1$ | 1995-1997 | $a_6$ | 1994-1995 |
| $a_2$ | 1993-1996 | $a_7$ | 1995-1995 |
| $a_3$ | 1993-1998 | $a_8$ | - |
| $a_4$ | 1990-1999 | $a_9$ | - |
| $a_5$ | 1994-1996 | $a_{10}$ | 1994-1997 |

**Tabela 5.2.** Conjunto de Termos de um Ano Base

| **Ano Base: 1995** | | | |
|---|---|---|---|
| Ano | % de Ocorrência | Ano | % de Ocorrência |
| 1995 | 70% | 1998 | 20% |
| 1996 | 50% | 1999 | 10% |
| 1994 | 50% | 1992 | 10% |
| 1997 | 40% | 1991 | 10% |
| 1993 | 30% | 1990 | 10% |

**Tabela 5.3.** Porcentagem de Ocorrência dos Anos nas Janelas Temporais dos Termos

Com o objetivo de encontrar o valor de $N$ e abordar o requisito Redução da Incerteza, propomos separar o conjunto de treinamento em dois subconjuntos: treinamento e validação. Para cada ano base $A_i$, o valor de $N$ precisa ser variado de 1, isto é, o ano mais freqüente nas janelas temporais de todos os termos (topo do *ranking*), até $Y$, que representa o número total de anos que ocorrem nas janelas temporais (período de estabilidade) de todos os termos. O valor de $N$ é definido como sendo o valor que maximiza o desempenho do classificador utilizando os documentos de validação que pertencem ao ano $A_i$. Ao final, um contexto temporal para cada ano base $A_i$ é determinado pelo conjunto de documentos de todo o conjunto de treinamento que pertence a essa janela temporal, definida pelo valor de $N$. Esses contextos temporais, os quais são simétricos, são utilizados como

conjunto de treinamento pelo classificador para os documentos de teste de cada ano base $A_i$, a fim de criar seus modelos de classificação.

Retomando o nosso exemplo anterior, o primeiro valor de $N$ a ser avaliado é, como vimos, 1. Pela Tabela 5.3 temos que o ano mais freqüente nas janelas temporais dos termos de 1995 é o próprio ano 1995. Dessa forma, utilizando apenas os documentos de 1995 do subconjunto de treinamento gera-se um modelo de classificação e avalia-se o mesmo utilizando os documentos de 1995 do subconjunto de validação. O próximo valor de $N$ a ser avaliado é 2, ou seja, os dois anos mais freqüentes nas janelas temporais dos termos (1995 e 1996). Assim, utilizando apenas os documentos de 1995 e 1996 do subconjunto de treinamento, gera-se um modelo de classificação e avalia-se o mesmo utilizando os documentos de 1995 do subconjunto de validação. Isso é feito para todos os possíveis valores de $N$ (10 no nosso caso) e o valor de $N$ que apresenta os melhores resultados no conjunto de validação é aplicado para o conjunto de teste. Em nosso caso, supondo que o valor de $N$ que obteve os melhores resultados seja 4, temos que o contexto temporal dos documentos de teste de 1995 será composto pelos documentos de treinamento que pertencem aos anos de 1995, 1996, 1994 e 1997, os 4 anos mais freqüentes da Tabela 5.3.

Uma busca completa pelos valores de $N$ que alcançam os melhores resultados em termos de qualidade de classificação precisa avaliar um número significativo de alternativas. Por exemplo, em uma coleção de documentos em que o espaço temporal é discretizado em dias e existem 365 dias base (i.e., uma coleção de documentos referentes a um ano de notícias de um jornal), 133.225 alternativas precisam ser avaliadas para se selecionar o contexto temporal de cada momento (dia) base (para cada dia base, podem ser avaliadas até 365 alternativas). Considerando-se que o espaço temporal pode aumentar (i.e., adicionam-se todas as notícias diárias de outros anos), a avaliação de todas as alternativas pode se tornar computacionalmente inviável. Uma estratégia que pode ser utilizada para reduzir o custo computacional é podar alternativas. Dessa forma, propomos parar a busca pelo melhor valor de $N$ se o classificador não melhorar seu desempenho após um certo número de tentativas. Apesar dessa estratégia ser simples, ela pode prover

uma boa aproximação comparada à solução que avalia todas as possibilidades, uma vez que os $N$ anos (momentos) com maior porcentagem de ocorrência, em cada ano base $A_i$, tendem a formar uma janela contígua, já que o relacionamento entre os termos e classes muda suavemente à medida que se aumenta a distância temporal entre os documentos de teste e os documentos dos contextos, conforme observamos no Capítulo 3.

## 5.3 Sumário

Neste capítulo propusemos duas heurísticas computacionalmente viáveis para a seleção de contextos temporais, a *GreedyChronos* e a *WindowChronos*, ambas derivadas do algoritmo geral *Chronos*. A principal diferença entre as duas heurísticas é referente aos contextos temporais gerados, que podem ser simétricos ou não em relação aos termos que compõem os documentos de treino e teste. Enquanto a *GreedyChronos* considera apenas as características presentes nos documentos de teste (termos de teste) a fim de selecionar contextos temporais, a *WindowChronos* gera contextos simétricos, pois considera todas as características (termos de teste e não-teste). A seguir, apresentamos os resultados experimentais referentes a avaliação dos contextos temporais selecionados por essas heurísticas em conjunto com diversos algoritmos de CAD.

# Capítulo 6

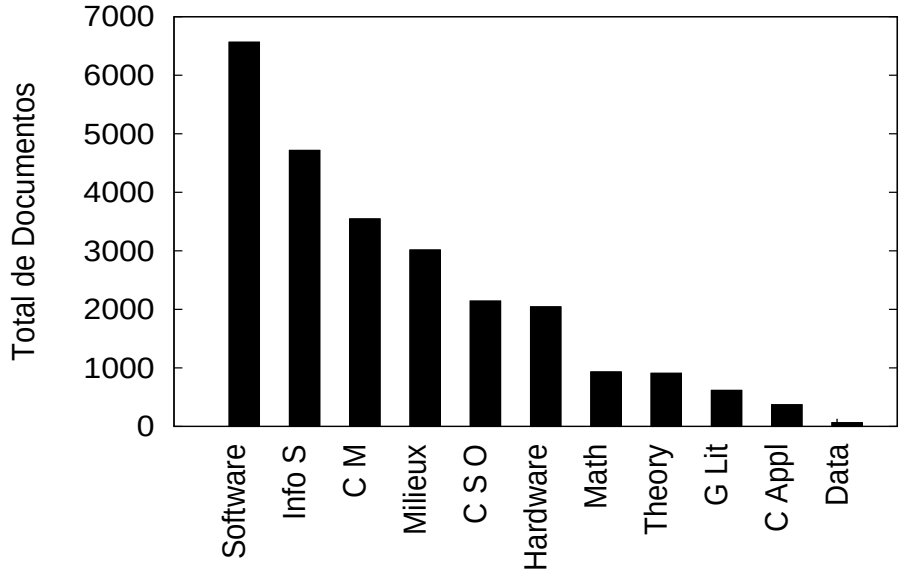
# Resultados Experimentais

Neste capítulo apresentamos os resultados da avaliação dos contextos temporais selecionados pelas heurísticas *GreedyChronos* e *WindowChronos* em conjunto com diversos algoritmos tradicionais de classificação automática de documentos: kNN, Naïve Bayes, Rocchio, e SVM. Todas as análises e resultados apresentados neste capítulo estão em processo de publicação no *Journal of American Society for Information Science and Technology* Rocha et al. (2009).

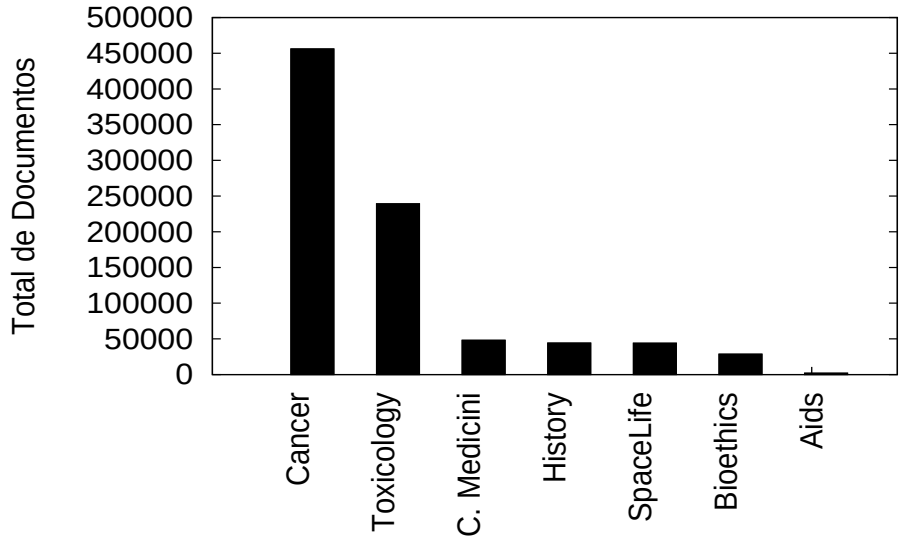
## 6.1 Ambiente Experimental

Em todos os experimentos utilizamos as mesmas duas coleções utilizadas nos experimentos dos capítulo anteriores: ACM-DL e MedLine. Entretanto, para reduzir a complexidade dos experimentos, descartamos parte da base MedLine e consideramos apenas o período entre 1970 e 1985. A Figura 6.1 ilustra a distribuição de documentos por classe para ambas as coleções utilizadas.

Em nossos experimentos utilizamos os algoritmos de classificação kNN (uma abordagem sob-demanda), Rocchio (uma abordagem vetorial) e Naïve Bayes (uma abordagem probabilística), implementados pelo arcabouço de classificação Libbow (McCallum, 1996). Além desses algoritmos, empregamos também o SVM-Perf (Joachims, 2006), um




(a) Coleção ACM-DL




(b) Coleção MedLine

**Figura 6.1.** Histograma de Documentos por Classe

pacote que implementa uma versão eficiente do algoritmo de classificação *Support Vector Machine* (SVM), que pode ser treinado em tempo linear. Além disso, utilizamos uma abordagem um-contra-todos (Vapnik, 1998) a fim de adaptar a versão binária do classificador SVM para classificação multi-classe, uma vez que nossas coleções possuem mais de duas classes. Diferentemente dos experimentos apresentados nos capítulos anteriores, utilizamos a informação referente à freqüência dos termos (*term frequency* - TF) para

todos os algoritmos avaliados neste capítulo.

O tempo pode ser visto como uma discretização natural das mudanças inerentes a qualquer área de conhecimento. Entretanto, essas mudanças podem ser detectadas em diferentes momentos do tempo, sob diferentes escalas, dependendo das características da área. Em nosso caso, as coleções de documentos utilizadas em nossos experimentos consistem de artigos científicos. Dessa forma, adotamos intervalos anuais para identificar as mudanças, ou seja, discretizamos as mudanças em intervalos anuais assim como nos experimentos apresentados nos Capítulos 3 e 4 (conferências, por exemplo, são usualmente anuais). entretanto, em outros cenários a granulação pode ser diferente.

A eficácia de nosso método foi avaliada utilizando métricas padrão da área de recuperação de informação: precisão, revocação e $F_1$ (Lewis, 1995). Revocação, $r$, é definida como a fração dos documentos de uma classe corretamente classificados. Precisão, $p$, por sua vez, é definida como a fração de documentos corretamente classificados dentre todos os documentos atribuídos pelo classificador a uma classe. Dessa forma, uma revocação perfeita para uma dada classe é alcançada caso todos os documentos da classe em questão sejam nela classificados, independentemente se outros documentos de outras classes sejam também atribuídos a ela. Por outro lado, uma boa precisão é atingida ao evitar que documentos oriundos de diferentes classes sejam atribuídos a uma só. Em decorrência dessa multiplicidade de aspectos de avaliação, uma métrica mais usual para avaliar a eficácia da classificação é $F_1$, uma combinação entre precisão e revocação dada pela média harmônica dessas duas métricas. Como resultado, dada uma classe, $c_j$ , sua respectiva pontuação $F_1$ é matematicamente definida como:

$$F_1(c_j) = \frac{2p_j r_j}{p_j + r_j} \tag{6.1}$$

Sumarizamos então nossos resultados gerais por meio de duas métricas tradicional-

mente usadas para esse fim: macro-média $F_1$ (macroF1) e acurácia. A macroF1 mensura a eficácia da classificação de cada classe individualmente e então calcula uma média aritmética dos $F_1$ de cada classe. A acurácia mensura a eficácia global de todas as decisões tomadas pelo classificador e é calculada pela razão entre o número total de documentos classificados corretamente e o número total de documentos classificados (Lewis, 1995). Dessa forma, enquanto a acurácia parte do princípio que todos os documentos são igualmente importantes, a macroF1 parte do princípio que cada classe é igualmente importante, e, por isso, atribui-lhes pesos iguais, independente da quantidade de documentos contidos em cada uma. Como resultado, se a maioria das classes em uma coleção contiver proporcionalmente poucos documentos em relação ao todo, então a macroF1 é uma métrica tipicamente mais relevante, pois são raros os casos em que é adequado subestimar a importância de uma vasta diversidade de classes. Caso contrário, a acurácia é uma métrica tipicamente mais significativa.

## 6.2 Avaliação da Heurística GreedyChronos

Primeiramente, realizamos um conjunto de experimentos que determinam os contextos temporais associados a cada documento de teste e avaliamos esses contextos juntamente com diversos algoritmos de classificação automática de documentos. Com esse objetivo, dividimos o conjunto de treinamento em dois subconjuntos: treinamento e validação. Em seguida, procuramos pelos valores do parâmetro Predominância que alcançam os melhores resultados de classificação utilizando o conjunto de validação como teste. Isso é feito para cada cenário, uma vez que as características específicas de cada coleção e algoritmo afetam a escolha desse parâmetro. Por fim, aplicamos a *GreedyChronos* para produzir os contextos temporais para cada um dos documentos de teste utilizando o melhor parâmetro Predominância encontrado em cada caso. Em todos os experimentos empregamos uma validação cruzada de 10 partes (*10-fold cross-validation*) (Brieman & Spector, 1992) e os resultados finais de cada experimento são a média das dez execuções.

Nas próximas seções apresentamos os resultados obtidos em nossos experimentos em cada um dos cenários avaliados. Além disso, apresentamos uma caracterização detalhada dos processos de criação de modelos de classificação usado por cada algoritmo e como eles são afetados pelos contextos temporais gerados pela *GreedyChronos*. O objetivo principal é entender a relação entre algumas premissas usadas por esses algoritmos e os contextos temporais selecionados. Definimos como premissas dos algoritmos características básicas que são direta ou indiretamente assumidas e utilizadas por eles em seus processos de criação de modelos de classificação. Por exemplo, a baixa ocorrência de um termo (i.e., sua raridade) é uma característica comumente utilizada por diversos algoritmos, sendo uma premissa importante para algumas implementações do Rocchio, por exemplo. A partir dessa análise esperamos compreender as razões dos ganhos ou não apresentados por cada algoritmo.

### 6.2.1 kNN

Os resultados alcançados utilizando os contextos temporais selecionados pela heurística *GreedyChronos* e o algoritmo kNN são apresentados na Tabela 6.1. As linhas com o rótulo “l.b.” (linha de base) apresentam os valores para as execuções utilizando o conjunto de treino completo sem levar em consideração os aspectos temporais, enquanto que as linhas com o rótulo “c.t.” apresentam os valores para as classificações utilizando os contextos temporais. As linhas com o rótulo “g.r.” representam a diferença percentual entre a classificação com o contexto temporal e a linha de base, ou seja, quantifica, para um dado classificador, se houve ganho positivo ou negativo ao utilizar contextos temporais na classificação. Finalmente, as linhas com o rótulo “t-t” descrevem se as variações produzidas utilizando-se contextos temporais representam diferenças estatisticamente significativas, dada uma confiança de 99% em um teste-t de dupla cauda (os ganhos positivos são representados por ▲, as perdas são representadas por ▼ e os resultados estatisticamente equivalentes são representados por ●). Como podemos observar, o uso de contextos temporais resultou em melhorias significativas do algoritmo kNN, quando comparado ao cenário em

que esses contextos não foram considerados.

| Coleção | | ACM-DL | | MedLine | |
|---|---|---|---|---|---|
| Métrica | | macF$_1$(%) | acc.(%) | macF$_1$(%) | acc.(%) |
| kNN | l.b. | 56,78 | 69,80 | 66,57 | 79,82 |
| | c.t. | 60,10 | 72,30 | 69,48 | 82,46 |
| | g.r. | +5,84 | +3,47 | +4,37 | +3,31 |
| | t-t. | ▲ | ▲ | ▲ | ▲ |

**Tabela 6.1.** Impacto dos Contextos Temporais (*GreedyChronos*) no kNN.

kNN (Yang & Liu, 1999) é um algoritmo sob-demanda, uma vez que retarda a construção do modelo de classificação até que um dado documento de teste seja apresentado para classificação. Baseando-se nos termos presentes no documento de teste, o algoritmo kNN obtém a amostra do conjunto de treinamento que será utilizada no processo de criação do modelo de classificação. Essa é uma estratégia bastante similar com nossa heurística de seleção de contextos temporais. Entretanto, o algoritmo kNN não considera explicitamente nenhum aspecto relacionado ao tempo para obter essa amostra do conjunto de treino, a qual consideraremos como "contextos não-temporais". O algoritmo kNN decide para qual classe $c_i$ um documento de teste $d_t$ deverá ser assinalado observando a classe dos $k$ documentos mais próximos (similares) de $d_t$ no contexto não-temporal, ponderando pela similaridade entre o documento de teste e aqueles $k$ documentos. O cálculo de similaridade normalmente utiliza apenas os termos presentes no documento de teste, a qual é uma premissa assimétrica, e pode ser calculada usando diferentes métodos, sendo que o mais comumente utilizado é a similaridade de cosseno (Salton & McGill, 1983).

Nossa hipótese para explicar porque o algoritmo kNN utilizando contextos temporais apresenta melhorias é que nesses contextos a confusão referente à ambigüidade dos termos de teste entre as classes é reduzido, melhorando assim a qualidade dos mesmos. Assim, com o objetivo de demonstrar nossa hipótese, avaliamos os documentos que compõem os contextos temporais relacionados a cada um dos documentos de teste. Encontramos que, enquanto na linha de base somente 30,15% e 51% da amostra (contextos não-temporais), para ACM-DL e MedLine, respectivamente, são compostos em sua

maioria por documentos relacionados à classe correta do documentos de teste, ao utilizar os contextos temporais esses valores aumentam para 52,4% e 67,5%, para ACM-DL e MedLine, respectivamente. Dessa forma, esses resultados mostram que os contextos temporais removem vizinhos "ruins" do conjunto dos $k$ documentos mais próximos. A seguir, avaliamos o impacto dessa remoção na qualidade de classificação do algoritmo kNN.

O algoritmo kNN cria, para cada documento de teste $d_t$, um *ranking* de classes baseado na pontuação assinalada e classifica $d_t$ para a classe que ocupa a primeira posição do *ranking*. A fim de avaliar o impacto dos contextos temporais, inspecionamos como eles alteram o *ranking* gerado pelo kNN analisando como a posição da classe verdadeira dos documentos de teste é afetada. Dessa forma, definimos uma variável aleatória $X$, a qual representa o número de posições do *ranking* que foram ganhos ou perdidos pela classe verdadeira ao utilizar os contextos temporais. Para avaliar $X$, documentos corretamente classificados tanto utilizando os contextos não-temporais quanto os contextos temporais não foram levados em consideração. Assim, mais formalmente, temos:

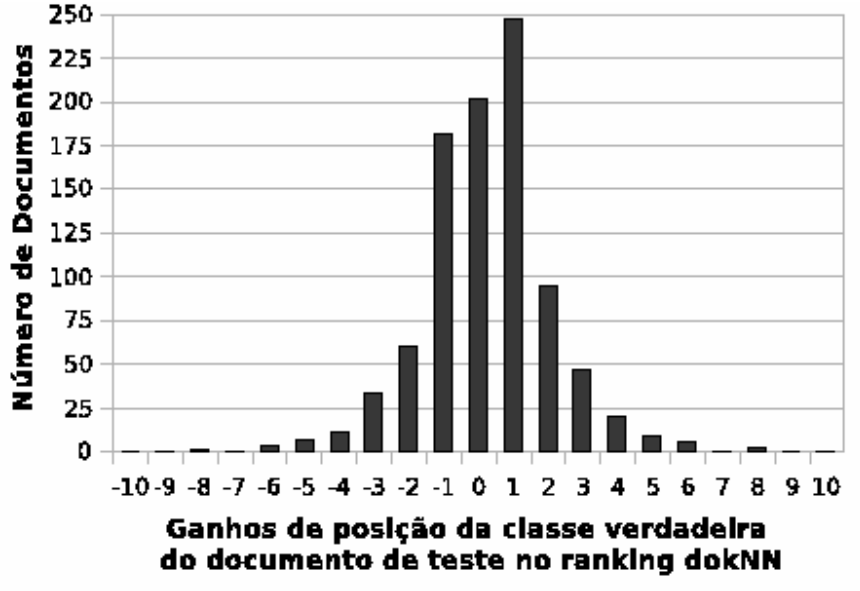
$$X(d_t) = Pos_{t.c.}(d_t) - Pos_{bl.}(d_t) \tag{6.2}$$

em que $Pos_{bl.}(d_t)$ é a posição no *ranking* da classe correta do documento de teste $d_t$ utilizando contextos não-temporais e $Pos_{t.c.}(d_t)$ é análogo ao raciocínio anterior, mas utilizando os contextos temporais.

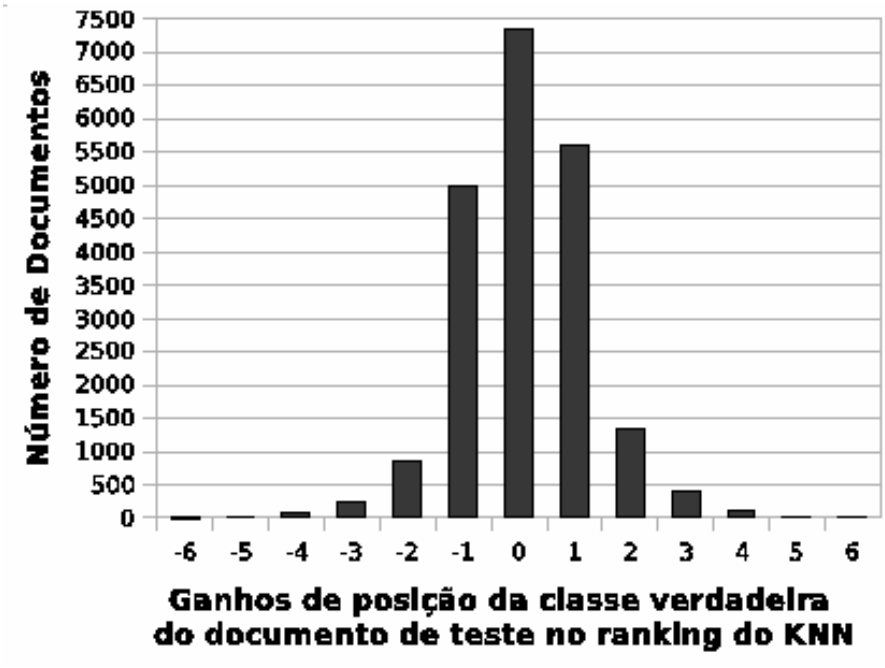
Dessa maneira, avaliamos a variável $X$ para ambas as coleções e os resultados são ilustrados na Figura 6.2. Como pode ser visto, o valor médio da distribuição é maior que zero, o que significa que o uso de contextos temporais melhora a qualidade da classificação para a maioria dos documentos da coleção. Além disso, também quantificamos a obliqüidade (*skewness*) da distribuição a fim de avaliar seu comportamento, utilizando um método popular conhecido como *Third Standardized Moment's Mode Skew-*

*ness* (Groeneveld, 1991). Um valor positivo para obliqüidade significa que a cauda da direita da distribuição é mais pesada que a da esquerda. O valor de obliqüidade calculado para ambas as coleções foi positivo (+0,20 para ACM-DL e +0,10 para MedLine), o que significa que a cauda da direita da distribuição é mais longa. Esse fato demonstra que, para o algoritmo kNN os contextos temporais selecionados pela heurística *Greedy-Chronos* gera um deslocamento positivo grande o suficiente para melhorar a posição no *ranking* das classes corretas. Como observamos na Tabela 6.1, essa melhora de posicionamento no *ranking* beneficia o algoritmo kNN, melhorando sua qualidade em ambas as coleções.

Apesar de haver um deslocamento positivo associado aos contextos temporais, podemos observar na Figura 6.2 que existem alguns casos de deslocamento negativo, que serão analisados a seguir. O uso de contextos temporais é baseado na intuição de que os termos de um documento de teste são bons indicadores para classificá-lo corretamente. Entretanto, alguns desses termos podem estar fortemente associados a classes que não sejam a classe correta do documento de teste. De fato, isto pode acontecer para a maioria dos algoritmos de CAD, e os contextos temporais amplificam isso. Essa distorção é anterior ao processo de classificação, como uma conseqüência da representatividade dos documentos de treinamento em relação a toda coleção, confusões naturais inerentes às coleções, mudanças temporais não consideradas, entre outras. Uma amostra grande da coleção pode ser mais robusta com relação a esse fato, permitindo uma qualidade de classificação melhor enquanto que nos contextos temporais essa diversidade intrínseca a grandes amostras é reduzida. Conseqüentemente, a efetividade dos algoritmos de CAD que utilizam contextos temporais tende a ser limitada pela ocorrência dessas distorções nas coleções. O número de casos em que os contextos não-temporais são melhores que os contextos temporais (i.e., a classe correta é encontrada apenas utilizando os contextos não-temporais ) não é grande. Por exemplo, na coleção ACM-DL esses casos representam menos de 6% dos documentos de teste.




(a) Coleção ACM-DL




(b) Coleção MedLine

**Figura 6.2.** Efeito dos Contextos Temporais sobre o kNN

### 6.2.2 Naïve Bayes

Os resultados alcançados utilizando os contextos temporais selecionados pela heurística *GreedyChronos* e o algoritmo Naïve Bayes são apresentados na Tabela 6.2. Como

podemos observar, os contextos temporais resultaram em melhorias estatisticamente significativas em termos de macro$F_1$ mas a diferença em termos de acurácia não foram estatisticamente significantes, de acordo com o teste-t.

| Coleção | | ACM-DL | | MedLine | |
|---|---|---|---|---|---|
| Métrica | | macro$F_1$(%) | acc.(%) | macro$F_1$(%) | acc.(%) |
| Naïve Bayes | l.b. | 56,87 | 74,00 | 65,64 | 79,65 |
| | c.t. | 58,90 | 73,30 | 66,68 | 80,36 |
| | g.r. | +3,47 | -0,96 | +1,75 | +0,44 |
| | t-t. | ▲ | ● | ▲ | ● |

**Tabela 6.2.** Impacto dos Contextos Temporais (*GreedyChronos*) no Naïve Bayes.

O algoritmo Naïve Bayes (Manning et al., 2008) é baseado em modelos probabilísticos, o qual calcula a pontuação de uma classe $c_i$ como sendo a probabilidade de um documento $d_t$ ser assinalado à classe $c_i$. Baseado nessa pontuação, é criado um *ranking* das classes e o Naïve Bayes classifica $d_t$ para a classe que ocupa a primeira posição desse *ranking*. Mais formalmente, tomando $P(c_i|d_t)$ como sendo a probabilidade de um documento de teste $d_t(a_1, a_2, \ldots, a_j)$ pertencer a uma classe $c_i$, em que o conjunto $(a_1, a_2, \ldots, a_j)$ é o vetor (binário ou ponderado) que representa o conjunto de termos de um documento $d_t$. Essa probabilidade é calculada utilizando o Teorema de Bayes e é definida como:

$$P(c_i|d_t) = \frac{P(c_i)P(d_t|c_i)}{P(d_t)} \tag{6.3}$$

O cálculo de $P(c_i|d_t)$ tem um custo computacional muito alto, dado que o número de possibilidades de vetores $d_t$ também é muito alto. Como conseqüência, o algoritmo Naïve-Bayes atenua esse problema assumindo que a ocorrência de todos os termos é independente. Isso permite computar a probabilidade de cada termo independentemente dos demais, tornando o processo de classificação computacionalmente viável. Assim, $P(c_i|d_t)$ pode ser definido como:

$$P(c_i|d_t) = \frac{P(c_i) \prod_{\forall j \in d_t} P(a_j|c_i)}{P(d_t)} \tag{6.4}$$

Existem, na literatura, duas principais abordagens mais gerais que são comumente utilizados. A primeira delas é chamada de Bernoulli (McCallum & Nigam, 1998), uma abordagem tradicional da área de redes Bayesianas, e é apropriada para tarefas de classificação que possuem um número fixo de atributos. A segunda delas, chamada de Multinomial (McCallum & Nigam, 1998), é mais tradicional em modelagem estatística de linguagem para o reconhecimento de fala e classificação de texto. Dessa forma, no escopo deste trabalho, o algoritmo Naïve Bayes adotado é baseado na abordagem Multinomial.

Observando a Equação 6.4, temos que a questão fundamental na definição de $P(c_i|d_t)$ está em como estimar $P(a_j|c_i)$. Assim, de acordo com (Manning et al., 2008) definimos:

$$P(a_j|c_i) = \frac{Tc_i(a_j)}{\sum_{v \in V} Tc_i(v)} \tag{6.5}$$

onde $Tc_i(a_j)$ é o número de ocorrências do termo $a_j$ na classe $c_i$ e $\sum_{v \in V} Tc_i(v)$ é o somatório do número de ocorrências de todos os termos que ocorrem em documentos da classe $c_i$. Para evitar inconsistências (i.e., divisão por zero), uma suavização de Laplace, a qual consiste simplesmente em adicionar 1 tanto ao denominador quanto ao numerador, é comumente utilizada:

$$P(a_j|c_i) = \frac{Tc_i(a_j) + 1}{\sum_{v \in V} Tc_i(v) + 1} \tag{6.6}$$

$$P(a_j|c_i) = \frac{Tc_i(a_j) + 1}{(\sum_{v \in V} Tc_i(v)) + V} \tag{6.7}$$

onde $V = |V|$ representa o tamanho do vocabulário. Assim, podemos concluir que $P(a_j|c_i)$ define a representatividade de um termo $a_j$ em uma classe $c_i$ como sendo a razão entre a freqüência do termo $a_j$ na classe $c_i$ e a freqüência total de todos os termos da classe $c_i$. Como todos os termos são considerados, podemos afirmar que o algoritmo Naïve Bayes utiliza uma premissa simétrica para calcular a representatividade de um termo $a_j$ e, conseqüentemente, também pode ser considerado como sendo um algoritmo simétrico.

A fim de explicar porque o algoritmo Naïve Bayes utilizando contextos temporais não apresentou melhoras em termos de acurácia mas apresentou ganhos em termos de macroF1, primeiramente avaliamos seu desempenho utilizando contextos não-temporais, isto é, testamos o comportamento de uma versão sob-demanda do algoritmo Naïve Bayes utilizando contextos não-temporais. Nessa versão, um modelo de classificação utilizando o Naïve Bayes é criado para cada documento de teste a ser classificado, utilizando a informação dos contextos não-temporais que são baseados apenas nos termos presentes no documento de teste. Em outra palavras, qualquer documento de treino que compartilhe um dos termos do documento de teste é selecionado para compor o contexto não-temporal. Nesse cenário, o algoritmo Naïve Bayes alcançou 55,13% e 71,81% em termos de macroF1 e acurácia, respectivamente, para a coleção ACM-DL, e 62,13% e 78,61% em termos de macroF1 e acurácia, respectivamente, para a coleção MedLine. Comparamos esses resultados com os resultados alcançados na linha de base apresentados na Tabela 6.2 e, dada uma confiança de 99% em teste-t de dupla cauda, podemos afirmar que esses resultados são estatisticamente inferiores que os resultados da linha de base. Comparamos também esses resultados com os alcançados utilizando contextos temporais (Tabela 6.2), e podemos afirmar que os resultados alcançados utilizando os contextos temporais são estatisticamente superiores àqueles que utilizaram contextos não-temporais, dada uma confi-

ança de 99% em teste-t de dupla cauda (ganhos de 6,84% e 2,1% para macroF1 e acurácia, respectivamente, na coleção ACM-DL, e ganhos de 7,32% e 2,2% para macroF1 e acurácia, respectivamente, na coleção MedLine). De acordo com essa observação, concluímos que o fato do algoritmo Naïve Bayes não apresentar melhoras não está relacionado ao uso ou não de contextos temporais mas sim ao fato que esses contextos, assim como os não-temporais, são assimétricos (compostos apenas por documentos de treino que contenham pelo menos um dos termos de teste).

Analisando a Equação 6.7, em ambas as versões do algoritmo Naïve Bayes (linha de base e sob-demanda), o numerador é o mesmo, uma vez que a probabilidade de um documento de teste $d_t$ pertencer a uma classe $c_i$ é calculada utilizando apenas os termos de $d_t$, isto é, os termos de teste. Como anteriormente mencionado, o denominador representa todos os termos que ocorrem em cada classe e esse valor será bem menor nas abordagens sob-demanda do que na versão utilizada na linha de base, uma vez que o número total de documentos nos contextos temporais e não-temporais é bem inferior ao número total de documentos de toda a coleção. Portanto, a representatividade dos termos nas classes aumentará bastante ao se utilizar os contextos, uma vez que o denominador será bem menor. No entanto, esse aumento da representatividade dos termos não é confiável, já que os contextos não-temporais, assim como os contextos temporais, são assimétricos, onde os termos que não ocorrem nos documentos de teste (termos de não-teste) não são levados em consideração no processo de seleção dos contextos. Esses outros termos influenciam na probabilidade de um determinado termo de teste em uma dada classe e, conseqüentemente, a premissa simétrica do Naïve Bayes em que a representatividade de um termo de teste na classe pode ser calculada em função dos demais termos da classe é seriamente afetada.

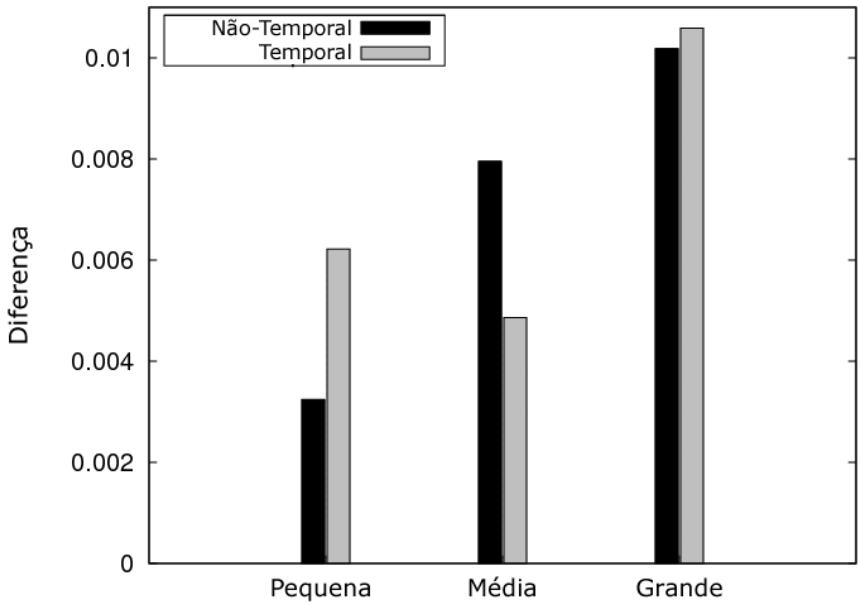
Com o objetivo de demonstrar nossas hipóteses, executamos um conjunto de experimentos em que os ganhos de representatividade dos termos em uma classe, utilizando contextos temporais e não-temporais, foram quantificados e comparados com a linha de base. Em todos os experimentos agrupamos as classes em três diferentes grupos (*Pe-*

*quena*, *Média* e *Grande*), de acordo com o tamanho das mesmas em termos de número de documentos. Avaliamos, para cada documento de teste, a diferença absoluta entre a probabilidade de cada termo utilizando contextos (temporais e não-temporais) e a linha de base. Analisamos a média desses resultados para cada documento e em seguida para cada classe. Ilustramos os resultados dessa análise por meio da Figura 6.3. Como podemos observar, existem diferentes ganhos de representatividade dos termos para todos os grupos de classe em ambas as coleções. No próximo conjunto de experimentos avaliamos o impacto desses ganhos de representatividade na pontuação das classes que são assinaladas pelo algoritmo Naïve Bayes.

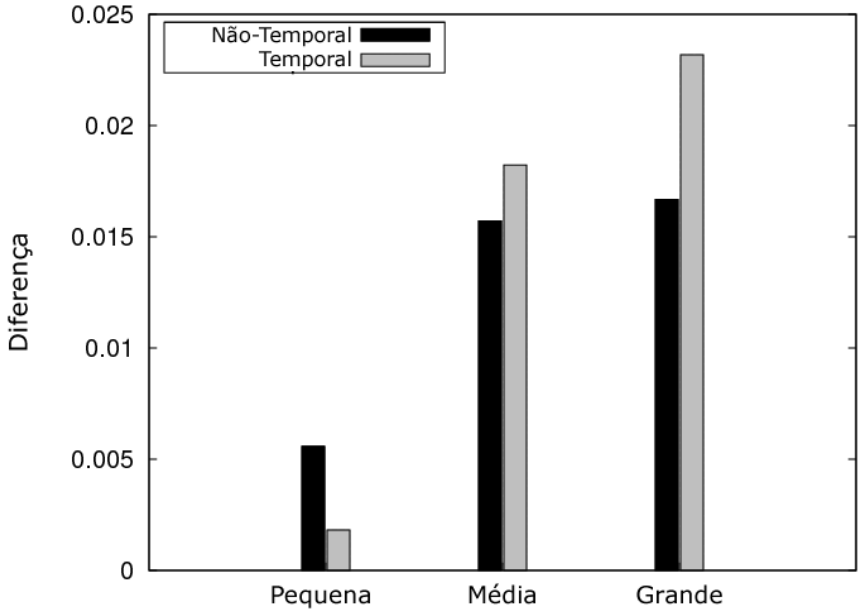
Primeiro, avaliando o *ranking* gerado pelo algoritmo Naïve Bayes (linha de base), calculamos a posição média da classe correta de cada grupo de classes. Para cada documento classificado, ordenamos a pontuação assinalada pelo Naïve Bayes para cada classe e em seguida agrupamos as mesmas pelo tipo. Então calculamos a posição média para cada grupo. Os resultados podem ser observados na Figura 6.4, onde podemos notar que, em média, em ambas as coleções, o grupo de classes *Pequena* tende ocorrer nas últimas posições do *ranking*, enquanto que as classes do grupo *Grande* tende ocorrer nas primeiras posições.

Para reforçar as evidências de nossas análises, novamente ordenamos de forma decrescente as pontuações assinaladas pelo algoritmo Naïve Bayes a cada classe para cada documento de teste. Em seguida, calculamos a porcentagem de ocorrência das classes de cada grupo na primeira posição do *ranking* utilizando: toda a coleção (linha de base); contextos não-temporais; e os contextos temporais. Os resultados são apresentados na Figura 6.5.

Como podemos observar na Figura 6.5, as classes maiores, com um número maior de documentos, aparecem na melhor posição do *ranking* mais freqüentemente que os outros grupos de classes em todos os experimentos. Para fins de análise, primeiro focaremos nos resultados relacionados aos contextos não-temporais. Comparando os resultados da linha de base com eles, podemos observar que a porcentagem de ocorrência das classes



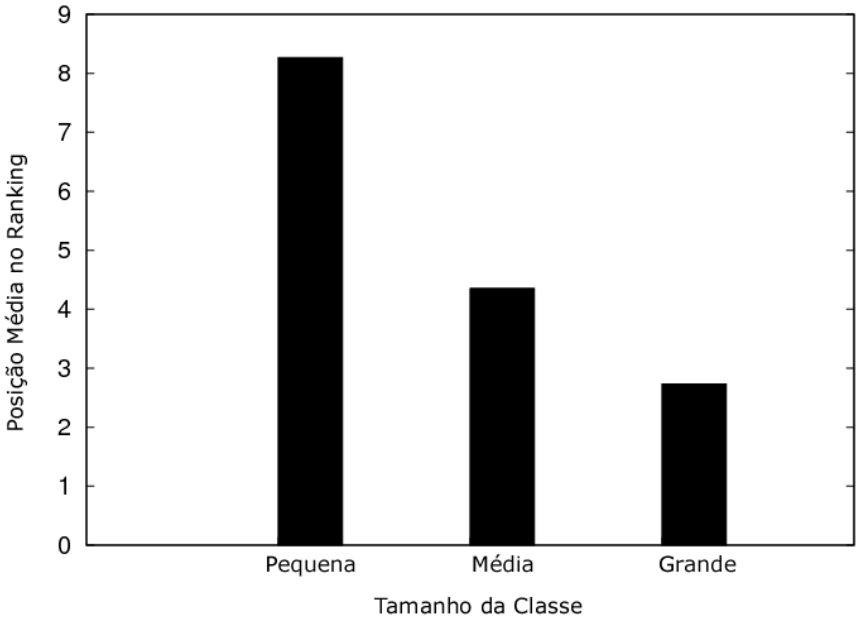

(a) Coleção ACM-DL



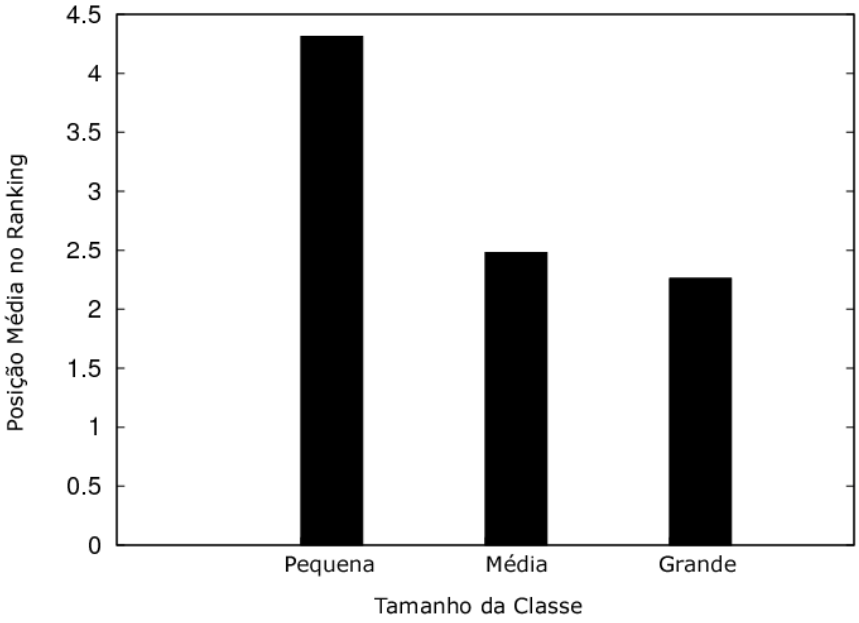

(b) Coleção MedLine

**Figura 6.3.** Ganhos de Representatividade dos Termos nas Classes

menores (*Pequena*) na melhor posição do *ranking* não aumentou em ambas as coleções. Isso pode ser explicado pelo fato que, apesar do aumento da representatividade dos termos dessas classes utilizando os contextos não-temporais, esse aumento não foi suficiente para deslocar essas classes para a melhor posição no *ranking*, uma vez que a posição mé-



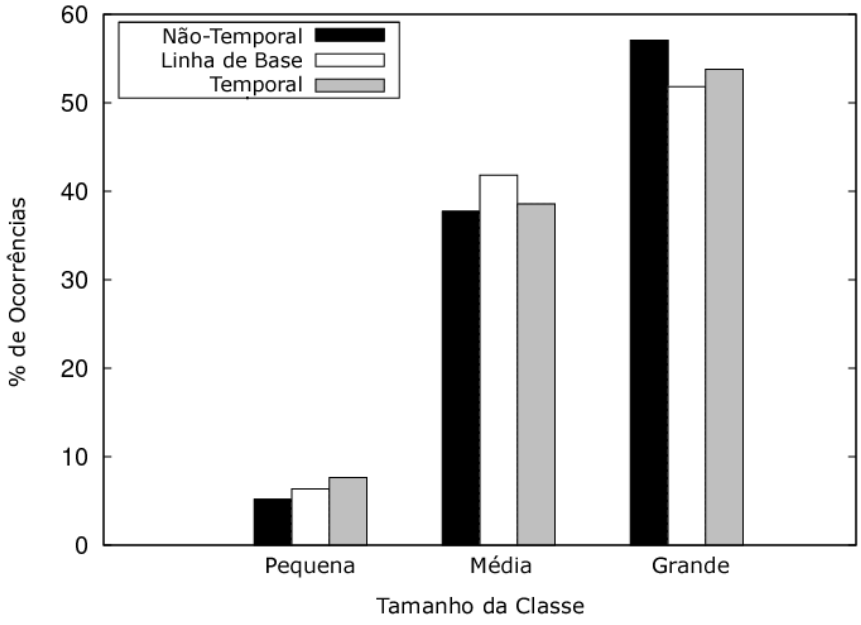

(a) Coleção ACM-DL



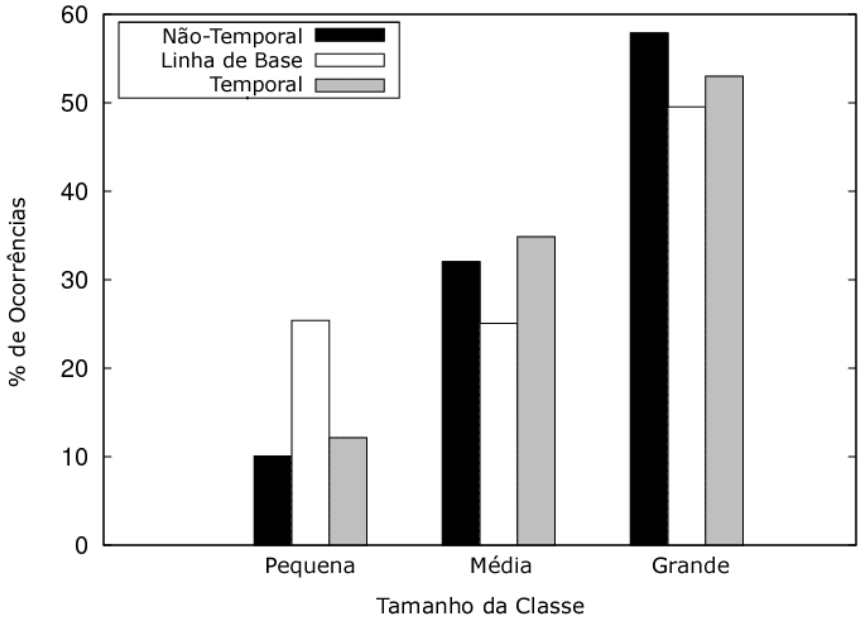

(b) Coleção MedLine

**Figura 6.4.** Posição Média no *Ranking* - Linha de Base

dia dessas classes no *ranking* era originalmente alta (últimas posições), como podemos observar na Figure 6.4. Entretanto, a porcentagem de ocorrência das classes classificadas como (*Grande*) na melhor posição do *ranking* apresentaram um ganho significativo em ambas as coleções. Essas classes estão, em média, nas primeiras posições do *ran-*




(a) Coleção ACM-DL




(b) Coleção MedLine

**Figura 6.5.** Ocorrência na Primeira Posição do *Ranking*

*king* (e.g., posição média 2,7 para ACM-DL), e a redução no denominador provido pelos contextos não-temporais é suficiente para colocá-las na primeira posição.

Ainda considerando os contextos não-temporais, a principal diferença entre as duas coleções está nos resultados relacionados às classes *Média*. Na coleção ACM-DL, os

ganhos de representatividade dos termos para essas classes não foi suficiente para deslocar essas classes para a melhor posição no *ranking*, uma vez que a posição média dessas classes era originalmente alta (i.e., 4,5). No caso da MedLine, observamos que o uso de contextos temporais resultou em um aumento significativo no número total de predições da classe correta como sendo uma das classes *Média*, uma vez que essas classes, assim como as *Grande*, também aparecem nas primeiras posições do *ranking* (i.e., 2,5). Em suma, a versão original do Naïve Bayes (linha de base), como observamos, tende a priorizar as classes com maior número de documentos, assinalando a elas, mais freqüentemente, os documentos de teste. Utilizando os contextos não-temporais, essas classes são ainda mais priorizadas, aumentando a tendência de prevê-las e assim degradando a qualidade da classificação.

Comparando os resultados da linha de base com os relacionados ao uso de contextos temporais, podemos observar que a porcentagem de ocorrências das classes *Grande* na primeira posição do *ranking* aumentou. Entretanto, esse aumento foi menor que o aumento alcançado com o uso de contextos não-temporais. Para a coleção ACM-DL, a porcentagem de ocorrência das classes *Pequena* no topo do *ranking* também aumentou. Isso pode ser explicado pelo ganho de representatividade dos termos das classes *Pequena*, que foram proporcionalmente maiores que as demais classes (veja Figura 6.3) e esse aumento foi grande o suficiente para deslocar essas classes para o topo do *ranking*, apesar da alta posição média dessas classes. Entretanto, para as classes *Média*, os ganhos de representatividade dos termos não foram tão grandes quanto para as classes *Pequena*, e, conseqüentemente, não foram grandes o suficientes para deslocar essas classes para a melhor posição do *ranking*.

Para a coleção MedLine observamos que a porcentagem de ocorrência das classes *Pequena* na primeira posição do ranking diminuiu, uma vez nesse grupo de classes o aumento da representatividade dos termos foi muito pequeno. Entretanto, a porcentagem de ocorrência das classes *Média* no topo do *ranking* apresentou um aumento significativo, maior que o aumento alcançado utilizando os contextos não-temporais. Dessa forma,

utilizando os contextos temporais, a tendência de classificação para as classes com um número grande de documentos é reduzido, uma vez que a representatividade dos termos para as classes *Pequena* (ACM-DL) e *Média* (MedLine) foram grandes o suficiente para aumentar o assinalamento dos documentos de teste para essas classes. Esse efeito provê uma acurácia maior que a alcançada utilizando contextos temporais, mas não tão grande quanto as alcançadas na linha de base. Assim, ao utilizar os contextos temporais identificamos uma redução da tendência de classificar os documentos de teste como sendo de uma das classes com maior número de documentos, o que explica os ganhos em termos de macroF1 em comparação com a linha de base, em ambas as coleções.

### 6.2.3 Rocchio

Os resultados alcançados utilizando contextos temporais selecionados pela heurística *GreedyChronos* e o algoritmo Rocchio são apresentados na Tabela 6.3. Como podemos observar, o uso de contextos temporais degradou o desempenho do algoritmo Rocchio.

| Coleção | | ACM-DL | | MedLine | |
|---|---|---|---|---|---|
| Métrica | | $macF_1$(%) | acc.(%) | $macF_1$(%) | acc.(%) |
| Rocchio | l.b. | 56,97 | 67,95 | 54,14 | 69,36 |
| | c.t. | 53,25 | 61,65 | 52,21 | 65,54 |
| | g.r. | -6,53 | -9,28 | -3,56 | -5,51 |
| | t-t. | ▼ | ▼ | ▼ | ▼ |

**Tabela 6.3.** Impacto dos Contextos Temporais (*GreedyChronos*) no Rocchio (TFIDF).

Rocchio é um algoritmo de classificação linear que emprega um modelo de espaço vetorial para representar cada classe por meio de um documento protótipo (Salton, 1971). Cada documento de treino é representado por um vetor em que cada posição do mesmo representa um dos seus termos. A fim de criar o vetor que representa uma determinada classe, uma soma de vetores é feita entre todos os documentos que pertencem à mesma no conjunto de treinamento. Assim, cada classe é representada por um grande vetor que contém informações relacionadas aos termos de todos seus documentos. Para classificar um

documento de teste, é calculada a distância entre o vetor que representa esse documento e os vetores protótipo de cada uma das classes. O documento de teste é assinalado para a classe mais próxima em termos de distância vetorial.

Cada termo de um documento, isto é, cada posição no vetor do documento, é freqüentemente representado utilizando a ponderação tf-idf, ou seja, a freqüência do termo no documento (tf) e o inverso da freqüência do termo entre documentos (idf) (Salton & Buckley, 1987). Diferentes métodos podem ser utilizados para calcular a distância vetorial entre os vetores das classes e os vetores dos documentos de teste, sendo que os mais comuns são o cálculo da distância Euclidiana e a similaridade de cosseno (Salton & McGill, 1983). Em nossos experimentos utilizamos o Rocchio implementado pelo arcabouço de classificação Libbow (McCallum, 1996), no qual o peso de cada termo é definido como:

$$Vector[j] = TFa_j \times \log(\frac{|D|}{DFa_j}) \tag{6.8}$$

onde $TFa_j$ é a freqüência do termo $a_j$, $|D|$ representa o número de documentos de treinamento, e $DFa_j$ o número de documentos em que o termo $a_j$ ocorre. Além disso, na implementação do arcabouço Libbow, a distância vetorial entre documentos de teste e as classes protótipos são calculadas utilizando similaridade de cosseno.

Com o objetivo de explicar por que o Rocchio utilizando contextos temporais não alcançou ganhos, avaliamos o comportamento da Equação 6.8 na linha de base utilizando os contextos temporais. Uma vez que os contextos temporais são baseados apenas nos termos de teste, o tamanho dos contextos é significativamente menor que o conjunto de treinamento completo (menos de 20%). Observando a Equação 6.8, temos que o idf varia entre a linha de base e os contextos temporais. Mais especificamente, o idf dos termos nos contextos temporais são menores que o idf dos termos no conjunto de treino completo.

Baseado nessas observações, nossa hipótese é que, como os contextos temporais se-

lecionados pela heurística *GreedyChronos* são assimétricos, o idf dos termos de teste são bem menores que o idf dos demais termos (termos não-teste) nesses contextos. Enquanto podemos afirmar que existe pelo menos um termo de teste em todos os documentos que compõem os contextos temporais de cada documento de teste, essa afirmação não é verdadeira para os outros termos sendo, conseqüentemente, menos freqüentes. Isso significa que, nos vetores protótipos de cada uma das classes, os termos de teste possuem um peso muito menor que os outros termos. Conseqüentemente, quando a distância entre os vetores que representam os documentos de teste e os vetores protótipo das classes é calculada, como o algoritmo Rocchio considera igualmente ambos os termos de teste e não-teste (uma premissa simétrica), os termos não-teste possuem uma influência maior do que eles realmente deveriam ter, reduzindo significativamente a similaridade entre os documentos de teste e as classes. Esse fato torna ainda mais complicada a tarefa de separação das classes, degradando assim a qualidade da classificação. Note que o fato que os tf dos termos de teste serem bem maiores que o tf dos termos não-teste não é suficiente para evitar esse problema, uma vez que existem muito mais termos não-teste do que de teste.

Para demonstrar nossas hipóteses, realizamos dois conjuntos de experimentos. Primeiro, calculamos a diferença entre o valor do idf dos termos (assinalado pela versão original do Rocchio na linha de base) e os valores assinalados utilizando contextos não-temporais. É importante notar que, utilizando os contextos, o valor do idf de cada termo varia entre os contextos de cada documento de teste. Assim, calculamos a média dos valores de idf para cada termo entre todos os contextos para encontrar um valor único para cada termo. Em seguida, em cada contexto, consideramos separadamente os termos que ocorriam no documento de teste (termos de teste) e aqueles que não ocorriam (termos não-teste). Para cada grupo de termos calculamos a diferença média entre os valores de idf na linha de base e os valores nos contextos, e relativizamos essa diferença pelo valor da linha de base. Finalmente, calculamos a média dessa diferença entre todos os contextos temporais. Para a coleção ACM-DL, o número total de termos de teste é 65.592 e o número total de termos não-teste é 64.008.785. A diferença média encontrada entre os

valores de idf foi de -21% e -9% para termos de teste e não-teste, respectivamente. Para a MedLine, o número total de termos de teste é 2.120.471 e o número total de termos não-teste é 3.821.256.424. A diferença média encontrada foi de -24% e -7% para termos de teste e não-teste, respectivamente. Aplicando um teste-t de dupla cauda, podemos afirmar que a redução dos valores de idf foram maiores para os termos de teste do que para os termos não-teste com uma confiança de 99% para ambas as coleções. Assim, o idf dos termos não-teste é maior que o idf dos termos de teste utilizando os contextos temporais, Além disso, percebemos que o uso de contextos temporais tende a piorar esse efeito colateral, uma vez que os contextos temporais possuem ainda menos documentos que os contextos não-temporais. Essa fato aumenta ainda mais a influência dos termos não-teste no processo de classificação. Essa efeito faz com que o processo de classificação seja quase aleatório, uma vez que a classificação dos documentos de teste são cada vez mais dependentes dos termos que não ocorrem nos mesmo documentos.

A fim de reforçar nosso argumento e demonstrar o impacto do aumento dos valores de idf dos termos não-teste, mudamos a implementação do algoritmo Rocchio no arcabouço Libbow para considerar apenas o tf dos termos, isto é, criamos uma versão do Rocchio em que cada posição do vetor é representada como:

$$Vector[j] = TFa_j \tag{6.9}$$

Em seguida realizamos os mesmos experimentos como anteriormente. Dividimos o conjunto de treinamento em dois subconjuntos, treinamento e validação, e procuramos pelos valores do parâmetro Predominância que alcançam os melhores resultados de classificação utilizando o conjunto de validação como teste para cada coleção (empregamos um validação cruzada de 10 partes). Então, aplicamos a heurística *GreedyChronos* para encontrar os contextos temporais para os documentos de teste utilizando o melhor valor encontrado. Os melhores resultados obtidos em nossos experimentos são apresentados

na Tabela 6.4. As linhas com o o rótulo “ct-tf.” apresentam os valores referentes a essa nova versão do Rocchio utilizando os contextos temporais, ou seja, utilizando contextos temporais sem considerar o idf dos termos.

| Coleção | | ACM-DL | | MedLine | |
|---|---|---|---|---|---|
| Métrica | | $macF_1$(%) | acc.(%) | $macF_1$(%) | acc.(%) |
| Rocchio | l.b. | 56,97 | 67,95 | 54,14 | 69,36 |
| | ct-tf. | 59,39 | 72,59 | 60,42 | 72,78 |
| | g.r. | +6,82 | +4,25 | +11,60 | +4,93 |
| | t-t. | ▲ | ▲ | ▲ | ▲ |

**Tabela 6.4.** Impacto dos Contextos Temporais (*GreedyChronos*) no Rocchio(TF)

Pelos resultados apresentados na Tabela 6.4, observamos que ganhos significativos foram obtidos se comparados com a versão original do Rocchio que utiliza todo o conjunto de treinamento, em ambas as coleções. Comparando esses resultados com os obtidos pela versão original do Rocchio utilizando contextos temporais (Tabela 6.3), temos que os ganhos são ainda mais significativos. Esses experimentos ajudam a demonstrar como as mudanças dos valores de idf dos termos nos contextos temporais afetam a qualidade da classificação.

### 6.2.4 SVM

O resultados obtidos utilizando os contextos temporais selecionados pela heurística *GreedyChronos* e o algoritmo SVM são apresentados na Tabela 6.5. Como podemos observar, os resultados para o algoritmo SVM usando esses contextos foram piores para ambas as medidas em ambas as coleções.

O algoritmo Support Vector Machine é baseado no princípio de Minimização do Risco Estrutural (*Structural Risk Minimization*) dentro da teoria de aprendizagem computacional (Vapnik, 1998). De acordo com (Joachims, 1998b), a idéia principal da minimização do risco estrutural é encontrar uma modelo para a qual podemos garantir o menor erro verdadeiro. O erro verdadeiro do modelo é a probabilidade dele não classificar corretamente um exemplo de teste randomicamente escolhido. Em outras palavras, todas as

| Coleção | | ACM-DL | | MedLine | |
|---|---|---|---|---|---|
| Métrica | | $macF_1$(%) | acc.(%) | $macF_1$(%) | acc.(%) |
| SVM | l.b. | 60,07 | 73,03 | 72,28 | 83,27 |
| | c.t. | 56,60 | 71,19 | 67,15 | 80,13 |
| | g.r. | -5,78 | -2,52 | -7,1 | -3,77 |
| | t-t. | ▼ | ▼ | ▼ | ▼ |

**Tabela 6.5.** Impacto dos Contextos Temporais (*GreedyChronos*) no SVM.

características dos documentos de treinamento (e.g., termos) são colocadas em um espaço vetorial e o algoritmo SVM tenta encontrar os hiperplanos (o modelo) que melhor delimitam as classes desses documentos. Documentos aleatórios do conjunto de treino são utilizados como teste (validação) para avaliar a qualidade dos hiperplanos gerados. Esse processo de otimização corresponde ao processo de aprendizagem.

Para o algoritmo SVM, uma função *Kernel* apropriada é responsável pelo processo de aprendizagem. Existem diversas funções, baseadas nos mais diversos princípios, tais como *three-layer sigmoid neural nets* e *maximum-margin* (o qual é utilizado em nossos experimentos). Ao final do processo de aprendizagem, o algoritmo SVM provê pesos para cada termo em cada classe diferente. Um termo com peso positivamente alto para uma determinada classe indica que documentos com esses termos estão provavelmente relacionados à classe em questão, enquanto que pesos negativos indicam exatamente o contrário. Todos os termos e seus pesos, positivos e negativos, são utilizados pelo algoritmo SVM para classificar os documentos de teste, o que claramente corresponde a uma premissa simétrica. Dessa forma, nossa hipótese é que utilizando-se contextos temporais assimétricos, selecionados baseados somente nos termos de teste, a informação que é provida pelos termos não-teste que permanecem nos contextos temporais pode estar distorcida se comparada com o resultado que utiliza toda a coleção (linha de base) e esse fato pode contribuir para a degradação dos resultados do algoritmo SVM.

A fim de demonstrar nossa hipótese, primeiramente avaliamos a qualidade do algoritmo SVM utilizando os contextos não-temporais, ou seja, o comportamento de uma abordagem sob-demanda do algoritmo SVM utilizando contextos não-temporais. Apesar

do custo computacional de tal abordagem ser muito alto e inviável, a avaliação de tal abordagem será muito importante para nos ajudar a entender e caracterizar o comportamento do uso de contextos temporais pelo algoritmo SVM. Nesse cenário, o SVM alcançou 70,03% e 55,61% para acurácia e macroF1, respectivamente, para a coleção ACM-DL e 79,13% e 67,15% de acurácia e macroF1 para a coleção MedLine. Comparamos esses resultados com a linha de base apresentada na Tabela 6.5 e, dada uma confiança de 99% em um teste-t de dupla cauda, podemos afirmar que esses resultados são estatisticamente inferiores a linha de base. Observando esses resultados, temos a primeira evidência que a falta de ganhos do algoritmo SVM não está relacionada com as características temporais mas, novamente, com o fato que contextos temporais e não-temporais são assimétricos, compostos apenas por documentos que contém pelo menos um dos termos de teste.

No segundo conjunto de experimentos avaliamos a distribuição de probabilidade de ocorrência dos termos não-teste entre as diversas classes, utilizando contextos temporais, contextos não-temporais e toda a coleção. Para isso, primeiro calculamos a freqüência de cada termo não-teste em cada uma das classes em toda a coleção (em ambas, ACM-DL e MedLine). Em seguida realizamos o mesmo cálculo em cada contexto (temporal e não temporal). Então, para cada termo não-teste, calculamos o diferença absoluta entre sua freqüência nas classes de um dado contexto e em toda a coleção. A distorção de cada termo é definida como sendo a média dessas diferenças entre todas as classes. Em seguida, calculamos a média das distorções entre os termos não-teste para encontrar a distorção de cada contexto. Por fim, calculamos a distorção média entre todos os contextos (temporais e não-temporais separadamente). Para os contextos temporais encontramos uma distorção de 53,9% e 63,5% para as coleções ACM-DL e MedLine, respectivamente. Para os contextos não-temporais encontramos uma distorção de 19,3% e 25,3% para as coleções ACM-DL e MedLine, respectivamente. Analisando os resultados relacionados à distorção dos termos não-teste, podemos concluir que a representatividade desses termos nos contextos (temporais ou não) é muito diferente da representatividade encontrada em toda a coleção. Além disso, essa distorção acontece arbitrariamente, sem nenhum con-

trole. No caso dos contextos não-temporais, a distorção da representatividade dos termos de teste é 0% (como vimos, contextos temporais são compostos por todos os documentos que contém pelo menos um dos termos do documento de teste) demonstrando que a perda de qualidade observada quando o SVM utiliza esses contextos é causada pela distorção de representatividade dos termos não-teste. Como podemos ver, a distorção é ainda maior para os contextos temporais. Apesar dos termos de teste apresentarem um bom poder discriminativo nos contextos temporais selecionados pela heurística *GreedyChronos*, como mostrado anteriormente, os mesmos não são suficientes para contornar a degradação da qualidade da classificação causada pela distorção dos termos não-teste.

Finalmente, no terceiro conjunto de experimentos, avaliamos se essas distorções dos termos não-teste foram sempre prejudiciais para o algoritmo SVM. Para isso, avaliamos o SVM usando contextos (temporais e não-temporais) removendo todos os termos não-teste. Utilizando os contextos não-temporais o SVM obteve uma acurácia de 60,59% (ACM-DL) e 69,14% (MedLine) e uma macroF1 de 42,72% (ACM-DL) e 52,76% (MedLine). Usando os contextos temporais o SVM alcançou uma acurácia de 67,14% (ACM-DL) e 74,89% (MedLine) e uma macroF1 de 49,21% (ACM-DL) e 59,24% (MedLine). Como podemos observar, os resultados obtidos foram piores que os resultados alcançados utilizandos todos os termos, o que significa que as distorções dos termos não-teste algumas vezes podem ajudar no processo de classificação. Além disso, comparando os resultados de contextos temporais e não-temporais sem utilizar os termos não-teste, podemos ver que os resultados obtidos utilizando-se contextos temporais foram bem melhores, o que significa, mais uma vez, que a heurística *GreedyChronos* é capaz de melhorar a representatividade dos termos de teste (bom poder discriminativo). Entretanto, os ganhos de qualidade não são grandes o suficiente para melhorar a qualidade do SVM se comparado com a linha de base, uma vez que a representatividade dos termos não-teste, que também são utilizados pelo SVM e são em maior número, é muito distorcida.

## 6.3 Avaliação da Heurística WindowChronos

O primeiro passo para selecionar os contextos temporais é, para cada conjunto de teste de cada ano base $A_i$, determinar a estabilidade dos termos, a qual é calculada utilizando a métrica Predominância. Com o objetivo de avaliar qual o valor de Predominância é o mais apropriado de ser adotado, realizamos um experimento em que variamos esse valor de 50%[1] até 90% e geramos diversos histogramas, um para cada ano base $A_i$, com o número de ocorrências dos anos nos períodos de estabilidade dos termos. Analisando esses diversos histogramas em cada uma das coleções, observamos que suas distribuições foram muito similares, ou seja, os anos com maior porcentagem de ocorrência nos histogramas de um ano base $A_i$ foram quase os mesmos para os diversos valores de Predominância, variando apenas a magnitude das freqüências.

A fim de ilustrar essa análise, nas Figuras 6.6, 6.7, 6.8 e 6.9 apresentamos alguns desses histogramas. Nas Figuras 6.6 e 6.7 apresentamos os histogramas para a coleção ACM-DL, com três diferentes valores de Predominância (50%, 70% e 80%), para os anos base 1981 e 1997, respectivamente. As Figuras 6.8 e 6.9 mostram os histogramas para a coleção MedLine para os anos base 1979 e 1981, respectivamente. Como mencionado anteriormente, os histogramas do mesmo ano base $A_i$ são similares para diferentes valores da métrica Predominância, para cada coleção.

Na Figura 6.10 sumarizamos a análise descrita para ambas as coleções, ACM-DL e MedLine. Utilizando o valor de Predominância igual à 50%, para cada ano base $A_i$ calculamos a popularidade de todos os anos que fazem parte do período de estabilidade de todos os termos que ocorrem em $A_i$. Para sumarizar os resultados de todos os anos base $A_i$, ao invés de utilizarmos efetivamente os valores dos anos que compõem os períodos de estabilidade, utilizamos a distância temporal relativa ao ano base $A_i$. Por exemplo, para o ano base 1985, as popularidades dos anos 1984, 1985 e 1986 no período de estabilidade dos termos são, respectivamente, 39%, 45% e 40%. Nesse caso, a distância temporal seria

[1] Esse valor garante que, independente do número de classes da coleção, um termo estará predominantemente associado a uma determinada classe

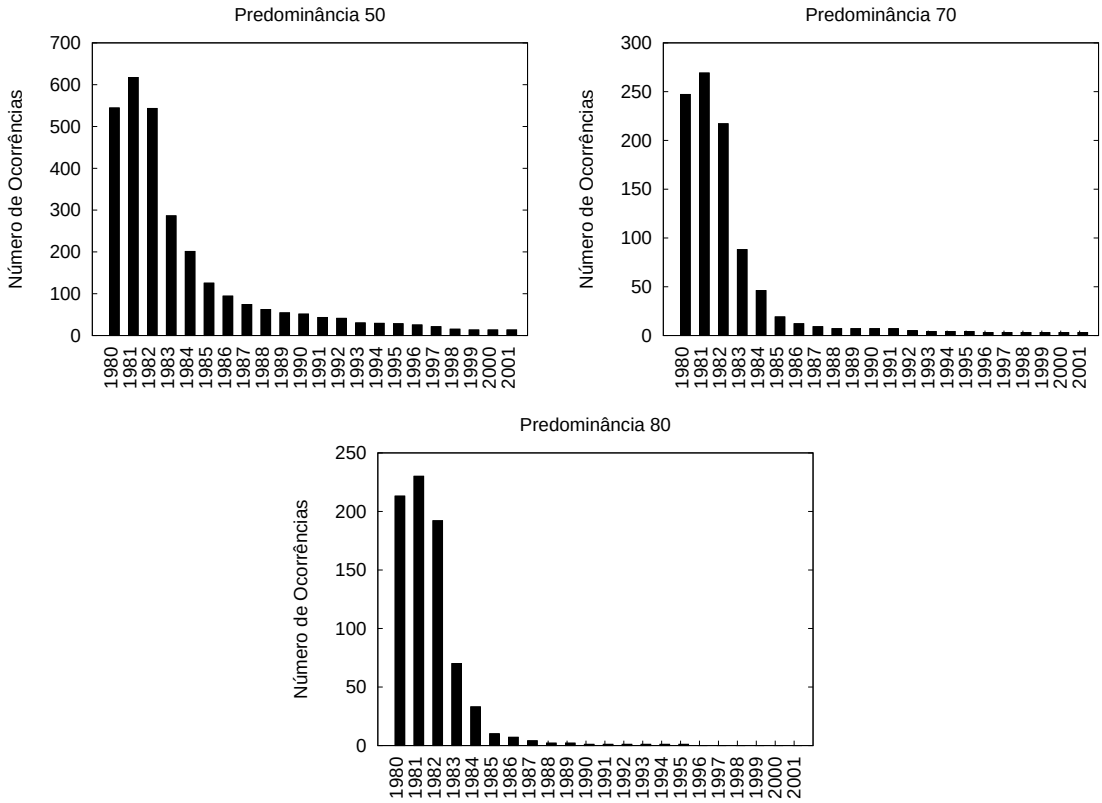



**Figura 6.6.** Popularidade dos Anos nas Janelas Temporais dos Termos (Ano 1981) - ACM-DL

-1 (1984), 0 (1985) e 1 (1986) onde os valores 39%, 45% e 40% representam a porcentagem dos termos de 1985 que possuem esses anos (distância temporal) em seus períodos de estabilidade. Por fim, computamos para cada distância temporal a porcentagem de ocorrência média entre todos os anos base $A_i$.

Observando a Figura 6.10, em ambas as coleções os anos que estão temporalmente perto do ano base são os mais freqüentes no período de estabilidade dos termos. À medida que aumenta a distância temporal (para o passado e para o futuro), a freqüência de ocorrência diminui, ou seja, existem poucos termos que permanecem estáveis por um longo período. Entretanto, podemos notar que, enquanto na ACM-DL o decaimento é muito acentuado, na MedLine esse decaimento é mais suave. Além disso, observando a diferença entre a porcentagem de ocorrência do ano mais freqüente e o ano menos freqüente em cada coleção, temos que, enquanto na MedLine o ano mais freqüente é aproximadamente duas vezes maior que o menos freqüente, na ACM-DL essa diferença é mais

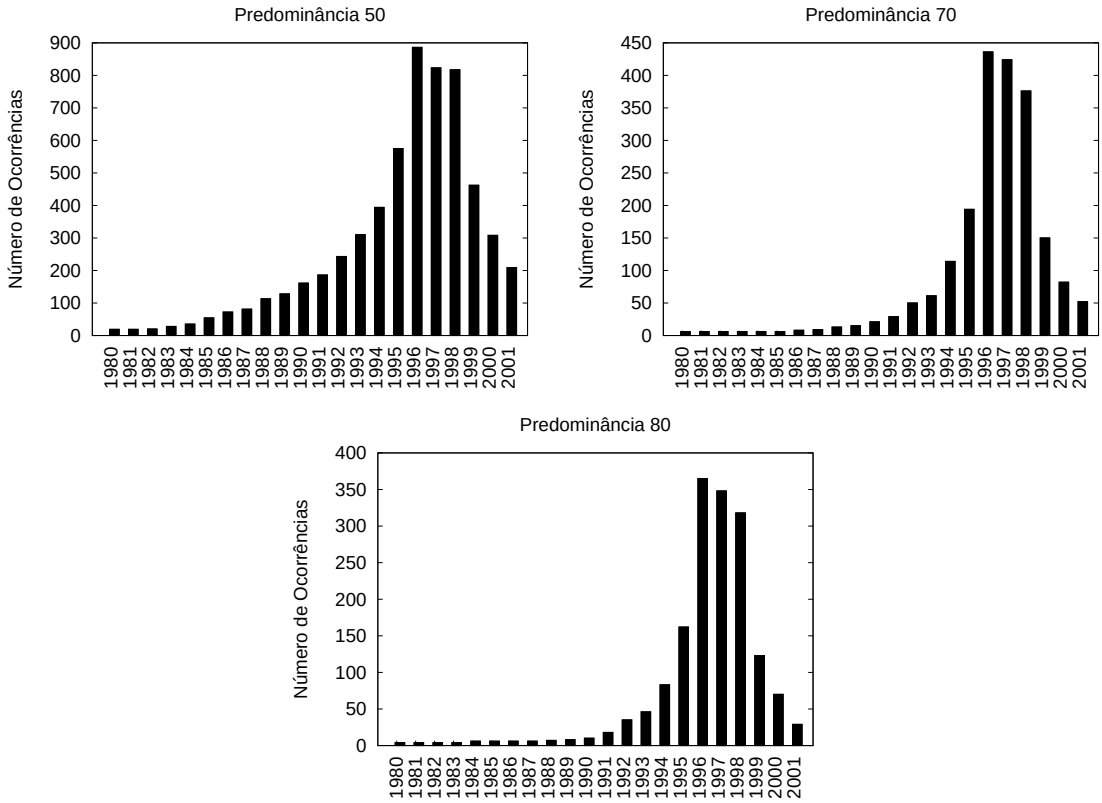



**Figura 6.7.** Popularidade dos Anos nas Janelas Temporais dos Termos (Ano 1997) - ACM-DL

acentuada, aproximadamente quatro vezes maior. Essas observações podem ser explicadas pelo fato que a coleção ACM-DL é bem mais afetada pelos efeitos temporais que a MedLine, conforme mostramos no Capítulo 3. Por fim, observando o decaimento da popularidade à medida que a distância temporal aumenta, podemos perceber que, enquanto na ACM-DL o decaimento do lado esquerdo da curva (passado) é muito parecido com o decaimento do lado direito (futuro), na MedLine o decaimento do lado esquerdo é mais acentuado que o direito. Isto ocorre uma vez que na MedLine um número maior de novos termos é introduzido ao longo dos anos, conseqüentemente, o período de estabilidade desses novos termos é composto mais freqüentemente por anos futuros.

Comparando os gráficos da Figura 6.10 com os gráficos das Figura 3.4 (Variação da Localidade Temporal) e 3.9 (Média da Distribuição dos Termos), apresentados no Capítulo 3, podemos concluir que a Predominância é uma métrica que pode ser muito bem utilizada para determinar o período de estabilidade dos termos. Além disso, conforme

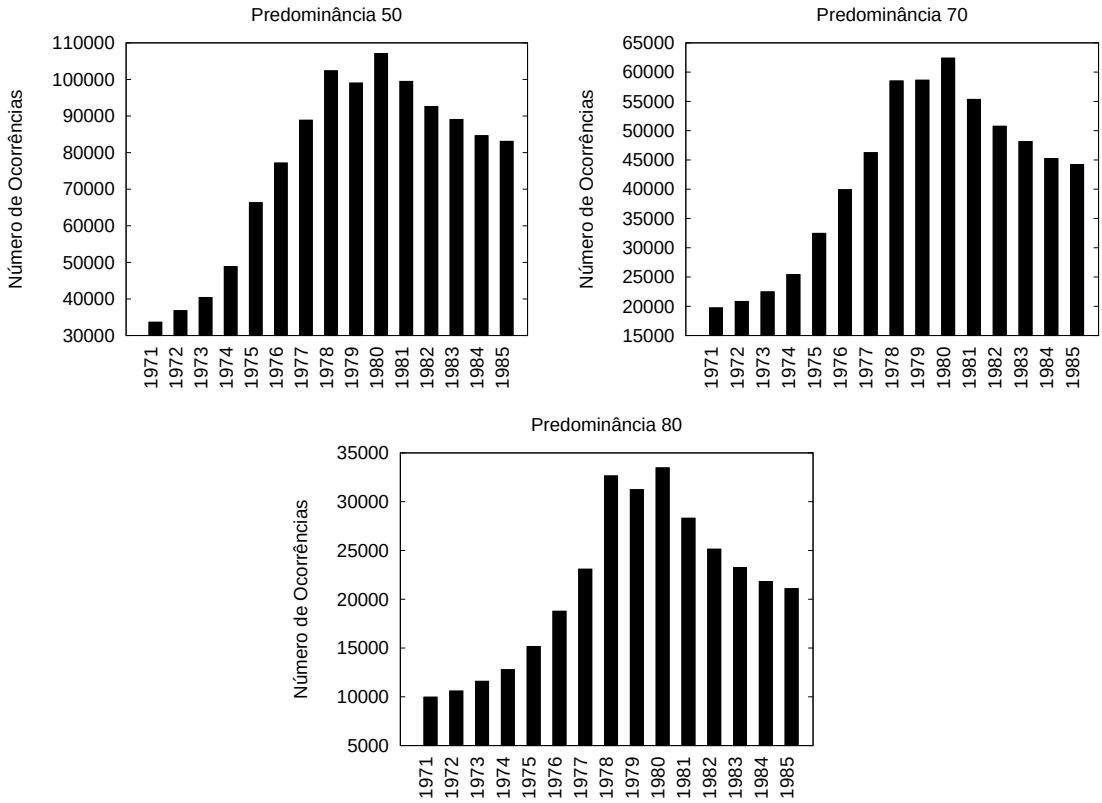



**Figura 6.8.** Popularidade dos Anos nas Janelas Temporais dos Termos (Ano 1979) - MedLine

observamos nas Figuras 6.6, 6.7, 6.8 e 6.9, os anos com maior porcentagem de ocorrência nos histogramas praticamente são os mesmos para os diversos valores de Predominância, em ambas as coleções. Dessa forma, qualquer valor de Predominância maior ou igual a 50% pode ser utilizado para determinar quais os anos irão compor os contextos temporais dos conjuntos de documentos de teste para cada ano base $A_i$. Intuitivamente, esse mesmo comportamento é esperado para outras coleções de documentos, uma vez que a qualidade dos classificadores está diretamente relacionada com a estabilidade das características nas coleções, a qual é assegurada por uma Predominância maior ou igual à 50%. Entretanto, uma análise similar pode ser feita caso a heurística *WindowChronos* seja aplicada em outras coleções de documentos quaisquer.

Feita a escolha de qual valor de Predominância que deverá ser utilizado, o próximo passo é realizar um conjunto de experimentos que determinam os contextos temporais associados aos conjuntos de documentos de teste de cada ano, para cada algoritmo de CAD.

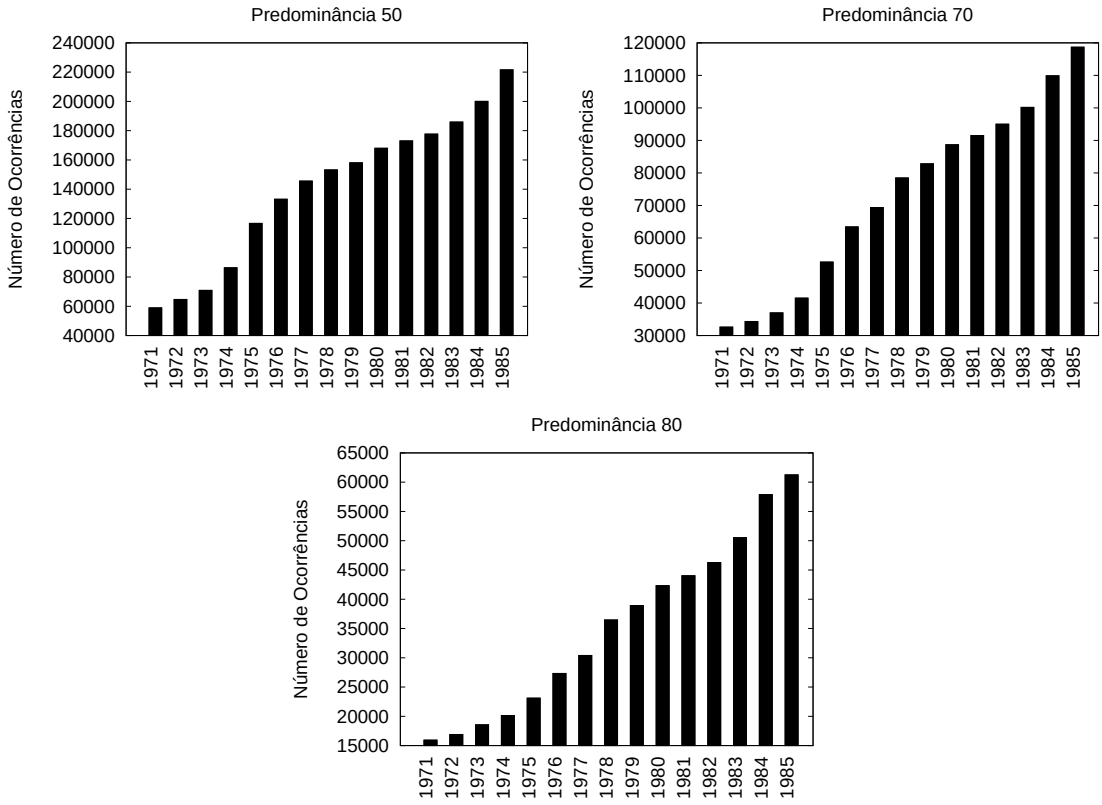
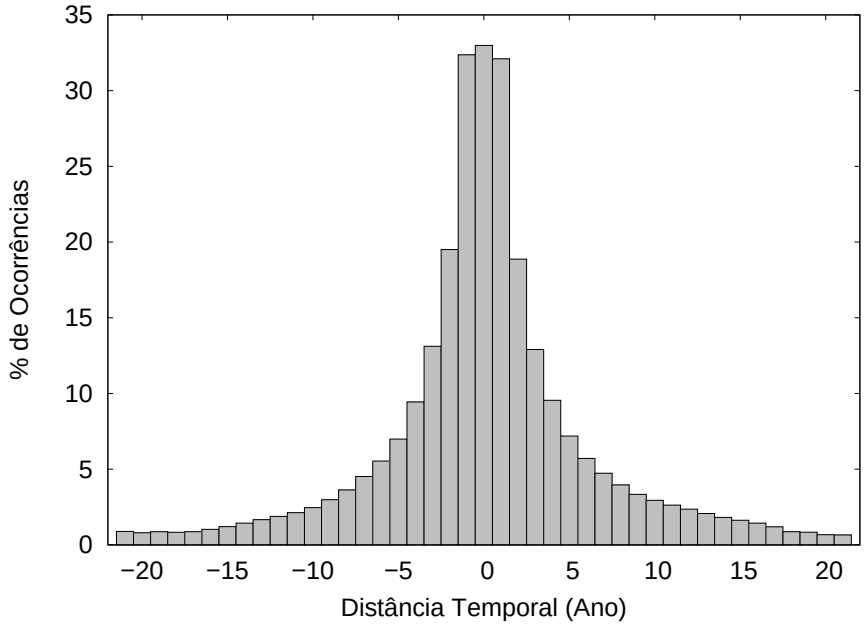


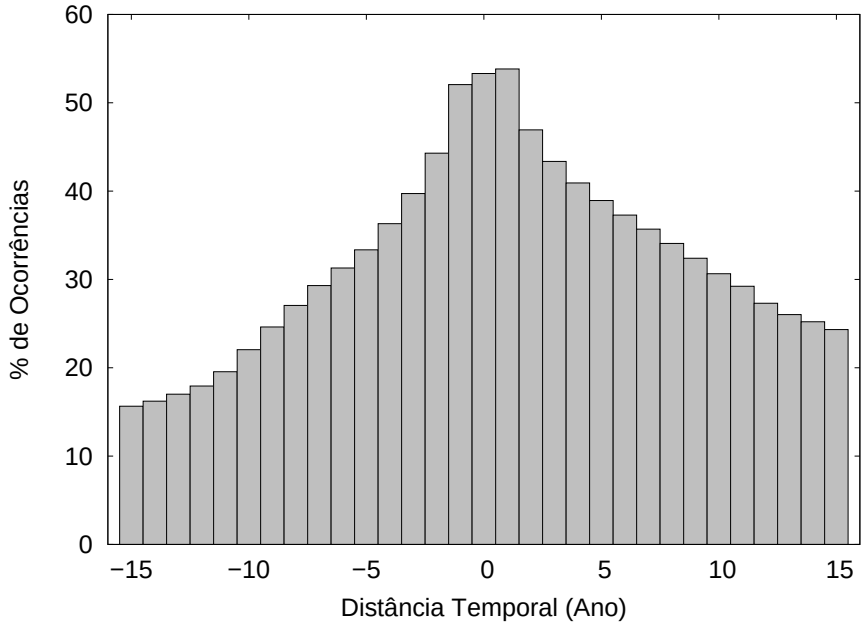

**Figura 6.9.** Popularidade dos Anos nas Janelas Temporais dos Termos (Ano 1985) - MedLine

Com esse objetivo, separamos o conjunto de treinamento em treinamento e validação. Em seguida, utilizando o valor de Predominância igual a 50% e uma validação cruzada de 10 partes, procuramos pelo valor de $N$ (os anos que irão compor a janela temporal dos contextos) que apresenta o melhor desempenho no conjunto de validação para cada combinação entre algoritmo e coleção, uma vez que as características específicas de cada coleção e algoritmo podem afetar a escolha de $N$. Aplicamos o melhor valor de $N$ encontrado em cada uma das combinações para produzir os contextos temporais dos conjuntos de documentos de teste.

Nesses primeiros experimentos, os valores de $N$ foram encontrados por meio de uma busca completa das alternativas, variando $N$ de 1 até $Y$ sem nenhum tipo de poda (e.g., na ACM-DL existem 22 anos base, conseqüentemente, no geral, 484 alternativas foram avaliadas). Além disso, empregamos uma validação cruzada de 10 partes (Brieman & Spector, 1992) e os resultados finais de cada experimento é a média das




(a) Coleção ACM-DL




(b) Coleção MedLine

**Figura 6.10.** Popularidade dos Anos nas Janelas Temporais dos Termos

dez execuções.

Os resultados obtidos em nossos experimentos em cada um dos cenários são apresentados na Tabela 6.6. Apresentamos os resultados das classificações utilizando as métricas macroF1 e acurácia. As linhas com o rótulo "l.b." (linha de base) apresentam

os valores referentes às execuções usando o conjunto de treino completo sem considerar os aspectos temporais, enquanto as linhas com o rótulo "c.t." apresentam os valores alcançados nas classificações utilizando os contextos temporais selecionados pela *WindowChronos*. As linhas com o rótulo "g.r." representam a diferença percentual entre a classificação com os contexto temporal e a linha de base. Por fim, as linhas com o rótulo "t-t" descrevem se as variações produzidas utilizando-se contextos temporais representam diferenças estatisticamente significativas, dada uma confiança de 99% em um teste-t de dupla cauda (os ganhos positivos são representados por ▲, as perdas são representadas por ▼ e os resultados estatisticamente equivalentes são representados por ●).

| Coleção | | ACM-DL | | MedLine | |
|---|---|---|---|---|---|
| Métrica | | $macF_1$(%) | acc.(%) | $macF_1$(%) | acc.(%) |
| kNN | l.b. | 56,78 | 69,80 | 66,57 | 79,82 |
| | c.t. | 61,90 | 74,44 | 68,97 | 81,87 |
| | g.r. | +9,01 | +6,53 | +3,61 | +2,57 |
| | t-t. | ▲ | ▲ | ▲ | ▲ |
| Naïve Bayes | l.b. | 56,87 | 74,00 | 65,64 | 79,65 |
| | c.t. | 59,76 | 76,56 | 68,95 | 81,88 |
| | g.r. | +5,09 | +3,45 | +5,04 | +2,80 |
| | t-t. | ▲ | ▲ | ▲ | ▲ |
| Rocchio | l.b. | 56,97 | 67,95 | 54,14 | 69,36 |
| | c.t. | 61,66 | 72,68 | 56,99 | 71,71 |
| | g.r. | +8,24 | +6,95 | +5,26 | +3,39 |
| | t-t. | ▲ | ▲ | ▲ | ▲ |
| SVM | l.b. | 60,07 | 73,03 | 72,28 | 83,27 |
| | c.t. | 62,32 | 76,50 | 73,98 | 84,90 |
| | g.r. | +3,75 | +3,80 | +2,35 | +1,96 |
| | t-t. | ▲ | ▲ | ▲ | ▲ |

**Tabela 6.6.** Impacto dos Contextos Temporais (*WindowChronos*).

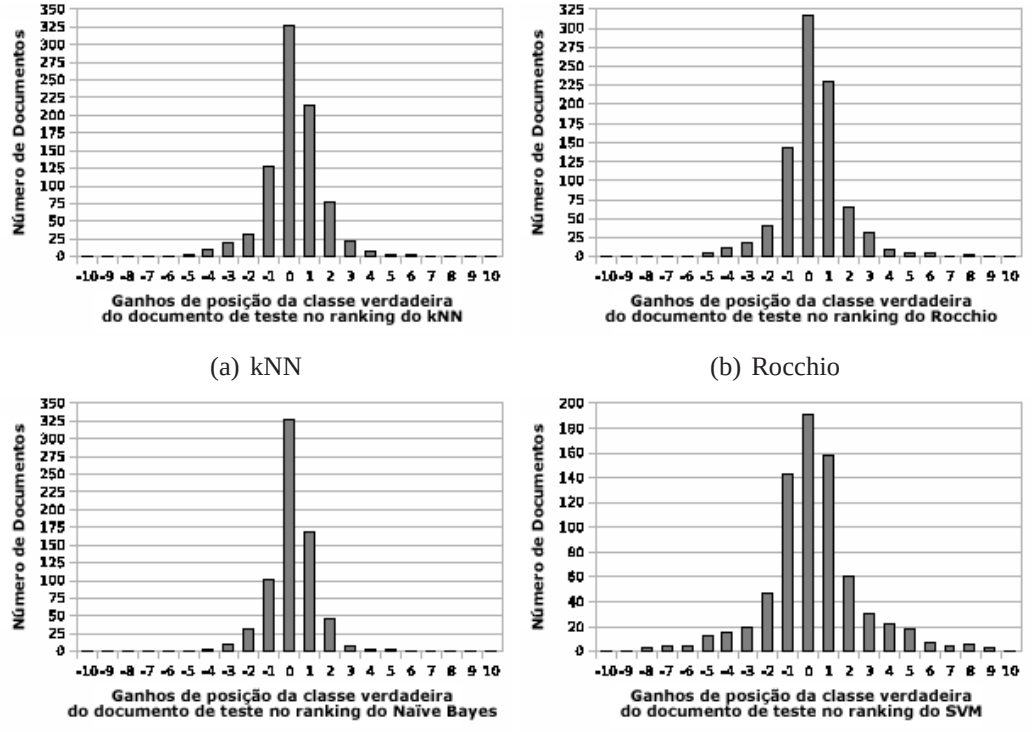
Como podemos observar, utilizando os contextos temporais selecionados pela heurística *WindowChronos* foi possível alcançar melhorias estatisticamente significativas em todos os algoritmos, quando comparados com os cenários em que esses contextos não foram utilizados. As melhorias obtidas pelo algoritmo KNN podem ser explicadas, assim como nos resultados relacionados à heurística *GreedyChronos*, pela melhoria da qualidade

dos termos de teste em contextos temporais, reduzindo a ambigüidade deles entre as classes. Para os demais algoritmos, a heurística *WindowChronos* considera todos os termos que ocorrem em um determinado ano (não somente os termos de teste) para selecionar os contextos temporais, que são simétricos. Conseqüentemente, as premissas adotadas por esses algoritmos de CAD, que também são simétricas, não são afetadas, conforme mostrado a seguir:

- **Naïve Bayes**: a premissa que a representatividade dos termos de teste nas classes é medida em função da ocorrência dos demais termos que ocorrem na classe permanece válida, uma vez que todos eles são igualmente considerados nos contextos.
- **Rocchio**: a raridade de todos os termos pode ser utilizada pelo Rocchio no cálculo dos vetores protótipos das classes, uma vez que a grande diferença entre a raridade dos termos de teste e não-teste que ocorre nos contextos temporais selecionados pela *GreedyChronos* não ocorre nos contextos selecionados pela *WindowChronos*.
- **SVM**: todos os termos e seus pesos, positivos e negativos, podem ser efetivamente utilizados pelo algoritmo SVM, uma vez que a informação que é provida por todos os termos que ocorrem nos contextos temporais selecionados pela heurística *WindowChronos* tende a ser menos confusa.

Todos esses algoritmos de CAD criam, para cada documento de teste, um *ranking* de classes baseados nas pontuações assinaladas por eles. O documento de teste é então classificado para a classe que se encontra na primeira posição desse *ranking*. Para analisar mais detalhadamente o impacto dos contextos temporais selecionados pela heurística *WindowChronos*, inspecionamos como esses contextos alteram o *ranking* gerado pelos algoritmos analisando como a posição da classe verdadeira dos documentos de teste é afetada. Para isso, consideramos a mesma variável $X$ definida pela Equação 6.2.1, apresentada na Seção 6.2.

Assim, avaliamos a variável $X$ para ambas as coleções e todos os algoritmos de CAD. Os resultados referentes às coleções ACM-DL e MedLine podem ser observados




(a) kNN (b) Rocchio

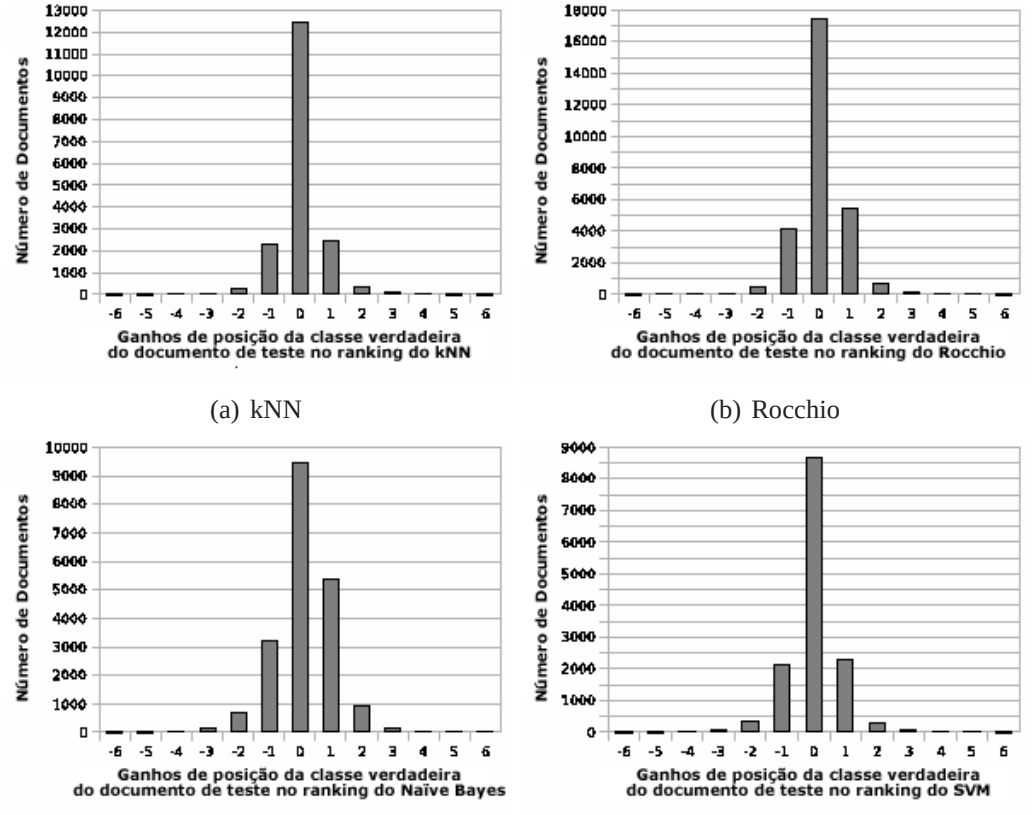
(c) Naïvez Bayes (d) SVM

**Figura 6.11.** Efeito dos Contextos Temporais sobre os Algoritmos - ACM

nas Figuras 6.11 e 6.12, respectivamente. Além disso, da mesma forma que foi feito no Capítulo 5, também quantificamos a obliqüidade (*skewness*) das distribuições utilizando o método *Third Standardized Moment's Mode Skewness* (Groeneveld, 1991). O valor encontrado foi positivo para todos os algoritmos em ambas as coleções, conforme podemos observar na Tabela 6.7. Esse fato demonstra que, para todos os cenários avaliados, os contextos temporais selecionados pela heurística *WindowChronos* gera um deslocamento positivo grande o suficiente para melhorar a posição no *ranking* das classes corretas.

| Algoritmo | | kNN | Rocchio | Naïve Bayes | SVM |
|---|---|---|---|---|---|
| Coleção | ACM-DL | +0,09 | +0,30 | +0,10 | +0,20 |
| | MedLine. | +0,20 | +0,10 | +0,09 | +0,10 |

**Tabela 6.7.** Valores de Obliqüidade




(a) kNN (b) Rocchio

(c) Naïve Bayes (d) SVM

**Figura 6.12.** Efeito dos Contextos Temporais sobre os Algoritmos - MedLine

### 6.3.1 Otimização da Heurística *WindowChronos*

Nesta seção avaliamos o impacto da estratégia de redução do custo computacional, previamente apresentada, na qualidade dos resultados. Essa estratégia é baseada em podas no número de alternativas a serem avaliadas para encontrar o valor de $N$ (os anos que irão compor a janela temporal dos contextos). Para isso, realizamos um novo conjunto de experimentos. Como antes, separamos o conjunto de treinamento em um conjunto de validação e outro de treinamento. Em seguida, utilizando um valor de Predominância igual à 50%, procuramos pelo valor de $N$ que apresenta o melhor desempenho no conjunto de validação para cada combinação entre algoritmo e coleção. Entretanto, nesses novos experimentos, a busca pelo melhor valor de $N$ foi interrompida quando o desempenho do classificador não apresentou melhoras após um determinado número de tentativas (5 tentativas para a coleção ACM-DL e 3 tentativas para a coleção MedLine). Aplicamos

também uma validação cruzada de 10 partes e os resultados finais de cada experimento representam a média das dez execuções.

Os melhores resultados alcançados em nossos experimentos estão apresentados na Tabela 6.8, onde as linhas com o rótulo "c.t." apresentam os valores para a classificação utilizando os contextos temporais gerados sem nenhum tipo de poda e as linhas com o rótulo "c.t.P" apresentam os valores para as execuções que aplicaram a estratégia de poda. As linhas com o rótulo "g.r." representam a diferença percentual entre a classificação utilizando contextos com com poda e sem poda. Por fim, as linhas com o rótulo "t-t" descrevem se as variações produzidas utilizando-se contextos temporais com poda ("c.t.P"), com respeito aos contextos gerados sem poda ("c.t.") e representam diferenças estatisticamente significativas, dada uma confiança de 99% em um teste-t de dupla cauda (os ganhos positivos são representados por ▲, as perdas são representadas por ▼ e os resultados estatisticamente equivalentes são representados por ●).

| Coleção | | ACM-DL | | MedLine | |
|---|---|---|---|---|---|
| Métrica | | macF$_1$(%) | acc.(%) | macF$_1$(%) | acc.(%) |
| kNN | c.t. | 61,90 | 74,44 | 68,97 | 81,87 |
| | c.t.P | 61,56 | 74,15 | 68,94 | 81,81 |
| | g.r. | -0,54 | -0,39 | -0,04 | -0,07 |
| | t-t. | ● | ● | ● | ● |
| Naïve Bayes | c.t. | 59,76 | 76,56 | 68,95 | 81,88 |
| | Ptc. | 59,21 | 76,23 | 68,75 | 81,78 |
| | g.r. | -0,92 | -0,43 | -0,29 | -0,12 |
| | t-t. | ● | ● | ● | ● |
| Rocchio | c.t. | 61,66 | 72,68 | 56,99 | 71,71 |
| | c.t.P | 61,44 | 72,35 | 56,90 | 71,68 |
| | g.r. | -0,36 | -0,45 | -0,16 | -0,04 |
| | t-t. | ● | ● | ● | ● |
| SVM | c.t. | 62,32 | 76,50 | 73,98 | 84,90 |
| | c.t.P | 62,03 | 76,01 | 73,75 | 84,40 |
| | g.r. | -0,64 | -0,47 | -0,31 | -0,59 |
| | t-t. | ● | ● | ● | ● |

**Tabela 6.8.** Impacto do uso de Poda na *WindowChronos*

O uso de podas na busca pelo melhor valor de $N$ provê uma redução média de

32% no número total de alternativas avaliadas. Além disso, como podemos observar, para todos os algoritmos de CAD, em ambas as coleções, o uso da estratégia de poda não resultou em reduções estatisticamente significativas no eficácia dos algoritmos, ou seja, a estratégia de poda possui uma boa aproximação comparada à solução que avalia todas as possibilidades. Esse fato pode ser explicado uma vez que o relacionamento entre os termos e as classes tende a se modificar suavemente ao longo do tempo, como apresentamos no Capítulo 3. Conseqüentemente, conforme podemos observar na Figura 6.10, os anos que compõem o melhor valor de $N$ tendem a ser contíguos e temporalmente próximos do conjunto de documentos de teste.

## 6.4 Sumário

Neste capítulo avaliamos duas heurística para seleção de contextos temporais denominadas: *GreedyChronos* e *WindowChronos*, ambas baseadas nos requisitos apresentados no Capítulo 4 e no Algoritmo *Chronos*. Avaliamos essas heurísticas utilizando diferentes algoritmos de CAD, como Naïve Bayes, kNN, Rocchio e SVM, usando duas coleções de documentos distintas: a biblioteca digital da ACM (que contém documentos relacionados à área de Ciência da Computação) e a MedLine (que contém documentos relacionados à Medicina). Além disso, realizamos também uma análise detalhada do comportamento de cada algoritmo de CAD utilizando contextos temporais selecionados por essas heurísticas.

Por meio de nossos resultados apresentados, concluímos que o uso dos contextos temporais selecionados pela heurística *GreedyChronos* se apresenta como uma boa alternativa para melhorar a qualidade do algoritmo KNN. Por se tratar de um classificador que utiliza uma premissa assimétrica, assim como a heurística *GreedyChronos*, baseia-se apenas nos termos dos documentos de teste. Esse fato sugere que a heurística de seleção de contextos em questão pode funcionar bem com outras abordagens que também sejam baseadas em premissas assimétricas similares. Isso é algo que pretendemos investigar em um futuro próximo. Entretanto, existem algoritmos de classificação automática de docu-

mentos que são baseados em premissas simétricas e, conseqüentemente, utilizam direta (SVM) ou indiretamente (Naïve Bayes e Rocchio) todos os termos. Apesar da redução de ambigüidade dos termos de teste entre as classes nos contextos temporais selecionados pela heurística *GreedyChronos*, o mesmo não pode ser garantido para os demais termos que são incluídos nos contextos. Conseqüentemente, as premissas dos algoritmos que consideram todos os termos podem ser afetados, como discutido a seguir:

- **Naïve Bayes**: é baseado na premissa principal e simétrica de que a representatividade dos termos de teste em uma dada classe é baseada em todos os termos que ocorrem nessa classe, ou seja, os termos não-teste que ocorrem nos contextos temporais afetam a probabilidade dos termos de teste em uma dada classe. Conseqüentemente, essa premissa principal é seriamente afetada.

- **Rocchio**: é baseado na premissa simétrica de que todas os termos são utilizadas para criar os vetores protótipos de cada uma das classes e também para calcular a similaridade entre os documentos de teste e as classes. Entretanto, a raridade dos termos não-teste é bem maior que a raridade dos termos de teste, uma vez que existem pelo menos um termo de teste em todos os documentos que formam os contextos temporais. Conseqüentemente, nos vetores protótipos que representam as classes, o peso dos termos não-teste é maior que dos termos de teste. Quando a distância vetorial entre os documentos de teste e os protótipos das classes é calculada, os termos não-teste possuem maior influência do que realmente deveriam ter.

- **SVM**: todos os termos e seus pesos, positivos ou negativos, são efetivamente utilizados pelo SVM para classificar os documentos de teste, o que é claramente uma premissa simétrica. Entretanto, a informação que é provida pelos termos não-teste que ocorrem nos contextos temporais podem estar distorcidas se comparadas com os resultados utilizando toda a coleção e esse fato contribuí para degradar os resultados do SVM.

Observando os resultados apresentados relacionados a heurística *WindowChronos*, podemos concluir que o uso dos contextos temporais selecionados por essa heurística se apresenta como uma boa estratégia para melhorar a qualidade de todos os algoritmos, uma vez que alcançamos melhorias estatisticamente significativas em todos os cenários avaliados. As melhorias obtidas pelo kNN podem ser explicadas, assim como nos resultados relacionados à *GreedyChronos*, pela melhoria da qualidade dos termos de teste em contextos temporais, referente à redução da ambigüidade deles entre as classes. Para os demais algoritmos de CAD, como os contextos temporais selecionados pela heurística *WindowChronos* são simétricos, considerando todos os termos que ocorrem em cada ano (não apenas os termos de teste), as premissas simétricas adotadas por esses algoritmos não são afetadas. Por fim, aplicamos uma estratégia de poda para reduzir as alternativas avaliadas pela heurística *WindowChronos* e, conseqüentemente seu custo computacional. Mostramos também que, apesar da nossa estratégia ser simples, ela apresenta uma boa aproximação comparada à solução que avalia todas as possibilidades.

# Capítulo 7

# Conclusões e Trabalhos Futuros

Neste capítulo apresentamos um sumário dos principais resultados alcançados nesta tese. Além disso, baseados nesses resultados, apresentamos as conclusões relacionadas aos potenciais e às limitações do uso de contextos temporais em classificação automática de documentos. Por fim, discutimos alguns trabalhos futuros.

## 7.1 Conclusões

Nesta tese mostramos evidências de que o tempo é realmente um fator importante e que deve ser considerado pelas algoritmos de classificação. Na primeira parte do trabalho demonstramos que a evolução temporal pode dificultar a construção de modelos de classificação de documentos, no sentido que os modelos baseados em toda coleção não funcionam bem, e considerar essa evolução é a chave para uma classificação efetiva. Decompomos essa evolução temporal em três diferentes efeitos que podem afetar o desempenho de classificadores automáticos. O primeiro efeito, chamado de distribuição de classes, está relacionado ao impacto da evolução temporal sobre as classes. Classes podem surgir ou desaparecer, o que pode acontecer como conseqüência da junção ou separação das classes atuais. O segundo efeito, denominado distribuição de termos, refere-se a como as relações entre termos e classes mudam ao longo do tempo, como conseqüên-

cia dos termos que surgem, desaparecem ou apresentam um poder discriminativo variável entre as classes. O terceiro efeito, a similaridade entre classes, refere-se a como duas classes podem ser similares em um determinado momento e não mais em outro. Por fim, nessa primeira etapa, propusemos e aplicamos uma metodologia para avaliar o impacto da evolução temporal sobre o desempenho dos classificadores.

Na segunda parte desta tese, propusemos uma estratégia para seleção de contexto que pode ser utilizada para a construção de modelos de classificação. Essa estratégia consiste em selecionar contextos, um conjunto de documentos pré-classificados, que minimizem os três efeitos temporais acima mencionados (distribuição de classe, distribuição de termos e similaridade entre classes), de modo que o modelo de classificação resultante do processo de aprendizado dos classificadores seja menos suscetível a variações temporais. Nesse sentido, primeiramente identificamos três requisitos que devem ser considerados ao selecionar os contextos temporais: ponto de referência, estabilidade das características e redução de incerteza. Por fim, propusemos o *Chronos*, um algoritmo que pode ser instanciado por propostas que visam selecionar contextos temporais.

Na terceira parte da tese, propusemos e avaliamos duas heurísticas computacionalmente viáveis para a seleção de contextos temporais, a *GreedyChronos* e a *WindowChronos*, ambas derivadas do algoritmo geral *Chronos*. A principal diferença entre as duas heurísticas é referente aos contextos temporais gerados, que podem ser simétricos ou não em relação aos termos que compõem os documentos de treino e teste. Enquanto a *GreedyChronos* considera apenas as características presentes nos documentos de teste (termos de teste) a fim de selecionar contextos temporais, a *WindowChronos* gera contextos simétricos, pois considera todas as características (termos de teste e não-teste). Avaliamos essas heurísticas utilizando diferentes algoritmos de CAD, como Naïve Bayes, kNN, Rocchio e SVM, usando duas coleções de documentos distintas: a biblioteca digital da ACM (que contém documentos relacionados à área de Ciência da Computação) e a MedLine (que contém documentos relacionados à Medicina). Além disso, realizamos também uma análise detalhada do comportamento de cada algoritmo de CAD utilizando

contextos temporais selecionados por essas heurísticas.

Nossos resultados mostraram que a heurística *GreedyChronos* funciona muito bem com algoritmos baseados em premissas assimétricas, por exemplo, algoritmos que utilizam somente termos de teste a fim de determinar a classe correta dos documentos de teste, mas pode não funcionar apropriadamente com algoritmos que consideram todos os termos, baseados em premissas simétricas. Realizamos uma caracterização detalhada desses resultados a fim de entender as razões desse comportamento, considerando as características intrínsecas dos algoritmos avaliados. Essa caracterização mostrou que, apesar do fato da confusão referente à ambigüidade dos termos de teste entre as classes ser reduzida nos contextos temporais selecionados pela heurística *GreedyChronos*, o mesmo não pode ser garantido para termos que não são de teste, incluídos nos contextos temporais. Conseqüentemente, as premissas desses algoritmos que consideram todos os termos podem ser afetadas: 1) Naïve Bayes utiliza uma premissa simétrica de que a representatividade dos termos de teste de uma dada classe é baseada em todos os termos que ocorrem nessa classe. 2) Rocchio baseia-se em uma premissa simétrica de que todos os termos devem ser utilizados para calcular o vetor protótipo de cada classe. 3) SVM efetivamente usa todos os termos para construir seu modelo de classificação, o que também é uma premissa simétrica.

Entretanto, utilizando os contextos temporais selecionados pela heurística *WindowChronos*, alcançamos melhorias estatisticamente significativas em todos os cenários avaliados. As melhorias obtidas pelo kNN podem ser explicadas, assim como nos resultados relacionados à *GreedyChronos*, pela melhoria da qualidade dos termos de teste em contextos temporais, referente à redução da ambigüidade deles entre as classes. Para os demais algoritmos de CAD, como os contextos temporais selecionados pela *WindowChronos* são simétricos, considerando todos os termos que ocorrem em cada ano (não apenas os termos de teste), as premissas simétricas adotadas por esses algoritmos não são afetadas.

Na próxima seção apresentamos algumas conclusões relacionadas aos potenciais e às limitações do uso de contextos temporais por cada um dos algoritmos de CAD avalia-

dos, correlacionando também com as características das coleções.

## 7.2 Contextos Temporais: Potenciais e Limitações

Analisando a caracterização dos efeitos temporais juntamente com os resultados relacionados à avaliação das duas heurísticas de seleção de contextos temporais, *GreedyChronos* e *WindowChronos*, discutimos, a seguir, em quais cenários cada um dos algoritmos de CAD possui um maior potencial para melhorar a qualidade de seus resultados utilizando os contextos temporais. Além disso, apresentamos uma análise que relaciona as características das coleções e a necessidade de se reconstruir os modelos de classificação periodicamente, a fim de evitar a degradação da qualidade da classificação, em um cenário de fluxo de documentos (*streams*).

O kNN (Yang & Liu, 1999) é um algoritmo sob-demanda que se baseia nos termos presentes no documento de teste para obter a amostra do conjunto de treinamento que será utilizada no processo de classificação. Esse algoritmo decide para qual classe um documento de teste será assinalado observando a classe dos $k$ documentos mais similares na amostragem. Dessa forma, esse algoritmo é muito dependente dos termos de testes. Conseqüentemente, em coleções de documentos em que as relações entre os termos e as classes mudam significativamente ao longo do tempo (efeito da distribuição dos termos), que é o caso das coleções que usamos (especialmente a ACM-DL), o uso dos contextos temporais pode melhorar significativamente a eficácia desses algoritmos, como podemos observar nas Tabelas 6.1 e 6.6.

O algoritmo Naïve Bayes, como mencionado anteriormente, baseia-se na representatividade dos termos nas classes. Em coleções de documentos em que o desbalanceamento entre as classes é muito alto, a versão original do Naïve Bayes (linha de base) tende a priorizar as classes maiores (com maior número de documentos), atribuindo mais freqüentemente a elas os documentos de teste. Em ambas as coleções, podemos observar que os resultados alcançados foram muito bons considerando a acurácia (tão bom quanto

o algoritmo SVM). No entanto, considerando a macro$F_1$ que mede a eficácia da classificação em cada classe individualmente, os resultados não foram tão bons, uma vez que ambas as coleções são muito desbalanceadas e, conseqüentemente, as classes maiores são priorizadas. Baseado nessas observações, podemos concluir que, em uma coleção de documentos onde a freqüência das classes varia consideravelmente ao longo do tempo (efeito da distribuição de classes), em que a classe mais freqüente muda muito ao longo do tempo (que não é o caso da ACM-DL e da MedLine, como podemos observar na Figura 3.5), o uso de contextos temporais pelo Naïve Bayes tem um grande potencial.

Rocchio é um classificador linear que emprega um modelo de espaço vetorial para representar cada classe por meio de um documento protótipo. Cada classe é representada por um grande vetor que contém informações relacionadas aos termos de todos seus documentos. Para classificar um documento de teste, é calculada a distância entre o vetor que representa esse documento e os vetores protótipo de cada uma das classes, assinalando o documento de teste à classe mais próxima em termos de distância vetorial. Portanto, a qualidade desse algoritmo é diretamente relacionada ao nível de similaridade entre as classes, onde a tarefa do classificador se torna mais difícil à medida que a similaridade entre as classes aumenta. Conseqüentemente, em coleções de documentos em que a similaridade entre as classes varia muito ao longo do tempo (efeito da similaridade entre classes), em função dos termos que ocorrem em seus documentos, os resultados desse algoritmo utilizando contextos temporais tendem a apresentar ganhos significativos.

O algoritmo Support Vector Machine (SVM) é baseado no princípio de Minimização do Risco Estrutural (*Structural Risk Minimization*) da teoria de aprendizagem computacional, em que a idéia principal é encontrar uma hipótese $h$ para a qual podemos garantir o menor erro verdadeiro através de um processo de otimização, que é também chamado de processo de aprendizagem. Uma função *Kernel* apropriada é responsável por esse processo de aprendizagem. Como mencionamos anteriormente, ao final do processo de aprendizagem, o SVM provê pesos para cada termo em cada classe diferente. Um alto peso positivo de um termo para uma determinada classe indica que documentos com

esses termos estão provavelmente relacionados com a classe em questão, enquanto que pesos negativos indicam exatamente o contrário. Os termos com pesos intermediários, que são confusos e insuficientes para identificar uma classe, normalmente apresentam relações com diferentes classes ao longo do tempo. Conseqüentemente, em coleções de documentos onde as relações entre os termos e as classes mudam significativamente ao longo do tempo (efeito da distribuição de termos), a utilização de contextos temporais pode melhorar significativamente a qualidade desse algoritmo, assim como no algoritmo kNN, uma vez que o número de termos com altos pesos positivos ou negativos tendem a aumentar. Além disso, analisando os bons resultados alcançados no Capítulo 4, em que o SVM que utiliza uma função Kernel muito robusta (*Radial Basis Function*) é avaliado em conjunto com uma heurística exaustiva de seleção de contextos temporais sensível ao tempo, muito semelhante a *WindowChronos*, podemos concluir que, quanto mais robusta é a função Kernel utilizada, melhores serão os resultados alcançados utilizando contextos temporais.

Por fim, analisamos a correlação entre o desempenho dos algoritmos de CAD, as heurísticas de seleção de contextos temporais e as características das coleções utilizadas em nossos experimentos. Na coleção ACM-DL, observamos que as relações entre os termos e classes variam ao longo do tempo de forma mais significativa do que na coleção MedLine (Figura 3.9), ou seja, a MedLine é menos afetada pelo efeito temporal. No entanto, o número de novos termos que são introduzidos ao longo dos anos é bem mais elevado na MedLine que na ACM-DL (Figura 6.10). Nosso trabalho, até o momento, não foi avaliado no cenário de fluxo de documentos (*streams*, conforme apresentado por alguns trabalhos recentes (Cohen & Singer (1999); Tsymbal (2004)), estando focado na melhoria de tarefas de re-classificação em geral, a partir de uma visão geral de uma coleção. Por exemplo, nosso trabalho pode ser uma boa alternativa para a tarefa de re-classificação de uma coleção baseada nas características e taxonomia de uma outra coleção (por exemplo, se o CiteSeer fosse adquirido pela ACM-DL). A visão da coleção que usamos é global, ou seja, apesar de considerarmos os aspectos temporais associados às mudanças na cole-

ção, não consideramos o processo de adição de novas informações (termos) ao longo do tempo, que é muito comum na coleção MedLine. Desse modo, nossos resultados apresentados foram obtidos sem tirar proveito desse processo, o que ajuda a explicar porque as melhorias na ACM-DL, de forma geral, são melhores que na MedLine.

Considerando a classificação automática para o problema de fluxos de documentos, temos que, em ambas as coleções, por motivos distintos, o modelo de classificação deve ser reconstruído periodicamente a fim de evitar a degradação da qualidade da classificação. Enquanto para a coleção ACM-DL o modelo precisa ser reconstruído principalmente como conseqüência das variações temporais das relações entre os termos e as classes, na coleção MedLine a reconstrução do modelo deve ser feita principalmente como conseqüência das novas informações que são adicionadas com o tempo. Conforme discutimos no Capítulo 2, os trabalhos relacionados à classificação de documentos adaptativa e incremental, assim como trabalhos sobre tendência de conceitos, lidam com esse problema, porém evitando a reconstrução dos modelos de classificação, aplicando mudanças (adições e remoções) no modelo inicial. Vários desses trabalhos consideram que esses ajustes precisam ser feitos por causa do surgimento de novas informações. Como mencionado nos capítulos anteriores e melhor discutido na próxima seção, nossas heurísticas podem facilmente ser adaptadas para esse problema, porém com a vantagem de que ambas as razões mencionadas anteriormente sejam consideradas.

## 7.3 Trabalhos Futuros

Durante a elaboração e execução desta tese, pudemos vislumbrar um conjunto grande de outras oportunidades de trabalhos que podem ser exploradas a partir das contribuições e resultados alcançados. Essas oportunidades não se restringem apenas ao problema de classificação de documentos, abrangendo outras linhas de pesquisa tais como comércio eletrônico e redes sociais. Nas seções seguintes descrevemos um pouco de cada uma dessas oportunidades nos diferentes cenários.

### 7.3.1 Classificação Automática de Documentos

No cenário de classificação automática de documentos, ainda há muito a ser explorado no que se refere à evolução temporal de coleções e em como lidar com essa evolução. Primeiro, nosso objetivo é aplicar a metodologia apresentada no Capítulo 3 em diferentes coleções Web visando investigar como a evolução temporal as afeta. Além disso, vamos aplicar as duas heurísticas apresentadas neste trabalho nessas coleções da Web, utilizando diferentes algoritmos CAD, a fim de mostrar o potencial de contextos temporais em vários cenários. Nesse sentido, atualmente estamos trabalhando na caracterização da evolução temporal da coleção de documentos publicados pela *Wikipedia*[1].

Além disso, pretendemos estender nossas heurísticas considerando, por exemplo, outros tipos de evidência, como *inlinks* e *outlinks*, dentro de contextos temporais. Nesse sentido, a idéia é ampliar a seleção dos contextos temporais, permitindo que as janelas temporais dos termos não sejam apenas contíguas, ampliando um pouco mais o espaço de busca por meio das citações (*inlinks* e *outlinks*), entretanto sem acarretar em um aumento muito grande do custo computacional.

Outra extensão do nosso trabalho é aplicar as técnicas apresentadas em outros cenários como classificação adaptativa de documentos e tendências de conceitos. Esses trabalhos consideram que a necessidade de reconstrução dos modelos de classificação para as coleções acontece apenas em função do surgimento de novas informações ou o desaparecimento de outras antigas. Para contornar esse problema, as propostas existentes sugerem que alterações no modelo original devem ser feitas a partir de remoções e adições das informações. Entretanto, conforme discutimos na seção anterior, a necessidade de re-treinamento pode surgir a partir de mudanças temporais que ocorrem nas coleções, como, por exemplo, migração de termos entre classes. Dessa forma, a idéia é estender esses trabalhos adaptando as técnicas apresentadas nessa tese para identificar essas mudanças, tornando assim, os modelos de classificação cientes também da evolução temporal da coleção.

[1] http://www.wikipedia.org

Por fim, baseado em todos os estudos realizados e resultados alcançados, novos algoritmos de classificação automática de documentos capazes de lidar com os efeitos temporais podem ser propostos. Atualmente estamos trabalhando na adaptação de um algoritmo para esse propósito, o Rocchio. Alteramos a implementação desse algoritmo para que um vetor protótipo de classe fosse gerado para cada momento distinto no espaço temporal (e.g., um vetor protótipo para cada classe em para cada ano). Para se classificar um documento de teste, deve-se calcular a distância vetorial entre o documento de teste e cada um desses vetores. Em seguida, para cada classe, uma soma com todas as distâncias encontradas para cada ano deve ser feita de tal forma que uma penalidade deve ser aplicada nesse soma. Quanto maior a distância temporal entre o documento de teste e o vetor da classe, maior deve ser a penalidade. Essa implementação encontra-se em fase de teste, entretanto resultados preliminares mostraram que se trata de uma técnica bastante promissora. Outros algoritmos como Naïve Bayes e SVM também podem ser alterados seguindo uma estratégia parecida com a adotada no Rocchio. Além disso, outras técnicas originais também podem ser elaboradas.

### 7.3.2 Comércio Eletrônico

Aplicações de comércio eletrônico estão relacionadas ao uso de comunicação digital aplicada aos negócios Jr. et al. (2002). O comércio eletrônico está hoje presente no dia-a-dia das empresas e dos consumidores, afetando e mudando as formas de relacionamento entre as partes envolvidas nas transações. Esse tipo de aplicação aumenta a possibilidade de alcançar um número maior de consumidores e por mais horas através dos sites na Internet. No Brasil, de 2001 a 2007 o faturamento alcançado por meio transações comerciais eletrônicas cresceu aproximadamente 1200%, saltando de R$ 0,54 bilhão em 2001 para R$ 6,40 bilhões em 2007[2].

O ambiente que caracteriza esse conjunto de aplicações é bem complexo e dinâmico, devido à presença de diferentes tipos de participante, diversidade de aplicações,

[2]http://e-commerce.org.br

fácil acesso à informação e interatividade. As principais linhas de pesquisa no cenário de comércio eletrônico estão focadas no objetivo de aperfeiçoar a eficiência dessas aplicações, a fim de torná-las mais interessantes (atrativas) e mais populares para os atores envolvidos na negociação eletrônica (compradores, vendedores, provedores de serviços e soluções tecnológicas). Uma das principais ferramentas utilizadas para se atingir esse objetivo é a modelagem da interação entre usuário-sistema e o uso dessa informação para propor modelos econômicos e estratégias de venda mais lucrativas, sistemas de recomendação de compras, sistemas de apoio a decisão aos atores envolvidos etc.

Analisando a complexidade das características que compõem os ambientes de comércio eletrônico, a tarefa de modelar a interação entre usuário-sistema se torna um desafio. Se consideramos que as ações e características observadas nesse conjunto de aplicações mudam com o tempo, ou seja, estão sujeitas à evolução temporal, o problema de modelagem das mesmas se torna ainda mais complexo. Por exemplo, os interesses dos usuários por produtos mudam ao longo do tempo, compradores e vendedores se tornam mais experientes e conseqüentemente mais exigentes em termos de eficiência e eficácia, a todo momento novos produtos, dos mais variados tipos, são comercializados utilizando essas aplicações. Esse é, portanto, um cenário propício para a criação e utilização de modelos e algoritmos temporais de mineração de dados.

### 7.3.3 Redes Sociais

Uma rede social consiste em pessoas que interagem umas com as outras por meio de comunidades *online*, compartilhando informações pela Web Falkowski et al. (2006b). À medida que aumenta o número de atividades cotidianas da sociedade conduzidas no domínio digital, em particular na Web, aumenta também a coleta de dados referentes à forma como as pessoas se comunicam e interagem umas com as outras, o que elas aprendem, como elas trabalham e se divertem. As linhas de pesquisa nesse cenário estão relacionadas à caracterização e modelagem de propriedades topológicas e funcionais de diferentes redes sociais, possibilitando a modelagem do comportamento social coletivo

dos usuários. Os resultados dessas pesquisas têm sido usados também para aumentar a eficiência, confiabilidade e segurança de sistemas de informação distribuídos em larga escala, como ocorre na Web.

As redes sociais tratam de relações complexas que envolvem interações entre um grande número de entidades heterogêneas. Essas interações, em sua grande maioria, são extraídas e registradas (logs) pelos diversos serviços relacionados, gerando um grande e difuso conjunto de dados. Dessa forma, caracterizar e modelar essas redes são tarefas desafiadoras e mais uma vez as técnicas e modelos de mineração de dados têm se mostrado promissoras nesse contexto Jeremy Kubica (2003); Wasserman et al. (1994).

Os comportamentos das entidades envolvidas em redes sociais variam no tempo e se manifestam em múltiplas escalas de tempo e múltiplos tipos de relacionamento, o que demanda o desenvolvimento de novas técnicas e modelos de mineração de dados que sejam mais robustos com relação a questão temporal. Por exemplo, diversas redes sociais emergem das interações entre pesquisadores de diferentes áreas do conhecimento. Em particular, redes podem ser usadas para representar as relações de co-autoria entre estes pesquisadores, as relações de colaboração entre departamentos acadêmicos, as interações entre grupos de pesquisadores de diferentes países em diferentes épocas no tempo. O estudo dessas redes pode levar ao entendimento da evolução de uma área de pesquisa, à determinação da importância e do impacto de certos autores ou grupos de autores e à difusão de novas teorias ou práticas experimentais.

# Referências Bibliográficas


Allan, J. (1996). Incremental relevance feedback for information filtering. In *SIGIR '96: Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 270--278, New York, NY, USA. ACM.

Allan, J.; Carbonell, J.; Doddington, G.; Yamron, J. & Yang, Y. (1998). Topic detection and tracking pilot study final report. In *In Proceedings of the DARPA Broadcast News Transcription and Understanding Workshop*, pp. 194--218, Lansdowne, VA.

Alonso, O.; Gertz, M. & Baeza-Yates, R. (2007). On the value of temporal information in information retrieval. *SIGIR Forum*, 41(2):35–41.

Arampatzis, A.; Beney, J.; Koster, C. & van der Weide, T. (2000a). Incrementality, half-life, and threshold optimization for adaptive document filtering. In *Nineth Text Retrieval Conference (TREC-9)*, pp. 589–600, Gaithersburg,MD.

Arampatzis, A. & van der Weide, T. (2001). Document filtering as an adaptive and temporally-dependent process. In *BCS-IRSG European Colloquium on IR Research*, pp. 213–224, Darmstadt,Germany.

Arampatzis, A.; van der Weide, T.; Koster, C. & van Bommel, P. (2000b). Term selection for filtering based on distribution of terms over time. In *RIAO'2000 Content-Based Multimedia Information Access*, pp. 1221–1237, Paris, France.

Baeza-Yates, R. A. & Ribeiro-Neto, B. A. (1999). *Modern Information Retrieval*. ACM Press / Addison-Wesley.

Berberich, K.; Vazirgiannis, M. & Weikum, G. (2004). T-rank: Time-aware authority ranking. *Algorithms and Models for the Web-Graph*, 3243:131–142.

Borko, H. & Bernick, M. (1963). Automatic document classification. *J. ACM*, 10(2):151–162.

Brewington, B. E. & Cybenko, G. (2000). How dynamic is the Web? *Computer Networks*, 33(1–6):257--276.

Brieman, L. & Spector, P. (1992). Submodel selection and evaluation in regression: The x-random case. *International Statistical Review*, 60:291–319.

Brinkmeier, M. (2006). Pagerank revisited. *ACM Trans. Internet Technol.*, 6(3):282--301.

Caldwell, N. H. M.; Clarkson, P. J.; Rodgers, P. A. & Huxor, A. P. (2000). Web-based knowledge management for distributed design. *IEEE Intelligent Systems*, 15(3):40–47.

Chawla, N. V. (2003). C4.5 and imbalanced data sets: investigating the effect of sampling method, probabilistic estimate, and decision tree structure. In *Proceedings of the Workshop on Learning from Imbalanced Data Sets at International Conference on Machine Learning*, pp. 72–80, Washington, DC US.

Cho, J. & Garcia-Molina, H. (2000). The evolution of the web and implications for an incremental crawler. In *Proceedings of the 26th International Conference on Very Large Data Bases*, pp. 200--209, San Francisco, CA, USA.

Cohen, W. W. & Singer, Y. (1999). Context-sensitive learning methods for text categorization. *ACM Trans. Inf. Syst.*, 17(2):141–173.

Dörre, J.; Gerstl, P. & Seiffert, R. (1999). Text mining: finding nuggets in mountains of textual data. In *KDD '99: Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 398--401, New York, NY, USA. ACM.

Dumais, S.; Platt, J.; Heckerman, D. & Sahami, M. (1998). Inductive learning algorithms and representations for text categorization. In *Proceedings of the 7th International Conference on Information and knowledge Management*, pp. 148–155, Bethesda, Maryland, United States. ACM.

Fahmi, I. (2004). Examining learning algorithms for text classification in digital libraries. In *Master's thesis*, University of Groninge, Netherland.

Faith, D. P. (1983). Asymmetric binary similarity measures. *Oecologia (Berlin)*, 57:287--290.

Falkowski, T.; Bartelheimer, J. & Spiliopoulou, M. (2006a). Mining and visualizing the evolution of subgroups in social networks. In *Proceedings of the IEEE/WIC/ACM International Conference on Web Intelligence*, pp. 52--58, Washington, DC, USA. IEEE Computer Society.

Falkowski, T.; Bartelheimer, J. & Spiliopoulou, M. (2006b). Mining and visualizing the evolution of subgroups in social networks. In *Proceedings of the 2006 IEEE/WIC/ACM International Conference on Web Intelligence*, pp. 52--58, Washington, DC, USA.

Fetterly, D.; Manasse, M.; Najork, M. & Wiener, J. L. (2004). A large-scale study of the evolution of web pages. *Softw. Pract. Exper.*, 34(2):213--237.

Folino, G.; Pizzuti, C. & Spezzano, G. (2007). An adaptive distributed ensemble approach to mine concept-drifting data streams. In *Proceedings of the 19th IEEE International Conference on Tools with Artificial Intelligence - Vol.2 (ICTAI 2007)*, pp. 183--188, Washington, DC, USA. IEEE Computer Society.

Forman, G. (2003). An extensive empirical study of feature selection metrics for text classification. *Journal of Machine Learning Research*, 3:1289–1305.

Forman, G. (2006). Tackling concept drift by temporal inductive transfer. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 252–259, Seattle, Washington, USA. ACM.

Friedman, J. H.; Kohavi, R. & Yun, Y. (1996). Lazy decision trees. In Shrobe, H. & Senator, T., editores, *Proceedings of 13th National Conference on Artificial Intelligence -AAAI*, pp. 717–724, Portland, Oregon, USA. AAAI Press.

Groeneveld, R. (1991). An influence function approach to describing the skewness of a distribution. *The American Statistician*, 45:97–102.

Hanani, U.; Shapira, B. & Shoval, P. (2001). Information filtering: Overview of issues, research and systems. *User Modeling and User-Adapted Interaction*, 11(3):203--259.

Jeremy Kubica, Andrew Moore, J. S. (2003). Tractable group detection on large link data sets. In *Proceedings of Third IEEE International Conference on Data Mining*, pp. 573–576. IEEE Computer Society.

Joachims, T. (1998a). Making large-scale support vector machine learning practical. In B. Schölkopf, C. Burges, A. S., editor, *Advances in Kernel Methods: Support Vector Machines*, pp. 169–184. MIT Press, Cambridge, MA.

Joachims, T. (1998b). Text categorization with support vector machines: Learning with many relevant features. In *10th European Conference on Machine Learning*, pp. 137--142, Chemnitz, Germany. Springer Verlag.

Joachims, T. (2006). Training linear svms in linear time. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 217–226, Philadelphia, PA, USA. ACM.

Jones, R. & Diaz, F. (2007). Temporal profiles of queries. *ACM Trans. Inf. Syst.*, 25(3):14.

Jr., W. M.; Murta, C. D.; Aguiar, S. V. & Neto, D. O. G. (2002). *Sistemas de Comércio Eletrônico, Projeto e Desenvolvimento*. Campus, Rio de Janeiro, Brasil.

Kim, Y. S.; Park, S. S.; Deards, E. & Kang, B. H. (2004). Adaptive web document classification with mcrdr. In *Proceedings of the International Conference on Information Technology: Coding and Computing Volume 2*, p. 476, Washington, DC, USA. IEEE Computer Society.

Kiritchenko, S.; Matwin, S. & Abu-Hakima, S. (2004). Email classification with temporal features. In Klopotek, M. A.; Wierzchon, S. T. & Trojanowski, K., editores, *Intelligent Information Systems*, Advances in Soft Computing, pp. 523–533, Zakopane, Poland. Springer.

Klinkenberg, R. (2004). Learning drifting concepts: Example selection vs. example weighting. *Intell. Data Anal.*, 8(3):281--300.

Klinkenberg, R. & Joachims, T. (2000). Detecting concept drift with support vector machines. In *Proceedings of 17th International Conference on Machine Learning*, pp. 487--494, Stanford, US.

Klinkenberg, R. & Renz, I. (1998). Adaptive information filtering: Learning in the presence of concept drifts. In *Workshop on Learning for Text Categorization at the 15th Conference of the American Association for Artificial Intelligence*, pp. 33–40, Madison, Wisconsin. AAAI Press.

Koll, M. B. (1981). Information retrieval theory and design based on a model of the user's concept relations. In *Proceedings of the 3rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 77–93, Kent, UK, UK. Butterworth & Co.

Kolter, J. & Maloof, M. (2003). Dynamic weighted majority: A new ensemble method for tracking concept drift. Technical Report CSTR-20030610-3, Department of Computer Science, Georgetown University, Washington, DC.

Lam, W. & Ho, C. Y. (1998). Using a generalized instance set for automatic text categorization. In *Proceedings of the 21st Annual International ACM SIGIR Conference on*

*Research and Development in Information Retrieval*, pp. 81–89, New York, NY, USA. ACM.

Lanquillon, C. & Renz, I. (1999). Adaptive information filtering: detecting changes in text streams. In *Proceedings of the Eighth International Conference on Information and knowledge Management*, pp. 538--544, New York, NY, USA. ACM.

Lauw, H. W.; Lim, E.-P.; Pang, H. & Tan, T.-T. (2005). Social network discovery by mining spatio-temporal events. *Comput. Math. Organ. Theory*, 11(2):97--118.

Lawrence, S. & Giles, C. L. (1998). Context and page analysis for improved web search. *IEEE Internet Computing*, 2(4).

Lazarescu, M. M.; Venkatesh, S. & Bui, H. H. (2004). Using multiple windows to track concept drift. *Intell. Data Anal.*, 8(1):29--59.

Lewis, D. D. (1995). Evaluating and optimizing autonomous text classification systems. In *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 246--254, Seattle, Washington, USA.

Liebscher, R. A. (2004). Temporal context: applications and implications for computational linguistics. In *Proceedings of the ACL 2004 Workshop on Student Research*, p. 43, Morristown, NJ, USA. Association for Computational Linguistics.

Liu, R. & Lu, Y. (2002). Incremental context mining for adaptive document classification. In *Proceedings of the 8th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 599–604. ACM Press.

Manning, C. D.; Raghavan, P. & Schütze, H. (2008). *Introduction to Information Retrieval*, chapter 13, pp. 234--265. Cambridge University Press.

McCallum, A. & Nigam, K. (1998). A comparison of event models for naive bayes text classification. In *Workshop on Learning for Text Categorization at the 15th Conference*

*of the American Association for Artificial Intelligence*, pp. 41--48, Madison, Wisconsin. AAAI Press.

McCallum, A. K. (1996). Bow: A toolkit for statistical language modeling, text retrieval, classification and clustering. http://www.cs.cmu.edu/ mccallum/bow.

Mladenic, D. & Grobelnik, M. (1998). Feature selection for classification based on text hierarchy. In *Proceedings of the Workshop on Learning from Text and the Web*, pp. 78–84, Pittsburgh, PA.

Molina, L.C.and Belanche, L. & Nebot, A. (2002). Feature selection algorithms: a survey and experimental evaluation. In *Proceedings of the IEEE International Conference on Data Mining*, pp. 306–313.

Nunes, S. (2007). Exploring temporal evidence in web information retrieval. In *Proceedings of the BCS IRSG Symposium: Future Directions in Information Access*, pp. 44–50, Glasgow, Scotland, UK. British Computer Society.

Olken, F. & Rotem, D. (1995). Random sampling from databases - a survey. *Journal of Statistics & Computing*, 5:25–42.

Riordan, C. O. & Sorensen, H. (2002). Information filtering and retrieval: An overview. Technical report, Department of Information Technology, National University of Ireland, Gakway, Ireland.

Rocha, L.; Mourao, F.; Araújo, R.; Couto, T.; Gonçalves, M. & Meira, W. (2008a). Understanding temporal aspects in document classification. In *Proceedings of ACM International Conference on Web Search and Data Mining*, Palo Alto, CA, USA. ACM Press.

Rocha, L.; Mourao, F.; Pereira, A.; Gonçalves, M. & Meira, W. (2008b). Exploiting temporal contexts in text classification. In *Proceedings of ACM International Conference*

*on Information and Knowledge Management*, pp. 159--170, Napa Valley, CA, USA. ACM Press.

Rocha, L.; Mourão, F.; Miranda, L.; Pereira, A.; Meira, W. & Gonçalves, M. (2009). Temporal Context: Effective Text Classification in Evolving Document Collections. Manuscript.

Rogati, M. & Yang, Y. (2002). High-performing feature selection for text classification. In *Proc. of the 11th CIKM Conference*, pp. 659–661. ACM Press.

Salton, G. (1971). *The SMART Retrieval System—Experiments in Automatic Document Processing*, chapter 14, pp. 313--323. Prentice-Hall, Inc., Upper Saddle River, NJ, USA.

Salton, G. & Buckley, C. (1987). Term weighting approaches in automatic text retrieval. Technical report, Cornell University, Ithaca, NY, USA.

Salton, G. & McGill, M. J. (1983). *Introduction to Modern Information Retrieval*. McGraw-Hill, New York, NY, USA.

Schlimmer, J. C. & Granger, R. H. (1986). Beyond incremental processing: Tracking concept drift. In *Proceedings of the 5th National Conference on Artificial Intelligence*, pp. 502–507, Philadelphia, Pennsylvania. AAAI Press.

Scholz, M. & Klinkenberg, R. (2007). Boosting classifiers for drifting concepts. *Intell. Data Anal.*, 11(1):3–28.

Sebastiani, F. (2002). Machine learning in automated text categorization. *ACM Computing Surveys*, 34(1):1–47.

Tan, A. (1999). Text mining: The state of the art and the challenges. In *Proceedings of Third Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 65--70, Beijing, China.

Terveen, L.; Hill, W. & Amento, B. (1999). Constructing, organizing, and visualizing collections of topically related web resources. *ACM Trans. Comput.-Hum. Interact.*, 6(1):67–94.

Tseng, V. S.; Chang, J.-C. & Lin, K. W. (2006). Mining and prediction of temporal navigation patterns for personalized services in e-commerce. In *Proceedings of the 2006 ACM Symposium on Applied Computing*, pp. 867--871, New York, NY, USA. ACM.

Tsymbal, A. (2004). The problem of concept drift: Definitions and related work. Technical Report TCD-CS-2004-15, Department of Computer Science, Trinity Colleg, Dublin, Ireland.

Vapnik, V. (1998). *Statistical Learning Theory*. Wiley-Interscience, New York, NY, USA.

Veloso, A.; Jr., W. M.; Cristo, M.; Gonçalves, M. & Zaki, M. J. (2006). Multievidence, multicriteria, lazy associative document classification. In *Proceedings of the 15th ACM international Conference on Information and Knowledge Management*, pp. 218 – 227, Arlington, Virginia, USA. ACM.

Wang, H.; Fan, W.; Yu, P. S. & Han, J. (2003). Mining concept-drifting data streams using ensemble classifiers. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 226–235, Washington, D.C. USA.

Wasserman, S.; Faust, K. & Iacobucci, D. (1994). *Social Network Analysis : Methods and Applications (Structural Analysis in the Social Sciences)*. Cambridge University Press, Cambridge, UK.

Widmer, G. & Kubat, M. (1996). Learning in the presence of concept drift and hidden contexts. *Machine Learning*, 23(1):69–101.

Yang, Y. (1994). Expert network: effective and efficient learning from human decisions in text categorization and retrieval. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 13–22, Dublin, Ireland. ACM.

Yang, Y. & Kisiel, B. (2003). Margin-based local regression for adaptive filtering. In *Proceedings of the 12th International Conference on Information and Knowledge Management*, pp. 191–198, New Orleans, USA. ACM.

Yang, Y. & Liu, X. (1999). A re-examination of text categorization methods. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 42–49, Berkeley, California, United States. ACM.

Yu, P. S.; Li, X. & Liu, B. (2004). On the temporal dimension of search. In *Proc. of the 13th WWW Conference on Alternate track papers & posters*, pp. 448–449, New York, USA. ACM.

Zaiane, O. R. & Antonie, M. L. (2002). Classifying text documents by associating terms with text categories. In *Proceedings of the 13th Australian Database Conference*, pp. 215–222, Darlinghurst, Australia.