

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL  
INSTITUTO DE INFORMÁTICA  
PROGRAMA DE PÓS-GRADUAÇÃO EM COMPUTAÇÃO

**Um Assistente de Feedback para o  
Serviço de Filtragem do Software Direto**

por

LUIS CESAR DE MELLO

Dissertação de mestrado submetida à avaliação  
como requisito parcial para a obtenção do grau de  
Mestre em Ciência da Computação

Prof. Dr. Cláudio Fernando Resin Geyer  
Orientador

Porto Alegre, outubro de 2002

## CIP – CATALOGAÇÃO NA PUBLICAÇÃO

Mello, Luis Cesar de

Um Assistente de Feedback para o Serviço de Filtragem do Software Direto/ por Luis Cesar de Mello.-Porto Alegre: PPGC da UFRGS, 2002.

114f.:il.

Dissertação (mestrado) – Universidade Federal do Rio Grande do Sul. Programa de Pós-Graduação em Computação, Porto Alegre, BR – RS, 2002. Orientador: Geyer, Cláudio F. R.

1.Assistente de *Feedback* 2.Classificador. 3.Perfil de Usuário 4.Naive Bayes. I.Geyer, Cláudio Fernando Resin. II.Título.

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

Reitora Prof<sup>a</sup>. Wrana Maria Panizzi

Pró-Reitor de Ensino: Prof. José Carlos Ferraz Hennemann

Pró-Reitor Adjunto de Pós-Graduação: Prof. Jaime Evaldo Fensterseifer

Diretor do Instituto de Informática: Prof. Philippe Olivier Alexandre Navaux

Coordenador do PPGC: Prof. Carlos Alberto Heuser

Bibliotecária-Chefe do Instituto de Informática: Beatriz Regina Bastos Haro

## Agradecimentos

*Gostaria de agradecer:*

*primeiramente a DEUS pelas oportunidades e pelas dificuldades, estas motivo de crescimento;*

*ao meu amigo e orientador Claudio Fernando Resin Geyer pelos valiosos conselhos, recomendações e orientações fornecidas;*

*aos professores do Instituto de Informática pelos ensinamentos concedidos;*

*aos doutorandos Clairmont Borges e Adenauer Correa Yamin pelos valiosos conselhos e atenção a mim concedidos;*

*aos professores Rosa Viccari e Paulo Engel pelos valiosos conselhos concedidos na semana acadêmica;*

*ao professor Tiarajú pela hospitalidade e atenção desde o primeiro dia que cheguei em Porto Alegre;*

*ao mestrando Elias Cesar de Araújo de Carvalho pela amizade, pelas leituras e opiniões;*

*a todos os funcionários da UFRGS, em especial os da secretaria do PPGC e da biblioteca pela atenção dispensada;*

*a todos os alunos de mestrado da UFRGS;*

*o Cesumar, UEM e a Univale pelo apoio destinado a essa pesquisa;*

*aos meu pai e minha mãe por me ensinar o valor do saber e apoiar em todos os momentos difíceis de minha vida e a minha família pelo apoio incondicional em todos os momentos;*

*a minha amada esposa por tanto apoio e dedicação dispensados e as minhas filhas Malu, Marina e Mariana por compreenderem a minha ausência durante todo esse tempo.*

## Sumário

<b>Lista de Abreviaturas.....</b>	<b>7</b>
<b>Lista de Símbolos .....</b>	<b>8</b>
<b>Lista de Figuras.....</b>	<b>9</b>
<b>Lista de Tabelas .....</b>	<b>10</b>
<b>Resumo .....</b>	<b>12</b>
<b>Abstract.....</b>	<b>13</b>
<b>1 Introdução .....</b>	<b>14</b>
<b>1.1 Tema .....</b>	<b>14</b>
<b>1.2 Motivação .....</b>	<b>14</b>
<b>1.3 Proposta.....</b>	<b>15</b>
<b>1.4 Organização do Texto.....</b>	<b>16</b>
<b>2 Classificadores Probabilísticos .....</b>	<b>17</b>
<b>2.1 Fundamentos Básicos de Probabilidades.....</b>	<b>17</b>
2.1.1 Modelos Matemáticos.....	17
2.1.2 Conjuntos .....	17
2.1.3 Frequência Relativa.....	18
2.1.4 Noções Fundamentais de Probabilidade .....	18
2.1.5 Probabilidade Condicionada .....	19
2.1.6 Teorema de Bayes.....	21
<b>2.2 O Raciocínio Probabilístico .....</b>	<b>22</b>
2.2.1 Raciocínio Incerto.....	23
2.2.2 Relacionamento Condicional.....	24
<b>2.3 Redes Bayesianas .....</b>	<b>24</b>
2.3.1 Histórico.....	24
2.3.2 Definição de Redes Bayesianas .....	25
2.3.3 Aprendizado em Redes Bayesianas .....	26
<b>2.4 Classificadores .....</b>	<b>27</b>
2.4.1 Aprendizagem de Máquina .....	28
2.4.2 Definição de Classe.....	30
2.4.3 Os Classificadores.....	30
<b>2.5 O Classificador Naive Bayes .....</b>	<b>31</b>
2.5.1 O Algoritmo Naive Bayes.....	34
2.5.2 Comentários Finais .....	40
<b>2.6 Desempenho dos Classificadores .....</b>	<b>40</b>
2.6.1 Resultados de Desempenho do Classificador Naive Bayes.....	42
<b>2.7 Modelo de Usuário .....</b>	<b>47</b>
2.7.1 Conceitos e Características .....	47
2.7.2 Importância do Modelo .....	48
2.7.3 Aprendizado do Modelo de Usuário .....	48
<b>2.8 Ferramentas de Modelagem de Usuário para Web.....</b>	<b>50</b>
<b>2.9 O Direto .....</b>	<b>53</b>
2.9.1 Histórico.....	54
2.9.2 O Projeto Direto.....	54
2.9.3 A Tecnologia usada no Direto .....	56

<b>2.10 Considerações Finais .....</b>	<b>59</b>
<b>3 Modelo Geral do Assistente de <i>Feedback</i> .....</b>	<b>60</b>
<b>3.1 Estrutura Geral do Modelo .....</b>	<b>60</b>
<b>3.2 Etapa de Coleta de Termos e Eventos .....</b>	<b>63</b>
3.2.1 Dados de entrada.....	64
3.2.2 Dados de saída .....	66
<b>3.3 Etapa de Classificação .....</b>	<b>67</b>
3.3.1 As classes .....	68
3.3.2 O Algoritmo Naive Bayes.....	69
3.3.3 Os Parâmetros das probabilidades condicionais .....	69
3.3.4 A classificação .....	72
<b>3.4 Etapa de <i>Feedback</i> .....</b>	<b>74</b>
3.4.1 Entrada de Dados .....	75
3.4.2 Processamento.....	75
3.4.3 Interface Gráfica de Sugestões.....	76
3.4.4 O <i>Feedback</i> .....	77
<b>3.5 Considerações Finais .....</b>	<b>78</b>
<b>4 Protótipo de Assistente de <i>Feedback</i> Implementado .....</b>	<b>79</b>
<b>4.1 Considerações Iniciais .....</b>	<b>79</b>
<b>4.2 Ambiente e Ferramentas Utilizadas .....</b>	<b>80</b>
<b>4.3 Estrutura da Ferramenta AF .....</b>	<b>81</b>
<b>4.4 Acoplando o AF no Ambiente de Correio Eletrônico do Direto ....</b>	<b>85</b>
<b>4.5 Interface Gráfica de Sugestões do AF.....</b>	<b>87</b>
<b>4.6 Considerações Finais .....</b>	<b>88</b>
<b>5 Avaliação do Protótipo do Assistente de <i>Feedback</i> .....</b>	<b>89</b>
<b>5.1 Considerações Iniciais .....</b>	<b>89</b>
<b>5.2 Avaliação da Precisão do AF .....</b>	<b>89</b>
5.2.1 Ambiente Utilizado nos Testes .....	90
5.2.2 Coleta e Preparação dos Dados.....	90
5.2.3 Escolha dos Casos para Treinamento .....	91
5.2.4 Geração dos Parâmetros.....	92
5.2.5 Simulação do AF no Ambiente de Correio Eletrônico .....	93
5.2.6 Análise dos Resultados .....	94
<b>5.3 Avaliação de Desempenho do AF .....</b>	<b>95</b>
5.3.1 Ambiente Utilizado nos Testes .....	95
5.3.2 Dados Considerados nos Testes.....	96
5.3.3 Testes e Resultados dos Experimentos .....	96
5.3.4 Análise dos Resultados .....	96
<b>5.4 Comparação com Trabalhos Similares.....</b>	<b>97</b>
<b>5.5 Considerações Finais .....</b>	<b>98</b>
<b>6 Conclusão .....</b>	<b>99</b>
<b>6.1 Considerações Finais .....</b>	<b>99</b>
<b>6.2 Trabalhos Futuros.....</b>	<b>100</b>
<b>Anexo 1 Exemplo de Mensagens do Questionário.....</b>	<b>101</b>
<b>Anexo 2 Arquivo <i>ascii</i> com as Respostas Tabuladas .....</b>	<b>102</b>

<b>Anexo 3</b>	<b>Código Fonte do Método <i>geraParametro</i> .....</b>	<b>103</b>
<b>Anexo 4</b>	<b>Código Fonte da Classe <i>StopList</i>.....</b>	<b>104</b>
<b>Anexo 5</b>	<b>Código Fonte do Método <i>classifica</i> .....</b>	<b>107</b>
<b>Anexo 6</b>	<b>Arquivo Historico_classificado.....</b>	<b>108</b>
<b>Bibliografia</b> .....		<b>109</b>

## Lista de Abreviaturas

AF	Assistente de <i>Feedback</i>
API	Application Program Interface
CSS	Cascading Style Sheets
DAG	Gráfico Acíclico Direcionado
fig.	Figura
GNU	GNU's Not Unix
GUI	Graphics User Interface
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
IA	Inteligência Artificial
IBM	International Business Machine
IMAP	Internet Message Access Protocol
JDBC	Java Database Connectivity
JNDI	Java Naming and Directory Interface
LDAP	Lightweight Directory Access Protocol
MBR	Memory-Based Reasoning
MIME	Multipurpose Internet Mail Extensions
MS	Microsoft
PINE	Pine's not Elm
POP	Post Office Protocol
PROCERGS	Companhia de Processamento de Dados do Estado do Rio Grande do Sul
RB	Redes Bayesianas
RFC	Request for Comments
RTF	Rich Text Format
SBC	Simple Bayesian Classifier
SBN	Seletivo Naive Bayes
SGML	Standard Generalized Markup Language
SMTP	Simple Mail Transfer Protocol
SSL	Secure Sockets Layer
tab.	Tabela
UFRGS	Universidade Federal do Rio Grande do Sul
W3C	World Wide Web Consortium
WEKA	Waikato Environment for Knowledge Analysis
WWW	World Wide Web
XML	Extensible Markup Language
XSL	Extensible Stylesheet Language

## Lista de Símbolos

+	Soma
-	Subtração
*	Multiplicação
/	Divisão
=	Igualdade
$\leq$	Menor e igual a
$\geq$	Maior e igual a
$<$	Menor que
$>$	Maior que
%	Percentual
$\Omega$	Conjunto Fundamental
$\Sigma$	Somatório
$\cup$	União
$\cap$	Interseção
$N_A$	Frequência Absoluta
=	Igual a
$P(A B)$	Probabilidade do evento A, quando B tiver ocorrido primeiro
U	Conjunto Fundamental
E	Experimento Probabilístico
$\Re$	Números Reais
S	Espaço Amostral



## Lista de Figuras

FIGURA 2.1 - Grafo da Rede Bayesiana que representa a distribuição $P(X_1, X_2, X_3, X_4, X_5, X_6)$ .....	25
FIGURA 2.2 - Gráfico do tamanho das amostras e o erro associado .....	27
FIGURA 2.3 - A estrutura da rede Naive Bayes em estrela .....	32
FIGURA 2.4 - Representação da dependência condicional $P(A_1 C_1)$ no grafo .....	33
FIGURA 2.5 - Rede da distribuição de probabilidade conjunta da classe JOGA .....	39
FIGURA 2.6 - Rede da distribuição de probabilidade conjunta da classe NÃO JOGA..	40
FIGURA 2.7 - Comparativo do percentual de erro de classificação do classificador Naive Bayes e a rede Bayesiana.....	43
FIGURA 2.8 - Média de precisão de cada classificador predizendo as preferências de usuários de páginas da Web .....	46
FIGURA 2.9 - Usos do Modelo de Usuário .....	48
FIGURA 2.10 - Arquitetura do Direto.....	55
FIGURA 2.11 - Interface do módulo de Correio do Direto.....	56
FIGURA 3.1 - Estrutura geral do modelo do AF .....	61
FIGURA 3.2 - Seqüência de passos do AF no ambiente de e-mail do Direto.....	63
FIGURA 3.3 - Etapa de coleta dos termos das mensagens e os eventos do usuário .....	64
FIGURA 3.4 - Exemplo de uma mensagem filtrada.....	65
FIGURA 3.5 - Exemplo de uma mensagem de outras origens com termos extraídos ...	65
FIGURA 3.6 - Representação da etapa de classificação.....	68
FIGURA 3.7 - Gráfico da rede Naive Bayes com os eventos e a classe A.....	70
FIGURA 3.8 - Fórmula do classificador Naive Bayes do AF .....	73
FIGURA 3.9 - Representação da etapa de <i>Feedback</i> .....	75
FIGURA 3.10 - Criação do vetor com as médias finais dos pesos dos termos .....	76
FIGURA 3.11- Lista de termos interessantes e a lista de termos não interessantes .....	77
FIGURA 4.1- Arquitetura do Assistente de <i>Feedback</i> .....	81
FIGURA 4.2 - Formulário do correio eletrônico do Direto.....	85
FIGURA 4.3 - Formulário da pasta caixa de entrada contendo a mensagem recebida ..	86
FIGURA 4.4 - Estrutura do AF acoplado ao Direto .....	87
FIGURA 4.5 - Tela de sugestões do Assistente de <i>Feedback</i> .....	88

## Lista de Tabelas

TABELA 2.1 - Classificação dos algoritmos de aprendizagem (classificadores).....	31
TABELA 2.2 - Conjunto de classe do sistema Jogo de Tênis .....	34
TABELA 2.3 - Conjunto de atributos que caracterizam as condições atmosféricas dos casos .....	34
TABELA 2.4 - Novo caso para sofrer predição pelo Classificador Naive Bayes .....	34
TABELA 2.5 - Conjunto de exemplo de casos selecionados por um especialista .....	36
TABELA 2.6 - Valores das probabilidades a priori $P(c_i)$ e probabilidades condicionais dos atributos $A_j$ dado as classes $c_i$ .....	37
TABELA 2.7 - Cálculo do numerador da equação (2) do classificador Naive Bayes....	39
TABELA 2.8 - Média de precisão de cada classificador aplicado nos conjuntos de dados .....	44
TABELA 2.9 - Média de precisão de cada classificador aplicado nos conjuntos de dados .....	45
TABELA 2.10 - Quadro comparativo entre as ferramentas de modelos de usuários.....	53
TABELA 3.1 - Eventos que o usuário pode executar para uma mensagem.....	64
TABELA 3.2 - Representação dos estados dos eventos executados pelo usuário.....	67
TABELA 3.3 - Classes com suas descrições, intervalos percentuais de preferência e seus pesos.....	69
TABELA 3.4 - Probabilidades condicionais para a ocorrência dos eventos dado as classes.....	71
TABELA 3.5 - Probabilidades condicionais para a NÃO ocorrência dos eventos dadas as classes .....	72
TABELA 3.6 - Exemplo do histórico de termos e eventos .....	72
TABELA 3.7 - Valores inferidos pelo classificador para a primeira linha do histórico	73
TABELA 3.8 - Histórico de termos e eventos com os pesos incluídos pela segunda etapa .....	73
TABELA 3.9 - Formas de interpretações do <i>Feedback</i> e as possíveis decisões .....	77
TABELA 4.1 - Descrição dos arquivos utilizados na implementação .....	82
TABELA 4.2 - Descrição das classes agrupadas pelas etapas do modelo AF .....	83
TABELA 4.3 - Tabela de métodos agrupados pelas classes implementadas .....	84
TABELA 5.1- Características de diferentes listas de mensagens utilizadas na experimentação .....	90
TABELA 5.2 - Conjunto de casos utilizados para o treinamento do AF .....	92
TABELA 5.3 - Parâmetros numéricos gerados pelo algoritmo de aprendizagem.....	93
TABELA 5.4 - Resultados percentuais da precisão do AF .....	94

TABELA 5.5 - Tempos médios para execução do AF em milésimos de segundos ..... 96

## Resumo

Este trabalho descreve a especificação e implementação do protótipo Assistente de *Feedback* que ajuda os usuários a ajustarem os parâmetros do serviço de filtragem de mensagens vindas do correio eletrônico de sistemas como o Direto. O Assistente de *Feedback* é instalado no computador do usuário do Direto para monitorar suas preferências representadas pelas ações aplicadas nas mensagens do correio eletrônico.

O trabalho apresenta, ainda, uma revisão bibliográfica sobre os conceitos gerais de probabilidades, redes Bayesianas e classificadores. Procura-se descrever as características gerais dos classificadores, em especial o Naive Bayes, sua lógica e seu desempenho comparado a outros classificadores. São abordados, também, conceitos relacionados ao modelo de perfil de usuário e o ambiente Direto.

O Naive Bayes torna-se atraente para ser utilizado no Assistente de *Feedback* por apresentar bom desempenho sobre os demais classificadores e por ser eficiente na predição, quando os atributos são independentes entre si.

O Assistente de *Feedback* utiliza um classificador Naive Bayes para prever as preferências por intermédio das ações do usuário. Utiliza, também, pesos que representarão a satisfação do usuário para os termos extraídos do corpo da mensagem. Esses pesos são associados às ações do usuário para estimar os termos mais interessantes e menos interessantes, pelo valor de suas médias finais. Quando o usuário desejar alterar os filtros de mensagens do Direto, ele solicita ao Assistente de *Feedback* sugestões para possíveis exclusões dos termos menos interessantes e as possíveis inclusões dos termos mais interessantes.

O protótipo é testado utilizando dois métodos de avaliação para medir o grau de precisão e o desempenho do Assistente de *Feedback*. Os resultados obtidos na avaliação de precisão apresentam valores satisfatórios, considerando o uso de cinco classes pelo classificador do Assistente de *Feedback*. Os resultados dos testes de desempenho permitem observar que, se forem utilizadas máquinas com configurações mais atualizadas, os usuários conseguirão receber sugestões com tempo de respostas mais toleráveis.

**Palavras-Chave:** Assistente de *Feedback*, Classificador, Perfil de Usuário, Naive Bayes.

**TITLE: “A FEEDBACK ASSISTANT FOR A FILTERING SERVICE OF THE DIRETO SOFTWARE”**

## **Abstract**

This work describes the specification and implementation of the prototype Feedback Assistant that helps users to update the parameters of the filtering services that comes from the electronic mail of systems as the Direto. The Feedback Assistant is downloaded into the Direto user's computer in order to monitor the user's preferences represented by the actions executed in the e-mail messages.

The work also presents a bibliographical revision on the general concepts of probabilities, Bayesian networks and classifiers. It tries to describe the general characteristics of the classifiers and especially Naive Bayes, its logic and its compared performance the other classifiers. It is also approached concepts related to the model of user's profile and the Direto environment.

Naive Bayes becomes attractive to be used in the Feedback Assistant by presenting good performance over the other classifiers and for being efficient in the prediction, when the attributes are independent amongst themselves.

The Feedback Assistant uses a Naive Bayes classifier to predict the preferences given the user's actions. Also uses of weights to represent the user's satisfaction about the terms extracted from the message. These weights are associated with the user's actions in order to estimate the probable useless or useful terms. Whenever the user decides to update message filters of Direto, the Feedback Assistant suggests the exclusion of the most probable useless terms and the inclusion of the most probable useful terms.

The prototype is tested using two evaluation methods to measure the degree of precision and the performance of the Feedback Assistant. The results obtained in the evaluation of precision presented satisfactory values, considering the use of five classes for the classifier of the Feedback Assistant. The results of the performance tests allow to observe that if machines to be used with more updated configurations, the users will get to receive suggestions with time of answers more tolerable.

**Keywords:** Feedback Assistant, Classifier, User Profile, Naive Bayes

# 1 Introdução

## 1.1 Tema

Este trabalho visa estudar algoritmos de classificação Naive Bayes e desenvolver uma proposta de um modelo utilizando esses algoritmos para inferir/classificar o comportamento do usuário com as mensagens vindas de uma ferramenta de correio eletrônico, tal como o Direto [DIR 2001]. Este modelo origina um protótipo para interpretar as preferências dos usuários, dado o seu comportamento enquanto consome informações via correio eletrônico, construindo uma representação estatística das preferências inferidas – servindo de base para retorno de sugestões a serem realizadas pelo assistente.

## 1.2 Motivação

Nos últimos anos, devido ao grande crescimento das redes de computadores, a Internet passou a ser um importante veículo de comunicação em massa e armazenamento de informações com o objetivo de disponibilizá-las para todos os usuários da Web, seja usando correio eletrônico, acessando servidores de notícias (*newsgroups*) ou servidores WWW.

Os usuários têm vários caminhos para encontrar essas informações, por exemplo: navegando com browsers (escolhendo os links que são interessantes para eles); usando serviços de consultas em máquinas de procura como o Altavista; consultando hierarquias de tópicos, como o Yahoo ou Lycos; e também pelo uso de serviços de filtragem de conteúdo [KIM 2000]. Para que os serviços de filtragem funcionem, o usuário precisa preencher os parâmetros dos filtros com seu perfil de preferência, assim as informações poderão ser filtradas durante a utilização desses serviços. Os usuários que lidam com esses serviços deparam freqüentemente com o problema de fazer ajustes contínuos nos parâmetros dos serviços, na medida em que as preferências dos usuários vão se modificando com o passar do tempo. Este problema é de grande importância, dado o grande volume de textos disponíveis através da WWW, suprimento de notícias, e-mails eletrônicos, banco de dados corporativos e bibliotecas digitais; pelo tempo e esforço gastos pelos usuários, mas sem encontrar informações de interesse; e a dificuldade que o usuário tem para encontrar as informações pretendidas, porque muitas vezes o usuário não consegue expressar precisamente o que ele quer e os serviços, como máquinas de procura e serviços de filtragem, não conseguem se adaptar com a estratégia de trabalho de cada usuário, como descrito em [CHE 97].

Os algoritmos estatísticos para classificação de textos, treinados para classificar documentos, dado um suficiente conjunto de exemplos treinados por algum mecanismo de aprendizagem, têm sido usados para automatizar catálogos de novos artigos, páginas Web, aprendizagem do perfil de interesse dos usuários e classificar automaticamente correio eletrônico [NIG 98]. As técnicas de classificação são cada vez mais utilizadas para organizar informações a fim de ajudar os usuários a selecionar, mais facilmente, informações de seu interesse. Mas para resolver problemas como os citados acima, as classificações devem ser feitas observando algumas características que identificam as informações importantes para o usuário. Essas características deveriam ser selecionadas, observando o comportamento do usuário durante a interação com as fontes de

informação ou serviços facilitadores. Essas características incluem, por exemplo, conjuntos de palavras-chave que vão representar o perfil de preferências dos usuários com relação a um ou mais temas (tal como esporte, economia, etc.) de qualquer tipo de informação oriunda da Web.

A classificação de dados é um importante tópico de pesquisa para as áreas de aprendizagem de máquina e mineração de dados. Nas últimas duas décadas, muitos algoritmos têm sido desenvolvidos para aprendizagem de árvores de decisão e classificadores de redes neurais, métodos estatísticos e sistemas baseados em regras. Do mesmo modo, as redes bayesianas são fortes ferramentas de representação de conhecimento e inferência sob condições de incertezas. Por exemplo, Naive Bayes, é uma simples técnica de redes bayesianas que assume que os atributos são independentes, dados os nós de classe. Alguns estudos mostram que a utilização de algoritmos de classificação Naive Bayes apresenta melhor desempenho por tratar uma característica como mais importante que outras se a probabilidade condicional da característica de uma dada classe diferenciar consideravelmente entre as classes [PAZ 96].

Atualmente, existem diversas aplicações e protótipos de modelos de classificação baseados em classificadores com algoritmo Naive Bayes. Estas aplicações possuem as mais variadas finalidades. Algumas classificações são baseadas no conhecimento das preferências de cada usuário [AKO 97, ASN 97, BEN 97, BIL 96, BIL 98, KRU 97, LIE 95, MIN 96, NIG 98, PAZ 96]. Porém estas preferências deveriam ser atualizadas por um mecanismo de aprendizagem contínua, conforme a mudança do perfil do usuário.

Neste trabalho, mais especificamente, constitui o foco principal a inferência/classificação das preferências do usuário dado o seu comportamento, enquanto ele consome informação em ambientes como, por exemplo, o software Direto [DIR 2001], que oferece serviços de correio, agenda e catálogo corporativos.

A necessidade do desenvolvimento de um aplicativo com as características citadas acima motiva a proposição de uma pesquisa, mais aprofundada, utilizando técnicas bayesianas para inferir as preferências do usuário. Essa pesquisa visa especificar e implementar um mecanismo de aprendizagem automática de preferências de usuários de ambientes com as características do *software* Direto.

O *software* Direto [DIR 2001] está sendo desenvolvido pela PROCERGS [PRO 2000] com a finalidade de ser utilizado como correio, agenda e catálogo corporativo. Foi construído utilizando a linguagem Java, sistema operacional *FreeBSD* e banco de dados *PostGres*, entre outros. Além disso, o Direto é um *software* livre, isto é, licenciado nos termos do *GNU General Public License* [GNU 2000]. O objetivo do Direto é ser executado em ambiente Internet de forma que o usuário apenas necessite de um *browser* para utilização do produto.

### 1.3 Proposta

O objetivo principal da pesquisa aqui proposta é a especificação e implementação de um Assistente de *Feedback*, o qual possui:

- ◆ um mecanismo de coleta automática de dados e eventos que refletem o comportamento do usuário;

- ◆ um modelo baseado em técnicas bayesianas para inferir as preferências do usuário, dado o seu comportamento durante o consumo de informações via correio eletrônico;
- ◆ uma representação estatística das preferências inferidas – servida de base para as sugestões de *feedback* a serem realizadas pelo assistente.

Pode-se identificar como principais contribuições deste trabalho:

- ◆ análise do domínio do serviço de e-mail do Direto, identificando padrões de comportamento pelas ações dos usuários com as mensagens (por exemplo, eventos como ler, encaminhar, arquivar em pastas e excluir mensagens) que representam as preferências e não preferências do usuário;
- ◆ pesquisa das técnicas e modelos baseados na estatística Bayesiana, identificando as mais apropriadas para inferir/classificar o comportamento de usuários dentro do domínio da proposta;
- ◆ pesquisa e especificação da interface do Assistente de *Feedback* para o usuário visualizar as estatísticas das preferências e escolher a configuração dos parâmetros do serviço de filtragem de *emails* do Direto;
- ◆ implementação e validação do protótipo deste classificador, atingindo o objetivo de inferir as preferências do usuário, dado o seu comportamento durante o consumo de informações via correio eletrônico;
- ◆ estudo de ambientes, preferencialmente o Direto, que serão utilizados como meio de simulação para o mecanismo de aprendizagem de perfil de usuários;
- ◆ criação de novas frentes de trabalho já que esta aplicação particular de classificação Naive Bayes não tinha sido abordada no contexto de ambientes como o Direto.

## 1.4 Organização do Texto

O restante deste texto está dividido em 6 capítulos. O capítulo 2 apresenta os principais conceitos que servirão de base para o entendimento dos classificadores Naive Bayes, introduzindo alguns conceitos básicos de probabilidades e o teorema de Bayes que fundamenta o funcionamento do classificador. Também no capítulo 2, descreve-se o classificador Naive Bayes, sua lógica e seu desempenho. No final deste capítulo é apresentado o ambiente Direto e a teoria dos modelos de usuários nas aplicações de agentes. As principais contribuições deste trabalho são apresentadas no capítulo 3 que trata da estrutura global do modelo do protótipo, apresentando detalhadamente cada uma das etapas que compõem o modelo, em especial a forma com que foram ajustadas para o objetivo pretendido.

No capítulo 4, é apresentada a ferramenta (protótipo) implementada para automação do modelo proposto, assim como sua interface e forma de manuseio. O capítulo 5 apresenta alguns resultados de diversos testes feitos para validação do modelo. Finalmente, no capítulo 6, são apresentadas as conclusões deste trabalho e possíveis direções de pesquisa para futuros trabalhos.



## 2 Classificadores Probabilísticos

Neste capítulo é apresentada uma revisão bibliográfica sobre os conceitos gerais de probabilidade, redes Bayesianas, algoritmos de aprendizagem e em especial o classificador Naive Bayes, que foi escolhido para ser utilizado no protótipo Assistente de *Feedback*, que será apresentado nos próximos capítulos. Inicialmente é feita uma descrição sobre os conceitos básicos de probabilidades e o teorema de Bayes. É abordado também sobre o raciocínio probabilístico, as redes Bayesianas e os algoritmos de aprendizagem, sendo mostrada uma classificação dos tipos mais comuns. Em seguida é detalhado o funcionamento do classificador Naive Bayes, mostrando alguns resultados do seu desempenho com relação a outros classificadores. Finalmente são mostrados os conceitos básicos do modelo de usuário e suas formas de aprendizagens e a arquitetura do ambiente do Direto.

### 2.1 Fundamentos Básicos de Probabilidades

#### 2.1.1 Modelos Matemáticos

Segundo Meyer [MEY 83], um modelo determinístico significa que o mesmo determina de alguma forma o resultado do experimento<sup>1</sup>, como, por exemplo, ao calcular o fluxo da corrente elétrica de uma bateria com a fórmula  $I=E/R$ . Ao colocar-se o valor de  $E$  e  $R$ , imediatamente tem-se o valor de  $I$ .

Para uma situação na qual não há condições de prever com certeza o valor do experimento, mesmo considerando todas as propriedades que poderiam influenciar no resultado final do experimento e não existindo modelo determinístico que forneça o resultado exato, deve-se chamar de modelo probabilístico (não determinístico). Segundo Lopes [LOP 99], o modelo probabilístico é um instrumento matemático que prevê a chance de um possível resultado sem que seja necessário repetir a experiência.

#### 2.1.2 Conjuntos

Alguns conceitos de conjuntos devem ser estudados para entender-se o modelo probabilístico. Conjunto é uma coleção de objetos e representam-se por uma letra maiúscula  $A, B$ . Pode ser descrito como  $A=\{1,2,3\}$  ou através de  $A=\{x|0<1 x <3\}$ .

O conjunto fundamental é o conjunto de todos os objetos que estejam sendo estudados. É, geralmente, representado pela letra maiúscula  $U$  ou pelo símbolo  $\Omega$ .

---

<sup>1</sup> Experimento: é o fenômeno que interessa ser observado, e cada realização dele é uma experiência. Também refere-se a qualquer processo de observação e medida.

Para cada experimento probabilístico  $E$  (por exemplo, jogar o dado e ver que valor aparecerá), define-se o *espaço amostral* como o conjunto de todos os resultados possíveis do experimento  $E$ . Geralmente, representa-se esses valores para o dado  $A$  como  $S = \{1,2,3,4,5,6\}$ .

Considerando um espaço amostral  $S$ , associado a um experimento  $E$ , um evento é um subconjunto do espaço amostral  $S$ , como, por exemplo,  $A = \{1,2,3\}$  para o caso do dado  $A$ .

*Evento composto* é aquele que apresenta vários resultados de um experimento. *Eventos mutuamente excludente* são aqueles que a ocorrência de um evento impede a ocorrência de outro.

### 2.1.3 Frequência Relativa

Considerando  $E$  como um experimento (por exemplo, jogar um dado), que será repetido  $n$  vezes,  $A$  como o evento associado a  $E$  e Frequência absoluta ( $N_A$ ) como o número de vezes que o evento  $A$  ocorre nas  $n$  repetições do experimento. Define-se como Frequência relativa de um evento  $A$  como o total da Frequência Absoluta ( $N_A$ ) dividido pelo número de repetições de  $E$  ( $n$ ).

A frequência relativa de  $A$  apresenta as seguintes propriedades:

- ◆  $0 \leq \text{Frequência}_A \leq 1$
- ◆  $\text{Frequência}_A = 1$  se  $A$  ocorrer em todas as repetições  $n$  de  $E$ .
- ◆  $\text{Frequência}_A = 0$  se  $A$  nunca ocorrer nas  $n$  repetições.

Exemplo [MEY 83]: ao jogar 8 vezes o dado ( $n$  é igual a 8), deparou-se com os seguintes resultados do experimento: 2,3,1,4,5,6,4,3. O evento  $A$  é {o dado cai com a face 4}. No final do experimento, a frequência relativa é igual a  $2/8$  (o valor 4 aparece 2 vezes, ou seja, a frequência absoluta é igual a 2).

### 2.1.4 Noções Fundamentais de Probabilidade

A forma mais comum de medir incertezas relacionadas a eventos (subconjuntos de resultados de um experimento) consiste em atribuir probabilidades ou especificar as chances de ocorrência do evento [FRE 2000].

Segundo Meyer [MEY 83], avaliar quão verdadeira é a ocorrência de  $A$ , utilizando a frequência relativa, pode ter alguns problemas:

- ◆ deverá se saber o tamanho de  $n$ ;
- ◆ caso considerar-se a sorte do experimentador, podem ocorrer variações na frequência relativa, pois pode-se, por sorte, repetir naquele experimento um resultado mais vezes, mesmo com o dado equilibrado. Em outro experimento, o resultado pode se inverter.

Quando não se pode ter qualquer informação a respeito dos resultados possíveis, pode-se descrevê-los como resultados igualmente prováveis. Caso for aplicar alguma experiência com elementos equilibrados que gerarão  $n$  resultados, a probabilidade de

cada resultado será de  $1/n$ . A probabilidade, *a priori*, é a razão do número de resultados igualmente prováveis favoráveis a uma determinada situação pelo número total de resultados [LOP 99].

Quando se deseja ter um resultado sem recorrer à experimentação deve se proceder da seguinte maneira:

- ◆ seja  $S$  um espaço amostral associado ao experimento  $E$ . A cada evento  $A$ , associa-se um número real  $P(A)$  e denomina-se probabilidade de  $A$ , ou probabilidade *a priori* de  $A$ , que satisfaça as seguintes propriedades:
  - $0 \leq P(A) \leq 1$ ;
  - $P(S)=1$  é a probabilidade do evento certo;
  - se os eventos  $A$  e  $B$  forem incompatíveis (mutuamente excludentes), então:  $P(A \cup B) = P(A) + P(B)$ ;
- ◆ se os eventos  $A_1, A_2, \dots, A_n$  forem, dois a dois, eventos mutuamente excludentes, então:

$$P\left(\bigcup_{i=1}^n A_i\right) = P(A_1) + P(A_2) + P(A_n) = \sum_{i=1}^n P(A_i)$$

- ◆ como calcular o  $P(A)$ :

$$P(A) = \frac{\text{número de casos favoráveis a } A \text{ que podem ocorrer em } E}{\text{número total de casos que } E \text{ pode ocorrer}}$$

Exemplo [MEY 83]: Ao se lançar um dado equilibrado, e se deseja verificar o evento  $A = \{5, 6\}$ , os valores maiores que 4, o resultado de  $P(A) = 1/6 + 1/6 = 2/6$ .

### 2.1.5 Probabilidade Condicionada

Sendo  $A$  e  $B$  dois eventos contidos dentro do experimento  $E$ , é possível condicionar  $A$  à ocorrência de  $B$ .

$$P(A|B) = \frac{P(A \cap B)}{P(B)}, P(B) > 0$$

A definição de  $P(A|B)$  pode ser definida como: dado que o evento  $B$  ocorreu, a probabilidade do evento  $A$  ocorrer é igual à probabilidade da interseção dos dois eventos ( $A$  e  $B$ ), dividida pela probabilidade de  $B$ , que deve ser maior que 0.

Então  $P(A|B)$  é definida como a probabilidade condicionada do evento  $A$ , quando  $B$  tiver ocorrido primeiro  $P(A \cap B) = P(B) * P(A|B)$  e a definição de  $P(B|A)$  à probabilidade condicionada do evento  $B$ , quando  $A$  tiver ocorrido primeiro  $P(A \cap B) = P(A) * P(B|A)$ .

Caso os eventos  $A$  e  $B$  forem independentes, ou seja,  $P(A|B) = P(A)$ , neste primeiro caso, o evento  $A$  não depende dos resultados do evento  $B$  e  $P(B|A) = P(B)$ , o evento  $B$  não

depende dos resultados do evento  $A$ . Neste dois casos pode-se dizer que:  $P(A \cap B) = P(A) * P(B)$ .

Exemplo [MEY83]: Caso se tenha o espaço amostral  $S$  de 100 peças e dentro desse lote 20 são defeituosas e 80 não defeituosas.  $A = \{a \text{ primeira peça defeituosa}\}$  e  $B = \{a \text{ segunda peça defeituosa}\}$ . Com reposição,  $P(A)$  terá probabilidade igual a  $P(B)$  com valor igual a  $20/100$ , mas sem reposição  $P(A)$  será  $20/100$  e  $P(B|A)$  será  $19/99$ . Isto ocorre porque houve uma redução do espaço amostral de uma peça, provocada pelo evento  $A$ .

O resultado de  $(A \cap B)$  ocorre somente se existe um resultado que satisfaça os dois eventos  $A$  e  $B$ . Neste caso o cálculo de  $(A \cap B)$  seria igual

$$a: \frac{\text{N}^\circ \text{ de resultados que satisfaz os eventos A e B}}{\text{N}^\circ \text{ total de resultados}}$$

Para entender o cálculo de  $P(A \cap B)$ , [MEY83] apresenta um exemplo de dois dados equilibrados sendo lançados, registrando o resultado como  $(x_1, x_2)$ . Ocorrendo 36 resultados igualmente prováveis:

$$S = \left\{ \begin{array}{l} (1,1)(1,2)\dots(1,6) \\ (2,1)(2,2)\dots(2,6) \\ (6,1)(6,2)\dots(6,6) \end{array} \right\}$$

Considerando os dois eventos:

$$A = \{(x_1, x_2) | x_1 + x_2 = 10\} \quad \text{e} \quad B = \{(x_1, x_2) | x_1 > x_2\}$$

$$\text{Assim, } A = \{(5,5), (4,6), (6,4)\} \quad \text{e} \quad B = \{(2,1), (3,1), (3,2), \dots\}$$

A interseção dos eventos  $A$  e  $B$   $P(A \cap B)$ , que representa a soma de dois dados for 10 e se o primeiro dado tiver apresentado um valor maior que o segundo dado, é representado por um resultado que encaixa nesta situação  $(6,4)$ , então o valor de  $(A \cap B)$  é  $1/36$ .

Pode-se obter as seguintes combinações de probabilidades condicionais entre dois eventos  $(A$  e  $B)$ :

$$P(A|B) = \frac{P(A \cap B)}{P(B)} \quad \text{para } P(B) > 0 \quad \text{e} \quad P(B|A) = \frac{P(A \cap B)}{P(A)} \quad \text{para}$$

$$P(A) > 0$$

Obs: Quando  $P(B) = 0$ ,  $P(A|B)$  não é definida, e quando  $P(A) = 0$ ,  $P(B|A)$  não é definida.

Existem duas formas para se calcular a probabilidade condicionada de  $P(B|A)$ :

- ◆ diretamente, pela consideração da probabilidade de  $B$  em relação ao espaço amostral reduzido  $A$ ;
- ◆ empregando a definição acima, onde  $P(A \cap B)$  e  $P(A)$  são calculados em relação ao espaço amostral original  $S$ .

### 2.1.6 Teorema de Bayes

Segundo Meyer [MEY 83], o teorema de Bayes é também chamado de probabilidades das *causas* ou dos *antecedentes*, esse teorema ainda é denominado como *regra de Bayes* ou *lei de Bayes* [CAR 99a].

Existe uma grande diferença na expressão  $P(A|B)$  e  $P(B|A)$ . A diferença é clara entre as expressões observando o primeiro exemplo da seção 2.1.5, quando se deseja formular esta questão para o exemplo: supor que ao tirar uma peça do depósito que mantém peças fabricadas pelas fábricas 1, 2 e 3, deseja-se saber qual a probabilidade que uma peça seja da fábrica 1? Nota-se que o raciocínio é inverso, isto é, do efeito para a causa.

Empregando a notação já introduzida, pede-se agora  $P(B|A)$ . Existem muitos problemas que envolvem os pares de probabilidades condicionais  $P(A|B)$  e  $P(B|A)$ . Igualando a expressão  $P(A \cap B)$ , das duas fórmulas dos pares das probabilidades condicionais da seção 2.1.5, obtém-se a seguinte expressão:

$P(A) \cdot P(B|A) = P(B) \cdot P(A|B)$  e assim,

$$P(B|A) = \frac{P(B) \cdot P(A|B)}{P(A)} \text{ ou}$$

$$P(A|B) = \frac{P(A) \cdot P(B|A)}{P(B)}$$

Pode-se calcular esta probabilidade como consequência da seguinte exposição: seja  $B_1, B_2, \dots, B_k$  uma partição do espaço amostral  $S$  e seja  $A$  um evento associado a  $S$ . Com a probabilidade condicionada pode-se escrever:

$$P(B_i|A) = P(A|B_i)P(B_i) / \sum_{j=1}^k P(A|B_j)P(B_j) \quad \text{onde } i=1,2,3,4,k$$

Para calcular, precisa-se do  $P(B_i)$ , mas muito frequentemente este valor é desconhecido. Por isso deve-se tomar muito cuidado com o cálculo do  $P(B_i)$  porque sua ausência pode limitar a aplicabilidade do teorema.

Exemplo [MEY83]: Uma determinada peça é fabricada por 3 fábricas, 1, 2 e 3. Sabe-se que 1 produz o dobro de peças que 2, e 2 e 3 produziram o mesmo número de peças (durante um determinado tempo). Sabe-se também que dois por cento das peças produzidas por 1 e por 2 são defeituosas, enquanto quatro por cento daquelas produzidas por 3 são defeituosas. Todas as peças produzidas são colocadas em um depósito e depois uma peça é extraída ao acaso. Qual a probabilidade da peça ter sido produzida pela fábrica 1?

Primeiramente, deve-se calcular os valores totais de peças contidas na fábrica 1, 2 e 3.

Como a quantidade da fábrica 1 apresenta o dobro de 2 e 3:

$1x + 1/2x + 1/2x = 2x$  o total de peças.

$P(B_1) = \{\text{provém de 1}\} = 1x/2x = 1/2;$

$P(B_2) = \{\text{provém de 2}\} = 1/2x/2x = 1/4;$

$P(B_3) = \{\text{provém de 3}\} = 1/2x/2x = 1/4;$

$$P(A|B_1)=0,02;$$

$$P(A|B_2)=0,02;$$

$$P(A|B_3)=0,04;$$

$$P(B_1|A)=\frac{(0,02)/(1/2)}{(0,02)(1/2)+(0,02)(1/4)+(0,04)(1/4)}=0,40$$

Segundo Freund [FRE00], como o teorema de Bayes envolve o raciocínio inverso do efeito para a causa, é que desenvolve um papel importante para as inferências bayesianas, que o raciocínio vai dos dados amostrais observados para as populações.

## 2.2 O Raciocínio Probabilístico

O raciocínio probabilístico é baseado na realização de inferências probabilísticas, que é o cálculo da probabilidade de um evento, dada toda evidência possível. Este cálculo é baseado em probabilidade condicional e no teorema de Bayes [RUS 95a] citado por [LAD 2000]. Como, em geral, existem diversas evidências, esta abordagem apresenta certa dificuldade para aplicações reais, porque exige que uma enorme matriz de probabilidades condicionais seja estimada e fornecida para o sistema, inviabilizando a aquisição de conhecimentos e implicando grande tempo, armazenamento e capacidade computacional para processar essas informações.

Para contornar este problema, os primeiros sistemas computacionais para suporte à decisão (anos 60) eram fundamentados na teoria da decisão e tratavam incerteza com uma forma restritiva, mas viável em computador, da teoria probabilística de Bayes. Em especial, os sistemas para diagnósticos requeriam que o conjunto de possíveis doenças a diagnosticar fosse mutuamente exclusivo. A evidência era assumida condicionalmente independente, dada qualquer hipótese (doença possível), e somente uma doença era assumida existir em cada paciente [JEN 90a] citado por [LAD 2000]. Essa abordagem apresentava muitos problemas, tais como:

- ◆ em domínios maiores, as técnicas adotadas apresentavam resultados matematicamente incorretos;
- ◆ não existiam mecanismos que explicassem os resultados para os não especialistas;
- ◆ não existiam meios de representar a estrutura do conhecimento humano.

Por esses motivos, houve uma diminuição do interesse pelo uso da probabilidade nessa abordagem.

Os sistemas especialistas dos anos 70 utilizaram uma linguagem para representar o conhecimento do especialista em uma forma semelhante aos predicados lógicos e empregavam algum método de raciocínio heurístico. Para manipular incerteza, utilizavam métodos derivados da teoria de probabilidade, mas em desacordo com os axiomas da probabilidade [LAD 2000]. Os problemas dos anos 70 são resolvidos nesta abordagem, como:

- ◆ trabalhavam com domínios de problemas maiores e mais complexos;
- ◆ provêm facilidades de explicação para os não especialistas;
- ◆ as características importantes deveriam ser citadas nesta abordagem;

- ◆ foram feitas fortes suposições de independência condicional que facilitariam o seu uso, mas criariam perigos se não fossem bem definidas as dependências importantes;
- ◆ deveriam ser evitadas causalidades inversas;
- ◆ os antecedentes de uma regra deveriam relacionar todas as condições dependentes;
- ◆ a rede de inferência deveria ter a forma de uma árvore.

No final dos anos 80, houve uma retomada dos interesses por abordagens probabilísticas, motivada pelo fato de apenas utilizar probabilidades condicionais entre variáveis diretamente dependentes, para representar relacionamento causal e a independência condicional entre variáveis do domínio.

Essa retomada está associada ao surgimento das representações gráficas de dependências probabilísticas chamadas de redes probabilísticas. As redes bayesianas (são adequadas para realizar inferências probabilísticas), as redes de *Markov* e os diagramas de influências (representam a formalização de um problema de decisão em um domínio com incerteza, onde o decisor é racional) são exemplos de redes probabilísticas. A principal característica desses tipos de redes é a habilidade de reduzir o cálculo a uma série de cálculos locais, usando somente variáveis obtidas de um objeto e seus vizinhos em uma estrutura de grafo [LAD 2000].

### 2.2.1 Raciocínio Incerto

Segundo Ladeira [LAD 2000], as pessoas resolvem problemas e tomam decisões em ambientes onde a informação é apresentada parcial ou de forma aproximada. A informação é parcial quando algumas respostas e questões relevantes não são conhecidas. São aproximadas quando as respostas são conhecidas, mas não são exatas (pouca confiabilidade). Tem-se tentado utilizar estas informações parciais e aproximadas em sistemas inteligentes, mas apenas pode-se obter soluções aproximadas, isto é, com incerteza.

É necessário dispor de formas para lidar com incertezas. Duas abordagens podem ser utilizadas [LAD 2000]:

- ◆ **abordagem simbólica:** utiliza uma teoria lógica que determina o mecanismo de inferência utilizado. Utiliza-se esta abordagem para tratar incerteza em informação parcial e justificar conclusões finais;
- ◆ **abordagem numérica:** representa incerteza como uma quantidade precisa em uma dada escala ou valor *fuzzy*<sup>2</sup>, permitindo definir o cálculo a utilizar para combinar e propagar a incerteza durante o processo de raciocínio. Esta quantidade pode ser um único valor ou um intervalo e o usuário ou especialista deve fazer uma atribuição numérica precisa e consistente para a incerteza dos dados atômicos e das suas

---

<sup>2</sup> Fuzzy: é uma técnica adequada para representar conceitos vagos inerentes a termos lingüísticos, tais como: imprecisão lingüística de predicados como 'alto', 'maior', 'próximo', 'jovem'; imprecisão como 'a caixa pesa entre 10 a 16 quilos' e quantificadores imprecisos como 'muitos', 'alguns', 'poucos', etc. [LAD 00]

relações. A agregação de incerteza permite classificar os fatos resultantes (com base no valor agregado), permitindo o seu uso na tomada de decisão.

### 2.2.2 Relacionamento Condicional

Os pesquisadores utilizam como base o relacionamento condicional para a organização do conhecimento humano. Em  $P(A|B)$ , o B serve como contexto para mensurar a crença de A. O conhecimento de um especialista é codificado através da probabilidade condicional. Por exemplo, em um diagnóstico de doença, o médico tem mais confiança em diagnosticar uma doença utilizando um sintoma (B) para medir a crença de uma doença (A), do que simplesmente utilizar uma probabilidade incondicional da doença  $P(A)$ .

Os pesquisadores utilizam também como base para expressar a noção quantitativa de importância de uma rede probabilística, os conceitos básicos de independência condicional e dependência condicional, definidos como:

- ◆ **independência Condicional:** a notação  $P(A,B,C)$  representa que A é independente de C, dado B, ou seja, o conhecimento de C não afeta a crença de A, quando já se conhece B. Intuitivamente, A é determinado por B sem levar C em conta;
- ◆ **dependência Condicional:** a ligação de um arco entre duas variáveis pode demonstrar alguma dependência entre essas duas variáveis. Pode-se definir algumas variações dessa dependência, como:

## 2.3 Redes Bayesianas

### 2.3.1 Histórico

Bayes, em 1763, sugeriu uma regra (regra de Bayes) possibilitando que a probabilidade de um evento possa ser dada com base no conhecimento humano, ou seja, em eventos nos quais não se pode medir a frequência com que ocorrem, a probabilidade pode ser dada com base no conhecimento que um especialista tem sobre o mesmo [HRU 97].

A estatística bayesiana passou a ser aplicada em sistemas de Inteligência Artificial (IA) no início dos anos 60 (Russel & Norvig,1995) citado por [HRU 97]. Naquela época, o formalismo da utilização de probabilidades condicionais ainda não estava bem definido. Além disso, a quantidade muito grande de dados impossibilitava a utilização dessa teoria. A partir dos anos 70 até a metade dos anos 80, a probabilidade bayesiana não foi muito utilizada nas pesquisas de IA.

No final dos anos 80, com as publicações de trabalhos que definiram de forma mais concisa o ferramental bayesiano e reduziram a quantidade de cálculos necessários, a teoria bayesiana sofreu um grande impulso no campo de IA.



### 2.3.2 Definição de Redes Bayesianas

Hruschka [HRU 97] define a probabilidade bayesiana como uma teoria consistente e que permite a representação de conhecimentos certos e incertos sobre condições de incerteza via distribuição conjunta de probabilidades. Tal distribuição conjunta pode ser representada pelo produto de distribuições condicionadas, como, por exemplo:

$$P(X_1, X_2, X_3, X_4, X_5, X_6) = P(X_6|X_5) P(X_5|X_2, X_3) P(X_2|X_1) P(X_4|X_1) P(X_3|X_1) P(X_1)$$

Uma variável é condicionada a uma ou mais variáveis numa relação causal. Uma distribuição pode ser representada por um grafo orientado. No grafo, cada nó representa uma variável do modelo, e os arcos ligam as variáveis que estão em relação direta causa/efeito. Por exemplo, se houvesse uma pesquisa com pessoas que passam pela rua, que perguntasse: que dia é hoje? Sem dúvida a resposta da maioria seria a mesma. Isso porque a maioria das pessoas segue um mesmo calendário e isso causa o fato das respostas serem aproximadamente iguais. Sendo assim, se se deseja saber qual será a resposta do próximo entrevistado, não é necessário observar todas as respostas anteriores; basta observar a causa, ou seja, o fato do calendário ser utilizado pela maioria da população. A esta estrutura gráfica, com a quantificação de crença nas variáveis e seus relacionamentos, dá-se o nome de redes bayesianas (RB) ou redes causais.

Uma rede bayesiana é um gráfico direcionado acíclico (DAG) com uma distribuição de probabilidades condicionais para cada nó. Cada nó  $x$  representando uma variável de domínio, e cada arco  $a_i$  representando uma dependência condicional entre os nós [CHE 99]. Quando se aprende conjunto de dados em redes bayesianas, usa-se o nó para representar atributos dos conjuntos de dados. (fig. 2.1)

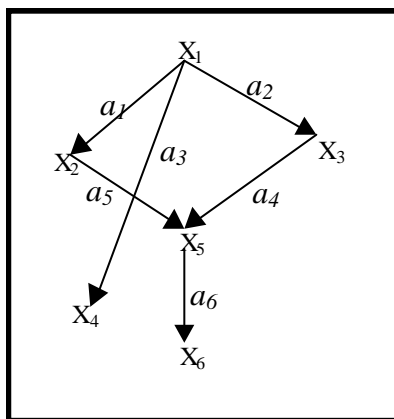


FIGURA 2.1 - Grafo da Rede Bayesiana que representa a distribuição  $P(X_1, X_2, X_3, X_4, X_5, X_6)$

Para verificar se um grafo DAG é uma rede bayesiana, precisa-se verificar se cada variável  $x$  do grafo deve ser condicionalmente independente de todos os nós que não são seus descendentes exceto seus pais (Perl,1988), citado por [HRU 97]. Esta condição permite reduzir consideravelmente o esforço computacional, porque existe uma explosão combinatório no cálculo da distribuição conjunta das probabilidades.

Para reduzir o esforço computacional, basta explorar as distribuições das relações entre as variáveis do problema (Perl,1988), citado por [HRU 97]. Graças a esta distribuição

foram desenvolvidos vários algoritmos de propagação de crença em RB, os quais permitiram, sobre uma rede, propagar o conhecimento de novos fatos. Propagação de crenças corresponde em estabelecer um procedimento que, explorando a conectividade da rede, permita a determinação das distribuições de probabilidades das variáveis, objetivo do diagnóstico, condicionadas aos valores das variáveis que representam evidências. Assim, o conhecimento incerto pode ser atualizado de forma simples e clara, mantendo a consistência e a confiabilidade.

As redes bayesianas podem ser visualizadas de duas maneiras:

- ◆ como uma estrutura que codifica um grupo de relacionamentos de independência condicional entre os nós, conforme conceito de *d-separação*<sup>3</sup>. A idéia é que uma estrutura de RB possa ser instruída pelo mecanismo de aprendizagem com a independência condicional entre os nós da rede. Usando alguns testes estatísticos, pode-se encontrar a relação de independência condicional entre os atributos e usar o relacionamento como forma para construção de uma RB;
- ◆ como uma estrutura que codifica a união distributiva dos atributos. Então uma RB pode ser usada como um classificador que dá a posterior probabilidade distributiva do nó de classificação, dados os valores de outros atributos.

Com a RB pode-se representar problemas do mundo real em que existam relações de causa e consequência entre as variáveis. Isso motiva o uso desses tipos de redes porque os seres humanos têm uma certa necessidade de representar o conhecimento utilizando fatos em forma de relacionamentos causais [HRU 97].

As RB's também possuem a possibilidade de realizar o aprendizado a partir dos dados. Nesse aprendizado, é dada uma amostra, e o sistema, através de um algoritmo, gera uma estrutura que melhor se adapta aos dados do problema (Heckerman, 1995), (Buntine, 1994 e 1995) citados por [HRU 97]. Existem vários algoritmos e métodos para o aprendizado de RB's a partir dos dados, sendo que cada um se adapta melhor a uma determinada classe de problema.

### 2.3.3 Aprendizado em Redes Bayesianas

O objetivo do aprendizado em redes bayesianas é possibilitar o menor esforço computacional possível para facilitar a recuperação dos dados na rede, através da criação de uma estrutura interna que representará as restrições de um determinado problema [PER 88]. Este fato motiva muitos autores a verem o processo de aprendizagem como um método de otimização. Para o trabalho do especialista, o aprendizado facilita bastante no desenvolvimento de um sistema, pois como o aprendizado cria uma estrutura gráfica que representa o sistema, o especialista precisa apenas analisar a estrutura gerada, que é de fácil compreensão.

O processo de aprendizagem em redes bayesianas resume-se a aprender as distribuições probabilísticas (parâmetros numéricos) de uma rede e identificar as relações de interdependência dadas pelos arcos (estrutura) [HRU 97].

---

<sup>3</sup> D-separação: Segundo Nilsson [NIL98], isto acontece quando uma Rede Bayesiana (RB) implica mais independência condicional do que geralmente aquelas independências envolvidas com os pais de um nó.

Quando se estudam problemas de tamanho exponencial, dificultando alcançar uma única estrutura representativa para o problema, deve-se, então, aplicar a aprendizagem em uma amostra que melhor representa o problema em questão. Quanto maior é a amostra do problema, o erro diminui para a escolha da melhor estrutura. Isso pode ser visto no gráfico da fig. 2.2.

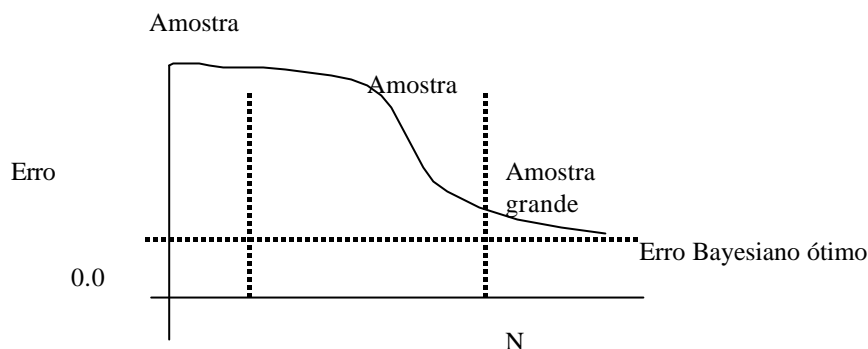


FIGURA 2.2 - Gráfico do tamanho das amostras e o erro associado

Fonte [HRU 97]

Por outro lado, as amostras também podem apresentar tamanho exponencial, tornando o processo de aprendizagem em um problema NP-completos [HRU 97].

Segundo Hruschka [HRU 97], quando a estrutura da rede é pré-fixada, o aprendizado passa a ser somente em relação aos valores das variáveis. Existem diversos algoritmos para aprendizado de parâmetros *Parameter Fitting* ((Edwards, 1989), (Jirousek & Preucil, 1995), (Lauritzen, 1995) e (Dempster entre outros, 1977)), *Maximum likelihood* (Casella & Berger, 1990) e *Bayesian Maximum A-Posterior* (MAP) ((Heckerman, 1995), (Howard, 1970)) todos citados por [HRU 97]. As maiores dificuldades neste aprendizado é a falta de dados em problemas do mundo real e os erros gerados pela aproximação.

## 2.4 Classificadores

Segundo [HEN 94], a tarefa de classificação é muito freqüente durante as atividades humanas. Quando se deseja decidir ou predizer com base em informações disponíveis, deve ser usado um método de classificação para julgamento de novas situações. Os problemas que serão tratados neste trabalho são constituídos de procedimentos que serão aplicados numa seqüência de casos, onde cada novo caso pode ser assinalado por um conjunto pré-definido de classes baseado na observação de atributos ou de características. A classificação recebe outras terminologias como reconhecimento de padrões, discriminação, ou aprendizagem supervisionada para os procedimentos de classificação de um conjunto de dados pelo qual as classes são conhecidas. Pode ser chamada de aprendizagem não supervisionada ou clusterização, onde as classes são inferidas de um conjunto de dados.

Muitos dos problemas que exigem decisão ou predição encontrados na ciência, indústria e comércio podem ser resolvidos com classificação usando, freqüentemente, muitos dados, como, por exemplo, nos procedimentos mecânicos para ordenar cartas nas máquinas de leituras de códigos postais, classificar o crédito com base no histórico

financeiro e informações pessoais do cliente e diagnósticos preliminares de doenças de pacientes através de resultados de exames.

### 2.4.1 Aprendizagem de Máquina

A área de aprendizado de máquina estuda a modelagem de sistemas de aprendizagem em computadores e constitui um dos maiores desafios para os pesquisadores na Inteligência Artificial. Atualmente o campo de aprendizagem de máquina se esforça para instruir computadores a realizar trabalhos, utilizando definições corretas e completas de um algoritmo para esta tarefa. Estas atividades envolvem um esforço grandioso para os especialistas. Mesmo assim, a pesquisa de aprendizado de máquina tem como objetivo básico explorar soluções de aprendizado, incluindo descobertas de diferentes algoritmos de indução, o escopo e limitações de certos métodos, as informações que devem estar disponíveis para os aprendizes. Mais especificamente, generaliza, classificando em expressões simples suficientes para serem entendidas por um ser humano. Tenta imitar o raciocínio humano para conseguir eficiência no processo de decisão e predição.

Diante disso o aprendizado humano não é a única forma de adquirir conhecimento e habilidade. Torna-se parte do grande domínio de possibilidades de métodos de aprendizagem.

Conforme Ramirez Pozo [POZ 2000], as técnicas de aprendizado utilizadas para incrementar um método podem ser classificadas em quatro grandes grupos, tais como:

- ◆ **aprendizado pela implantação direta de novos conhecimentos:** nenhuma inferência do conhecimento é requerida por parte do aprendiz. Pode variar de aprendizagem de uma programação convencional construída por uma entidade externa; e aprendizagem por memorização de fatos e dados utilizando um sistema convencional de banco de dados;
- ◆ **aprendizagem por instrução:** requer que o aprendiz transforme o conhecimento em uma representação interna, integrando ao conhecimento anterior. O aprendiz aplica alguma inferência, mas a grande responsabilidade fica por conta do instrutor que deve estar presente para incrementar o conhecimento anterior do aprendiz;
- ◆ **aprendizagem por analogia:** adquirir novos fatos ou habilidades pela transformação e aumento do conhecimento existente através da forte similaridade para as novas situações. Exige mais inferência que a aprendizagem por instrução;
- ◆ **aprendizagem por exemplos:** pegando uma série de exemplos e contra-exemplos de um conceito, induzindo uma descrição geral que descreve todos os exemplos positivos do conceito. A quantidade de inferência executada é maior que os tipos de aprendizagem citados anteriormente. As origens desses exemplos podem ser feitas pelo próprio aprendiz, por um especialista ou de um ambiente externo. Este trabalho tratará especificamente deste tipo de aprendizagem.

O aprendizado de máquinas é um conjunto de técnicas que executam duas tarefas fundamentais: generalizar a partir de um conjunto de exemplos conhecidos e detalhar a estrutura destas generalizações. Há uma forma de classificar os algoritmos de aprendizagem de máquina da forma como eles aprendem, utilizando aprendizado supervisionado, ou um aprendizado não supervisionado[FEL 98]:

- ◆ **aprendizado supervisionado**: a idéia básica é de um classificador. Os padrões são encontrados explorando o número de casos conhecidos que implicam padrões bem definidos e, conseqüentemente, as generalizações são formadas. Uma forma de representar o aprendizado supervisionado é usar a generalização a partir de casos conhecidos, de tal forma que, dado um conjunto de observações no qual as classes (categorias) são conhecidas, o objetivo está em encontrar regras capazes de classificar novas observações entre as classes já existentes;
- ◆ **aprendizado não supervisionado**: os padrões dos exemplos não são conhecidos em sua estrutura. Os padrões dos dados são encontrados começando de alguma caracterização lógica de regularidade. O ponto de partida desse tipo de aprendizado é alguma descrição do tipo de padrão requerido. Pode-se usar representar o aprendizado não supervisionado como um processo analítico, ou seja, expandindo alguma descrição implícita, de tal forma que, dado um conjunto de observações, o objetivo é tentar estabelecer a existência de classes ou *clusters* nos dados.

Os algoritmos de aprendizagem de máquina, também chamados de classificadores, podem ser agrupados de acordo com seu tipo de método de aprendizado [FEL 98]:

- ◆ **método simbólico**: busca aprender através da construção de uma representação de um conceito utilizando análise de exemplos e contra exemplos desses conceitos. Essa representação é uma estrutura simbólica que pode ser uma expressão lógica, árvores de decisão, regras de produção, redes semânticas, etc;
- ◆ **método estatístico**: as técnicas estatísticas procuram focar funções em que todos os atributos têm valores contínuos ou ordinais. Um atributo é uma possível característica relevante do conceito a ser aprendido. Esses atributos podem ser paramétricos, assumindo alguma forma de modelo, onde seus valores são encontrados a partir de um conjunto de dados. Os classificadores estatísticos freqüentemente assumem que valores de atributos estão normalmente distribuídos, e então usam os dados fornecidos para determinar médias, variâncias e co-variâncias da distribuição;
- ◆ **método *Instance-based***: Uma forma de classificação de casos é usar a similaridade de casos. A função básica desse método é encontrar a classe de um novo caso através da lembrança de um caso similar onde a classe é conhecida;
- ◆ **método conexionista**: Esse método utiliza estruturas matemáticas, onde é conectado um grande número de pequenas unidades de processamento ligadas em rede. Um exemplo desse método são as redes neurais, onde seus elementos são modelados como neurônios e foram inspirados no modelo biológico do sistema nervoso;
- ◆ **método genético**: Um classificador genético consiste de uma população de elementos de classificação que competem para fazer uma predição. Elementos com baixo desempenho são descartados, enquanto que elementos com bom desempenho se proliferam, produzindo variações de si mesmos.

Os classificadores podem ser agrupados também pelos tipos de linguagens de descrição que são utilizadas para representar exemplos ou instâncias, conceitos aprendidos ou conhecimento de fundo [FEL 98], tais como:

- ◆ **linguagem de descrição de exemplos ou instâncias**: descreve os exemplos utilizados pelo programa para aprender conceitos, estabelecendo limites sobre os tipos de padrões que o sistema pode aprender;

- ◆ **linguagem de descrição de hipóteses ou conceitos aprendidos**: descreve o estado interno de um programa de aprendizado, correspondente à teoria dos conceitos ou padrões que existem nos dados, estabelecendo limites sobre o que pode ser aprendido; Estruturas como regras de produção, árvores de decisão e redes semânticas utilizam esse tipo de linguagem;
- ◆ **linguagem de teoria de domínio ou conhecimento de fundo**: descreve todo o conhecimento prévio que o programa possui sobre o domínio.

Felix [FEL 98] classifica os algoritmos de aprendizagem de máquina segundo o modo de inclusão de novos exemplos, tais como:

- ◆ **não incremental**: necessita que todos os exemplos de treinamento estejam disponíveis para que seja induzindo um conceito;
- ◆ **incremental**: Este modo revê o conceito corrente a cada momento em que ocorre a classificação de um exemplo. Os exemplos observados são considerados um a um, isto é, considera o primeiro exemplo, construindo uma hipótese; a seguir considera o segundo exemplo, que pode ou não modificar a primeira hipótese.

## 2.4.2 Definição de Classe

Uma importante questão é o entendimento do conceito de classes nos estudos de classificações, considerando a natureza e o caminho que elas são definidas. Existem três casos bem definidos de classes, descritos em [HEN 94]:

- ◆ **classes como rótulos de diferentes populações**: neste tipo de classe não leva em consideração associações de várias populações. Por exemplo, cachorro e gato formam duas diferentes classes ou populações, e como é conhecido, com certeza, ou o animal é um cachorro ou um gato (ou nenhum). Associação de uma classe ou população é determinada por uma autoridade independente, a distribuição para uma classe é independentemente determinada de quaisquer atributos ou variáveis;
- ◆ **classes resultantes de um problema de predição**: esta classe é uma consequência do resultado de uma predição através do conhecimento dos atributos. Em termos estatísticos a classe é uma variável randômica. Um exemplo típico é em aplicações de predições de taxas de juros. Frequentemente a questão é: as taxas de juros subirão (classe=1) ou não (classe=0)?;
- ◆ **classes determinadas pelos valores espaciais dos seus atributos**: pode-se dizer que a classe é uma função dos atributos. Então um item manufaturado pode ser classificado como falho se alguns atributos estão fora dos limites pré-determinados, e não falhos de outra forma. Existem regras que classificam os valores dos atributos. O problema é criar uma regra que imite a realidade o mais próximo possível. Muitos sistemas de crédito são desse tipo de classe.

## 2.4.3 Os Classificadores

A partir do formalismo apresentado na seção anterior pode-se agora classificar os tipos de classificadores mais citados nas bibliografias relacionadas ao assunto. Estas definições foram baseadas no trabalho de [FEL 98]. A tab. 2.1 mostra os algoritmos e

suas classificações quanto ao método, tipo de aprendizagem, linguagem de descrição, incrementabilidade, características e sua referência bibliográfica.

TABELA 2.1 - Classificação dos algoritmos de aprendizagem (classificadores)

Classificador	Tipo	Método	Linguagem	Incrementabilidade	Característica	Ref.
ID3	Superv.	Proposicional	Árvore de decisão	Não Incremental	Algoritmo básico que constrói uma árvore de decisão onde determina a classificação de objetos, testando os valores das propriedades.	[QUI86]
C4.5	Superv.	Proposicional	Árvore de decisão	Não Incremental	Generaliza através de classificação por árvore de decisão considerando todos os possíveis testes que o conjunto de dados usados pode apresentar, escolhendo o que apresentar melhor informação. É uma evolução do ID3.	[QUI93]
Naives Bayes	Não superv.	Estatístico			Classifica um determinado valor de um dado desconhecido baseado nas probabilidades observadas das classes dado um conjunto de casos treinados. (ver detalhes seção 2.5).	[PAZ97], [LAN92]
Aha-IB	Superv.	Instance-Based	Exemplos	Incremental	Classifica um novo caso a partir da forte similaridade em outro caso cuja classe é conhecida assumindo que o novo caso terá a mesma classe.	[AHA92]
CART	Superv.	Estatístico	Árvore de decisão	Não incremental	Monta árvores de decisão, utiliza técnicas estatísticas enfocando tarefas em que todos os atributos têm valores contínuos ou ordinais.	[FRI84]
Nearest-Neighbor	Não superv.	Estatístico		Incremental	Classifica armazenando todos os exemplos dos conjuntos de dados treinados. Ele assinala a classe que seja mais similar ao novo caso, buscando o exemplo que apresentar a maior quantidade de características em comum. Usa a similaridade como o Aha- IB.	[AHA92]

## 2.5 O Classificador Naive Bayes

As redes bayesianas têm sido usadas em processos de classificação há muitos anos. Quando sua estrutura apresentar uma classe como nó pai de todos os outros nós e nenhuma outra conexão é permitida, torna-se ideal para os processos de classificação. Esta estrutura é comumente chamada de redes Naive Bayes, que é um caso especial de redes probabilísticas ou redes bayesianas.

Uma rede bayesiana é um gráfico acíclico onde cada nó representa um atributo (interpretado como uma variável randômica), que é usado para descrever um domínio de interesse, e cada ligação representa uma dependência entre os atributos. O gráfico mostra uma particular junção de probabilidade distributiva, onde cada nó da rede representa uma probabilidade condicional distributiva entre os valores dos atributos. Uma rede bayesiana como classificador Naive Bayes apresenta o gráfico em forma de estrela, no qual o centro da estrela é a classe que será classificada. Como pode ser

observado na fig. 2.3, os atributos formam as pontas da estrela ( $A_1$  a  $A_n$ ). A única conexão possível é cada atributo com a classe ( $C_i$ ). Nenhuma outra conexão é permitida na rede Naive Bayes.

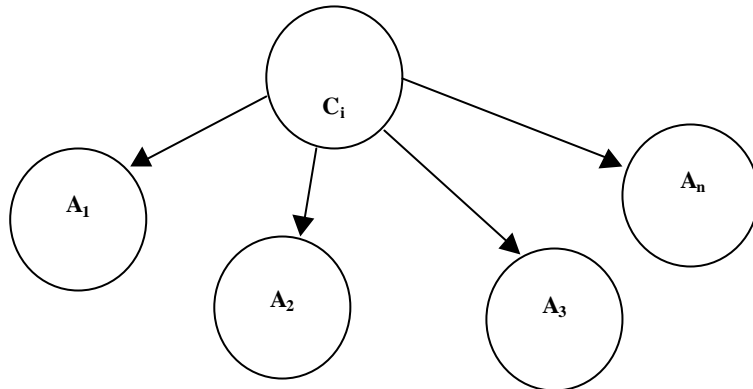


FIGURA 2.3 - A estrutura da rede Naive Bayes em estrela

Será necessário definir algumas notações para facilitar o entendimento, tal como  $C$ , que representará um conjunto de classes que assume várias características que sofrerão a predição. Esse conjunto de classes  $C$  com finito domínio,  $\text{Dom}(C)=\{c_1, \dots, c_t\}$ , onde cada  $c_i$  representa uma classe do domínio de  $t$  classes. Define-se também que  $A_1, \dots, A_n$  como um conjunto de atributos valorados usados para descrever um caso do universo de casos. Esses atributos terão a função de prever as classes, podendo assumir valores simbólicos  $(A_j)=\{a_1^{(j)}, \dots, a_m^{(j)}\}$  ou valores numéricos  $(A_j)=\mathcal{R}$ . Define-se  $a_i^{(j)}$  para o valor de um atributo  $A_j$  independente se ele é numérico ou simbólico. A variável  $j$  indicará o atributo da quantidade total de  $n$  atributos e  $m$  a quantidade de valores para um determinado atributo.

Em especial, neste trabalho, serão apresentados exemplos de atributos usando apenas valores simbólicos, já que na aplicação a ser apresentada nos próximos capítulos utilizará apenas este tipo de atributo.

A distribuição probabilística do classificador Naive Bayes, representando o grafo definido na fig. 2.3, pode ser visualizada na equação (1).

$$P(C = c_i | A_1 = a_{i_1}^{(1)}, \dots, A_n = a_{i_n}^{(n)}) = P(A_1 = a_{i_1}^{(1)}, \dots, A_n = a_{i_n}^{(n)} | C = c_i) \quad (1)$$

onde:

$P(C = c_i | A_1 = a_{i_1}^{(1)}, \dots, A_n = a_{i_n}^{(n)})$  será o valor percentual calculado pelo classificador para a classe  $c_i$ , os atributos valorados  $(A_1 = a_{i_1}^{(1)}, \dots, A_n = a_{i_n}^{(n)})$  será o conjunto de valores de todos os atributos que representam um novo caso a ser classificado.

Para compreender melhor a representação da distribuição das probabilidades conjuntas da equação (1), é interessante lembrar que uma rede Naive Bayes é um Grafo Acíclico Direcionado (DAG), onde os nós representam as variáveis de interesse de um domínio e os arcos representam a dependência condicional entre as variáveis. Entre os nós, somente são permitidas ligações de dependência entre as classes e os atributos, como é mostrado na fig. 2.3. A força da dependência é representada pelas probabilidades condicionais que são associadas à ligação de cada nó pai-filho [CAR 99a].



Como as variáveis do grafo devem ser condicionalmente independentes de todos os nós que não são seus pais, pode-se derivar um procedimento recursivo para construção da rede Naive Bayes. Dada a distribuição da probabilidade conjunta da equação (1), inicia-se a construção do grafo escolhendo como nó-raiz a classe  $C_1$ , atribuindo a ela a probabilidade a priori  $P(C_1)$ . Em seguida acrescenta-se mais um nó  $A_1$  no grafo. Como  $A_1$  será dependente de  $C_1$ , então traça-se um arco ligando ambas as variáveis com a seta apontada para  $A_1$  e quantifica-se este arco com a probabilidade condicional  $P(A_1|C_1)$ , visualizada na (fig. 2.4). Repete-se esta operação para as demais variáveis de interesse (atributos) e, então, obtém-se uma rede Naive Bayes, cujo grafo está representado na fig. 2.3.

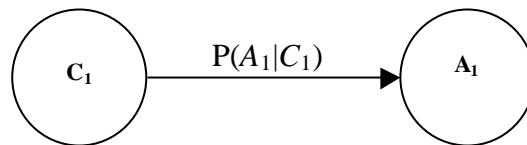


FIGURA 2.4 - Representação da dependência condicional  $P(A_1|C_1)$  no grafo

Se o processo for observado na ordem inversa, percebe-se que as probabilidades condicionais dos arcos contêm todas as informações necessárias para a reconstrução da função da distribuição original da equação (1).

O termo classificador Naive Bayes foi citado primeiramente por Kononenko [KON 90]. Outros nomes foram incluídos para a estrutura, como, por exemplo, *Classificador Bayesiano Simples* por Langley [LAN 93].

A rede Naive Bayes tem sido usada por pesquisadores, em classificações há muitos anos, por apresentar características vantajosas sobre outros tipos de classificadores, tais como:

- ◆ **facilidade na construção de seu algoritmo:** pela simplicidade do seu algoritmo, estimulou muitos pesquisadores a aplicar este método em muitas ferramentas [CHE 97]. Detalhes do algoritmo será apresentado na seção 2.5.1;
- ◆ **o processo de classificação é muito eficiente quando os atributos são independentes entre si:** em situações onde os atributos não são correlacionados, o classificador Naive Bayes sobressai surpreendentemente sobre muitos sofisticados classificadores. Esta característica é rara na prática de aprendizagem. Isso ocorre porque a rede Naive Bayes apresenta uma limitação nas ligações entre os nós (ver fig. 2.3). Quando ocorre dependência entre os atributos, torna-se inadequada para o uso do Naive Bayes. Por exemplo, um nome de um peixe pode sofrer predição pelas características de peso e tamanho. Dado um peixe de uma particular espécie, seu peso obviamente depende consideravelmente do seu tamanho e vice-versa. Neste caso, o uso do classificador Naive Bayes se torna inviável;
- ◆ **é muito rápido na aprendizagem e predição:** seu tempo de aprendizagem e de predição é linear independentemente do número de exemplos [BIL 96]. Muitos pesquisadores têm desenvolvido trabalhos, comparando o desempenho do classificador Naive Bayes com outros modelos de classificação complexos. Entre os trabalhos, destacam-se os de [FRI 97], [KOH 97], [KON 97] e [LAN 92].

### 2.5.1 O Algoritmo Naive Bayes

Classificadores Naive Bayes são programas de indução que nomeiam uma classe de um conjunto pré-definido de casos, os quais são representados por atributos valorados que têm a função de descrever cada caso do conjunto de casos. Esses classificadores, usando conceitos probabilísticos, calculam probabilidades condicionais das classes, dados os atributos, para então prever a classe mais provável, pelo maior valor probabilístico inferido.

Para facilitar a compreensão do algoritmo, considere que um sistema contém duas classes e quatro atributos. Este sistema terá a função de classificar uma classe que determinará se uma pessoa jogará ou não tênis (valores do domínio do conjunto de classes  $C$ ), em função das condições atmosféricas [ENG 2000], representadas pelos atributos. Na tab. 2.2 são apresentadas as duas classes do sistema.

TABELA 2.2 - Conjunto de classe do sistema Jogo de Tênis

$c_i$	Dom(C)=Jogo de Tênis
1	Joga
2	Não Joga

O conjunto de atributos, responsáveis em caracterizar cada caso e prever as classes, está detalhado na tab. 2.3:

TABELA 2.3 - Conjunto de atributos que caracterizam as condições atmosféricas dos casos

J	$\Omega = \text{Dom}(A_j)$	$\mathbf{w} = (a_{i_1}^{(1)}, \dots, a_{i_n}^{(n)})$
1	Tempo	Ensolarado, encoberto, chuva
2	Temperatura	baixa=até 17°C;média=acima de 17° até 25°C;alta=acima de 25°C
3	Umidade	baixa=até 70%;média=acima de 70% até 85%;alta=acima de 85%
4	Ventoso?	Sim ou Não

Como um valor de um atributo é representado por  $a_i^{(j)}$ , onde  $i$  pode variar de 1 a  $m$ , um caso pode ser descrito por uma instânciação  $\mathbf{w} = (a_i^{(1)}, \dots, a_i^{(j)})$  para os atributos  $A_1, \dots, A_n$ , que determina uma classe  $c_i$  do conjunto de classes  $C$ . Por exemplo, os atributos valorados Tempo<sub>ensolarado</sub>, Temperatura<sub>média</sub>, Umidade<sub>baixa</sub>, Ventoso?<sub>sim</sub>, pode-se determinar que o jogador tem condições atmosféricas para jogar tênis (classe  $c_1$ ).

Através do exemplo Jogo de Tênis, com o objetivo de exemplificar uma predição do classificador, cria-se um novo caso, cujos valores estão apresentados na tab. 2.4.

TABELA 2.4 - Novo caso para sofrer predição pelo Classificador Naive Bayes

Tempo	Temperatura	Umidade	Ventoso?	Classe
Chuva	Baixa	Baixa	Não	?

O classificador, como seu nome indica, utiliza o teorema de Bayes

$$P(X | Y) = \frac{P(Y | X) \cdot P(X)}{P(Y)}, \quad (2)$$

onde  $X$  e  $Y$  são eventos. As probabilidades condicionais na fórmula do classificador Naive Bayes são calculadas de forma inversa, ou seja, para chegar aos valores  $P(X|Y)$  deve-se saber as probabilidades condicionais de  $P(Y|X)$ , conforme foi demonstrado nas seções anteriores. Sendo assim, a decomposição da equação (2) será:

$$P(C = c_i | A_1 = a_{i_1}^{(1)}, \dots, A_n = a_{i_n}^{(n)}) = \frac{P(A_1 = a_{i_1}^{(1)}, \dots, A_n = a_{i_n}^{(n)} | C = c_i) \cdot P(C = c_i)}{P(A_1 = a_{i_1}^{(1)}, \dots, A_n = a_{i_n}^{(n)})} \quad (3)$$

O denominador da equação (3),  $P(A_1 = a_{i_1}^{(1)}, \dots, A_n = a_{i_n}^{(n)})$  utiliza probabilidades para expressar os atributos  $A_j$ , considerando que, neste exemplo (Joga Tênis), o valor do denominador sempre conterá um valor fixo para todo o processo de classificação de um determinado caso, ou seja, o valor calculado para o denominador será usado para ambos os cálculos das probabilidades condicionais das classes de  $C$ . Deste modo, o valor do denominador poderá ser descartado porque não influenciará nos resultados finais da classificação. Assim, utiliza-se apenas o numerador da fórmula (3) para calcular a proporção da probabilidade das classes de  $C$  e em seguida fazer uma comparação relativa entre os valores das classes, para escolher a classe que apresentar o maior valor.

Para que seja possível calcular a equação (3), os valores das probabilidades condicionais deverão ser calculados com base no conjunto de exemplos de casos pré-classificados. Definiu-se um conjunto de 14 casos pré-classificados escolhidos por um especialista do assunto, com o objetivo de inferir as classes para novos casos que contenham atributos valorados de condições atmosféricas. Estes casos estão apresentados na tab. 2.5.

Neste caso é necessário usar um algoritmo de *maximum likelihood estimation* (máxima estimação de probabilidade) [BOR 99], cuja fórmula é representada na equação (4):

$$\hat{P}(A_j = a_{i_j}^{(j)} | C = c_i) = \frac{\#(A_j = a_{i_j}^{(j)}, C = c_i)}{\#(C = c_i)} \quad (4)$$

Este algoritmo tem como objetivo “aprender” as distribuições probabilísticas de uma rede e identificar a interdependência entre os arcos [HRU 97]. Como a estrutura da rede está pré-definida pelas probabilidades condicionais  $P(A_1 = a_{i_1}^{(1)}, \dots, A_n = a_{i_n}^{(n)} | C = c_j)$ , resta fazer o aprendizado dos parâmetros numéricos, onde  $\#(C = c_i)$  é um número de exemplos de casos no conjunto pré-classificado, pertencentes à classe  $c_i$ . Como o exemplo apresenta apenas duas classes, o valor de  $\#(c_1 = \text{Joga})$  será de nove ocorrências desta classe, no conjunto de casos pré-classificados. Para a classe (Não Joga) o valor de  $\#(c_2 = \text{Não Joga})$  será de cinco ocorrências, conforme a tab. 2.5. Como

o total de casos é igual a quatorze, o cálculo da probabilidade a *priori* das duas classes será igual a:  $P(\text{Joga})=0.64$  ou 64% e  $P(\text{Não Joga})=0.36$  ou 36%.

TABELA 2.5 - Conjunto de exemplo de casos selecionados por um especialista

Caso	Tempo	Temperatura	Umidade	Ventoso?	Classe
1	Ensolarado	Média	Baixa	Sim	Joga
2	Ensolarado	Alta	Alta	Sim	Não joga
3	Ensolarado	Alta	Média	Não	Não joga
4	Ensolarado	Média	Alta	Não	Não joga
5	Ensolarado	Média	Baixa	Não	Joga
6	Encoberto	Média	Alta	Sim	Joga
7	Encoberto	Alta	Média	Não	Joga
8	Encoberto	Baixa	Baixa	Sim	Joga
9	Encoberto	Alta	Alta	Não	Joga
10	Chuva	Média	Média	Sim	Não joga
11	Chuva	Média	Baixa	Sim	Não joga
12	Chuva	Média	Média	Não	Joga
13	Chuva	Alta	Média	Não	Joga
14	Chuva	Média	Alta	Não	Joga

Note que o somatório das duas probabilidades a *priori* totaliza-se 1. Após serem calculadas as ocorrências das classes no conjunto de exemplos de casos, passa-se para o cálculo de  $\#(A_j = a_{i_j}^{(j)}, C = c_i)$  da fórmula (4). Para chegar a um valor a partir desta expressão, deve-se contar o número de exemplos de casos pertencentes à classe  $c$  e tendo o valor  $a_{i_j}^{(j)}$  para o atributo  $A_j$ . Por exemplo, o atributo (Tempo) apresenta três valores simbólicos (Ensolarado, Encoberto e Chuva). Dentro do domínio de ocorrência da classe (Joga) o valor do atributo (Tempo=Ensolarado) ocorre duas vezes. Assim, o valor da probabilidade condicional  $P(\text{Tempo}_{\text{Ensolarado}}/c_1=\text{Joga})$  será de  $\frac{2}{9} = 0.22$  ou 22%.

Este cálculo deve-se repetir para todos os valores dos 4 atributos do exemplo Jogo de Tênis. Considerando que todos os fatores atmosféricos (atributos) são condicionalmente independentes dado as classes, os parâmetros numéricos do sistema deverão dispor de duas probabilidades a *priori* das classes e de 22 probabilidades condicionadas ou *posteriori*, cujos valores já calculados estão apresentados na tab. 2.6.

TABELA 2.6 - Valores das probabilidades a priori  $P(c_i)$  e probabilidades condicionais dos atributos  $A_j$  dado as classes  $c_i$ 

<b>i</b>	<b><math>c_1=Joga</math></b>	<b><math>c_2=N\tilde{a}o Joga</math></b>
$P(c_i)$	9 ocorrências= 0.64	5 ocorrências=0.36
$P(\text{Tempo}_{\text{Ensolarado}} c_i)$	$\frac{2}{9}$ ocorrências = 0.22	$\frac{3}{5}$ ocorrências=0.6
$P(\text{Tempo}_{\text{Ensoberto}} c_i)$	$\frac{4}{9}$ ocorrências =0.45	$\frac{0}{5}$ ocorrências=0.03*
$P(\text{Tempo}_{\text{Chuva}} c_i)$	$\frac{3}{9}$ ocorrências =0.33	$\frac{2}{5}$ ocorrências=0.4
$P(\text{Temperatura}_{\text{Baixa}} c_i)$	$\frac{1}{9}$ ocorrências =0.11	$\frac{0}{5}$ ocorrências=0.03*
$P(\text{Temperatura}_{\text{Médial}} c_i)$	$\frac{6}{9}$ ocorrências =0.67	$\frac{3}{5}$ ocorrências=0.6
$P(\text{Temperatura}_{\text{Alta}} c_i)$	$\frac{2}{9}$ ocorrências =0.22	$\frac{2}{5}$ ocorrências=0.4
$P(\text{Umidade}_{\text{Baixa}} c_i)$	$\frac{3}{9}$ ocorrências =0.33	$\frac{1}{5}$ ocorrências=0.2
$P(\text{Umidade}_{\text{Médial}} c_i)$	$\frac{4}{9}$ ocorrências =0.45	$\frac{2}{5}$ ocorrências=0.4
$P(\text{Umidade}_{\text{Alta}} c_i)$	$\frac{2}{9}$ ocorrências =0.22	$\frac{2}{5}$ ocorrências=0.4
$P(\text{Ventoso?}_{\text{Sim}} c_i)$	$\frac{3}{9}$ ocorrências =0.33	$\frac{3}{5}$ ocorrências=0.6
$P(\text{Ventoso?}_{\text{N\tilde{a}o}} c_i)$	$\frac{6}{9}$ ocorrências =0.67	$\frac{2}{5}$ ocorrências=0.4

Caso não existirem exemplos de uma particular classe, no conjunto de dados, esta é simplesmente removida do domínio das classes.

Note que se um valor de atributo não ocorrer dado qualquer classe, esta probabilidade também é fixada em  $\frac{1}{2N}$ , onde N é o número de casos de exemplos [BOR 99][FEL 98].

Isso pode ser visto observando a linha três e cinco da tab. 2.5, cujos valores foram fixados em  $\frac{1}{2*14}=0.03$  por não apresentarem ocorrências de atributos, dada uma determinada classe.

Agora que foram calculados os valores das probabilidades condicionais e a *a priori*, pode-se voltar à predição do novo caso da tab. 2.4. O classificador usará como dados de entrada, além do novo caso, as probabilidades condicionais e a *a priori* da tab. 2.6.

Antes de apresentar o cálculo do numerador da equação (3), é preciso fazer algumas considerações a respeito de como serão computados os valores das probabilidades condicionais das classes, que darão a predição para o novo caso:

- ◆ deve-se aplicar a equação (3), para todas as classes do exemplo (Joga Tênis), ou seja, calcular a probabilidade condicional da classe ( $c_1$ =Joga), dado os valores dos atributos do novo caso e deve ser calculada a probabilidade condicional da classe ( $c_2$ =Não Joga), dados, também, os valores dos atributos do novo caso. Isso pode ser compreendido facilmente observando essas duas expressões de probabilidades condicionais:
  - a)  $P(c_1=\text{Joga}|A_1=\text{Tempo}_{\text{chuva}}, A_2=\text{Temperatura}_{\text{baixa}}, A_3=\text{Umidade}_{\text{baixa}}, A_4=\text{Ventoso?}_{\text{não}})$
  - b)  $P(c_2=\text{NãoJoga}|A_1=\text{Tempo}_{\text{chuva}}, A_2=\text{Temperatura}_{\text{baixa}}, A_3=\text{Umidade}_{\text{baixa}}, A_4=\text{Ventoso?}_{\text{não}})$ ;
- ◆ o numerador da fórmula do classificador na equação (3) é gerado a partir de um importante ingrediente do classificador Naive Bayes, a independência condicional. Define-se que dado o valor do atributo, qualquer atributo  $A_j$  é independente de todos os outros atributos. Assume-se que para conhecer a classe é suficiente apenas conhecer a probabilidade de um valor  $a_i^{(j)}$ . Assim não necessitaria conhecer os valores de todos os outros atributos, por causa da independência que existe entre eles. Este fato simplifica de modo considerável o estado do numerador da equação (3).

Utilizando os valores dos atributos e classes contidos nas expressões (a e b), pode-se agora passar para a decomposição do numerador:

$$P(A_1 = a_{i_1}^{(1)}, \dots, A_n = a_{i_n}^{(n)} | C = c_i).P(C = c_i) \quad (7)$$

Decompondo a equação (7), obtendo o seguinte produtório das probabilidades condicionais e a *a priori*:

$$\begin{aligned} &= P(A_1 = a_{i_1}^{(1)} | C = c_i) && (8) \\ &.P(A_2 = a_{i_2}^{(2)} | C = c_i) \\ &\dots \\ &.P(A_n = a_{i_n}^{(n)} | C = c_i) \\ &.P(C = c_i). \end{aligned}$$

A equação (7) é a base da fórmula fundamental do classificador Naive Bayes. Para um atributo simbólico  $A_j$  a probabilidade condicional  $P(A_j = a_i^{(j)} | C = c_i)$  é correspondente à armazenada na tabela de probabilidades condicionais (parâmetros numéricos) (tab. 2.6).

Substituindo os valores da tab. 2.6 na equação (7), aplicadas nas duas expressões (a) e (b), obtêm-se os seguintes valores para os numeradores, (tab. 2.7):

TABELA 2.7 - Cálculo do numerador da equação (2) do classificador Naive Bayes

Expressão	Equação (6)	Substituição pelos valores da tab. 2.5
(a)	$P(c_1=\text{Joga}   \text{Tempo}_{\text{chuva}})^*$	0.33*
	$P(c_1=\text{Joga}   \text{Temperatura}_{\text{baixa}})^*$	0.11*
	$P(c_1=\text{Joga}   \text{Umidade}_{\text{baixa}})^*$	0.33*
	$P(i_1=\text{Joga}   \text{Ventoso?}_{\text{não}})^*$	0.67*
	$P(c_1=\text{Joga})=$	0.64=0.0051365
(b)	$P(c_2=\text{Não Joga}   \text{Tempo}_{\text{chuva}})^*$	0.4*
	$P(c_2=\text{Não Joga}   \text{Temperatura}_{\text{baixa}})^*$	0.03*
	$P(c_2=\text{Não Joga}   \text{Umidade}_{\text{baixa}})^*$	0.2*
	$P(c_2=\text{Não Joga}   \text{Ventoso?}_{\text{não}})^*$	0.4*
	$P(c_2=\text{Não Joga})=$	0.36=0.0003456

Para o novo caso da tab. 2.4 , o processo de classificação, utilizando o método Naive Bayes, computou os valores para as classes de  $C$ , conforme pode ser observado na tab. 2.7. Note que para a classe ( $c_1=\text{Joga}$ ) apresentou valor 0.0051365 e a classe ( $c_1=\text{Não Joga}$ ) apresentou valor 0.0003456. Conclui-se que a predição computada pelo classificador Naive Bayes, para determinar se o jogador deva jogar ou não, dados as condições atmosféricas do novo caso, foi satisfatória para que o jogador deva jogar tênis por apresentar maior valor.

As redes das duas expressões, com sua distribuição probabilística e seus respectivos valores inferidos  $P(c_i)$ , são apresentados nas fig. 2.5 e fig. 2.6:

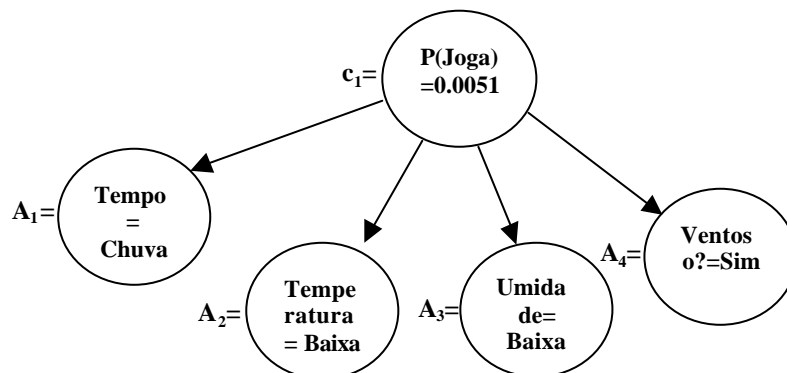


FIGURA 2.5 -Rede da distribuição de probabilidade conjunta da classe JOGA

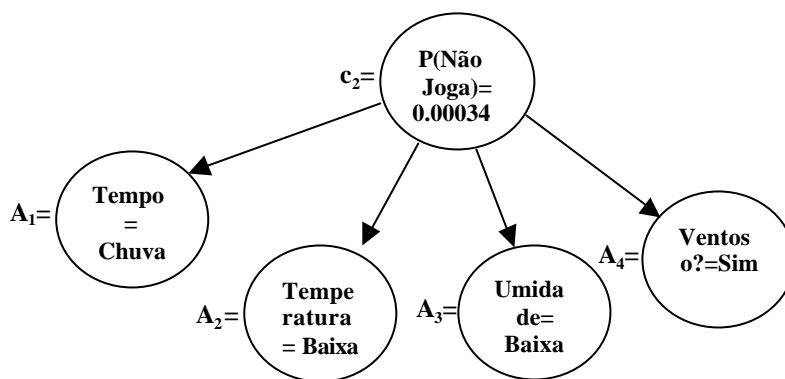


FIGURA 2.6 -Rede da distribuição de probabilidade conjunta da classe NÃO JOGA

## 2.5.2 Comentários Finais

Nos casos citados no exemplo, pode-se identificar as relações e as probabilidades condicionais para a tabela de parâmetros numéricos, mas podem ocorrer casos que não existem dados suficientes para determinar essas relações, e, assim, o trabalho do especialista se torna fundamental para definir as dependências e seus valores de probabilidades condicionais. Como foi citado nas seções anteriores, as relações de independência podem reduzir o esforço no cálculo da distribuição de probabilidade conjunta. Entretanto, nem sempre é possível a identificação de dependência entre as variáveis. Nestes casos usa-se o julgamento humano para definir essas probabilidades [CAR 99a]. Também pode-se usar o conhecimento humano com eficiência quando o domínio da aplicação é bem delimitado [HEC 98].

É interessante lembrar que existem fatos que podem gerar problemas nas predições do classificador Naive Bayes, como, por exemplo:

- ◆ trabalhar com valores com casas decimais. Os erros causados por arredondamentos podem causar variações na predição;
- ◆ os exemplos devem apresentar atributos com independência condicional, caso contrário, o Naive Bayes se torna inviável.

É evidente que, apesar desses fatores, o classificador Naive Bayes ainda se torna bem atraente para utilização em aplicações que envolvem predição. Sua facilidade de implementação e seu excelente desempenho, como será visto na próxima seção, colocam-no como um dos mais citados classificadores nas pesquisas na área de Inteligência Artificial.

## 2.6 Desempenho dos Classificadores

A área de aprendizagem de máquina fornece muitos algoritmos para classificar conjuntos de dados com o objetivo de analisar informações, reconhecer padrões e melhorar a precisão das predições. Preocupados com a qualidade dessas classificações, muitos pesquisadores têm examinado vários tipos de classificadores, observando sua aplicabilidade e sua precisão para classificar um conjunto de dados de domínios específicos. Segundo Kohavi [KOH 97], os resultados mostram que não há um único classificador que pode ser mais preciso do que outros em todos os domínios. Sabe-se muito pouco sobre qual dos algoritmos deverá ser escolhido para um problema



específico. Alguns critérios devem ser levados em considerações para medir a qualidade de um classificador, segundo [HEN 94] e [KOH 97]:

- ◆ **precisão:** é o grau de confiabilidade das regras, ou a precisão das predições feitas sobre uma instância. Geralmente é representado pela taxa de erro, embora em alguns casos existam erros mais sérios que outros, e em certos casos pode ser importante controlar a taxa de erro em algumas classes chaves. Por exemplo, se um cliente será capaz de pagar um empréstimo. Usando métodos como *holdout*, *bootstrap*, e *cross-validation* (Weiss & Kulikowski 1991, Efrom & Tibshirani, Kohavi 1995) citado por [HEN 94], a pessoa pode calcular bem a precisão de predição futura em dados não vistos na prática. Segundo [HEN 94], deve ter em mente que avaliar a precisão em conjunto de dados treinados e avaliar precisão em um conjunto de dados de teste, onde a classificação é desconhecida, são, muitas vezes, bem diferentes e decepcionantes. Isso ocorre porque para conjunto de dados treinados o valor da precisão pode ser perfeitamente ajustado, mas para conjunto de teste pode ser muito difícil. Geralmente, a precisão é aplicada em conjunto de dados de testes, onde a verdadeira classificação é desconhecida, o que de fato é relevante. O método geralmente aceito para calcular a precisão seria usar um determinado conjunto de dados, onde todas as associações de classes são conhecidas. Parte desse conjunto é selecionada randomicamente e definida como conjunto de dados treinados, sendo usado para treinar o procedimento. Esta regra é, então, testada nos dados restantes (conjunto de teste), e os resultados comparados com as classificações conhecidas. As proporções corretas do conjunto de testes é uma estimativa parcial da precisão da regra;
- ◆ **velocidade:** em algumas circunstâncias a velocidade de um classificador é o fator mais importante. Um classificador que é 90% preciso pode ser preferido a outro que é 95% preciso se ele for 100 vezes mais rápido nos testes. Tais considerações podem ser importantes para aplicações que possuem sérias restrições de tempo nas suas tarefas;
- ◆ **compreensibilidade:** se é um operador humano que deve aplicar o procedimento de classificação, o procedimento e os dados devem ser facilmente compreensíveis ou senão enganos podem ser feitos na aplicação das regras;
- ◆ **tempo de aprendizado:** especialmente em um ambiente que muda rapidamente, pode ser necessário aprender uma regra de classificação rapidamente, ou fazer ajustes a uma regra existente em tempo real. “Rapidamente” pode implicar também que seja necessário somente um pequeno número de observações para estabelecer a regra;
- ◆ **treinamento e tempo de classificação:** o tempo é tomado para classificar versus o tempo de treinamento. Alguns classificadores, como nas redes neurais, são rápidos para classificar mas lentos para treinar. Outros classificadores, como *nearest-neighbor* e outros algoritmos são, geralmente, rápidos para treinar mas lentos na classificação.

Resultados de pesquisas mostram que classificadores simples podem ter bom desempenho quando a medida de desempenho é esperada pela precisão de generalização do classificador. Na prática, o usuário de ferramentas de classificação se preocupa com a precisão, eficiência e compreensibilidade para especificar um domínio. Em certos domínios, um grupo de classificadores pode apresentar um bom desempenho

comparado com outros tipos de classificadores. Por exemplo, em domínio médico, muitos atributos descritos por médicos durante anos têm características independentes.

Conclui-se que, através de um simples algoritmo de aprendizagem, não se pode apresentar uma boa classificação para todas as situações. Estes resultados de desempenho tornam-se claros quando é possível medir a precisão dos classificadores em vários domínios. Esta necessidade motivou o desenvolvimento de muitas ferramentas que permitem medir o desempenho de vários classificadores em diversos domínios. Por exemplo, *MLC++* [KON 97] é uma biblioteca de aprendizagem de máquina com classes em C++, gratuita e de código aberto, que foi desenvolvida para ajudar a escolher um algoritmo de classificação apropriado para um dado domínio, facilitando a comparação de diferentes algoritmos num determinado domínio de interesse (disponível em [www.sgi.com/Technology/mlc](http://www.sgi.com/Technology/mlc)). Outra ferramenta que apresenta as mesmas características é a *WEKA* (*Waikato Environment for Knowledge Analysis*), é uma biblioteca de aprendizagem de máquina com classes em Java, onde se encontram muitos algoritmos, inclusive com interfaces para algoritmos externos, disponível gratuitamente em <http://www.cs.waikato.ac.nz/~ml/project.html>.

### **2.6.1 Resultados de Desempenho do Classificador Naive Bayes**

Vários autores demonstram o bom desempenho do Classificador Naive Bayes sobre outros classificadores. Nesta seção serão retratados alguns resultados dessas pesquisas, dado um determinado domínio de dados e o grau de independência dos atributos.

O Friedman [FRI 97] demonstra a influência das grandes independências dos atributos e as variações do tamanho dos conjuntos de atributos que influenciam no desempenho dos classificadores. Primeiramente, comparando o classificador Naive Bayes com uma rede Bayesiana, o experimento rodou 25 conjuntos de dados e como resultado a rede Bayesiana apresentou melhor precisão que o Naive Bayes em seis conjuntos de dados, mas teve seu desempenho baixo em outros seis conjuntos de dados. Examinando o gráfico da fig. 2.7, onde os valores verticais definem os percentuais de erros e os horizontais a quantidade de atributos por conjunto de dados, percebe-se que nos conjuntos onde apresentavam mais que quinze atributos, a rede Bayesiana apresentou baixo desempenho.

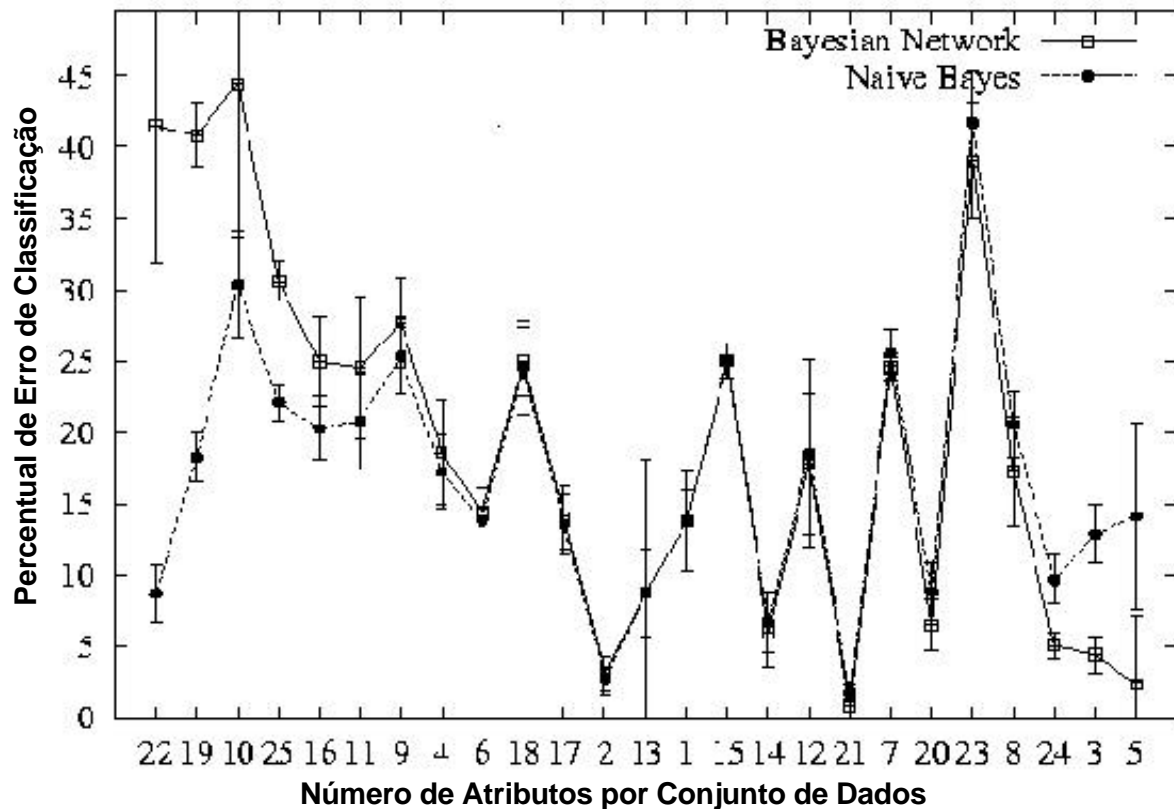


FIGURA 2.7 - Comparativo do percentual de erro de classificação do classificador Naive Bayes e a rede Bayesiana

(\*) O eixo x é ordenado de tal forma que as linhas se cruzam apenas uma vez

Fonte [FRI 97]

A tab. 2.8 mostra a comparação de vários classificadores discutidos em [FRI 97], onde se utilizam abreviações como: Classificador Naive Bayes (NB); redes Bayesianas (RB); Extensão do classificador Naive Bayes, onde ocorre relaxamento da independência dos atributos (TAN); classificador *multinet* (CL); classificador de estrutura em árvore de decisão [C4.5] desenvolvido por [QUI 93]; Seletivo classificador Naive Bayes [SBN], onde se selecionam atributos para o classificador Naive Bayes, usado por [KON 97].

TABELA 2.8 - Média de precisão de cada classificador aplicado nos conjuntos de dados

Data Set	NB	BN	TAN	CL	C4.5	SNB
Australian	86.23	86.23	84.20	85.07	85.65	<b>86.67</b>
Breast	<b>97.38</b>	96.92	96.92	97.07	94.73	96.19
Chess	87.15	95.59	92.31	92.40	<b>99.53</b>	94.28
Cleve	<b>83.76</b>	81.39	81.76	80.73	73.31	78.06
Corral	85.88	97.60	96.06	<b>99.23</b>	97.69	83.57
Crx	<b>86.22</b>	85.60	85.76	<b>86.22</b>	<b>86.22</b>	85.92
Diabetes	74.48	75.39	75.52	74.74	<b>76.04</b>	<b>76.04</b>
Flare	79.46	82.74	82.27	82.18	82.55	<b>83.40</b>
German	<b>74.70</b>	72.30	73.10	73.90	72.20	73.70
Glass	69.66	55.57	67.78	70.58	69.62	<b>71.98</b>
Glass2	<b>79.17</b>	75.49	77.92	<b>79.19</b>	76.67	<b>79.17</b>
Heart	81.48	82.22	<b>83.33</b>	82.22	81.11	81.85
Hepatitis	91.25	91.25	91.25	<b>92.50</b>	86.25	90.00
Iris	93.33	<b>94.00</b>	<b>94.00</b>	93.33	<b>94.00</b>	<b>94.00</b>
Letter	74.96	75.02	85.86	<b>88.02</b>	77.70	75.36
Lymphography	79.72	75.03	<b>85.03</b>	81.75	77.03	77.72
Mofn-3-7-10	86.46	85.94	91.11	<b>91.50</b>	85.55	87.50
Pima	<b>75.51</b>	75.00	<b>75.52</b>	75.39	75.13	74.86
Satimage	81.75	59.20	<b>87.20</b>	87.00	83.15	82.05
Segment	91.17	93.51	<b>95.58</b>	93.12	93.64	93.25
Shuttle-small	98.34	99.17	<b>99.53</b>	99.02	99.17	99.28
Soybean-large	91.29	58.54	92.17	91.46	92.00	<b>92.89</b>
Vehicle	58.28	61.00	69.63	67.15	<b>69.74</b>	61.36
Vote	90.34	94.94	93.56	94.71	<b>95.63</b>	94.71
Waveform-21	77.89	69.45	<b>78.38</b>	<b>78.36</b>	74.70	76.53

Fonte: [FRI 97]

Os valores da tab. 2.8 mostram que o classificador Naive Bayes apresenta melhor precisão em comparação à rede Bayesiana quando os conjuntos de dados apresentam vários atributos. Mostra-se também que o Naive Bayes apresenta uma boa precisão com relação ao C4.5 e o SNB na maioria dos conjuntos de dados. Já com relação ao TAN, percebeu-se que nos conjuntos de dados onde se caracterizava uma certa dependência entre os atributos, o Naive Bayes não apresentou bom desempenho.

Em [DOM 96], mostra-se o surpreendente desempenho do classificador Naive Bayes, como pode ser observado na tab. 2.9. A comparação foi entre os classificadores Naive Bayes (SBC), *Instance-based* (PEBLS), indução por árvores de decisão (C4.5), indução por regras (CN2) e um classificador *default* onde se assinala a mais freqüente classe para todos os exemplos testados. O teste foi feito processando vinte vezes cada conjunto de dados, selecionando randomicamente dois terços do conjunto de dados para treinamento e o restante para teste.

TABELA 2.9 - Média de precisão de cada classificador aplicado nos conjuntos de dados

Dominio	SBC	Default	C4.5	PEBLS	CN2
Audiology	73.9	21.3	72.5	<b>75.8</b>	71.0
Annealing	93.5	76.4	91.3	<b>98.7</b>	81.2
Breast cancer	68.7	67.6	<b>70.1</b>	65.8	67.9
Credits creening	<b>85.2</b>	57.4	85.0	81.3	82.0
Chess endgames	88.0	52.0	99.2	96.9	<b>98.1</b>
Pima diabetes	<b>74.4</b>	66.0	72.4	71.4	73.8
Echocardiogram	66.7	67.8	65.8	64.1	<b>68.2</b>
Glass	50.4	31.7	<b>66.1</b>	65.8	63.8
Heart disease	83.1	55.0	74.2	79.2	<b>79.7</b>
Hepatitis	<b>81.2</b>	78.1	78.7	79.9	80.3
Horse colic	77.8	63.6	<b>83.6</b>	76.3	82.5
Thyroid disease	97.3	95.3	99.1	97.3	<b>98.8</b>
Iris	89.0	26.5	<b>93.4</b>	91.7	93.3
Labor neg.	<b>92.6</b>	65.0	79.7	91.6	82.1
Lung cancer	<b>46.4</b>	26.8	40.9	42.3	38.6
Liver disease	61.8	58.1	63.7	60.1	<b>65.0</b>
LED	<b>66.8</b>	8.0	61.2	55.3	58.6
Lymphography	81.5	57.3	75.3	<b>82.9</b>	78.8
Post-operative	61.8	<b>71.2</b>	70.2	58.8	60.8
Promoters	87.6	43.1	74.3	<b>91.7</b>	75.9
Primary tumor	<b>44.9</b>	24.6	35.9	30.9	39.8
Solar flare	68.0	25.2	<b>70.6</b>	67.6	70.4
Sonar	24.1	50.8	64.7	<b>73.3</b>	66.2
Soybean	<b>100.0</b>	30.0	95.0	<b>100.00</b>	96.9
Splice junctions	<b>95.4</b>	52.4	93.4	94.3	81.5
Voting records	91.2	60.5	<b>96.3</b>	94.9	95.8
Wine	90.9	36.4	91.7	<b>96.9</b>	90.8
Zoology	91.9	39.4	89.6	<b>94.6</b>	90.6

Fonte: [DOM 96]

Os autores observaram nestes resultados que o classificador Naive Bayes apresentou bom desempenho mesmo em situações onde existe um certo grau de dependência entre os atributos.

Em [PAZ 97], mostra-se uma comparação de desempenho de vários classificadores (ID3, PEBLS, Decision Tree e Nearest Neighbor) com o Naive Bayes. Neste trabalho os classificadores foram usados a fim de medir a precisão de preferências de usuários que acessam páginas cujo assunto envolve Inteligência Artificial e Medicina. O teste consistiu em entrevistar um usuário interessado em IA e Medicina para criar uma lista de interesse e outra lista com páginas sem interesse localizada em <http://golgi.harvard.edu/biopages/all.html>. A lista de interesse contém 38 itens e a de sem interesse contém 82. Essas páginas foram usadas como treinamento e dados de testes para uma avaliação experimental. Para uma tentativa individual de um experimento, randomicamente selecionou-se um percentual de páginas para usar como conjunto de treinamento, e reservou o resto dos dados como conjunto de testes. Dos conjuntos de treinamento, foram encontrados 128 características, transformando o conjunto de treinamento como um vetor de características para ser usado no algoritmo de aprendizagem. Foram aplicados quatro algoritmos de aprendizagem. Os algoritmos criaram uma representação da preferência do usuário. Após o processamento dos algoritmos, os dados de teste foram convertidos para um vetor de características usando as características encontradas nos conjuntos de dados. Finalmente, o aprendizado de preferências de usuário foi usado para determinar se as páginas no conjunto de teste interessariam ao usuário. Para cada tentativa, gravou-se a precisão da preferência

aprendida (percentual de exemplos de testes por onde a preferência aprendida concordou com o interesse do usuário). Foram rodadas 50 tentativas para cada algoritmo em 7 diferentes tamanhos de conjunto de treinamento (10,15,25,35,50,75,100). A fig. 2.8 mostra a média de precisão de cada algoritmo em função dos números de exemplos treinados. É interessante observar a surpreendente superioridade do desempenho do classificador Naive Bayes sobre outros classificadores. Com 15 exemplos ele apresentou 84,5% de precisão; com 50 exemplos ele apresentou 89,1% de precisão e com 100 exemplos 91,2%, em contraste com outros algoritmos, onde nenhum excedeu os 78% de precisão. Um exemplo da superioridade do desempenho do Naive Bayes, quando comparado ao *Decision Tree*, é que o Naive Bayes examina todas características, enquanto que a árvore de decisão somente examina algumas características. O *Nearest Neighbor* também examina todas as características, entretanto ele não distingue entre a mais importante e a menos importante delas. Em contraste, com o Naive Bayes ele trata uma das características como mais importante que outras se a probabilidade condicional de uma característica dada à classe diferir substancialmente entre as classes [PAZ 97].

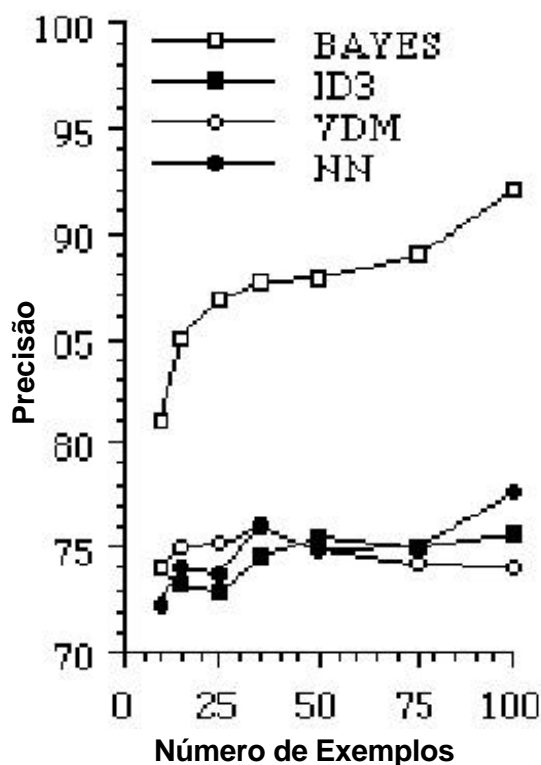


FIGURA 2.8 - Média de precisão de cada classificador predizendo as preferências de usuários de páginas da Web

Fonte: [PAZ 97]

## 2.7 Modelo de Usuário

As aplicações que efetuam buscas, classificações e aplicam regras inteligentes em banco de dados ou ambientes como a Web, utilizando sistemas inteligentes, necessitam ter conhecimento sobre o modelo do usuário, para que estas pesquisas resultem em dados relevantes aos interesses dos usuários. Para isso os sistemas inteligentes precisam ter conhecimento sobre o perfil e as preferências pessoais de cada usuário. Nesta seção serão abordado de forma rápida os principais conceitos, tipos e mecanismos de aquisição de conhecimentos sobre modelo de usuário.

### 2.7.1 Conceitos e Características

O modelo do usuário é um conjunto de informações de todos os aspectos relevantes do usuário para a conduta de um determinado sistema inteligente. O conteúdo do modelo pode variar desde idade, educação, nível de experiência e preferência do usuário. Muitas vezes estes modelos apresentam estruturas totalmente diferentes. Por estes motivos, [FIN 89] citado por [CAR 99b] classificou o modelo de usuário em vários tipos, como:

- ◆ **tipos de indivíduos:** os sistemas podem trabalhar com mais de um tipo de usuário, sendo necessária, portanto, a capacidade de modelar mais de um tipo de indivíduo;
- ◆ **vários modelos para cada usuário:** o sistema pode trabalhar com mais de um domínio, isto é, não considerar apenas um único domínio da aplicação. Sendo assim, o usuário poderá possuir mais de um modelo, tendo um domínio diferente para cada um;
- ◆ **grau de especialização:** um sistema pode trabalhar considerando o modelo como genérico ou individual. O modelo genérico assume um conjunto homogêneo de usuários, onde todos os indivíduos que utilizam o sistema são similares nos aspectos relevantes à aplicação, podendo ter um tratamento único. O modo individual contém informações específicas para um único usuário, tornando necessária, em sistemas que o utilizam, a existência de um modelo separado para cada um;
- ◆ **extensão temporal:** a extensão temporal refere-se ao tempo de duração do modelo, podendo ser de curto ou longo-prazo. Os modelos de curto prazo são construídos no início da execução do sistema e expiram ao final, isto é, ao final da sessão. Por sua vez, os modelos de longo-prazo caracterizam-se por serem mantidos de sessão para sessão;
- ◆ **modificabilidade:** de acordo com a modificabilidade, o modelo do usuário pode ser estático ou dinâmico, dependendo se é ou não modificado durante a sessão ou entre sessões.

O modelo do usuário para ser eficiente deve observar alguns requisitos básicos [FAO 2000]:

- ◆ **representar múltiplos interesses em um único usuário:** o modelo deve ser capaz de representar um usuário com vários interesses em diferentes assuntos, porque muitas vezes um usuário possui interesse em mais de uma área totalmente independentes umas das outras;

- ◆ **adaptar-se rapidamente às mudanças de interesse do usuário:** o modelo deve ser flexível ao ponto de adaptar-se às mudanças de interesse do usuário rapidamente, mesmo depois de um longo período de treinamento;
- ◆ **interagir com o usuário para ampliar o conteúdo do perfil:** o modelo deve levar em conta a interação do usuário para adicionar os conhecimentos, obtidos através desta interação, à base que armazena o perfil de cada usuário em cada tema.

### 2.7.2 Importância do Modelo

O modelo de usuário fornece informações que podem ser utilizadas para múltiplas finalidades, dependendo do tipo de sistema. Em seguida, são descritos alguns possíveis usos do modelo e ilustrados na fig. 2.9, [FIN 89] citado por [CAR 99b]:

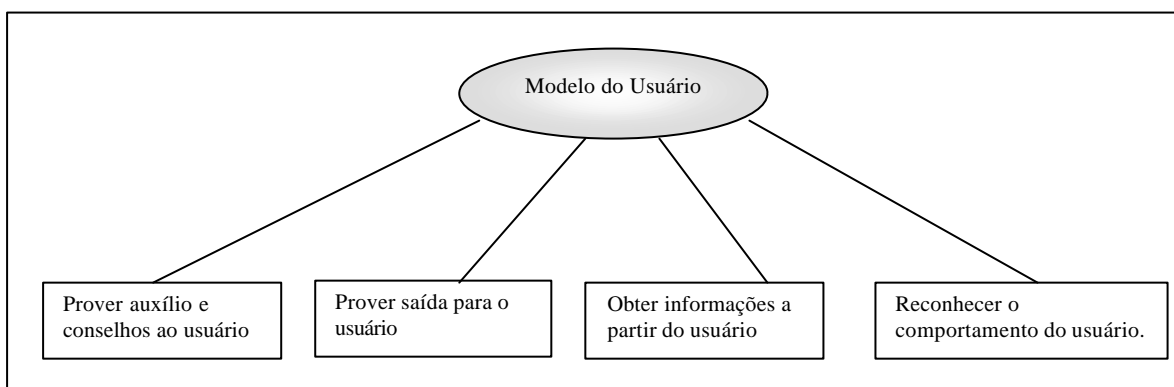


FIGURA 2.9 - Usos do Modelo de Usuário

Fonte: [CAR 99b]

- ◆ **prover auxílio e conselhos ao usuário:** o modelo fornece informações que permitem ao sistema avaliar o grau de importância do auxílio e o momento certo para tal;
- ◆ **prover retorno ao usuário:** o modelo fornece informações que permitem ao sistema decidir que resultados e como apresentá-los ao usuário;
- ◆ **obter informações a partir do usuário:** o modelo fornece informações que permitem ao sistema decidir o que e como perguntar e como interpretar as respostas do usuário;
- ◆ **reconhecer o comportamento do usuário:** o modelo fornece informações que permitem ao sistema reconhecer objetivos e planos do usuário.

### 2.7.3 Aprendizado do Modelo de Usuário

Para construir um modelo de usuário, deve ser aplicado um processo de aprendizado do modelo de perfil de cada usuário em diferentes áreas. Esse processo envolve mecanismos para aquisição do conhecimento sobre o usuário, a fim de gerar um modelo. Em alguns casos envolve também o processo de manutenção desses modelos.



[FAO 2000], descreve os mecanismos de aprendizagem mais relevantes entre a grande diversidade de mecanismos existentes nas bibliografias:

- ◆ **aprendizado a partir de amostras:** para a implementação do processo de aprendizado do modelo de usuário, as aplicações utilizam algoritmos que requerem conjuntos de exemplos positivos sobre os interesses dos usuários (“o usuário está interessado em ...”) e conjuntos de exemplos negativos, ou seja, assuntos em que o usuário não tem interesse (“o usuário não está interessado em ...”). A partir dos documentos submetidos como exemplos positivos e negativos, as aplicações procedem o aprendizado em dois passos: o primeiro é a avaliação das características de todos os documentos submetidos e o segundo é a aplicação de algoritmos de aprendizado sobre as características extraídas. Os conhecimentos obtidos pelo algoritmo de aprendizado, a partir dos exemplos submetidos, serão armazenados e servirão de base ao modelo de usuário e passarão a ser utilizados durante a execução do sistema inteligente, por exemplo, buscas de documentos na Web. Durante as execuções do sistema, o modelo pode ser incrementado aplicando-se os mesmos mecanismos citados acima sobre os documentos recuperados e classificados como positivos ou negativos através de interação com o usuário;
- ◆ **aprendizado a partir do histórico:** a utilização do histórico do consumo de informações pelos usuários é outra fonte de referência muito utilizada para a aquisição dos conhecimentos sobre as preferências pessoais e conseqüente formação do modelo de usuário. Por exemplo, no ambiente Web, este mecanismo utiliza como base para aquisição do conhecimento as páginas, os caminhos, os *links* e outros locais por onde o usuário já passou.

Outra abordagem interessante de técnicas para aquisição de conhecimento sobre o usuário a fim de construir um modelo foi feita por [FIN 89] e citada por [CAR 99b]:

- ◆ **aquisição explícita:** o projetista utiliza a aquisição explícita de conhecimento quando constrói estereótipos ou modelos genéricos de acordo com suas expectativas quanto aos usuários do sistema. Por sua vez, quando o usuário fornece informações diretas que descrevem seus objetivos e suas características particulares, através de questionários e/ou formulários, também se tem aquisição explícita;
- ◆ **aquisição implícita:** o modelo é construído através da observação do comportamento do usuário ao longo do uso do sistema. Com isto, o sistema torna-se hábil em inferir fatos sobre o usuário a partir de suas ações. Por esta razão, a aquisição de informações implicitamente é mais difícil.

No entanto, estas duas formas podem ser tratadas em conjunto no mesmo sistema. Para isso, muitos sistemas iniciam o processo de construção do modelo com a modelagem explícita, buscando do usuário suas características. Depois, o sistema passa a observar o usuário, acrescentando, ao longo da interação, novas informações ao seu modelo ([Bertoletti, 1997] citado por [CAR99b]).

A manutenção do modelo do usuário envolve incorporar novas informações ao mesmo, o que muitas vezes pode criar inconsistências. Neste sentido, manter o modelo significa atualizar e restaurar a consistência, se necessário ([Kass, 1988] citado por [CAR 99b]).

## 2.8 Ferramentas de Modelagem de Usuário para Web

Enquadram neste enfoque aquelas ferramentas que utilizam técnicas de modelagem para aprender o perfil de interesse do usuário sobre documentos trafegados na Web, a fim de organizá-los ou filtrá-los, fundamentados no perfil de interesse do usuário. Nesta seção serão apresentadas algumas aplicações que se enquadram nestas características e objetivos e ainda serão feitas comparações entre elas.

Existem muitas técnicas que podem ser usadas para filtrações de mensagens. Uma técnica é excluir a mensagem antes desta ser depositada na caixa de entrada do usuário. Isso ocorre geralmente no servidor de mensagens. Outra técnica é armazenar as mensagens indesejáveis em uma pasta para que ocasionalmente o usuário possa verificá-las a fim de confirmar a legitimidade da filtração. Em alguns casos, esta última técnica também pode ser usada para a organização de mensagens em pastas.

Em ambas as técnicas os filtros de mensagens permitem ao usuário filtrar informações de seu interesse. Isto ocorre porque esses sistemas contêm mecanismos que classificam informações tratadas pelos usuários e retiram resultados dos seus perfis que serão utilizados para as filtrações.

A maioria desses sistemas geradores de perfis de usuários está integrada com os filtros de mensagens. Essa forma de integração dificulta a interferência do usuário nas mudanças das regras mensuradas pelos classificadores de perfis, caso se desejar modificar algumas regras que não serão utilizadas por esses filtros. Porém facilita o usuário, evitando que o mesmo faça a tarefa de ajuste dos parâmetros dos filtros. Existem também sistemas de filtrações não integrados aos geradores de perfis, necessitando da interferência do usuário para ajustar os parâmetros de filtração. Neste caso a ferramenta apresenta um comportamento de assistente.

O papel do assistente melhora o trabalho do usuário nas mudanças dos parâmetros dos filtros de mensagens. Em [PAY 95b] mostra-se a importância do assistente nas ferramentas que utilizam o perfil de usuário, adquirindo o conhecimento observando e analisando o comportamento do usuário. Assistentes (*Feedback*) de usuário podem ser usados para refinar o perfil de um usuário, no início do desenvolvimento de um perfil. O usuário só consegue reconhecer certas situações quando recebe a ajuda de um assistente e poderia ter condições de anular decisões ou aprender a partir dos assistentes. Deste modo a interferência do usuário nas parametrizações das regras de filtração podem melhorar significativamente a precisão desses sistemas.

Alguns trabalhos mostram o uso das técnicas de modelos de usuários com a finalidade de assistir os mesmos na tarefa de organizar ou filtrar mensagens de correio eletrônico. Pode-se citar as principais, tais como:

- ◆ **filtração de *junk* E-mail:** Em [SAH 98] citado por [ITS 2001] apresenta-se um sistema que aplica filtração de mensagens não interessantes para os usuários. Esse filtro utiliza um classificador probabilístico usando apenas duas classes para classificação (mensagens interessantes e não interessantes). O sistema incorpora características específicas com as características do texto de cada mensagem. Características específicas são frases incompletas como “FREE!”, “only \$”, “be over 21”, características não textuais como o domínio de um remetente, o percentual de caracteres não alfanuméricos no campo assunto ou ainda a presença de um arquivo anexado. Para denotar a presença e a ausência de características específicas

são usadas variáveis adicionadas em um vetor para cada mensagem. As vantagens encontradas neste sistema é que os experimentos mostraram que a incorporação das características específicas melhoraram significativamente a precisão do sistema. Por outro lado, os próprios autores admitiram que na tentativa de executar o sistema utilizando três classes, os resultados mostraram-se com menores precisões. Uma desvantagem encontrada neste filtro é que os usuários que utilizarão o mesmo deverão se enquadrar nas regras impostas pelo sistema. Neste caso, o usuário não poderá aplicar o seu próprio perfil de seleção de mensagens com características para as duas classes;

- ◆ **Sortmail:** Em [CHA 99] descreve o *sortmail* que filtra mensagens dos usuários do sistema *Pine* do UNIX, fundamentado em regras específicas de usuários. O classificador Naive Bayes pode classificar várias mensagens em diferentes classes ou pastas. Para treinamento o *sortmail* utiliza um conjunto de mensagens rotuladas com as classes corretas, selecionadas através do próprio usuário que classifica um conjunto de mensagens em um grupo de pastas rotuladas. Uma vantagem observada para esta ferramenta é que quando ocorrem reorganizações de pastas, o *sortmail* treina rapidamente essas novas pastas;
- ◆ **IFile:** A ferramenta *iFile* [REN 2000] citada por [ITS 2001] classifica as mensagens em múltiplas categorias e as armazenam em pastas. O sistema é dinâmico porque sempre é alterado quando uma mensagem nova é movida para outra pasta. Todas as mensagens são adicionadas no conjunto de treinamento com rótulo assinalado pelo classificador ou, em caso de ser avaliado errado, pelo usuário. A simplicidade e a eficiência do classificador Naive Bayes permitem pouca interferência junto ao usuário. O critério utilizado pelo selecionador de características é a frequência de termos. Termos frequentes são guardados enquanto não frequentes são descartados. Isso fica mais evidente quando são acrescentadas mensagens novas ao conjunto de treinamento. Assim termos raros podem se tornar frequentes com a inclusão constante das mensagens. Experimentos apresentam valores de desempenho de 86% a 91% de precisão e o autor acredita que eles seriam até mais altos se, em vez de nomear um tópico, o sistema oferecesse a um usuário vários tópicos mais prováveis para se escolher. O uso da pasta de mensagens pode-se tornar o sistema de filtragem mais prático e interativo com o usuário. Por outro lado, sistemas de filtragens que utilizam pastas como classes, pode-se ocorrer que nem sempre o usuário consiga expressar qual assunto se refere a um determinado grupo de mensagens para nomear uma pasta. Isso pode dificultar o processo de treinamento do classificador. O usuário pode definir pastas com rótulos que não condizem com o assunto das mensagens;
- ◆ **Mailcat:** Em [SEG 99] mostra-se a ferramenta *Mailcat*, um agente de software do tipo assistente pessoal que, integrado ao Lotus Notes, reduz a carga cognitiva e o tempo para organização de mensagens nas pastas. Usando os classificadores de texto AIM da IBM que usa como conjunto de treinamento mensagens pré-existentes nas pastas para prever os destinos das mensagens da caixa de entrada. Assim a aplicação se adapta dinamicamente aos usuários, observando os seus hábitos ao arquivar mensagens, predizendo as três pastas que são mais comumente usadas e apropriadas para alojá-las. Fornecem botões de atalhos que facilitam o arquivamento das mensagens nas pastas preditas pelo *Mailcat*, simplificando o trabalho do usuário. No lugar de ter que escolher entre várias pastas, o usuário pode confirmar

uma das escolhas sugeridas pelo *Mailcat* e, além disso, ele pode expressar aquela confirmação com um único clicar do mouse;

- ◆ ***Maxims***: Em [PAT 94], apresenta-se o sistema *Maxims* que auxilia os usuários a desempenhar uma variedade de operações nas mensagens eletrônicas, como organizar em pastas, excluir ou responder. *Maxims* monitora cada interação entre o usuário e o *Eudora*, e armazena um registro de cada interação. O algoritmo MBR (Memory-based reasoning) é usado para antecipar as ações do usuário. MBR busca resultados entre a situação atual e situações previamente encontradas. *Maxims* prediz a ação, predita automaticamente ou provê um único atalho ao usuário facilitando aquela ação. As desvantagens apresentadas pelo autor é que ele pode tomar algum tempo para fazer seu algoritmo de aprendizagem ganhar bastante experiência para ser útil. *Maxims* tenta resolver este problema permitindo que um agente recentemente instanciado aprenda a partir de agentes estabelecidos a mais tempo. Porém, isto é duvidoso porque esquemas de arquivamentos são baseados em preferências individuais.

Através das ferramentas apresentadas, pode-se, agora, fazer um quadro comparativo entre elas. Os critérios para comparação escolhidos foram os objetivos de cada proposta, o método de classificação, as características principais e as vantagens e desvantagens. A tab. 2.10 apresenta um quadro de comparação fundamentado nestes critérios.

TABELA 2.10 - Quadro comparativo entre as ferramentas de modelos de usuários

Ferramenta	Método de Classificação	Característica da Ferramenta	Vantagens	Desvantagens	Objetivo
Filtragem de Junk E-mail	Métodos probabilísticos. Utilizam duas classes para classificação das mensagens (interessantes e não interessantes)	Incorpora características específicas do texto e características não textuais para classificação da mensagem.	O uso de características específicas melhorou o desempenho do sistema.	O usuário tem que se enquadrar nas regras impostas pelo sistema.	Filtragem de e-mails não interessantes
iFile	Método probabilístico (Naive Bayes), usam-se pastas para classificar mensagens.	Usa critério de frequência de termos do texto para rotular mensagens.	O uso de pastas de mensagens pode-se tornar o sistema de filtragem mais prático e interativo com o usuário.	O uso de pastas pode dificultar o processo de classificação, porque o usuário pode definir pastas com rótulos que não condizem com o assunto das mensagens.	Classifica as mensagens em categorias e as armazenam em pastas.
Mailcat	Usa o classificador de texto AIM da IBM e prediz as pastas destinos das mensagens contidas na caixa de entrada. Usam-se como treinamento mensagens pré-existent no correio eletrônico.	Usa critério de frequência de termos do texto para classificação das mensagens. É integrado ao Lotus Notes. Aprende observando a forma como o usuário arquiva as mensagens nas pastas	Boa precisão na classificação. O usuário não precisa aprender nada de novo para operar o sistema, usam-se somente botões. Por usar mensagens já arquivadas para treinamento, demanda pouco trabalho do usuário para uso da aplicação.		Ajuda o usuário a organizar as mensagens em pastas.
Sortmail	Método probabilístico (Naive Bayes), usam-se pastas para classificar mensagens.	Usa-se critério de frequência de termos do texto para classificação das mensagens.	Quando ocorre reorganização de pastas, a aplicação treina rapidamente o novo conjunto de dados.	Armazena todas as mensagens nas pastas, mesmos as indesejadas.	Filtra mensagens do Pine da Unix a fim de organizar as mensagens em pastas.
Maxims	O algoritmo MBR (Memory-based reasoning) é usado para antecipar as ações do usuário.	Monitora cada interação entre o usuário e o Eudora, e armazena um registro de cada interação. O classificador é usado para antecipar as ações do usuário, buscando resultados entre a situação atual e situações previamente encontradas.	Cria atalhos rápidos para agilizar ações do usuário.	Ele pode tomar algum tempo para fazer seu algoritmo de aprendizagem ganhar bastante experiência para ser útil.	Auxilia os usuários a desempenharem uma variedade de operações nas mensagens eletrônicas, como organizar em pastas, excluir ou responder.

## 2.9 O Direto

O *software* Direto [DIR 2001] foi sendo desenvolvido pela PROCERGS [PRO 2000] com a finalidade de ser utilizado como correio, agenda e catálogo corporativo. Foi construído utilizando a linguagem Java, sistema operacional *FreeBSD* e banco de dados

*PostGres*, entre outros. Além disso, o Direto é um *software* livre, isto é, licenciado nos termos do *GNU General Public License* [GNU 2000]. Nesta seção será apresentado o projeto Direto, assim como suas funcionalidades e tecnologias e em especial as justificativas para a implementação do modelo do assistente de *Feedback*, proposto neste trabalho.

### 2.9.1 Histórico

A PROCERGS atua desde 28 de dezembro de 1972 como órgão executor da política de informática do Estado. Desde 1992 a PROCERGS e a maioria de seus clientes do Estado trabalhavam com as ferramentas de correio eletrônico MEMO e Notes, com o objetivo de suprir o Estado com uma ferramenta adequada para comunicação entre os órgãos. Alguns problemas e dificuldades dos usuários com o uso destas ferramentas. Os pontos mais críticos foram [FAP 2000]:

- ◆ falta de uma ferramenta padrão de Correio Eletrônico para o Governo do Estado que facilitasse a comunicação interna e externa;
- ◆ convivência de duas ferramentas de Correio Eletrônico adotadas pelo Estado (Memo e Notes), uma com interface caractere e outra com interface gráfica;
- ◆ ferramentas de Correio Eletrônico existentes estavam sub-implementadas, pois alguns recursos do Notes como replicação, trabalho off-line e acesso remoto não foram implementados por problemas técnicos de segurança. No Memo, alguns recursos também não foram implementados;
- ◆ falta de conhecimento dos usuários das ferramentas utilizadas;
- ◆ correio Eletrônico Memo é visto como ferramenta com obsolescência tecnológica pelo tipo de interface;
- ◆ montagem de uma infra-estrutura de *hardware* e *software* para o Estado depende da disponibilidade de recursos financeiros elevados;
- ◆ familiaridade dos usuários do Estado com correio Internet e ferramentas de Correio Eletrônico como Exchange, Outlook, e Eudora;
- ◆ falta de integração entre as agendas do Memo e do Notes;
- ◆ a PROCERGS decidiu por desenvolver sua própria solução, materializada através do projeto Direto.

### 2.9.2 O Projeto Direto

O Direto é uma ferramenta corporativa que inclui serviços de correio eletrônico, agenda e catálogo. Baseou-se totalmente em protocolos-padrão Internet. O Direto é um produto modularizado que utiliza os protocolos *vCalendar* para agenda, *SMTP* e *IMAP* para o correio e *LDAP* para o catálogo. A fig. 2.10 mostra a arquitetura básica do Direto.

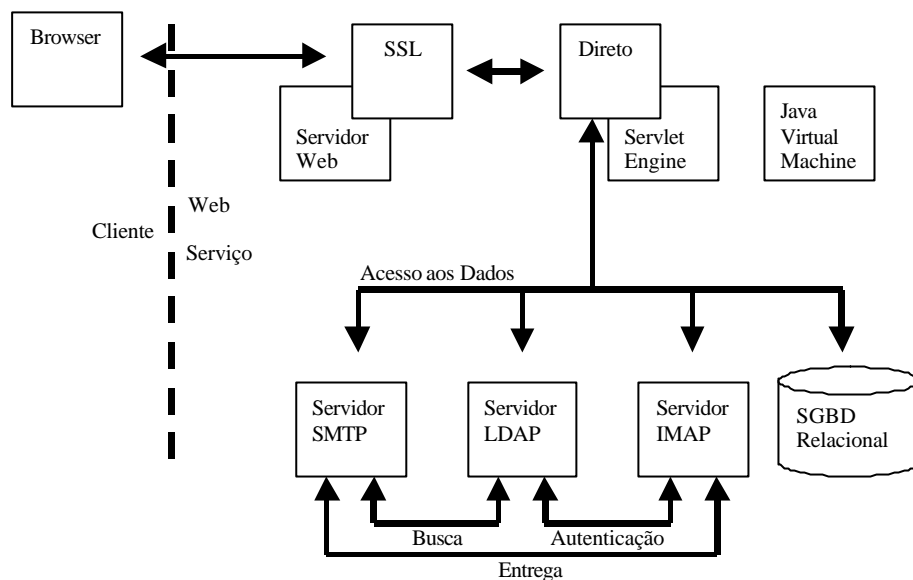


FIGURA 2.10 - Arquitetura do Direto

Fonte [TUT 01]

O objetivo do Direto é ser executado em ambiente Internet de forma que o usuário apenas necessite de um browser para utilização do produto. Todo o processamento é centralizado no servidor e o usuário precisa estar on-line para acessar os dados. A segurança dos dados pode ser garantida através da criptografia fornecida pelo protocolo *SSL*, suportado pelos principais browsers.

Sua estrutura modular apresenta código fonte disponível, tornando o Direto um *software* com grande adaptabilidade aos diversos ambientes, permitindo sua customização conforme as necessidades do usuário e que outros módulos sejam livremente desenvolvidos e incluídos ao produto.

A seguir são apresentados os módulos disponíveis para o usuário:

- ◆ **módulo Hoje** : é a página de abertura. Através dela o usuário tem a visão geral das mensagens e a listagem de compromisso do dia, serviços, notícias e uma seção de sugestões sobre o produto;
- ◆ **módulo Correio**: funciona como um serviço de correio eletrônico normal, com as facilidades da internet. Neste módulo o usuário pode ler, enviar, receber e encaminhar mensagens para qualquer contato, dentro ou fora da empresa. Especificamente neste módulo o protótipo desenvolvido neste trabalho será acoplado para servir de assistente para os usuários do Direto. A interface do correio está ilustrada na fig. 2.11.

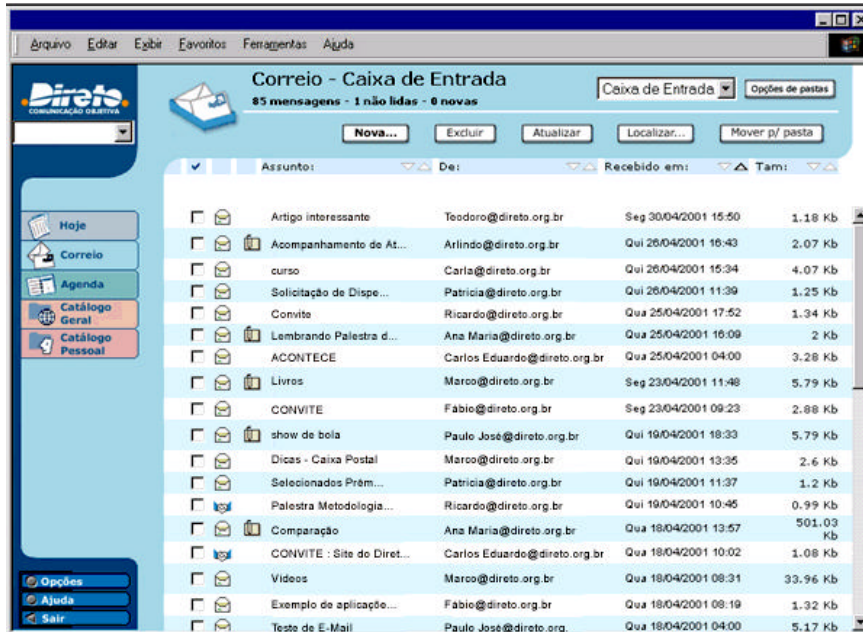


FIGURA 2.11 - Interface do módulo de Correio do Direto

- ◆ **módulo Agenda:** neste módulo, o usuário tem a oportunidade de agendar e acompanhar seus compromissos;
- ◆ **módulo Catálogo Geral:** é um catálogo de contatos da empresa, onde pode-se localizar os usuários previamente cadastrados;
- ◆ **módulo Catálogo Pessoal:** é o catálogo de contatos pessoais do usuário. Com ele é possível consultar informações contatos ou grupo de contatos.

### 2.9.3 A Tecnologia usada no Direto

O Direto utiliza várias tecnologias como serviços integrados, implementados por produtos de *software* livre. Esses produtos, por utilizarem protocolos Internet, podem ser substituídos sem que isso afete a solução.

O desenvolvimento do projeto foi motivado por alguns requisitos tecnológicos:

- ◆ seguir os protocolos-padrão Internet;
- ◆ apresentar independência de plataforma no cliente, bastando um browser Web para acessá-lo;
- ◆ prover independência de Gerenciador de Banco de Dados e do Servidor Web;
- ◆ ser baseado em *software* livre, diminuindo e viabilizando os custos do projeto;
- ◆ ser modularizado, permitindo que novas funcionalidades pudessem ser adicionadas no futuro.

Como a solução utiliza protocolos abertos, o acesso aos dados pode também ser disponibilizado para clientes de fornecedores que implementem esses protocolos. Outra consideração importante é a portabilidade da solução proposta. Qualquer um dos produtos envolvidos pode ser substituído sem alterar a estrutura geral do sistema.



Abaixo são apresentados os serviços e tecnologias utilizados no desenvolvimento do projeto Direto.

- ◆ **serviço de entrega de mensagens ( SMTP ):** O protocolo SMTP (*Simple Mail Transfer Protocol*), definido na RFC 821 ([Postel, 1982]), tem como objetivo transferir mensagens de forma confiável e eficiente. Uma das principais características do SMTP é a sua capacidade de transmitir mensagens entre ambientes diversos de serviços de transporte. O E-mail é uma aplicação de comunicação que utiliza o protocolo SMTP. O protocolo SMTP trabalha em conjunto com os protocolos de acesso a mensagens como IMAP e POP. O protocolo SMTP, pela sua eficiência e simplicidade, se torna uma opção extremamente vantajosa no que diz respeito a serviço de roteamento e entrega de mensagens;
- ◆ **serviço de acesso a mensagens (POP e IMAP):** Considerando que o computador esteja sempre disponível, o processo de armazenamento de mensagens é feito em um servidor. As mensagens chegam ao computador do destinatário com o uso de protocolos de acesso, como [BAL 2001]:
  - **POP3:** o protocolo POP foi projetado especificamente para o acesso off-line de mensagens, ou seja, o servidor de mensagens é apenas um depositário temporário das mensagens, devendo a manipulação das mensagens e pastas ficar sob responsabilidade do cliente (cliente POP);
  - **IMAP4 :** o protocolo IMAP (*Internet Message Access Protocol*), versão 4, permite que um cliente acesse e manipule mensagens em um servidor. Permite a manipulação de pastas de mensagens (“mailboxes”), de uma maneira funcionalmente equivalente à manipulação de pastas locais. IMAP4 também provê a capacidade de um possível cliente off-line ressincronizar suas mensagens com o servidor. O protocolo prevê funções como : criação, exclusão e renomeação de pastas; checagem de novas mensagens; remoção permanente de mensagens; pesquisa de atributos ou texto de mensagens.
- ◆ **serviço de diretório (LDAP):** Um ou mais servidores LDAP constituem uma árvore de diretórios. Um servidor LDAP é um servidor especializado para diretórios e não um tipo de banco de dados. A diferença de um servidor de diretório para um banco de dados genérico pode ser evidenciada pelo padrão de uso. Um diretório possui informações que são extensivamente pesquisadas, mas muito pouco atualizadas. Servidores LDAP são projetados para esse perfil de uso, enquanto um banco de dados relacional preocupa-se com a manutenção de forma geral dos dados;
- ◆ **serviço de agenda (vCalendar e iCalendar):** Define o formato da informação de agendamento e calendário, como, por exemplo, assunto de uma reunião, lista de convidados e data. Esse formato de dados permitirá, por exemplo, a troca de dados de um sistema de agendamento para outro na forma de um arquivo no formato vCalendar. O iCalendar, uma revisão do vCalendar, permite a captura e troca de informações entre aplicações de agendamento. O formato é definido como um tipo MIME o que permitirá que um objeto seja trocado entre sistemas por diversos transportes, incluindo SMTP, HTTP, um sistema de arquivos, protocolos de iteração desktop como drag-and-drop, entre muitos outros.

Além dos serviços representados por produtos de *software* livre, o Direto utilizou outras tecnologias como:

- ◆ **linguagem Java:** criada pelo grupo liderado por James Gosling na Sun Microsystems, é uma linguagem computacional completa, independente de plataforma e com uma série de facilidades para a integração com a Internet;
- ◆ **classes Java:** os componentes de negócio são constituídos de classes Java APIs que se comunicam com outros serviços para realizar determinada tarefa. As principais APIs utilizadas são:
  - **JavaMail:** é utilizada para a comunicação com os serviços de correio, acesso à caixa postal e serviço de transporte. O serviço de transporte utilizado é o SMTP, e a API é utilizada no envio de mensagens de correio eletrônico. No acesso à caixa postal, a API é utilizada para verificar mensagens novas, listar o cabeçalho de mensagens de determinada pasta, ler mensagens e excluir mensagens. Desta forma é necessário adicionar uma outra API, um *provider* específico para cada protocolo de acesso (POP3 e IMAP4);
  - **JNDI :** é utilizada para o acesso ao LDAP. A API JNDI é utilizada para consultas, inclusões e exclusões sobre a base LDAP do Direto;
  - **JDBC :** é utilizada para o acesso ao banco de dados. Trabalha em conjunto com o driver do banco de dados. É utilizada principalmente no módulo de agenda em consultas, inclusões e exclusões de registros;
- ◆ **Servlet:** devido à importância da geração de conteúdo dinâmico para o desenvolvimento da Web, foi natural que a Sun propusesse extensões para Java em seu domínio. Da mesma forma que a Sun introduziu *applets* como pequenas aplicações baseadas em Java para adicionar funcionalidade interativa aos navegadores da Web, em 1996 a Sun introduziu *servlets* como pequenas aplicações baseadas em Java para adicionar funcionalidade dinâmica a servidores Web;
- ◆ **XML:** XML (*eXtensible Markup Language*): é um padrão para publicação, combinação e intercâmbio de documentos multimídia, desenvolvido pelo consórcio W3C (*World Wide Web Consortium*). Assim como outras linguagens de marcação, XML lida com instruções embutidas no corpo de documentos chamadas tags, que permitem a descrição de dados. XML tem como base linguagens mais antigas como SGML e HTML, sendo atualmente empregada na representação de estruturas de dados estruturados e semi-estruturados e seu intercâmbio na Web;
- ◆ **XSL:** um documento XSL é um XML que consegue transformar um documento XML em outros formatos de documentos (HTML, TeX, PostScript, RTF), além de poder utilizar alguns elementos de estilo disponíveis em Cascading Style Sheets). A linguagem XSL pode ser utilizada para acrescentar aspectos de apresentação aos elementos de um documento XML. Desta forma, é possível criar múltiplas representações da mesma informação a partir de vários documentos XSL diferentes aplicados a um único documento XML;
- ◆ **licença do produto:** O Direto, além de utilizar *software* de distribuição livre, também é disponibilizado como um *software* livre. O Direto possui licença de *copyleft* e possui distribuição e licenças gratuitas, a partir do site do produto ([www.direto.org.br](http://www.direto.org.br)) mantido pela PROCERGS, seguindo as regras de *software* livre.

Atualmente o Direto está na versão 2.0. Esta versão possui várias melhorias em relação à versão anterior, como, por exemplo, o aprimoramento no seu desempenho, o

desenvolvimento de uma ferramenta para administração de usuários, a disponibilidade de novas funcionalidades, etc.

## 2.10 Considerações Finais

Este capítulo apresentou conceitos básicos sobre fundamentos da probabilidade, raciocínio probabilístico, redes bayesianas. Em seguida apresentou conceitos sobre classificadores mostrando suas classificações e os tipos mais citados nas bibliografias da área de aprendizagem de máquina. Especificamente detalhou o classificador Naive Bayes que será usado neste trabalho para inferir as preferências do usuário pelo Assistente de *Feedback*. Nesta apresentação foram mostrados seu algoritmo, os critérios para se medir a qualidade do mesmo, alguns resultados do seu desempenho com relação aos demais classificadores e suas extensões.

Em seguida foram apresentados os conceitos básicos do modelo de usuário e suas formas de aprendizagens e, finalmente, foi mostrado o ambiente Direto detalhando sua arquitetura e implementação.

No próximo capítulo será apresentado o modelo do Assistente de *Feedback* e suas características que utilizará como núcleo uma simples rede bayesiana, chamada Naive Bayes, para classificação das preferências do usuário.

### 3 Modelo Geral do Assistente de *Feedback*

O classificador Naive Bayes pode ser aplicado em uma ferramenta da *Web* de forma produtiva, como no ambiente Direto, de modo a assistir o comportamento do usuário durante o consumo de informações via correio eletrônico. Esta ferramenta gera uma base estatística que será interpretada a fim de retornar sugestões para ajustes dos parâmetros dos filtros das mensagens no ambiente do Direto.

Neste capítulo, é apresentada uma proposta de um modelo de assistente de *Feedback* para o serviço de filtragem de mensagens do *software* Direto. O modelo, projetado a partir do levantamento bibliográfico do classificador Naive Bayes, utiliza como núcleo uma simples rede bayesiana, chamada Naive Bayes, permitindo classificar/inferir as preferências dos usuários.

#### 3.1 Estrutura Geral do Modelo

A essência deste trabalho é apresentar a proposta de um modelo que terá a função de assistir o usuário a fim de interpretar a sua preferência, através do seu comportamento com as mensagens trafegadas em ambientes como o Direto. O projeto do modelo levou em consideração que seria implementado um protótipo que automatizasse o seu funcionamento.

São definidas algumas características básicas para o modelo, tais como:

- ◆ **simplicidade de uso:** a fundamental característica do Assistente de *Feedback* (AF) é a simplicidade. Ele poderá disponibilizar vários benefícios ao usuário sem demandar qualquer trabalho extra para o mesmo;
- ◆ **suporte à decisão do usuário:** o usuário é especialista do domínio de suas próprias características;
- ◆ **usuário não sofrerá interrupções durante a execução do AF:** o protótipo que será implementado funcionará em segundo plano com o correio eletrônico do ambiente Direto. Assim, o usuário não sofrerá interferência do AF durante a utilização do correio;
- ◆ **o AF interpreta ações aplicadas pelo usuário às mensagens:** o AF monitorará automaticamente as ações (p. ex. exclusão, encaminhamentos, respostas ao remetente) que serão aplicadas às mensagens durante a utilização do correio;
- ◆ **retorno de informações sobre as interpretações do comportamento do usuário às mensagens:** o modelo utilizará técnicas de inferência das preferências do usuário e retornará os resultados através de uma interface GUI no ambiente Direto.

O modelo proposto será dividido em três etapas com funções distintas, conforme descritas abaixo e visualizadas na fig. 3.1.

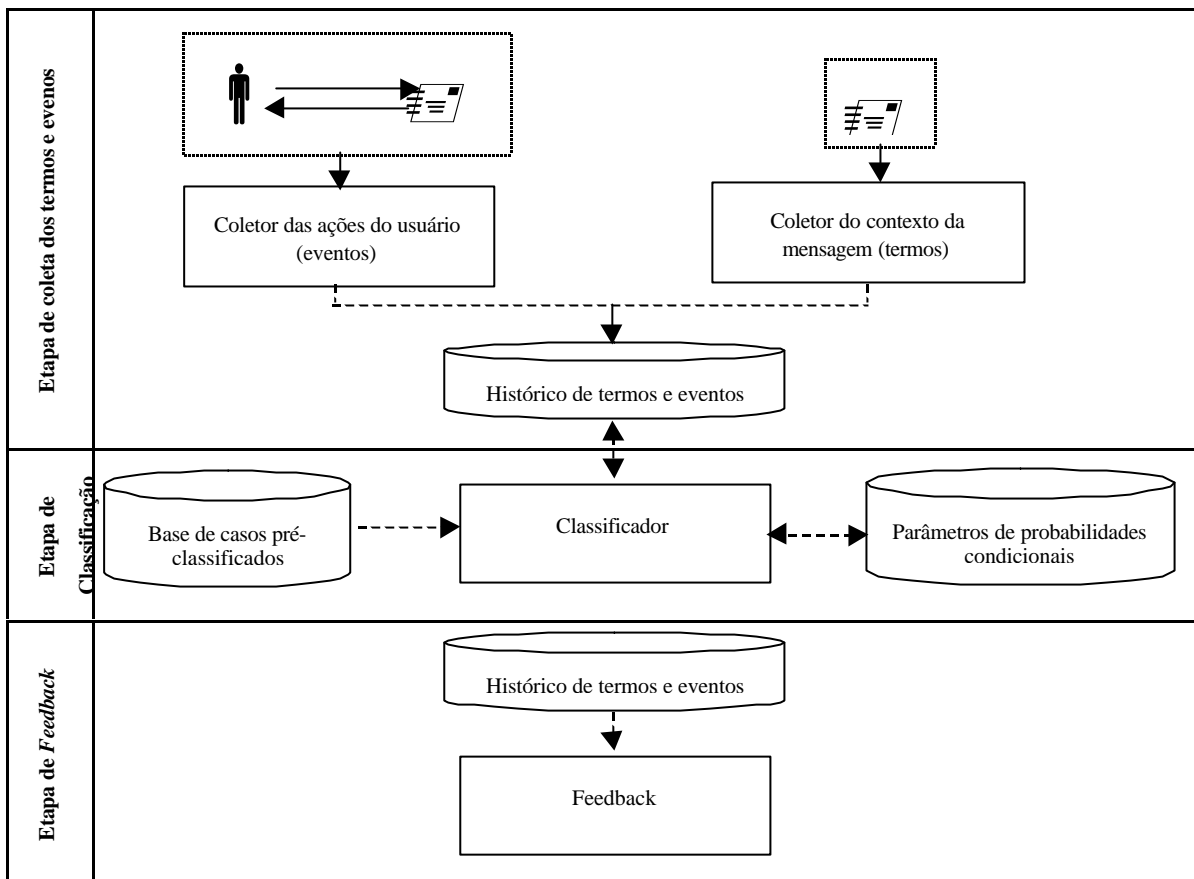


FIGURA 3.1 - Estrutura geral do modelo do AF

- ◆ **etapa de coleta dos termos e eventos das mensagens:** tem como função observar as ações dos usuários sobre as mensagens vindas do serviço de filtragem do Direto. A mensagem contém vários termos que representam o tema (ex. economia, saúde e informática). Os eventos são as ações que o usuário executou sobre a mensagem, durante a utilização do correio eletrônico (ex. leu, excluiu e encaminhou). O produto desta etapa será um histórico que conterá as identificações das mensagens com seus termos e os eventos executados pelo usuário. No histórico, cada mensagem será representada por registros contendo os termos mais relevantes com valores binários denotando se os eventos foram executados ou não durante o contato do usuário com a mensagem;
- ◆ **etapa de Classificação:** sua função é de inferir/classificar as preferências do usuário sobre os termos das mensagens, contidos no histórico, através dos eventos que representam as ações do usuário. Cada evento relacionado a uma mensagem é um indicador do interesse do usuário com relação à mensagem. Esse interesse é representado por probabilidades condicionais associando os eventos às classes de preferências do usuário. As classes são representadas pelas letras A,B,C,D,E que determinam o percentual de preferência do usuário por um determinado termo. Também cada classe contém um valor de peso que indica o grau de interesse que ela representa para os termos da mensagem. Os dados de entrada desta fase são os históricos dos termos e eventos das mensagens e os parâmetros de probabilidades

condicionais que contêm os valores das probabilidades dos eventos associados às classes. Essas probabilidades condicionais são geradas e ajustadas, usando técnicas de aprendizagem por indução na base de casos pré-classificados, definidos por um especialista do domínio do Direto. Como uma mensagem pode receber várias ações do usuário, um termo pode conter vários eventos. Assim a fórmula Naive Bayes utiliza esses eventos para calcular a probabilidade do termo relacionado a esses eventos. Isso é possível porque a fórmula assume uma relação de independência entre os eventos a fim de determinar a probabilidade que determina a classe de preferência do usuário sobre aquele termo. Ao executar a fórmula para cada termo do histórico, geram-se probabilidades para cada classe (A,B,C,D,E) e aquela que apresentar o maior valor probabilístico será a que representará a preferência do usuário para o termo inferido. O AF, ao assinalar a classe de maior valor probabilístico, finaliza a segunda etapa gravando no histórico o peso correspondente à classe de maior inferência;

- ◆ **etapa de *Feedback***: esta etapa tem a função de interpretar as classes de preferência dos termos, representadas pelos pesos, assinaladas pelo classificador Naive Bayes na segunda etapa. Esta interpretação cria retornos de sugestões para o usuário, a fim de permitir ajustes dos parâmetros dos filtros de mensagens no ambiente do Direto. Para apresentar sugestões, o AF apresentará uma interface gráfica que conterá duas listas de termos com suas respectivas médias dos pesos. Para cada termo contido no histórico, o assistente calcula a média dos pesos, resultando numa média final do peso para cada termo. Uma lista contendo os termos com as maiores médias indicará para o usuário os termos que mais gostou e a outra lista, contendo os termos com as menores médias, indicará para o usuário os termos que menos gostou, a fim de ajudá-lo nas eventuais mudanças e ajustes do filtro de mensagens. Os termos que contiverem as maiores médias são aqueles para os quais o usuário apresentou maior preferência durante o uso do correio eletrônico, sendo um grande candidato à inclusão ou confirmação na lista de termos no filtro do correio eletrônico. Os termos que contiverem as menores médias são aqueles que o usuário apresentou menor preferência durante o uso do correio eletrônico, sendo um grande candidato à exclusão ou não inclusão na lista de termos no filtro do correio eletrônico.

As etapas apresentam processos com funções bem definidas e de ordem sequencial, ou seja, os dados de saída da primeira etapa servirão de entrada para a segunda etapa e os dados de saída da segunda etapa servirão de entrada para a terceira. Na fig. 3.2 é mostrada a ordem sequencial dos passos que são executados pelo AF quando acoplado na ferramenta de e-mail do Direto.

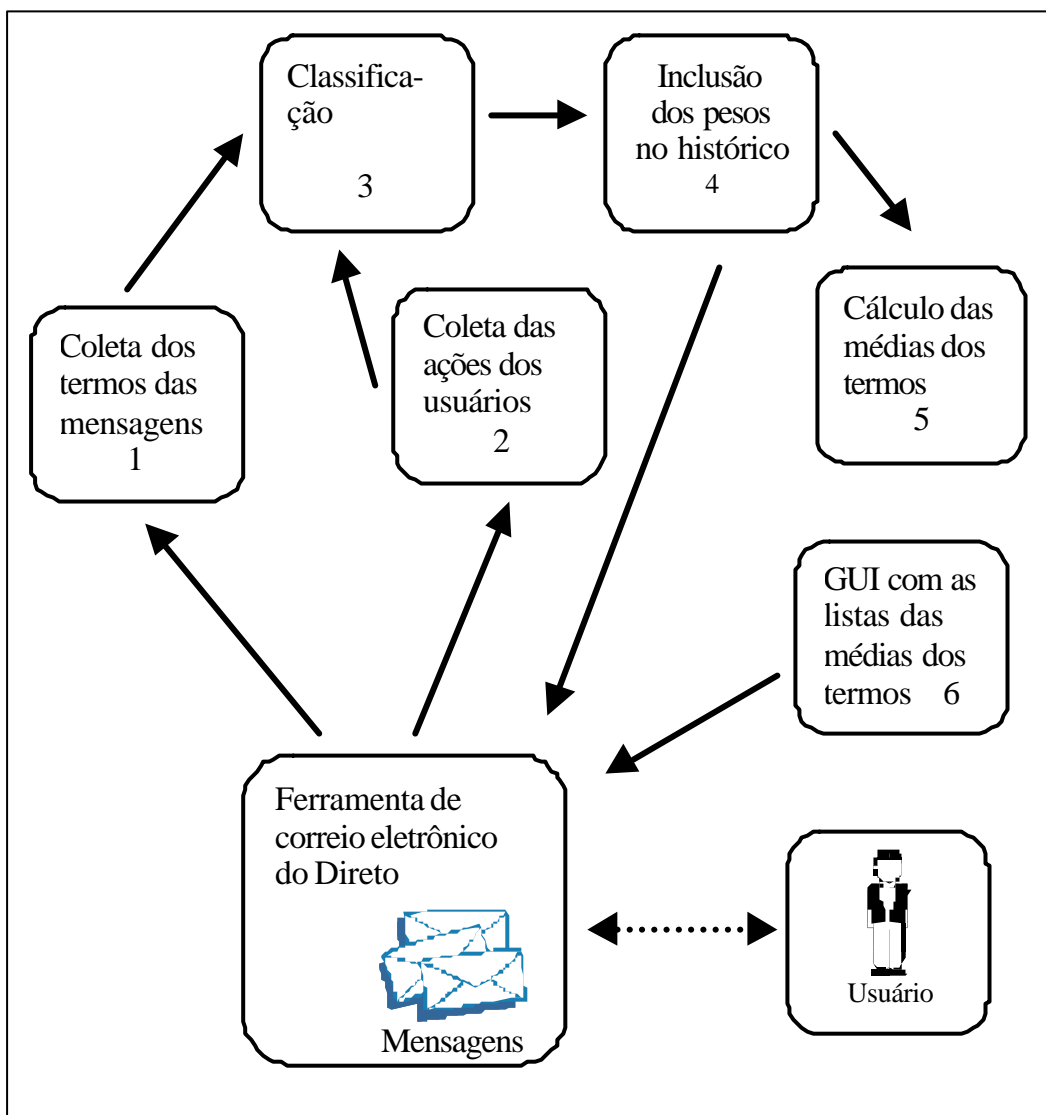


FIGURA 3.2 - Sequência de passos do AF no ambiente de e-mail do Direto

Nas próximas seções será detalhada cada etapa e apresentadas as relações existentes entre elas.

### 3.2 Etapa de Coleta de Termos e Eventos

Esta etapa corresponde à extração e ao armazenamento das ações do usuário (eventos) e dos termos que representam o tema das mensagens durante o uso do correio eletrônico do Direto pelo usuário. Esta coleta tem por finalidade formar a base de dados, chamada de histórico de termos e eventos, conforme pode ser visto na fig. 3.3.

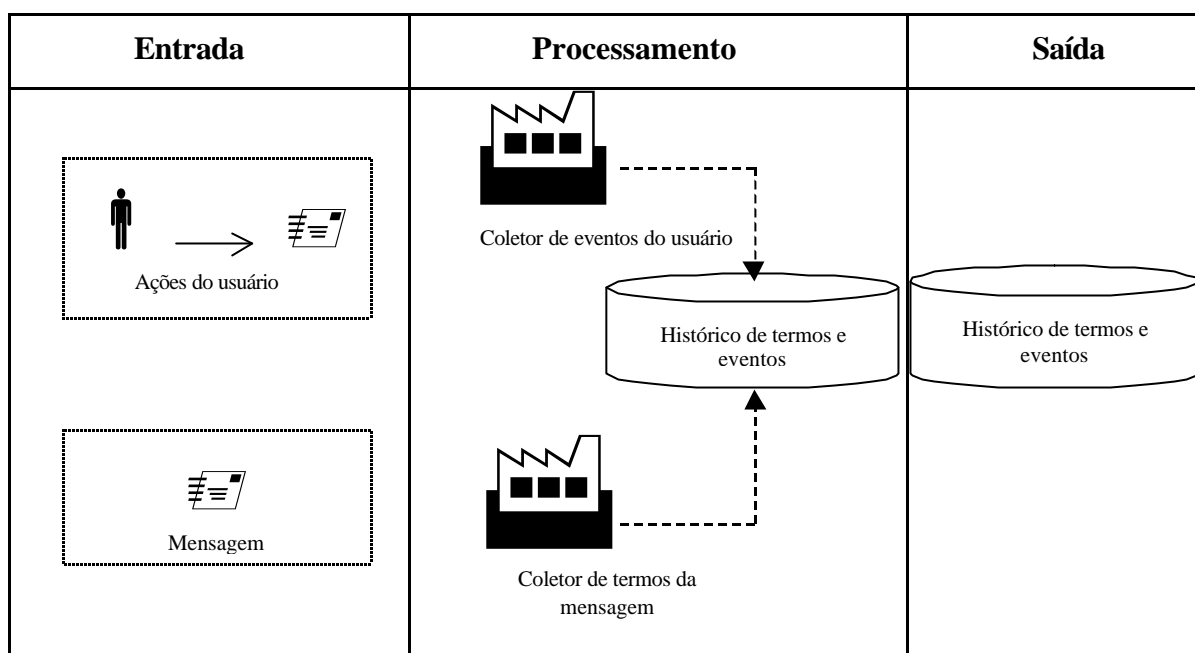


FIGURA 3.3 - Etapa de coleta dos termos das mensagens e os eventos do usuário

### 3.2.1 Dados de entrada

As origens dos dados de entrada são:

- ♦ **eventos do usuário:** o usuário do correio eletrônico executa operações que geram eventos para cada mensagem. Cada evento relacionado a uma mensagem é um identificador do interesse do usuário com relação à mensagem. A inferência desse interesse será tratada na etapa de classificação. Uma mensagem pode ter mais de um evento relacionado. Por exemplo, o usuário pode imprimir uma mensagem após tê-la lido. Algum tempo depois o usuário pode respondê-la ou encaminhá-la para um destinatário. O conjunto de eventos está apresentado na tab. 3.1:

TABELA 3.1 - Eventos que o usuário pode executar para uma mensagem

Identificador	Evento
Excluir	Excluiu mensagem
Encaminhar	Encaminhou mensagem
Responder	Respondeu mensagem
MoverPasta	Moveu para pasta
Imprimir	Imprimiu mensagem
LerCorpo	Leu corpo da mensagem



- ◆ **mensagens filtradas que o usuário recebe do serviço de filtragem do Direto:** essas mensagens são originadas do serviço de filtragem do correio eletrônico do Direto. Esse serviço, desenvolvido através de um trabalho de mestrado na UFRGS por Ricardo Balinski, para a filtragem de mensagens do correio eletrônico [BAL 2001]. As mensagens filtradas, que o usuário recebe do serviço de filtragem do Direto, são selecionadas conforme o vetor de termos que o usuário especifica para o canal de notícias, ao solicitar um serviço de filtragem para um canal. Cada mensagem filtrada traz consigo palavras-chave que causaram a sua seleção, assim como o canal de origem, como pode ser visto na fig. 3.4. Essas palavras-chave serão inseridas no histórico como termos representantes dessa mensagem.

---

*Id:* <2001111402.OAA2324@direto.inf.ufrgs.br>

---

*De:* [Filtro canal saúde@direto.inf.ufrgs.br](mailto:Filtro_canal_saude@direto.inf.ufrgs.br)

*Termos* **concurso / oftalmologia / pediatria / enfermeiros / médico-cirurgiões**

*Assunto:* Hospital de Clínicas anuncia edital de **concurso**

*(corpo da mensagem)* O Hospital de Clínicas anuncia o edital para o concurso de **médico-cirurgiões e enfermeiros** instrumentalistas.

O concurso será o dia dd/mm/aa das 8:00 às 12:00, abrindo para especialistas em **pediatria, oftalmologia,**

---

FIGURA 3.4 - Exemplo de uma mensagem filtrada

- ◆ **mensagens com outras origens:** mensagens vindas de outros locais, sem passar pelo filtro do Direto, poderão ser tratadas pelo AF. Neste caso, o AF aplica um algoritmo de extração de termos para o campo *Assunto* e o *corpo* da mensagem, a fim de coletar os termos relevantes. Este algoritmo basicamente extrai termos através da eliminação de palavras comuns encontradas numa lista, chamada de *stop-list* (por exemplo tal, como, e, através, entretanto, o, as, etc.). As palavras restantes formarão o conjunto de termos que representarão o tema da mensagem [PAY 95a]. Um exemplo dessa extração em uma mensagem pode ser visto na fig. 3.5, onde as palavras em negrito representam os termos coletados pelo algoritmo de extração.

---

*Id:* <2001111402.OAA2324@inf.ufrgs.br>

---

*De:* [Fulano@inf.ufrgs.br](mailto:Fulano@inf.ufrgs.br)

*Assunto:* **Artigos** sobre **Classificadores Naive Bayes**

*(corpo da mensagem)* Com relação a sua **pesquisa** sobre **classificadores**, existe mais **informações** nos **tópicos** de **aprendizagem**, **filtragens** de **informações** e **aprendizagem** com **agentes** utilizando **classificadores Naive Bayes**.

---

FIGURA 3.5 - Exemplo de uma mensagem de outras origens com termos extraídos

### 3.2.1.1 Filtro do Ambiente Direto

Foi desenvolvido por meio de um trabalho de mestrado na UFRGS por Ricardo Balinski, um serviço de filtragem que utiliza como linguagem o *Sieve*. Esse serviço apresenta dois tipos de filtros:

- ◆ filtro para mensagens de usuários;
- ◆ filtro para o serviço de canais de informação.

De modo geral, o filtro armazena os termos das mensagens com seus respectivos pesos. Os dados do filtro são lidos e o resultado é apresentado por meio de uma *String* separando por “/”, os termos com o seu valor de peso, como, por exemplo:

“*concurso=4/oftalmologia=6/pediatria=3/enfermeiros=4*”

A criação dos filtros do Direto será feita por meio de regras geradas, em forma de *scripts*, a partir das opções selecionadas por intermédio de uma interface com o usuário, utilizando *javascript*. As *scripts* geradas são enviadas para o servidor, aonde serão armazenadas para o uso do serviço de filtragem.

### 3.2.2 Dados de saída

Os dados de saída do coletor de eventos e termos irão compor o histórico de eventos e termos do AF.

Cada linha deste histórico de eventos e termos consiste na tupla (*IdMen, Termo, Eventos (Exclui, Encaminha, Responde, MovePasta, Imprime, Ler), PesoClasse*):

- ◆ *IdMen*: identificador da mensagem;
- ◆ *Termo*: este campo contém a palavra-chave (termo) da mensagem. Uma mensagem pode conter vários termos no arquivo *Historico*, um para cada tupla;
- ◆ *Eventos*: os eventos representam as ações executadas quando o usuário entra em contato com a mensagem. Cada evento pode assumir dois estados, representados por valores binários. Quando o usuário executar um determinado evento na mensagem, a ocorrência desse evento será indicada pelo valor binário um (1), nas linhas do histórico que contém os termos desta mensagem. Quando o usuário deixar de executar algum evento numa mensagem, os eventos não executados serão indicados pelo valor binário zero (0). Por exemplo, caso o usuário leia o corpo de uma mensagem e imprimir, os campos *LerCorpo* e *Imprimir* conterão valor binário um (1). Mas os demais eventos (*Excluir, Encaminhar, Responder, MoverPasta*) que não foram executados pelo usuário na mensagem conterão valor binário igual a zero (0). Os campos do histórico que poderão apresentar esses valores, pelas ações que o usuário poderá executar, são descritos abaixo e visualizados na tab. 3.2:

TABELA 3.2 - Representação dos estados dos eventos executados pelo usuário

Evento	Ocorreu	Não Ocorreu
Excluiu a mensagem	1	0
Encaminhou a mensagem	1	0
Respondeu a mensagem	1	0
Moveu para pasta	1	0
Imprimiu a mensagem	1	0
Leu corpo da mensagem	1	0

- *Excluir*: assume valor 1 quando o usuário exclui a mensagem, caso contrário permanece o valor 0;
  - *Encaminhar*: assume valor 1 quando o usuário encaminhou a mensagem, caso contrário permanece o valor 0;
  - *Responder*: assume valor 1 quando o usuário respondeu a mensagem, caso contrário permanece o valor 0;
  - *MoverPasta*: assume valor 1 quando o usuário moveu para a pasta a mensagem, caso contrário permanece o valor 0;
  - *Imprimir*: assume valor 1 quando o usuário imprimiu a mensagem, caso contrário permanece o valor 0;
  - *LerCorpo*: assume valor 1 quando o usuário leu o corpo da mensagem, caso contrário permanece o valor 0;
- ◆ *PesoClasse*: este campo será a representação numérica das classes utilizadas neste modelo. Quanto maior o valor, maior será o grau de preferência e vice-versa. Cada termo do histórico, após a etapa de classificação, conterá um valor que representará a densidade de preferência do usuário pelo termo, baseado nos eventos executados. Como este campo só receberá valores na saída da segunda etapa, não será tratado neste momento.

### 3.3 Etapa de Classificação

Esta etapa é considerada a mais importante das três identificadas no modelo. É nela que assinalarão as classes que indicarão o grau de preferência das mensagens manipuladas pelo usuário no correio eletrônico do Direto. Será incluído nesta etapa o processo de classificação, que corresponde à aplicação do algoritmo de classificação Naive Bayes, detalhado no capítulo anterior. Os detalhes desta etapa são descritos a seguir e o esboço das fases pode ser visualizada na fig. 3.6:

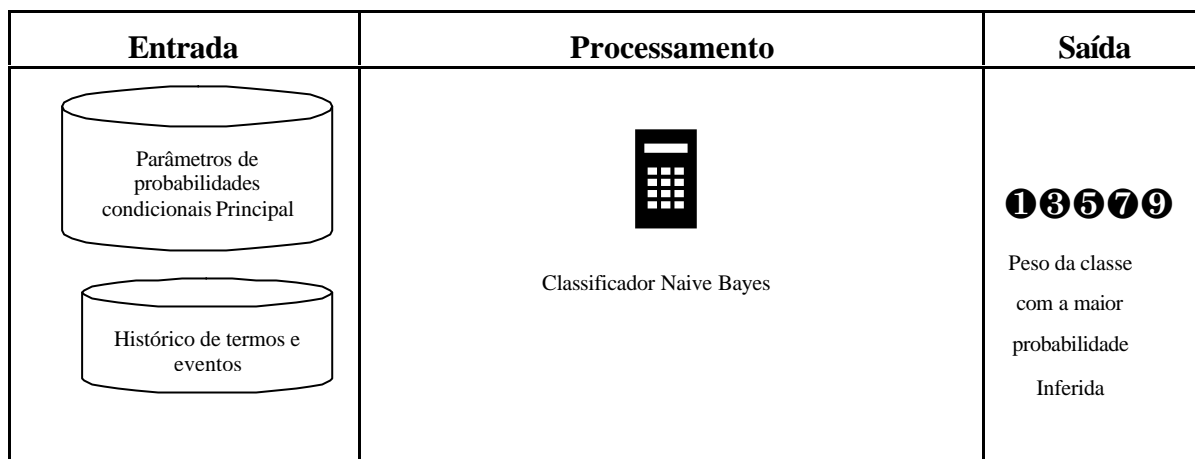


FIGURA 3.6 - Representação da etapa de classificação

Como entrada para esta etapa, além do histórico de eventos e termos, também se previu uma tabela de parâmetros de probabilidades condicionais dos eventos associados às classes. Como saída esta etapa armazenará o peso da classe com maior valor probabilístico, inferido pelo classificador, no campo *PesoClasse*, que será usado posteriormente na etapa de *Feedback*.

### 3.3.1 As classes

Como foi visto na seção 2.4.2, uma classe tem como função rotular uma determinada população de características. Estas classes podem ser resultado de uma predição pelo conhecimento dos valores dos atributos (características) ou ainda podem ser definidas pelo valor espacial dos seus atributos.

As classes do AF indicarão a preferência do usuário com os termos das mensagens, as quais serão inferidas pelo classificador através das ocorrências dos eventos executados pelo usuário. Esses eventos representam a população de características (atributos) que através dos seus valores predirão uma determinada classe.

Para representar o percentual de preferência, que varia de (0 a 100%), definiram-se as classes (A,B,C,D,E) que indicarão a densidade de preferência do usuário pela mensagem. Cada classe terá um peso que indicará o grau de preferência, entre o intervalo de 1 a 9 e quanto maior o peso maior a preferência do usuário com o termo da mensagem. Esses pesos serão incluídos na etapa de classificação após o processo de inferência do classificador Naive Bayes, e serão tratados estatisticamente na terceira etapa a fim de apresentar sugestões sobre os termos mais interessantes e menos interessantes para o usuário do correio eletrônico. As classes, suas descrições, seus intervalos percentuais de preferência e seus pesos podem ser vistos na tab. 3.3.

Para facilitar a compreensão da tab. 3.3, se uma predição do classificador indicasse a classe B para o termo “x”, isso indicaria que o usuário apresentou um grau de preferência entre 60% a 80%, com peso 7 (sete) da densidade de preferência para o termo “x”. Já para o termo “y”, a predição do classificador indicou a classe D de 20% a 40% da preferência, com peso 3 (três) da densidade de preferência. Isso significa que o usuário apresentou maior preferência para o termo “x” por conter maior valor de peso.

TABELA 3.3 - Classes com suas descrições, intervalos percentuais de preferência e seus pesos

Classe	Descrição	Intervalos de preferência de (0 a 100%)	Peso
A	Gostou Muito	Acima de 80% de preferência	9
B	Gostou	Acima de 60% até 80% de preferência	7
C	Gostou pouco	Acima de 40% até 60% de preferência	5
D	Gostou muito pouco	Acima de 20% até 40 de preferência	3
E	Não Gostou	De 0% até 20% de preferência	1

### 3.3.2 O Algoritmo Naive Bayes

O algoritmo escolhido para a classificação/inferência dos termos das mensagens foi o Naive Bayes (ver seção 2.5), opção que foi motivada pelo fato de ser citado nas literaturas [PAZ 96], [PAZ 97], [FRI 97] e [DOM 96] como estado da arte e por apresentar um desempenho mais satisfatório sobre as demais propostas. Muitas ferramentas utilizam o classificador Naive Bayes para inferir/classificar nas tarefas de perfis de usuário como em [PAZ 96] e [PAZ 97]. Além disso, as redes Bayesianas podem modelar um grande espectro de atividades cognitivas e sua maior força está em raciocínio causal, que facilita o raciocínio sobre as ações, explicações e preferências [CAR 99a]. Em especial, este trabalho apresenta um perfil de modelo cognitivo.

O Classificador Naive Bayes é um método probabilístico para classificação. Neste trabalho, ele pode ser usado para prever uma classe  $c_i$  dos termos extraídos de uma mensagem  $j$ , dados os valores dos eventos (atributos) aplicados nesta mensagem:

$$P(c_i | E_1 = V_{1_j} \dots E_n = V_{n_j})$$

Se os eventos valorados são independentes, esta probabilidade é proporcional a:

$$P(c_i) \prod_k P(E_k = V_{k_j} | c_i)$$

A probabilidade condicional do evento de uma mensagem, dado que o usuário tenha GostadoMuito ( $P(E_k = V_{k_j} | c_i)$ ) e a probabilidade a priori que o usuário tenha GostadoMuito ( $P(c_i)$ ) da mensagem podem ser estimadas através dos conjuntos de dados treinados. O conjunto de todos esses valores estimados das probabilidades a priori e condicionais é chamado de parâmetros das probabilidades condicionais. Em síntese, o classificador, ao aplicar a fórmula Naive Bayes, determina a classe mais provável para cada termo extraído de uma mensagem  $j$ , computando a probabilidade de cada classe (*GostouMuito*, *Gostou*, *GostouPouco*, *GostouMuitoPouco*, *NãoGostou*). A classe que apresentar o maior valor probabilístico será assinalada como aquela que representará o conjunto de termos da mensagem  $j$ .

### 3.3.3 Os Parâmetros das probabilidades condicionais

Como foi visto na seção 2.5.1, os valores das probabilidades condicionais podem ser obtidos ou a partir de dados dos exemplos de casos (enfoque frequencialista), ou a partir

do conhecimento [CAR 99a]. Em [HEC 98], diz-se que pode-se usar o conhecimento humano com eficiência quando o domínio da aplicação é bem delimitado, como é o caso dos eventos do usuário com relação às mensagens do correio eletrônico do Direto. Então o AF usará o conhecimento humano para determinar as distribuições probabilísticas que contêm as relações de dependência entre os eventos e as classes de preferência. Cada relação entre um evento e uma classe contém um valor probabilístico do evento condicionado a uma classe como  $P(\text{evento}|\text{classe})$ . Na fig. 3.7 pode ser vistas as probabilidades condicionais projetadas em um grafo acíclico direcionado (DAG). Este gráfico exibe uma rede Naive Bayes com a classe  $A$  sendo o nó pai de todos os outros nós, representados pelos eventos. Essa rede contém apenas ligações simples entre a classe e os eventos.

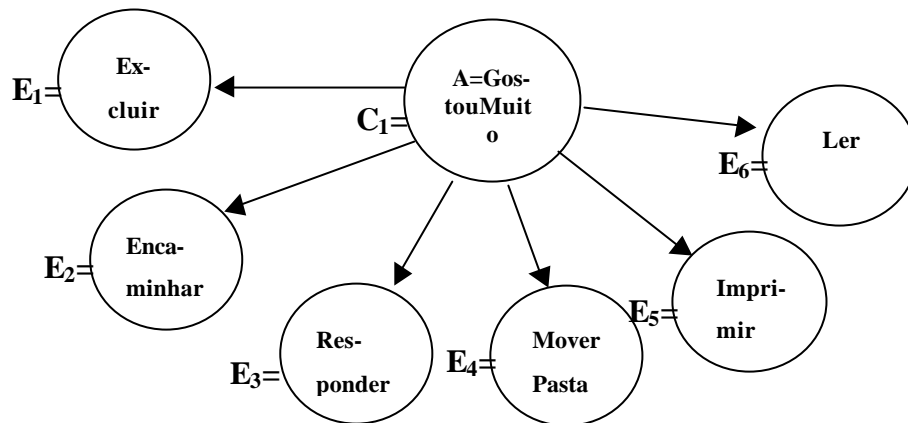


FIGURA 3.7 - Gráfico da rede Naive Bayes com os eventos e a classe  $A$

Com esta representação gráfica fica fácil observar a existência da dependência entre os eventos e a classe  $A$ . Essa dependência é definida pelo arco ligando a classe com os eventos. Essa ligação representa a probabilidade condicional entre o evento dado à classe. Por exemplo,  $P(E_1=\text{Excluir}|C_1=A)$  indica a probabilidade condicional da ligação da classe  $A$  com o evento  $E_1 = \text{Excluir}$ . Porém, é importante lembrar que existe entre os eventos valorados uma total independência, ou seja, um evento pode ocorrer independente se outro evento ocorreu ou não. Essa independência é responsável pela redução do esforço no cálculo das distribuições das probabilidades conjuntas no classificador Naive Bayes.

A rede da fig. 3.7 exibe a seguinte distribuição da probabilidade conjunta:

$$P(C_1, E_1, E_2, E_3, E_4, E_5, E_6) = P(C_1) \cdot P(E_1 | C_1) \cdot P(E_2 | C_1) \cdot P(E_3 | C_1) \cdot P(E_4 | C_1) \cdot P(E_5 | C_1) \cdot P(E_6 | C_1)$$

Sendo assim, considerando o uso do conhecimento de um especialista do domínio do ambiente Direto como requisito para criar as probabilidades condicionais, que serão utilizadas como entrada na fórmula Naive Bayes, conseguiu-se especificar todas as probabilidades condicionais e a *priori*, chamado de parâmetro das probabilidades condicionais, conforme mostra a tab. 3.4.

TABELA 3.4 - Probabilidades condicionais para a ocorrência dos eventos dado as classes

P(evento Classe)	Valor especificado pelo especialista P(evento c <sub>i</sub> )				
	c <sub>1</sub> =A	c <sub>2</sub> =B	c <sub>3</sub> =C	c <sub>4</sub> =D	c <sub>5</sub> =E
P(c <sub>i</sub> )	0.50	0.50	0.50	0.50	0.50
P(Excluir c <sub>i</sub> )	0.05	0.1	0.3	0.7	0.9
P(Encaminhar c <sub>i</sub> )	0.4	0.3	0.2	0.1	0.05
P(Responder c <sub>i</sub> )	0.7	0.6	0.5	0.2	0.05
P(MoverPastal c <sub>i</sub> )	0.9	0.7	0.5	0.2	0.1
P(Imprimir c <sub>i</sub> )	0.9	0.7	0.5	0.2	0.1
P(Ler c <sub>i</sub> )	0.5	0.4	0.3	0.2	0.05

Os valores das probabilidades condicionais e a *priori* da tab. 3.4 foram definidos por um especialista, que neste trabalho será representado pelo autor, que conhece o domínio da aplicação. Para chegar aos valores utilizou-se a técnica apresentada na seção 2.5.1, que usa um conjunto de casos validados analisados pelo especialista contendo eventos executados e a classe de preferência que esses eventos representam. A partir destes casos aplicou-se o algoritmo para aprender as distribuições probabilísticas da rede e identificar a interdependência entre os arcos [HRU 97]. Esse algoritmo utiliza o cálculo da máxima estimação de probabilidade, mostrado por [BOR 99] e [TIR 99], calculando as probabilidades a *priori* e condicionais.

Voltando ao exemplo da probabilidade condicional  $P(E_1=Excluir|C_1=A)$ , o especialista definiu que para essa relação o valor definido foi de 0.05 ou 5%. Este valor representa a probabilidade de 5% do usuário ter excluído a mensagem para a classe *GostouMuito*, caso ainda não existir disponível nenhuma outra ação aplicada por este usuário.

Como o domínio de valores para os eventos apresentam dois estados (Ocorreu=1 e NãoOcorreu=0), os parâmetros das probabilidades condicionais também deverão conter valores para as não ocorrências dos eventos, ou seja, quando o usuário não aplicar determinadas ações em uma mensagem. Determinou-se que um evento não ocorrido, dada uma determinada classe, será representado por uma probabilidade condicional com valor complementar ao valor da ocorrência desse evento, dada uma determinada classe. Por exemplo, aproveitando o valor apresentado anteriormente da probabilidade condicional de  $P(E_1=Excluir|c_1=A)$  para a não ocorrência desse evento, dada a classe A, ou seja, quando o evento Excluir apresentar valor igual a 0 no histórico, o valor de  $P(E_1=NãoExcluir|c_1=A)$  será de 95%. Conclui-se que o usuário ao apresentar o grau de preferência da classe *gostouMuito* para uma mensagem e ainda não existir disponível nenhuma outra informação sobre as ações que este usuário aplicou sobre a mesma, então a probabilidade de terem ocorridos todos os outros eventos, excetuando o evento *Excluir* na mensagem é de 0.95 ou 95%. Isso ocorre pelo fato de *Excluir* e *NãoExcluir* serem complementares, isto é,  $P(E_1=Excluir|c_1=A)+P(E_1=NãoExcluir|c_1=A)=1$  [CAR 99a].

Para completar os parâmetros das probabilidades condicionais como entrada da segunda etapa, geraram-se os seguintes valores para as não ocorrências dos eventos, conforme podem ser vistos na tab. 3.5.

TABELA 3.5 - Probabilidades condicionais para a NÃO ocorrência dos eventos dadas as classes

P(evento Classe)	Valor especificado pelo especialista P(evento c <sub>i</sub> )				
	c <sub>1</sub> =A	c <sub>2</sub> =B	c <sub>3</sub> =C	c <sub>4</sub> =D	c <sub>5</sub> =E
P(NãoExcluir c <sub>i</sub> )	0.95	0.9	0.7	0.3	0.1
P(NãoEncaminhar c <sub>i</sub> )	0.6	0.7	0.8	0.9	0.95
P(NãoResponder c <sub>i</sub> )	0.3	0.4	0.5	0.8	0.95
P(NãoMoverPasta c <sub>i</sub> )	0.1	0.3	0.5	0.8	0.9
P(NãoImprimir c <sub>i</sub> )	0.1	0.3	0.5	0.8	0.9
P(NãoLer c <sub>i</sub> )	0.5	0.6	0.7	0.8	0.95

### 3.3.4 A classificação

Na etapa de classificação, os dados de entrada serão o histórico dos termos e eventos e os parâmetros das probabilidades condicionais. Desta forma o classificador inicia o trabalho de classificação dos termos das mensagens pelos eventos associados contidos em cada linha do *Histórico*. Um exemplo do conteúdo do histórico pode ser visto na tab. 3.6.

TABELA 3.6 - Exemplo do histórico de termos e eventos

IdMensagem	Termos	Exclui	Encaminha	Responde	MovePasta	Impime	Ler	PesoClasse
402	Artigos	1	1	0	0	0	0	
402	Classificadores	1	1	0	0	0	0	
403	Agentes	0	1	1	1	0	1	
403	Artigos	0	1	1	1	0	1	
404	Agentes	1	0	0	0	0	0	
405	Artigos	1	0	0	0	0	0	
405	Classificadores	1	0	0	0	0	0	

Pode-se observar pela tab. 3.6 que cada tupla do histórico é diferenciada pela concatenação do campo identificador da mensagem (*IdMensagem*) e do campo (*Termo*). Cada mensagem pode ter vários termos, como podem ser vistos nas últimas duas linhas da tab. 3.6, onde a mensagem 405 contém os termos *Artigos* e *Classificadores*. Cada mensagem pode também receber vários eventos executados ao mesmo tempo, como, por exemplo, na linha 4, onde o usuário encaminhou, respondeu, moveu para uma pasta e leu a mensagem 403.

O interesse principal do classificador é prever uma classe de preferência, utilizando o algoritmo Naive Bayes, para cada tupla do histórico. Para isso, deve-se aplicar a fórmula do classificador para cada classe usando como dados de entrada os eventos executados e não executados, contidos na tupla do histórico. O resultado será um valor probabilístico para cada uma das classes e aquela que apresentar o maior valor inferido será assinalada para representar a preferência do usuário para cada tupla.

Para exemplificar a aplicação do classificador, usaram-se os dados da tab. 3.4 e tab. 3.5 para calcular as probabilidades das classes, usando como exemplo a primeira linha da tab. 3.6. Como foi apresentado na seção 2.5, o classificador Naive Bayes utiliza o



teorema de Bayes como fórmula geral. Como o cálculo do denominador da fórmula Naive Bayes é constante, decidiu-se por desconsiderar seu valor a fim de simplificar os cálculos. A fórmula utilizada neste modelo pode ser visualizada na fig. 3.8 e os cálculos da primeira linha do histórico são apresentados na tab. 3.7.

Na tab. 3.7 é apresentado o cálculo da proporção de probabilidade final para cada classe.

$$P(C = c_i | A_1 = a_{i_1}^{(1)}, \dots, A_n = a_{i_n}^{(n)}) \approx P(A_1 = a_{i_1}^{(1)}, \dots, A_n = a_{i_n}^{(n)} | C = c_i) \cdot P(C = c_i)$$

FIGURA 3.8 - Fórmula do classificador Naive Bayes do AF

TABELA 3.7 - Valores inferidos pelo classificador para a primeira linha do histórico

Classes	Aplicação da Fórmula Naive Bayes	Valor Inferido
A <sub>Gostou Muito</sub>	$0.5 \times 0.05 \times 0.4 \times 0.3 \times 0.1 \times 0.1 \times 0.5 = 0.000015$	<b>0.000015</b>
B <sub>Gostou</sub>	$0.5 \times 0.1 \times 0.3 \times 0.4 \times 0.3 \times 0.3 \times 0.6 = 0.000324$	<b>0.000324</b>
C <sub>Gostou Pouco</sub>	$0.5 \times 0.3 \times 0.2 \times 0.5 \times 0.5 \times 0.5 \times 0.7 = 0.002625$	<b>0.002625</b>
D <sub>Gostou Muito Pouco</sub>	$0.5 \times 0.7 \times 0.1 \times 0.8 \times 0.8 \times 0.8 \times 0.8 = 0.014336$	<b>0.014336</b>
E <sub>Não Gostou</sub>	$0.5 \times 0.9 \times 0.05 \times 0.95 \times 0.9 \times 0.9 \times 0.95 = 0.016448$	<b>0.016448</b>

É importante lembrar que a fórmula Naive Bayes se aplica a todas as classes, e no final pega-se a classe que apresentar o maior valor da proporção de probabilidade calculado. No exemplo acima, a mensagem 402 resultou na classe E<sub>NãoGostou</sub>, porque apresentou o maior valor calculado para as classes. Após a conclusão desse cálculo, o campo *PesoClasse* receberá o valor 1 correspondente à classe E.

Este processo de inferência é repetido para as demais linhas do histórico. Ao final desta fase de classificação, o histórico conterá valores em toda a coluna *PesoClasse*, como pode ser visto na tab. 3.8.

TABELA 3.8 - Histórico de termos e eventos com os pesos incluídos pela segunda etapa

IdMensagem	Termos	Exclui	Encaminha	Responde	MovePasta	Imprime	Ler	PesoClasse
402	Artigos	1	1	0	0	0	0	1
402	Classificadores	1	1	0	0	0	0	1
403	Agentes	0	1	1	1	0	1	7
403	Artigos	0	1	1	1	0	1	8
404	Agentes	1	0	0	0	0	0	2
405	Artigos	1	0	0	0	0	0	2
405	Classificadores	1	0	0	0	0	0	2

Caso o usuário voltar a tratar antigas mensagens armazenadas no correio, o histórico será alterado através da inclusão de novos eventos e, assim, o processo de classificação é executado novamente para essas mensagens. Por exemplo, se o usuário no primeiro momento apenas leu o corpo da mensagem, o classificador infere utilizando o evento

*LerCorpo* com valor 1 e os demais campos com valor igual a 0, movendo um valor para o campo *PesoClasse*. Depois, em outro momento, o usuário resolve imprimir a mesma mensagem, alterando o valor do evento *Imprimir* de zero (0) para o estado um (1). Conseqüentemente, o classificador deve refazer o cálculo, modificando o valor do *PesoClasse* da mensagem no arquivo histórico. Este processo pode ser repetido, conforme os valores dos eventos dos termos das mensagens são alterados pelas novas ações do usuário.

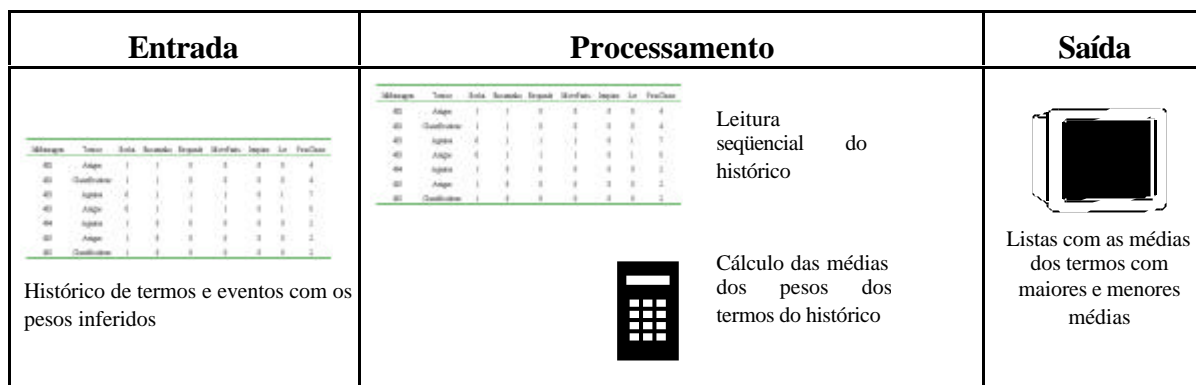
### 3.4 Etapa de *Feedback*

A etapa de *Feedback* é responsável pela análise estatística do histórico dos termos e eventos, a fim de apresentar, através de uma interface gráfica, termos de interesse do usuário. Esta análise ajudará o usuário com sugestões para possíveis mudanças nos parâmetros no filtro de mensagem do Direto. É importante lembrar que o usuário não será interrompido, pela interface do AF, durante o uso do correio eletrônico. Só a visualizará quando achar necessário, solicitando ao AF a apresentação da mesma, através de um atalho disponibilizado no correio eletrônico.

A interface gráfica basicamente conterá duas listas de termos com as médias dos pesos inferidos na segunda etapa pelo classificador Naive Bayes. A média indicará o grau de interesse do usuário em um certo assunto, representado pelo termo da mensagem. Em geral, as listas apresentam duas colunas, uma com os termos e a outra coluna com as médias dos pesos, conforme descrição abaixo:

- ◆ **lista dos termos mais interessantes:** conterá os termos com as maiores médias de peso, sugerindo ao usuário os termos candidatos a permanecerem ou a serem incluídos nos parâmetros do filtro de mensagem do Direto. As maiores médias dos termos indicam que o usuário apresentou maior interesse com certas mensagens que continham esses termos durante o uso do correio eletrônico;
- ◆ **lista com os termos menos interessantes:** conterá os termos com as menores médias de peso, sugerindo ao usuário a exclusão dos termos, contidos nos parâmetros do filtro de mensagem. Caso alguns termos não existirem nos parâmetros do filtro de mensagens, este fato reforçará a sugestão para não incluí-los futuramente nos parâmetros. As menores médias indicam que o usuário apresentou menor ou nenhum interesse com certas mensagens que continham esses termos durante o uso do correio eletrônico.

Para criar essas listas, o AF deverá tratar os pesos contidos no campo *PesoClasse* do histórico de termos e eventos. Durante o processamento desta etapa, serão calculadas as médias dos pesos contidos no histórico resultantes das inferências feitas pelo classificador. Como saída, apresentam-se para o usuário duas listas de termos com suas respectivas médias de pesos. As fases dessa etapa são ilustradas na fig. 3.9.

FIGURA 3.9 - Representação da etapa de *Feedback*

### 3.4.1 Entrada de Dados

Os dados de entrada para a etapa de *Feedback* serão apenas o histórico de termos e eventos. Quando o usuário requisitar sugestões do AF, o histórico será lido sequencialmente e serão criadas as médias dos pesos dos termos. Como esse histórico é constantemente incrementado pelas ações do usuário quando utiliza o correio eletrônico do Direto, o procedimento de criação dessas médias somente será processado nesta etapa, quando solicitado pelo usuário.

### 3.4.2 Processamento

O objetivo dessa fase da etapa de *Feedback* é criar um vetor de termos com as médias de pesos finais calculadas através do histórico de termos e eventos. Essa etapa é inicializada quando o usuário solicita sugestões do AF. Caso contrário, essa etapa não é processada.

Os passos básicos do processamento dessa fase são:

- ♦ **leitura sequencial do histórico:** quando o usuário requisitar alguma sugestão do AF, a etapa de *Feedback* fará uma leitura sequencial do histórico e criará um vetor com todos os termos. Esse vetor contém dois campos, o campo *Termo* e o campo *MédiaPeso* que será valorado na próxima fase;
- ♦ **cálculo da média dos pesos:** durante o processo de leitura, os pesos das repetições de cada termo são somados e o total da soma é dividido pelo número de ocorrências desse termo. Simplificando, a estratégia para se chegar às médias dos pesos dos termos inferidos pelo classificador é basicamente o cálculo da média dos pesos dos termos. No método de condicionamento global [PEA 88] citado por [CAR 99a], o cálculo da média é usado para combinar crenças parciais para obter uma crença final. No AF, o cálculo de média é usado para combinar o peso de cada termo distinto, ocorrido no histórico, para gerar um peso final do termo. Por exemplo, o histórico apresenta os termos com seus pesos inferidos durante a etapa de classificação. A fase de processamento da etapa de *Feedback*, através da leitura sequencial do histórico, montará um vetor contendo os termos e os pesos finais. Esses pesos finais são calculados usando a fórmula da média, como pode ser vista na fig. 3.10.

IdMensagem	Termos	Entra	Encaminha	Responde	MorrePasta	Impime	Leu	PesoClasse	
402	Artigos	1	1	0	0	0	0	4	Histórico de termos e eventos
402	Classificadores	1	1	0	0	0	0	4	
403	Agentes	0	1	1	1	0	1	7	
403	Artigos	0	1	1	1	0	1	8	
404	Agentes	1	0	0	0	0	0	2	
405	Artigos	1	0	0	0	0	0	2	
405	Classificadores	1	0	0	0	0	0	2	

↓ Cálculo da média dos pesos dos termos

Termos	Peso Final (Média)	
Agentes	4.5	Vetor com as médias dos termos
Artigos	4.6	
Classificador	3.0	

FIGURA 3.10 - Criação do vetor com as médias finais dos pesos dos termos

Para calcular os valores das médias finais dos termos *Agentes*, *Artigos* e *Classificador*, foram utilizados os seguintes passos:

- ◆ **média de peso para o termo *Agentes*:** o AF, lendo seqüencialmente o histórico, localizou duas ocorrências desse termo em duas mensagens. Em seguida somaram-se os pesos e dividiram-se pelas ocorrências,  $MédiaFinal = (7+2)/2$ , totalizando o valor 4.5 como média final dos pesos desse termo. Este valor final representa o grau de interesse do usuário sobre o termo *Agente* durante toda a utilização do correio eletrônico do Direto;
- ◆ **média de peso para o termo *Artigo*:** o AF, lendo seqüencialmente o histórico, localizou três ocorrências desse termo em três mensagens. Em seguida somaram-se os pesos e dividiram-se pelas ocorrências,  $MédiaFinal = (4+8+2)/3$ , totalizando o valor 4.6 como média final dos pesos desse termo. Este valor final representa o grau de interesse do termo *Artigo* para o usuário durante a utilização do correio eletrônico do Direto;
- ◆ **média de peso para o termo *Classificador*:** o assistente, lendo seqüencialmente o histórico, localizou duas ocorrências desse termo em duas mensagens. Em seguida somaram-se os pesos e dividiram-se pelas ocorrências,  $MédiaFinal = (4+2)/2$ , totalizando o valor 3.0 como média final dos pesos desse termo. Esse valor final representa o grau de interesse do termo *Classificador* para o usuário durante a utilização do correio eletrônico do Direto.

### 3.4.3 Interface Gráfica de Sugestões

Após a fase anterior gerar o vetor com todos os valores das médias dos pesos dos termos contidos no histórico, o próximo passo é montar as listas com os termos com as maiores e menores médias dos pesos e apresentá-las na interface gráfica do Direto.

Esta fase subdivide-se em dois passos:

**ordenar de forma decrescente o vetor pelas médias dos pesos:** esse passo permitirá que o vetor apresente na sua extremidade superior os termos com as maiores médias

finais e na sua extremidade inferior os termos com as menores médias finais. Esse processo facilitará muito o próximo passo que tratará da apresentação dos termos no vídeo;

**apresentação das listas na interface gráfica do correio eletrônico do Direto:** a utilização de recursos compatíveis ao ambiente Direto permitiu o desenvolvimento de uma interface para apresentar as duas listas contendo os termos com suas respectivas médias finais dos pesos. Estas listas podem ser visualizadas na fig. 3.11:

Termos Interessantes		Termos não Interessantes	
Termos	PesoFinal	Termos	PesoFinal
Artigos	4,6	Classificador	3,0
Agentes	4,5		

FIGURA 3.11- Lista de termos interessantes e a lista de termos não interessantes

### 3.4.4 O *Feedback*

Ao receber retornos de sugestões propostas por um sistema de *Feedback*, o usuário pode responder ou tomar alguma decisão. A linha do raciocínio para compreensão desses retornos de sugestões segue uma forma de interpretação, que seria basicamente um modo de pensamento para ajudar o usuário a entender o raciocínio do sistema e fornecer um caminho para verificar a validade de uma decisão. Deste modo um sistema torna-se útil para o usuário, fornecendo informações para auxiliar na tomada de decisões ou aplicar modificações como no caso dos parâmetros do filtro de mensagens do Direto.

No caso do AF apresentam-se duas formas de interpretações básicas, descritas na tab. 3.9, para que o usuário possa aproveitar as informações contidas nas listas de termos interessantes e não interessantes.

TABELA 3.9 - Formas de interpretações do *Feedback* e as possíveis decisões

Forma de Interpretação	Decisão
1. Os termos de maior preferência (maior média) da lista de termos interessantes, são aqueles que estavam contidos nas mensagens sobre as quais o usuário expressou interesse pelo seu conteúdo por meio de ações.	<ul style="list-style-type: none"> <li>• prováveis inclusões dos termos de maiores médias no filtro de mensagens, se esses termos ainda não estiverem incluídos;</li> <li>• não exclusão dos termos de maiores médias no filtro, confirmando o interesse da preferência do usuário sobre esses termos, se estes estiverem incluídos no filtro de mensagens.</li> </ul>
2. Os termos de menor preferência (menor média) da lista de termos não interessantes, são aqueles que estavam contidos nas mensagens sobre as quais o usuário expressou desinteresse pelo seu conteúdo por meio de ações.	<ul style="list-style-type: none"> <li>• exclusões dos termos de menores médias no filtro de mensagens, se esses termos já estiverem incluídos;</li> <li>• não inclusão dos termos de menores médias no filtro, confirmando o desinteresse da preferência do usuário sobre esses termos, se estes ainda não estiverem incluídos no filtro de mensagens.</li> </ul>

Tipicamente, um sistema é criado da necessidade de retornar informações pertinentes a fim de serem apresentadas ao usuário. É importante que o usuário tenha conhecimento dessas formas de interpretações porque geralmente é comum para os sistemas devolverem informações relevantes na forma de um único resultado, mas o usuário não recebe explicações ou justificações de como o resultado foi computado. Sem esclarecer essas formas de interpretações fica difícil para o usuário compreender esses retornos de informações e aproveitá-los da melhor forma.

### 3.5 Considerações Finais

O objetivo principal deste capítulo foi apresentar um novo modelo de Assistente de *Feedback* que utiliza o classificador Naive Bayes aplicado em tipos de ferramentas como o Direto, de modo a assistir o comportamento do usuário durante o consumo de informações via correio eletrônico. Esta ferramenta gera uma base estatística que será interpretada a fim de retornar sugestões para ajustes dos parâmetros dos filtros das mensagens no ambiente do Direto.

A primeira etapa do modelo, chamada de coleta dos termos e eventos das mensagens, usa a técnica para coletar as ações do usuário durante o contato com a mensagem e os termos relevantes contidos dentro das mensagens. No final desta etapa cria-se uma base contendo os históricos dos termos e eventos capturados.

Na segunda etapa, considerada a mais importante, apresenta uma alternativa de uso do classificador Naive Bayes para classificar os termos do histórico, utilizando os eventos executados pelo usuário. Essa classificação assinala a classe com maior probabilidade e classifica os termos utilizando os pesos das classes.

Na terceira e última etapa criam-se duas listas de termos mais interessantes e menos interessantes através do cálculo da média dos pesos dos termos do histórico. O usuário receberá na interface do Direto as duas listas contendo os termos e suas respectivas médias finais.

As principais contribuições deste trabalho estão inseridas na segunda etapa, chamada de classificação. Foi aplicado o algoritmo de classificação Naive Bayes para prever classes de preferência das mensagens trafegadas no ambiente como o Direto. As classes são inferidas através das ocorrências dos eventos executados pelo usuário. Esses eventos apresentam características bem estáveis possibilitando que as predições sejam bem delimitadas.

A partir da especificação deste modelo, foi implementado um protótipo para assistir o usuário no ambiente Direto, capaz de automatizar as etapas previstas e que pudesse ser utilizado para a validação do mesmo. O mesmo será descrito no próximo capítulo.

## 4 Protótipo de Assistente de *Feedback* Implementado

Após a especificação do novo modelo do Assistente de *Feedback* para o ambiente do Direto, foi construído um protótipo para a automação de cada uma das etapas do modelo.

Este capítulo descreve o protótipo implementado. Inicialmente serão relacionadas as principais características pretendidas para o mesmo. Em seguida, serão apresentados o ambiente e as ferramentas utilizadas para o seu desenvolvimento. Finalmente serão apresentadas a estrutura da implementação efetuada e a interface disponibilizada para as sugestões estatísticas que representa o interesse do usuário.

### 4.1 Considerações Iniciais

O protótipo implementado a partir do modelo apresentado seria utilizado, a princípio, para a validação do mesmo através da instalação e testes em situações reais de uso. Entretanto, buscou-se desde o início da implementação obter como resultado um *software* eficiente, estável e suficientemente genérico para que, ao final do desenvolvimento, o mesmo pudesse ser acoplado em qualquer ambiente que apresentar as mesmas características do Direto. Por se tratar de um modelo geral e amplo, optou-se pela implementação total, mas sem acoplá-lo no ambiente Direto.

Sendo assim, buscou-se implementar o protótipo com o objetivo principal de validar o modelo através de simulações. Desta forma, qualquer interessado que queira utilizar o protótipo para monitorar o comportamento dos usuários de um ambiente de correio eletrônico conseguirá ajustá-lo e, em seguida acoplá-lo ao seu ambiente.

Para implementação desse protótipo, levaram-se em consideração alguns requisitos, os quais foram decisivos na escolha das ferramentas, ambientes e interface, tais como:

- ◆ **automatizar o modelo sem prejudicar a interação entre o ambiente e o usuário:** o usuário não sofrerá interrupções durante o uso do correio eletrônico quando o AF estiver em execução. O protótipo funciona em segundo plano com o correio eletrônico e somente retornará resultados se o usuário solicitar;
- ◆ **simplicidade nas apresentações dos resultados:** não exige do usuário conhecimento prévio sobre técnicas de interpretações estatísticas de processos de classificação. As sugestões são apresentadas de forma simples para facilitar o procedimento de mudança dos parâmetros do filtro de mensagens;
- ◆ **ambiente de instalação somente no cliente:** ao solicitar instalação do protótipo, o ambiente do cliente recebe e executa todas as classes do AF. Portanto, o servidor não processa nenhuma etapa do modelo, diminuindo, assim, o servidor, da carga computacional exigida para execução das etapas. O histórico também é armazenado no cliente, evitando um consumo excessivo de disco do servidor;
- ◆ **interface de utilização através da Web:** por se tratar de um ambiente Web, o retorno de sugestões através deste ambiente facilitará o usuário durante a solicitação dos serviços do AF, porque o usuário não precisará mudar de ambiente caso desejar utilizar o assistente.

O protótipo tem essas características, baseadas nas funções de ambientes como o Direto, que é um sistema corporativo o qual disponibiliza vários tipos de serviços aos seus usuários. Sendo assim, a opção em disponibilizar a instalação do protótipo no cliente foi devido à grande sobrecarga dos serviços essenciais no servidor e também por ser o AF apenas um serviço opcional solicitado pelo usuário quando houver necessidade. Entretanto em trabalhos futuros, poderia ser implementado um *software* como o AF, para ser executado totalmente em um servidor Web.

## 4.2 Ambiente e Ferramentas Utilizadas

Decidiram-se na escolha das ferramentas a serem empregadas na construção do protótipo, considerando-se as características do ambiente em que se pretende acoplá-lo, e que seja executado na máquina do cliente juntamente com os dados de execução.

A codificação do protótipo a ser executado no cliente foi toda desenvolvida em Java, utilizando o pacote JDK 1.3 da Sun Microsystems<sup>®</sup>. A escolha da linguagem Java deve-se ao fato de a mesma ser eficiente e portátil para o ambiente Web, apresentando recursos de segurança para aplicações executadas no cliente e principalmente que todo o ambiente Direto é desenvolvido nesta linguagem.

Utilizou-se especificamente o *applet* Java, que são programas que podem ser carregados no navegador Web a partir de um computador remoto e descartados quando se completa a execução. O *applet* Java permite também disponibilizar a ferramenta no cliente de forma segura e eficiente a fim de manipular dados em arquivos localmente.

O ambiente de desenvolvimento e teste da ferramenta limitou-se ao Windows NT da Microsoft Corp., ao servidor Apache, *plug-in* Java e o JSDK 2.1 da Sun Microsystems<sup>®</sup>. A razão do uso desse ambiente Windows se deve muito mais à falta de disponibilidade de outros ambientes do que questões relacionadas à produtividade. Como o objetivo principal é a validação do modelo, as implementações futuras poderão usar outros ambientes.

O *browser* provê a capacidade de navegação padrão e permite visualizar as mensagens do correio eletrônico, como o Direto ou outro ambiente similar. A única exigência para o *browser* a ser utilizado é que ele deve ter capacidade de suportar aplicações Java, devido ao fato do AF ser implementado nesta linguagem. O uso do *plug-in* Java permite que os *applets* implementados possam ser executados em qualquer *browser*, porque alguns *browser* suportam somente Java 1.0 e muitos suportam Java 1.1, mas poucas versões suportam Java 2 *Platform* [DEI 2001].

Todos os algoritmos foram implementados pelo autor, incluindo o classificador Naive Bayes utilizado na etapa de classificação, por não se encontrar disponível nas literaturas algum algoritmo implementado que se enquadrasse no perfil do AF. Os algoritmos do classificador foram baseados nas definições dos trabalhos de [BOR 99] e [TIR 99].

A interface de sugestões também foi construída utilizando *applets* Java usando os pacotes gráficos *Swing* da Sun.



### 4.3 Estrutura da Ferramenta AF

O protótipo foi dividido em duas classes principais, dois arquivos para armazenamento de dados e a interface para o retorno de sugestões. As classes internas e métodos foram condensados nas duas classes principais para facilitar o carregamento quando o usuário solicitar o serviço do AF. As classes principais são os programas *applets*, com extensões (.class), que serão instalados no cliente, e as classes internas são as classes contidas nas classes principais [DEI 2001]. Nas classes principais existem cinco classes internas e trinta e quatro métodos que compõem todo o protótipo. Não foram utilizados arquivos temporários. Apenas vetores, matrizes e tabelas *hash* foram utilizados para aumentar a velocidade de processamento. A estrutura do Assistente de *Feedback* implementado pode ser visualizada na fig. 4.1.

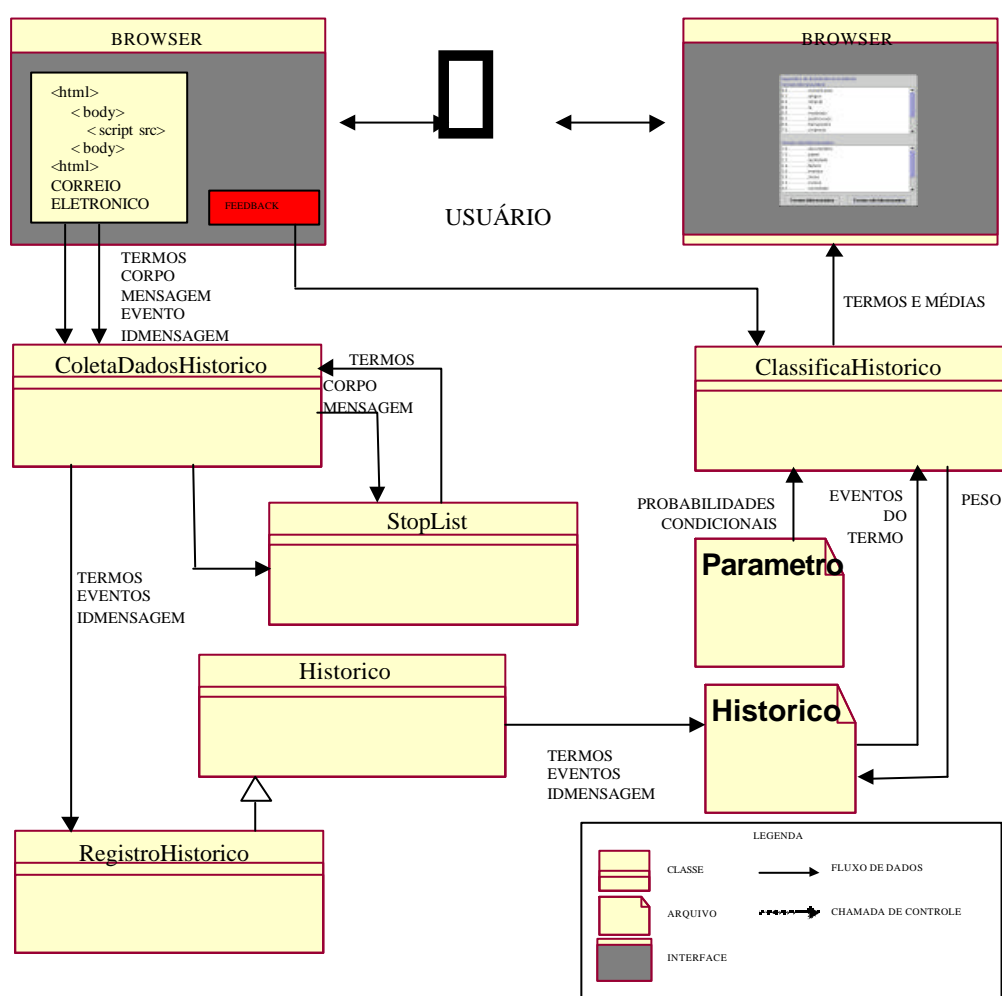


FIGURA 4.1- Arquitetura do Assistente de *Feedback*

A fig. 4.1 descreve a estrutura geral do AF, apresentada em forma de classes principais, classes internas, arquivos de dados e interfaces, todos ligados por chamadas de controle e pelo fluxo de dados.

Todos os módulos do AF foram desenvolvidos em *applet* Java usando assinaturas para acessar arquivos no cliente. O modelo de segurança implementado pela plataforma Java é centrado na idéia de que um código remoto (*applet*) não é confiável e pode acessar apenas recursos limitados, fornecidos dentro de uma área do servidor Web dedicado àquele *applet*. Porém, no contexto do trabalho, os *applets* deverão ser confiáveis para fornecer um determinado serviço, como, por exemplo, gravar arquivos no cliente. Uma solução para este problema foi utilizar assinaturas do *applet* digital, o qual é, então, tratado como um código local confiável, caso a assinatura seja reconhecida pelo sistema que recebeu o *applet*.

A base de dados do AF que será gravada no cliente foi definida utilizando-se os recursos de arquivo de acesso aleatório, implementados pelos objetos da classe *RandomAccessFile*, por permitir rapidez no acesso aos registros. Na tab. 4.1 mostram-se os arquivos de dados utilizados pelas classes implementadas.

TABELA 4.1 - Descrição dos arquivos utilizados na implementação

Arquivo	Descrição
<i>Historico</i>	Armazena os termos extraídos das mensagens e os eventos executados pelo usuário na etapa de coleta dos termos e eventos. Armazena os pesos inferidos durante o processo de classificação.
<i>Parametro</i>	Armazena os parâmetros das probabilidades condicionais <i>a priori</i> e <i>posteriori</i> definidos por um especialista para utilização pelo classificador Naive Bayes na etapa de classificação.

Implementaram-se as etapas do modelo em apenas duas classes principais, por questões exclusivamente voltadas ao desempenho do AF.

Observou-se que a etapa de coleta de termos e eventos utiliza pouco recurso de processamento porque apenas recebe os parâmetros enviados do formulário *HTML/javascript*, para gravá-los no arquivo *Historico*. Como o protótipo deve apresentar o requisito de não prejudicar a interação entre o ambiente do correio eletrônico e o usuário, optou-se em implementar esta etapa somente na classe *ColetaDadosHistorico*, que será chamada toda vez que o usuário executar alguma ação sobre a mensagem.

Porém, as duas últimas etapas do modelo exigem uma quantidade considerável de recursos de processamento. O motivo do consumo do processamento é porque a etapa de classificação do Naive Bayes é executada uma vez para cada registro do arquivo *Historico* e na etapa de *Feedback*. Calculam-se as médias dos termos do *Historico* para apresentá-las na interface gráfica de sugestões do AF. Por apresentar essas características, definiu-se que as últimas etapas seriam implementadas na classe principal *ClassificaHistorico*, para ser executada apenas quando o usuário solicitar as sugestões do AF.

Na tab. 4.2 mostra-se a distribuição das classes implementadas agrupadas nas etapas do modelo proposto do AF.

TABELA 4.2 - Descrição das classes agrupadas pelas etapas do modelo AF

<b>Etapas do Modelo</b>	<b>Classe</b>	<b>Descrição</b>
Coleta de termos e eventos	ColetaDadosHistorico (principal)	Trata dados como termos, eventos e identificadores das mensagens. Criam-se e gravam-se dados no arquivo <i>Historico</i> ;
	Historico	Encapsula as informações de registro do arquivo <i>Historico</i> ;
	RegistroHistorico	Herda as implementações da classe Historico que inclui as variáveis de instância e os métodos <i>get</i> e <i>set</i> . É utilizada pelos programas do AF tanto para ler registros do arquivo <i>Historico</i> como para gravar registros.
	StopList	Extrai termos relevantes de uma <i>String</i> contendo o corpo da mensagem e o assunto, os quais ficam concatenado na mesma <i>String</i> ;
Classificação e <i>Feedback</i>	ClassificaHistorico (principal)	Inferre os pesos dos termos do <i>Historico</i> através do classificador Naive Bayes, utilizando os eventos associados aos termos. Grava os pesos no arquivo <i>Historico</i> , calcula as médias dos pesos de cada Termo e apresenta as listas de sugestões no browser.
	Historico	Encapsula as informações de registro do arquivo <i>Historico</i> ;
	RegistroHistorico	Herda as implementações da classe Historico que inclui as variáveis de instância e os métodos <i>get</i> e <i>set</i> . É utilizada pelos programas do AF tanto para ler registros do arquivo <i>Historico</i> como para gravar registros.

Na tab. 4.3 são apresentados os métodos agrupados por classes e uma descrição das suas funções e arquivos utilizados por cada método.

TABELA 4.3 - Tabela de métodos agrupados pelas classes implementadas

Classe	Método	Função
ColetaDadosHistorico	abreHistorico	Abre o arquivo <i>Historico</i> , caso não exista cria-se um novo;
	leRegistroHistorico	Lê um registro no arquivo <i>Historico</i> ;
	imprimeRegistroHistorico	Lista os registros do arquivo <i>Historico</i> em um arquivo texto do formato <i>ascii</i> ;
	adicionaRegistroHistorico trataMensagem	Adiciona um registro no arquivo <i>Historico</i> ; Controla a inclusão do registro <i>Historico</i> , caso já exista altera o registro incluindo novos eventos;
	fechaHistorico	Fecha o arquivo <i>Historico</i> ;
StopList	incluiTableHash	Carrega uma tabela <i>Hash</i> com os termos a serem excluídos na mensagem pela classe <i>StopList</i> ;
	tokening	Retira os caracteres especiais, símbolos e números das mensagens;
	buscaResposta	Busca os eventos de uma mensagem em um arquivo texto do formato <i>ascii</i> ;
ClassificaHistorico	abreHistorico	Abre o arquivo <i>Historico</i> , caso não existir cria-se um novo;
	leRegistroHistorico ImprimeRegistroHistorico	Lê um registro no arquivo <i>Historico</i> ; Lista os registros do arquivo <i>Historico</i> em um arquivo texto do formato <i>ascii</i> ;
	adicionaRegistroHistorico leParametro	Adiciona um registro no arquivo <i>Historico</i> ; Lê as probabilidades condicionais do arquivo <i>Parametro</i> e carrega em uma matriz para ser utilizado pelo método <i>classifica</i> ;
	classifica	Calcula as probabilidades das classes usando o classificador Naive Bayes;
	geraListaMedia	Gera as listas de Médias de pesos dos termos, ordena os termos em ordem decrescente pelo peso e apresenta-os em duas listas de sugestões;
	fechaHistorico	Fecha o arquivo <i>Historico</i> ;
Historico	setIdMensagem, setTermoMensagem, setExcluiMensagem, setEncaminhaMensagem, setRespondeMensagem, setMovePastaMensagem, setImprimeMensagem, setLeMensagem, setValorInferencia	Métodos mutatórios que atribuem valores para as variáveis de instância <i>Private</i> do registro do arquivo <i>Historico</i> ;
	getIdMensagem, getTermoMensagem, getExcluiMensagem, getEncaminhaMensagem, getRespondeMensagem, getMovePastaMensagem, getImprimeMensagem, getLeMensagem, getValorInferencia	Métodos de consultas para obtenção dos valores das variáveis de instância <i>Private</i> dos registros do arquivo <i>Historico</i> ;
	read	Lê um <i>RegistroHistorico</i> de um <i>RandomAccessFile</i>
	padName	substitui bytes nulos por espaços;
	write	Grava um <i>RegistroHistorico</i> de um <i>RandomAccessFile</i>
	writeName	Formata uma <i>string</i> para o tamanho de 20 caracteres;

## 4.4 Acoplado o AF no Ambiente de Correio Eletrônico do Direto

A interação do usuário com cada uma das etapas do modelo do AF deveria ser feita no formulário do correio eletrônico do Direto, conforme apresentado na fig. 4.2.

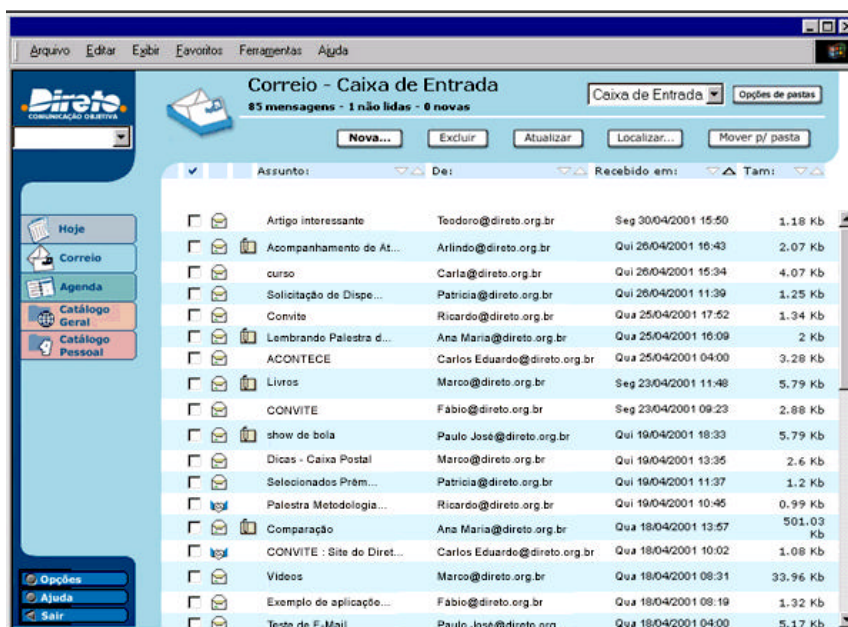


FIGURA 4.2 - Formulário do correio eletrônico do Direto

Como foi adiada a acoplagem do AF junto ao Direto, esta seção, de uma forma geral, mostrará as alterações básicas para permitir que o AF seja utilizado juntamente com o ambiente do Direto que poderá ser implementado em trabalhos futuros.

Os formulários usados na interação com o usuário e as mensagens são providos de recursos das linguagens *HTML* e *javascript*. Esses formulários contam com um conjunto de botões e âncoras que permitem ao usuário aplicar ações em uma determinada mensagem, como, por exemplo: para ler a mensagem o usuário deve clicar sobre uma âncora no formulário *Correio - Caixa de Entrada* (fig. 4.2); para mover uma mensagem para alguma pasta, utiliza-se um botão no formulário *Mover...* do formulário *Pasta Caixa de Entrada* ( fig. 4.3).

De forma geral, para chamar a classe *ColetaDadosHistorico*, os códigos dos botões devem receber comandos de chamadas para esse *applet* com passagens dos parâmetros: identificador de mensagens, termos oriundos do filtro de mensagens, código do evento correspondente ao botão, como, por exemplo, um para *Excluir*, dois para *Encaminhar*, três para *Responder*, quatro para *Mover para Pasta*, cinco para *Imprimir* e seis para *Ler* e finalmente o corpo da mensagem.

Seria criado um frame somente para o *applet ColetaDadosHistorico* que poderia ser inicializado juntamente com uma sessão do Direto. Para que este *applet* capture os dados necessários, seriam incluídas chamadas *javascript* em alguns formulários.

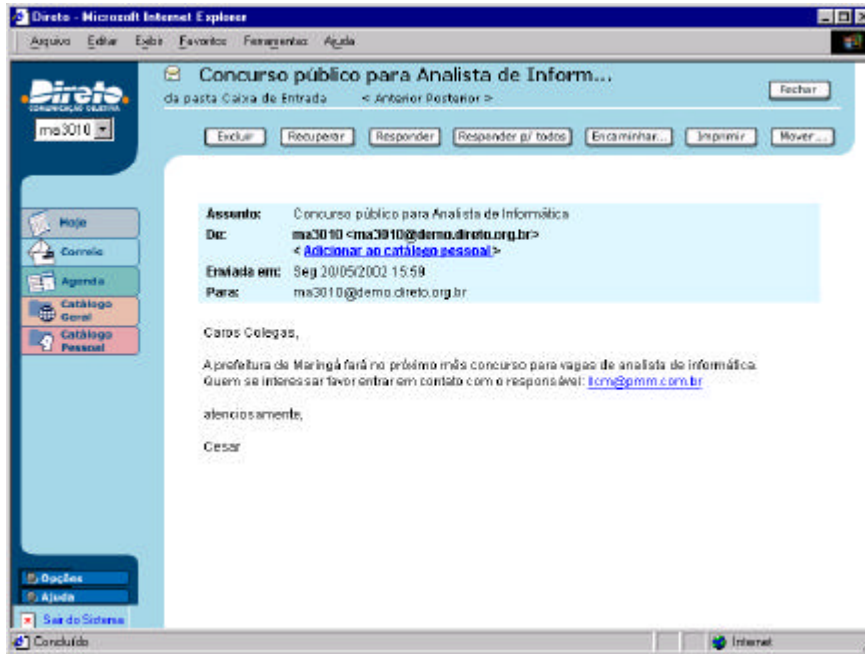


FIGURA 4.3 - Formulário da pasta caixa de entrada contendo a mensagem recebida

Por exemplo, a página que exibe o conteúdo de uma mensagem, quando o usuário estiver lendo a mesma, passaria as informações para o *applet* no evento, da seguinte forma:

```
<body onLoad=" javascript:chamadaColetaDados();">
```

A função *javascript* poderia ser implementada desta forma:

```
function chamadaColetaDados() {
    parent.frameApplet.applet1.ColetaDadosHistorico(idMensagem, termoFiltro,
6, corpoMensagem);
}
```

Nos botões como *excluir*, *responder*, *encaminhar*, *imprimir* e *mover para pasta* devem haver chamadas para o *applet ColetaDadosHistorico*.

Para a chamada de sugestões do AF, deve-se incluir no formulário *HTML/javascript* um botão, denominado 'Feedback', contendo códigos de chamada para o *applet ClassificaHistorico*, sem passagens de parâmetros.

A fig. 4.4 apresenta a estrutura do protótipo AF quando acoplado ao Direto.

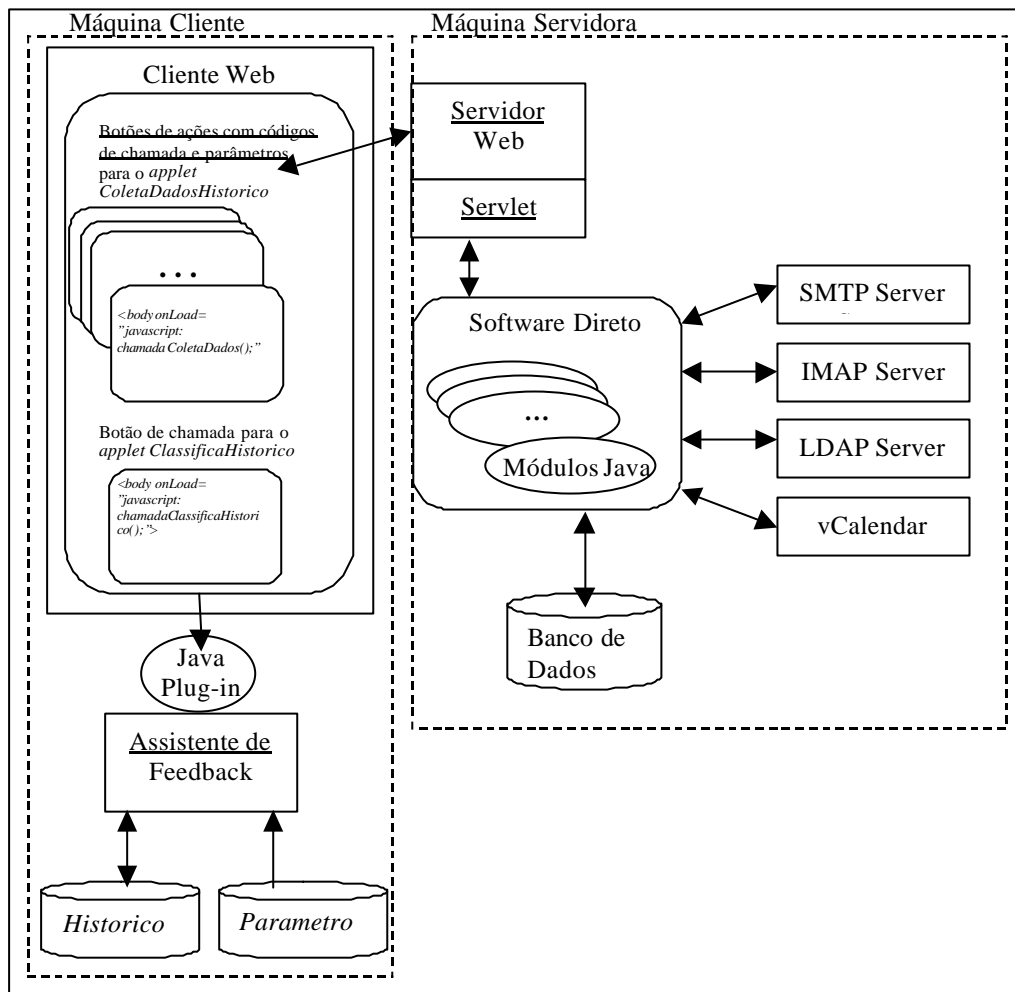


FIGURA 4.4 - Estrutura do AF acoplado ao Direto

## 4.5 Interface Gráfica de Sugestões do AF

O projeto da interface gráfica do AF foi influenciado pelas características recebidas durante a implementação e citado na seção 4.1. O AF trabalha em segundo plano de forma discreta com a ferramenta do correio eletrônico do Direto, utilizando como meio de entrada de dados a própria interface que o usuário utiliza para tratar as mensagens. A preocupação em definir o layout da interface gráfica levou a um esforço maior na etapa em que as sugestões do AF forem requisitadas pelo usuário.

A interface gráfica da ferramenta foi construída usando as classes *Swing* da Sun. Através destas classes, é possível desenvolver uma interface amigável e de fácil leitura pelo usuário. A interface da ferramenta é simples, composta de um painel com duas listas de palavras (termos), com suas médias de pesos calculadas na classe *ClassificaHistorico*. A lista superior contém termos que apresentam maior interesse pelo usuário e a inferior a que contém menor interesse. Na fig. 4.5 é apresentado um *snapshot* da tela de sugestões do AF.

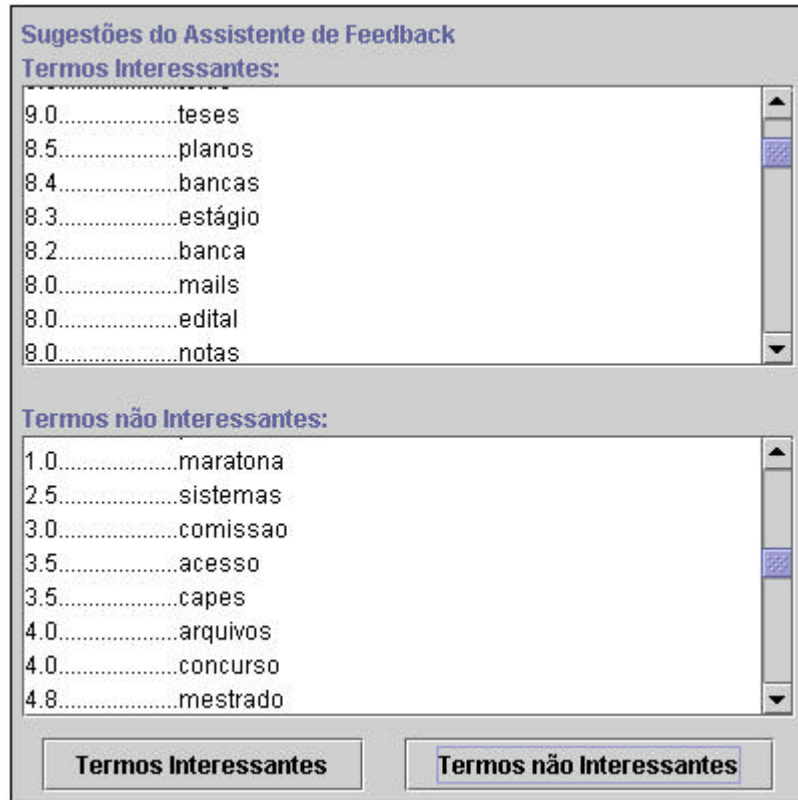


FIGURA 4.5 - Tela de sugestões do Assistente de *Feedback*

Cada termo apresenta um valor de média de peso em um formato legível para o usuário. Para interpretação das informações da interface, o usuário deve estar consciente que os termos os quais mostraram maior média, são aqueles que pertenciam a mensagens que receberam ações que representaram maior preferência do usuário. Os que mostraram menor média são aqueles que pertenciam a mensagens que receberam ações que representaram menor preferência do usuário.

## 4.6 Considerações Finais

Este capítulo descreveu a implementação do protótipo do modelo Assistente de *Feedback* para o ambiente do Direto. Esse protótipo, a princípio, foi construído para ser acessado pelo ambiente Direto utilizando um navegador compatível. Depois decidiu-se apresentar um protótipo com características genéricas, passivo de ser acoplado em qualquer ambiente com características similares às do Direto.

O resultado das etapas do modelo do Assistente de *Feedback* é apresentado no formulário com as listas de interesse dos termos de maior preferência e de menor preferência do usuário. Após a conclusão do mesmo, o próximo capítulo apresentará a avaliação do protótipo, a fim de que o modelo proposto e o protótipo construído pudessem ser validados.



## 5 Avaliação do Protótipo do Assistente de *Feedback*

Após a apresentação da implementação do protótipo do Assistente de *Feedback* (AF), realizou-se uma simulação por meio de um conjunto de mensagens apresentadas a um grupo de usuários a fim de validar a precisão do modelo proposto. Efetuaram-se também testes para avaliar o desempenho do AF quando o usuário solicitar sugestões do mesmo.

Este capítulo descreve a avaliação aplicada ao protótipo implementado, sendo que inicialmente é demonstrada a avaliação da precisão, descrevendo o ambiente, as ferramentas aplicadas, o método para a coleta dos dados, a forma como foram aplicados os experimentos e os resultados obtidos durante esse processo. Em seguida, é apresentada a avaliação do desempenho, descrevendo o ambiente, os dados utilizados, os experimentos e os resultados obtidos. Finalmente, é apresentada a análise dos resultados e as comparações com trabalhos similares.

### 5.1 Considerações Iniciais

Foi decidido primeiramente aplicar a avaliação através da simulação do ambiente de um correio eletrônico como o Direto, concentrando-se nos testes do classificador para validar a precisão do AF, pelos seguintes motivos:

- ◆ sem medir a precisão do classificador do AF, fica impossível avaliar o grau de confiança dos retornos de sugestões do AF;
- ◆ como não foi possível acoplar o AF no Direto para execução dos testes em ambiente real, não se possibilitou instalar um servidor de mensagens e criar canais de notícias, centralizando vários usuários para validação do protótipo;
- ◆ as avaliações que serão aplicadas são bastante utilizadas pelas literaturas como em [BIL 96], [PAY 95a], [PAZ 97], [PAZ 2000] e [SEG 99].

Considerou-se também como forma de avaliação o desempenho do AF, no instante em que o usuário solicita as sugestões. Essa etapa foi escolhida porque exige uma significativa carga de processamento pelo ambiente cliente e o uso do AF em condições reais, e que pode-se tornar impraticável pelo usuário. Por esse motivo tornou-se importante medir o tempo de resposta para receber as sugestões do AF. Tal validação foi submetida em várias configurações de *hardware* no ambiente cliente, com o objetivo de medir o tempo de execução do AF e sugerir a configuração mínima necessária para execução do protótipo.

### 5.2 Avaliação da Precisão do AF

Esta seção apresentará os detalhes do experimento e os resultados dos testes de precisão do AF com o objetivo de mensurar a qualidade da sua inferência.

### 5.2.1 Ambiente Utilizado nos Testes

Para realização dos testes o protótipo foi instalado sobre os seguintes ambientes de *hardware* e *software*:

- ◆ **hardware**: Microcomputador com processador K6-2 475 Mhz com 64 Mb RAM e HD de 10 Gb;
- ◆ **sistema operacional**: MS Windows NT 4 (SP 6).

O ambiente utilizado foi selecionado em função da disponibilidade de meios e recursos no período em que se realizaram os testes.

### 5.2.2 Coleta e Preparação dos Dados

O componente central do AF é o algoritmo de classificação usado pelo protótipo para inferir a preferência do usuário sobre os termos das mensagens. Para testá-lo dentro do domínio que o AF pretende ser aplicado, utilizaram-se 111 mensagens extraídas de uma caixa de correio eletrônico real. Essas mensagens foram originadas de seis diferentes listas de mensagens, as quais foram rotuladas por tópicos. Esses tópicos estão listados na tab. 5.1, com os totais de mensagens e sua descrição.

TABELA 5.1- Características de diferentes listas de mensagens utilizadas na experimentação

Tópico	Número	Descrição
Biblioteca UFRGS	30	Mensagens emitidas pela direção da biblioteca da UFRGS para as listas dos mestrados;
Coordenador Mestrado UFRGS	11	Mensagens emitidas pelo coordenador do mestrado da UFRGS para as listas dos mestrados;
Secretaria Mestrado UFRGS	20	Mensagens emitidas pela secretaria da pós-graduação da UFRGS para as listas dos mestrados ;
Estágio Supervisionado Graduação	21	Mensagens emitidas pelo coordenador do estágio supervisionado do curso de tecnólogo em Processamento de Dados do CESUMAR;
Coordenação Graduação	22	Mensagens emitidas pelo coordenador do curso de tecnólogo em Processamento de Dados do CESUMAR;
Centro de Empreendimentos	7	Mensagens emitidas pelos membros do Centro de Empreendimentos do Instituto de Informática da UFRGS;

Para obtenção dos dados, as mensagens foram apresentadas para um grupo de quatro usuários em forma de questionário, contendo como opções de respostas as ações (eventos) existentes no ambiente do correio eletrônico do Direto e as classes utilizadas pelo AF. O objetivo desse questionário é obter os eventos que os usuários desejariam aplicar sobre as mensagens e a classe que representaria a preferência do usuário sobre a mesma. Desta forma, conseguiria-se rotular, por intermédio das classes, um conjunto de eventos associado aos termos extraídos das mensagens. O anexo I mostra alguns exemplos das mensagens apresentadas para o grupo de usuários.

Para aplicar o questionário junto aos usuários, o especialista, representado pelo autor deste trabalho, explicou a importância do trabalho e a forma como deveriam assinalar as respostas. As restrições impostas pelo especialista para selecionar as opções de respostas foram:

- ◆ o usuário poderia assinalar uma ou mais ações para cada mensagem;
- ◆ o usuário poderia assinalar apenas uma classe para representar o interesse sobre a mensagem.

Após cada usuário ter respondido o questionário, os resultados foram tabulados para cada usuário e criou-se um arquivo que segue o padrão para entrada de dados dos algoritmos do AF. Esse padrão define que o arquivo deve ser *ascii*, e tabulado, ou seja, contendo as colunas: Identificador da Mensagem; Identificador do Usuário, seis colunas com valores binários que representam a presença (valor 1) ou ausência (valor 0) das ações do usuário (*Excluir*, *Encaminhar*, *Responder*, *MoverPasta*, *Imprimir*, *LerCorpo*) e o *Peso* que representa a classe. Um exemplo do arquivo *ascii* contendo as respostas tabuladas, é apresentado no anexo II.

Do mesmo modo, as 111 mensagens são agrupadas em um único arquivo-texto denominado *Mensagens*, contendo o identificador da mensagem, assunto e corpo.

### 5.2.3 Escolha dos Casos para Treinamento

Como os eventos executados pelo usuário no correio eletrônico determinam o interesse do usuário pelo assunto da mensagem, observou-se neste estudo que, em geral, os usuários do correio eletrônico utilizam-se das mesmas ações para expressar algum interesse pelo assunto da mensagem. Através dessa observação decidiu-se definir um perfil genérico para os usuários representados pela tabela de parâmetro numérico, que será gerada por um conjunto de casos de treinamento.

Os casos de treinamento, por conterem variedades de ocorrências de eventos relacionados aos pesos de interesse sobre as mensagens, dão condições de construir parâmetros que representarão, de forma satisfatória, o domínio de interesse do ambiente que será aplicado no AF.

Utilizando-se o conhecimento do autor sobre o domínio do ambiente do correio eletrônico, verificaram-se as respostas contidas nos quatro arquivos de respostas *ascii* e escolheram-se 25 casos que representarão a base de conhecimento do AF. Foi considerada como critério para a escolha de um determinado caso a sua alta frequência em todos os arquivos de resposta. A tab. 5.2 mostra os casos que foram utilizados como base de conhecimento do AF.

TABELA 5.2 - Conjunto de casos utilizados para o treinamento do AF

Casos	Excluir	Encaminhar	Responder	Mover	Imprimir	Ler	Peso da
				Pasta		Corpo	Classe
1	1	0	0	0	0	0	1
2	0	0	0	1	0	0	3
3	0	0	0	0	1	0	5
4	0	1	0	0	0	1	7
5	0	1	0	1	0	0	5
6	1	1	0	0	0	1	5
7	1	0	0	0	0	1	7
8	1	0	0	1	0	1	5
9	0	1	0	0	0	0	3
10	0	0	1	0	0	0	3
11	0	1	0	0	1	0	5
12	0	0	0	0	0	1	5
13	0	0	0	0	1	1	7
14	0	0	0	1	0	1	7
15	0	0	1	0	0	1	7
16	0	0	1	1	0	1	7
17	0	1	0	1	1	0	5
18	0	1	0	0	1	1	7
19	1	0	1	0	0	1	7
20	0	0	0	1	1	1	9
21	0	0	1	1	1	1	9
22	0	1	0	1	0	1	9
23	0	1	1	1	1	1	9
24	1	1	1	1	1	1	9
25	0	0	0	1	1	0	7

### 5.2.4 Geração dos Parâmetros

Foi executado o algoritmo de aprendizagem, para que, a partir dos casos de treinamento, fossem gerados os parâmetros numéricos que representarão o conhecimento *a priori* sobre o domínio do modelo para avaliação deste trabalho.

Para isso foi criado um arquivo que segue o padrão de entrada do algoritmo de aprendizagem do AF. Esse padrão define que o arquivo deve ser *ascii* e tabulado contendo sete colunas com a associação dos eventos com o peso da classe. Em seguida, executou-se o método *geraParametro* contido na *ClassificaHistorico* sobre o arquivo-texto gerado (veja código-fonte no anexo III). Essa execução criou um arquivo, denominado *parametro*, contendo as probabilidades *a priori* das classes e as

probabilidades a *posteriori* dos eventos, dada uma determinada classe do conjunto de classes (A,B,C,D e E). Esses valores podem ser vistos na tab. 5.3.

TABELA 5.3 - Parâmetros numéricos gerados pelo algoritmo de aprendizagem

P(evento Classe)	Valores P(evento C <sub>i</sub> )				
	A	B	C	D	E
P(C <sub>i</sub> )	0.2	0.36	0.28	0.12	0.04
P(Excluir C <sub>i</sub> )	0.2	0.223	0.285	0.02	1.0
P(Encaminhar C <sub>i</sub> )	0.6	0.223	0.571	0.334	0.02
P(Responder C <sub>i</sub> )	0.6	0.334	0.020	0.334	0.02
P(MoverPasta C <sub>i</sub> )	1.0	0.334	0.428	0.334	0.02
P(Imprimir C <sub>i</sub> )	0.8	0.334	0.428	0.02	0.02
P(Ler C <sub>i</sub> )	1.0	0.889	0.428	0.02	0.02
P(NãoExcluir C <sub>i</sub> )	0.8	0.777	0.715	1.0	0.02
P(NãoEncaminhar C <sub>i</sub> )	0.4	0.777	0.429	0.666	1.0
P(NãoResponder C <sub>i</sub> )	0.4	0.666	1.0	0.666	1.0
P(NãoMoverPasta C <sub>i</sub> )	0.02	0.666	0.572	0.666	1.0
P(NãoImprimir C <sub>i</sub> )	0.2	0.666	0.572	1.0	1.0
P(NãoLer C <sub>i</sub> )	0.02	0.111	0.572	1.0	1.0

### 5.2.5 Simulação do AF no Ambiente de Correio Eletrônico

Uma vez que a base inicial está definida, o sistema está pronto para prever as classes dos casos exemplos. Um caso exemplo é caracterizado pelos eventos executados por um usuário, associados aos termos extraídos das mensagens.

Para criar os casos exemplos, executou-se a classe *ColetaDadosHistorico* uma única vez para cada arquivo *ascii*, contendo as respostas de cada usuário com o arquivo *Mensagens*. Como resultado obtiveram-se quatro arquivos *Historico*, um para cada usuário. Cada arquivo *Historico* contém 389 termos (casos exemplos) mais relevantes das mensagens com os eventos executados e os pesos das classes escolhidas pelos usuários, vindos dos arquivos *ascii* de respostas. Utilizou-se a classe *StopList* para extrair os termos mais relevantes. Esta classe utiliza como critério para extração, as maiores frequências dos termos que não estão presentes na lista de 3706 palavras não relevantes (veja código-fonte no anexo IV). Esses arquivos *Historicos*, foram reservados para futuras comparações.

Em seguida, aplicou-se o algoritmo de classificação para cada registro do arquivo *Historico*, executando o método *classifica* contido na classe *ClassificaHistorico* (veja código fonte no anexo V). Usaram-se os parâmetros numéricos da tab. 5.3.

Os resultados dos pesos classificados nos arquivos agora nomeado *Historico\_classificado* (veja arquivo classificado *Historico\_classificado* impresso no anexo VI) foram comparados com os pesos nos arquivos *Historicos* reservados. Foram

computados os resultados que apresentavam valores iguais e foram assinalados os acertos. Finalmente são computados os erros cometidos pelo classificador.

A tab. 5.4 mostra os resultados dos percentuais de acertos e os percentuais de erros, todos agrupados por usuário.

TABELA 5.4 - Resultados percentuais da precisão do AF

	Usuário A	Usuário B	Usuário C	Usuário D	Média Final
Acertos	67.87	75.84	61.70	69.41	68.70
Erros	32.13	24.16	38.30	30.59	31.30

### 5.2.6 Análise dos Resultados

Nesse teste preliminar de precisão, observou-se que o AF apresentou resultados médios satisfatórios para a inferência do classificador. Observou-se uma média final de 68,70% de acerto com 31,30% de erros na predição. Esses valores foram obtidos com uso de um pequeno domínio de casos. Porém, se fosse utilizado um grande grupo de usuários para aumentar o número de ocorrências nos casos de treinamento, os resultados do experimento poderiam ser melhorados.

A precisão do AF, se for comparada com outros protótipos similares, apresentou valores inferiores nos seus percentuais. Este fato não invalida o modelo porque o AF utiliza 5 classes, as quais representam cada uma um grau de interesse do usuário. Diante disso, os resultados do AF durante a classificação podem se aproximar do limite da classe correta, mas não entrar no intervalo da mesma. Outro fator que pode ter influenciado nesses valores percentuais é que alguns eventos contidos nos dados de treinamento apresentaram pouca ou nenhuma ocorrência, colaborando de alguma forma para que os percentuais de acertos diminuíssem.

Por outro lado, considerando os acertos, o AF apresentou resultados satisfatórios nas sugestões, principalmente por utilizar médias de pesos para apresentar as sugestões dos termos.

A fig. 5.1 mostra os resultados finais das sugestões para os quatro usuários. Os termos apresentados nas listas de termos interessantes são os que contiveram maiores médias de peso. As listas de termos não interessantes mostram os termos que contiveram menores médias de peso.

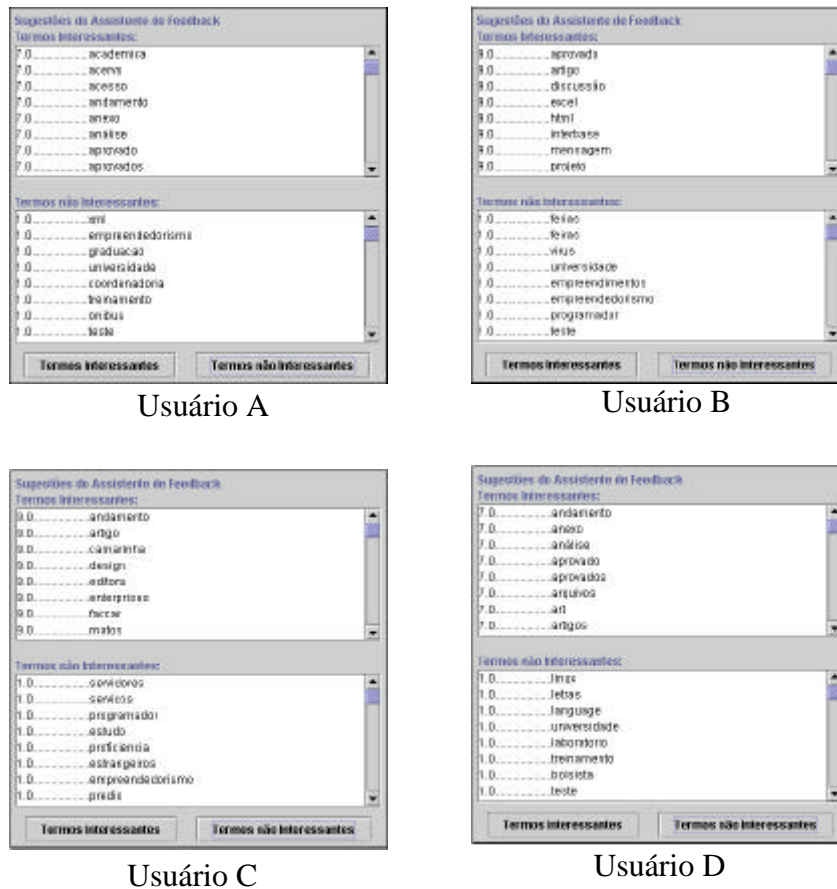


FIGURA 5.1 - Resultado das sugestões de cada usuário

Um problema encontrado na construção desse tipo de sistema é a obtenção de casos para construção da base de conhecimento utilizada pelo classificador. Um sistema bayesiano pode aprender a partir de um conjunto de casos e, dessa forma, facilitar bastante a interface com o especialista. Porém a tarefa do especialista nem sempre é simples, porque existe uma certa dificuldade na escolha desse conjunto de casos que será responsável pela precisão do classificador.

## 5.3 Avaliação de Desempenho do AF

Esta seção apresentará os detalhes do experimento, que tem como objetivo avaliar o protótipo implementado considerando o desempenho da etapa de classificação e *Feedback* em várias configurações de *hardware*. Para esse teste foram especificados alguns arquivos contendo termos e foram submetidos a uma quantidade de execuções em caráter experimental.

### 5.3.1 Ambiente Utilizado nos Testes

Para realização dos testes o protótipo foi instalado sobre os seguintes ambientes de *hardware*, *software* e comunicação:

- ◆ **Hardware (cliente-1):** Microcomputador com processador *Pentium* 200 Mhz com 96 Mb RAM e HD de 4 Gb SCSI;
- ◆ **Hardware (cliente-2):** Microcomputador com processador K6 475 Mhz com 64 Mb RAM e HD de 10 Gb IDE;
- ◆ **Hardware (cliente-3):** Microcomputador com processador *Athlon* 900 Mhz com 256 Mb RAM e HD de 10 Gb IDE;
- ◆ **Hardware (servidor):** Microcomputador com processador K6 475 Mhz com 64 Mb RAM e HD de 10 Gb IDE;
- ◆ **Sistema Operacional:** Todas as máquinas instaladas com MS Windows NT 4 (SP 6);
- ◆ **Outros software:** *Apache* como servidor Web; Plug-in Java 1.3.1.02 da Sun Microsystems; MS Windows Internet-Explorer 5.5 SP2;
- ◆ **Comunicação:** Rede Internet com protocolo TCP/IP.

O ambiente utilizado foi selecionado em função da disponibilidade de meios e recursos no período em que se realizaram os testes.

### 5.3.2 Dados Considerados nos Testes

Para realização do teste de desempenho, aqui descrito, utilizaram-se sete arquivos *Historico* contendo em cada um as seguintes quantidades de casos exemplos: 1000, 2000, 3000, 4000, 5000, 6000 e 7000, sendo que os dados contidos nestes casos foram escolhidos aleatoriamente.

### 5.3.3 Testes e Resultados dos Experimentos

Para testar o desempenho do AF, usaram-se as três configurações de *hardware*, como cliente. Em cada máquina, executou-se o *browser* chamando 10 vezes o *Applet ClassificaHistorico* para cada arquivo *Historico*. Tabulou-se para cada execução o tempo gasto em milésimos de segundos para apresentar as sugestões no *browser* e calculou-se a média final. Os resultados obtidos são apresentados na tab. 5.5.

TABELA 5.5 - Tempos médios para execução do AF em milésimos de segundos

	1000	2000	3000	4000	5000	6000	7000
<b><i>Pentium</i> 200</b>	9566.7	21062.1	36759	45896	60607.3	80166.1	98355.5
<b>K6-2 475</b>	2543.5	5284.5	10009.5	14346.8	17852.6	24697.4	31007.6
<b>K7 <i>Athlon</i></b>	631.9	1313.9	2117.1	3135.4	4825	7167.4	9747.9

### 5.3.4 Análise dos Resultados

Observou-se por meio desse experimento que máquinas com processadores de baixo desempenho, como o *Pentium* 200, apresentaram tempo médio gasto impraticável para



o usuário receber as sugestões do AF. Conforme a quantidade de casos exemplos aumenta, cresce o tempo de espera, ficando ainda mais inviável para o uso. Já para as duas últimas configurações de máquinas, o tempo de espera ficou tolerável para o uso do AF.

Conclui-se que a classificação do AF exige uma certa configuração mínima para apresentar um tempo de retorno satisfatório, porque o classificador exige um certo custo de processamento para inferir as classes e apresentar a interface com as sugestões, mesmo que o processo de aprendizagem não seja executado nessa etapa avaliada.

Na fig. 5.2 é mostrado o gráfico contendo os resultados do tempo de espera das três máquinas pela quantidade de casos exemplos em cada arquivo *Historico*.

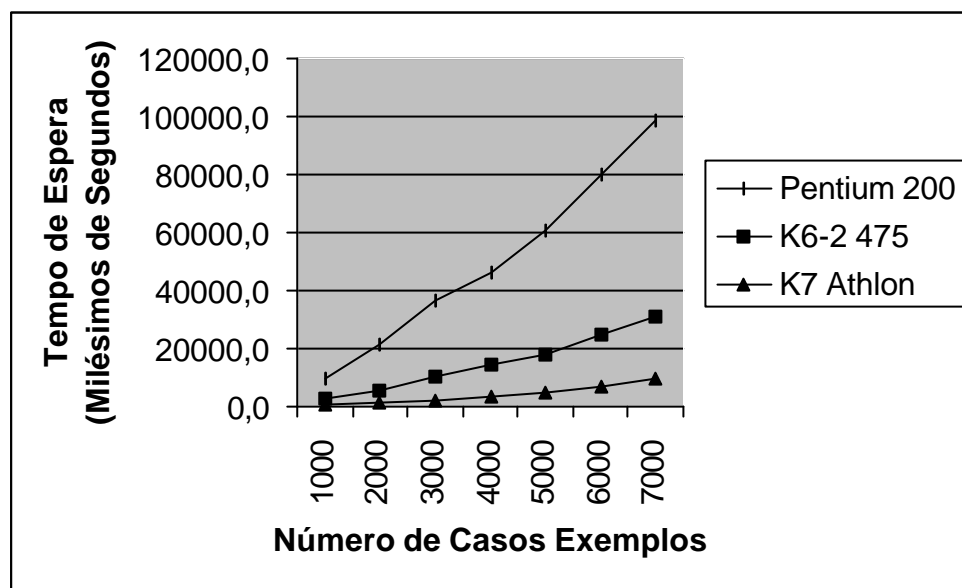


FIGURA 5.2 - Gráfico dos tempos médios das máquinas pelo número de registro

## 5.4 Comparação com Trabalhos Similares

Comparou-se o AF com sistemas similares, conforme descritos na seção 2.8, que utilizam técnicas de modelos de usuários com a finalidade de assistir os mesmos na tarefa de organizar ou filtrar mensagens de correio eletrônico. Observa-se que o modelo proposto neste trabalho pode apresentar boas características que validam este trabalho, tais como:

Em [SAH 98], os autores admitiram que na tentativa de executar o sistema utilizando três classes, os resultados mostraram-se com menores precisões na filtragem de mensagens. O AF utiliza cinco classes para representar a preferência do usuário com os termos. Apesar de não apresentar resultados nos níveis dos sistemas que utilizam apenas duas classes como (*interessante e não interessante*), em contrapartida o AF permite apresentar mais estados de preferência do usuário com a mensagem, ou seja, apresenta mais níveis de interesse como (*NãoGostou, GostouMuitoPouco, GostouPouco, Gostou, GostouMuito*).

Em sistemas descritos em [SAH 98] e [REN 2000], o uso das pastas de mensagens pode tornar o sistema de filtragem mais prático e interativo com o usuário. Porém, em

sistemas de filtragens que utilizam pastas como classes nem sempre o usuário consegue expressar qual assunto se refere a um determinado grupo de mensagens a fim de rotular uma pasta para armazenar este grupo de mensagens. Isso pode dificultar o processo de treinamento do classificador. O usuário pode definir pastas com rótulos que não condizem com o assunto das mensagens. O AF não depende do usuário para nomear classes para um grupo de mensagens. Pela simplicidade de uso, o usuário precisa somente aplicar ações às mensagens recebidas, para que o AF possa classificá-las. Como as ações são mais estáveis, o classificador poderá prever com mais segurança as classes.

Em sistemas como *Maxims* [PAT 94], o processo de aprendizagem pode tomar algum tempo para fazer seu algoritmo de aprendizagem ganhar bastante experiência para ser útil. O AF utiliza ações para expressar as preferências dos usuários sobre as mensagens. Para esse objetivo, as ações se tornam mais estáveis e, em geral, os usuários aplicam ações semelhantes para um determinado tipo de interesse por uma mensagem. Por esse motivo, o AF utiliza um perfil genérico para a base de conhecimento do classificador e, dessa forma, não necessita de um tempo mínimo para adquirir conhecimento a fim de melhorar a precisão das classificações.

Outra vantagem do AF é a utilização de pesos para representar o interesse dos termos contidos nas mensagens. Alguns trabalhos mostram bons resultados, utilizando como forma de apresentação os resultados das inferências médias de pesos e representando o grau de preferência sobre um determinado termo. Em [GOL 96] apresenta-se um *software* MUSAG que constrói um dicionário de expressões para um dado conceito específico e outro agente, nomeado SAG, fica responsável para encontrar os documentos, dado um conjunto de palavras-chaves com pesos relativos, treinado pelo MUSAG. Em outro trabalho em [VEL 98], são utilizados pesos para um sistema de agente com o objetivo de sugerir novos documentos pelos perfis dos usuários.

Outro fator interessante que o AF apresenta é o uso do classificador Naive Bayes como responsável pela inferência das preferências dos usuários. Muitos outros trabalhos utilizam esse classificador como em [AND 2000], [BIL 96], [PAZ 96], [PAZ 97] e [REN 2000]. Todos apresentaram resultados satisfatórios na predição do perfil dos usuários.

## 5.5 Considerações Finais

Neste capítulo, foram descritas as avaliações do protótipo de Assistente de *Feedback* para o ambiente Direto. Os experimentos utilizados na avaliação tinham como objetivo medir a precisão do AF, bem como o seu desempenho. Os resultados obtidos na avaliação de precisão apresentaram resultados satisfatórios comparados com outros sistemas similares. Os resultados poderiam ter sido melhores caso fosse utilizado um domínio maior de mensagens e um grupo maior de usuários. Foi também avaliado o desempenho do AF no retorno de sugestões para usuário. Verificou-se que o protótipo tem um tempo de retorno de sugestões toleráveis a partir de configurações mínimas de *hardware*. No próximo capítulo serão apresentadas as conclusões finais do trabalho bem como os projetos futuros, que, no entendimento deste autor, poderão ser desenvolvidos a partir deste trabalho.

## 6 Conclusão

### 6.1 Considerações Finais

Este trabalho destinou-se a encontrar uma solução para os usuários que utilizam os serviços de filtragens e precisam preencher os parâmetros do filtro com seu perfil de preferência, para as informações serem filtradas durante a utilização do correio eletrônico de sistemas como o Direto. Os parâmetros do filtro do Direto são conjuntos de regras geradas em forma de *scripts* a partir de opções selecionadas por intermédio de uma interface com o usuário. Essas *scripts* são enviadas para o servidor do Direto para o uso do serviço de filtragem. Os usuários que usam os filtros deparam frequentemente com o problema de fazer ajustes contínuos nos parâmetros, na medida em que suas preferências vão se modificando com o passar do tempo. Este problema é de grande importância, dado o grande volume de informações disponíveis por meio da WWW. Mesmo pelo tempo e esforço gasto pelos usuários, muitas vezes eles não conseguem expressar precisamente o que querem filtrar e os serviços de filtragem não conseguem se adaptar com a estratégia de trabalho de cada usuário.

Outros estudos já desenvolvidos realizam a classificação de informações com o objetivo de auxiliar o usuário a organizar e filtrar mensagens vindas da WWW. Essas aplicações possuem as mais variadas finalidades. Algumas classificações são baseadas no conhecimento das preferências de cada usuário. Entretanto, estas preferências deveriam ser atualizadas por um mecanismo de aprendizagem contínua, conforme a mudança do perfil do usuário.

Diante disso, desenvolveu-se um modelo Assistente de *Feedback* (AF) para o serviço de filtragem do Direto, com o objetivo de resolver os problemas apresentados. O AF permite auxiliar os usuários a parametrizar o filtro de mensagens do Direto por meio das suas preferências representadas pelas ações aplicadas no correio eletrônico.

O capítulo 2 apresentou os principais conceitos básicos de probabilidades e o teorema de Bayes para a compreensão do classificador Naive Bayes que foi utilizado no AF. Também no capítulo 2, procurou-se descrever as características gerais dos classificadores e em especial o Naive Bayes, sua lógica e seu desempenho comparado a outros classificadores. Foram abordados também conceitos relacionados ao modelo de perfil de usuário e o ambiente Direto. Observou-se que o Naive Bayes tornou-se atraente para ser utilizado no AF, por apresentar bom desempenho sobre os demais classificadores e por ser eficiente na predição, quando os atributos são independentes entre si, como ocorre com o modelo implementado neste trabalho.

O modelo Assistente de *Feedback* foi apresentado no capítulo 3. Foi detalhada a sua estrutura global, descrevendo cada uma das etapas, e a forma como foram ajustadas para os objetivos pretendidos.

O protótipo implementado para automação do modelo proposto foi apresentado no capítulo 4. Detalhou-se sua estrutura geral, a implementação, utilizando os recursos da linguagem Java, as classes principais e classes internas, os métodos e os arquivos utilizados para armazenamento dos dados.

Os testes para avaliação do modelo e os resultados foram abordados no capítulo 5. Os resultados obtidos na avaliação de precisão apresentaram valores satisfatórios,

considerando o uso de cinco classes pelo classificador do AF. Concluiu-se que um problema encontrado na construção desses sistemas é a obtenção de casos para construção da base de conhecimento utilizada pelo classificador. A utilização de um especialista para ajustes da base de conhecimento nem sempre é simples. Um sistema bayesiano pode aprender a partir de um conjunto de dados e, dessa forma, facilitar bastante a interface com o especialista. Porém, a escolha desse conjunto de casos é importante para o resultado da precisão do classificador. Os resultados dos testes de desempenho permitiram observar que se forem utilizadas máquinas de configurações obsoletas, o tempo de espera para os retornos de sugestões do AF pode, de alguma forma, aumentar ao ponto de se tornar impraticável com o crescimento da base de dados. Conclui-se que o usuário, utilizando o AF em máquinas com configurações mais atualizadas, conseguirá receber sugestões com tempos de respostas mais toleráveis.

As principais contribuições deste trabalho foram:

- ◆ implementação e validação de um protótipo que é instalado no computador do usuário para monitorar as preferências representadas pelas ações aplicadas nas mensagens do correio eletrônico, atingindo o objetivo de assistir o usuário, nas seleções de regras para os ajustes dos parâmetros do filtro de mensagens de ambientes como o Direto;
- ◆ análise do domínio do serviço de e-mail do Direto, no qual identificaram-se padrões de comportamento pelas ações dos usuários com as mensagens que representam as preferências e não preferências do usuário. Foi utilizado também como meio de simulação para o mecanismo de aprendizagem de perfil de usuários;
- ◆ pesquisa das técnicas e modelos baseados na estatística Bayesiana, identificando o classificador Naive Bayes como a mais apropriada para inferir/classificar o comportamento de usuários dentro do domínio da proposta;
- ◆ especificação de uma interface do Assistente de *Feedback* para o usuário visualizar as estatísticas das preferências, sem exigir do usuário conhecimento prévio sobre técnicas de interpretações estatísticas de processos de classificação. As sugestões são apresentadas de forma simples para facilitar o procedimento de mudança dos parâmetros do filtro de mensagens.

## 6.2 Trabalhos Futuros

Os trabalhos futuros provenientes desta dissertação estão relacionados com o modelo descrito no capítulo 3.

Pode ser modelado como trabalho futuro um avaliador de confiança da predição. Dessa forma, poderia-se criar um modelo que disponibilize ao usuário avaliar a predição, onde se verificaria a precisão do assistente e permitiria melhorias nos parâmetros para se adequar à realidade das ações dos usuários.

Outras frentes de pesquisas sugeridas seriam as inclusões de outros tipos de classificadores, a fim de avaliar a precisão e desempenho em condições reais de uso, bem como acoplar o AF no ambiente Direto e também a aplicação do assistente, em outros ambientes com características funcionais diferentes do Direto.

## Anexo 1 Exemplo de Mensagens do Questionário

CA 03

Assunto: [mestrado] ERRATA > Re: [Profpos] Entrega de dissertações/teses  
De: Carla Dal Sasso Freitas <carla@inf.ufrgs.br>

> Respeitados os padrões de qualidade, é óbvio, as defesas em dezembro são  
> duplamente benéficas para as estatísticas de nosso programa de PG: são  
agregadas  
> ao nosso índice de TITULADOS NO ANO de 2001 e, para os alunos, que ingressaram  
> em 2000, reduzem nosso TEMPO MÉDIO DE TITULAÇÃO, que é bastante alto em  
relação  
> aos padrões desejáveis.

Isto para MESTRANDOS que ingressaram em 2000 E doutorandos que ingressaram após  
1996.

Carla

Carla Maria Dal Sasso Freitas  
Instituto de Informatica - UFRGS  
Caixa Postal 15064  
91501-970 Porto Alegre, RS Brasil

<b>A-GostouMuito</b>	<b>B-Gostou</b>	<b>C-GostouPouco</b>	<b>D-GostouMuitoPouco</b>	<b>E-NãoGostou</b>
<b>1-Excluir</b>	<b>2-Encaminhar</b>	<b>3-Responder</b>	<b>4-MoverPasta</b>	<b>5-Imprimir</b>
				<b>6-LerCorpo</b>

CA 04

Assunto: [mestrado] Semana Acadêmica  
De: Carlos Alberto Heuser <heuser@inf.ufrgs.br>

Em nome da Coordenação do Programa, gostaria de agradecer a todos,  
professores e alunos, pela participação na Semana Acadêmica que hoje  
encerramos.

Um agradecimento especial vai para as Professoras Ingrid e Lúcia Lisboa  
e à Lourdes que fizeram todo trabalho de programação e organização da  
Semana Acadêmica.

Tivemos mais de 60 apresentações, com poucas faltas ou atrasos e tenho  
certeza de que todos tiraram proveito do evento.

Heuser  
(pela Coordenação do Programa)

<b>A-GostouMuito</b>	<b>B-Gostou</b>	<b>C-GostouPouco</b>	<b>D-GostouMuitoPouco</b>	<b>E-NãoGostou</b>
<b>1-Excluir</b>	<b>2-Encaminhar</b>	<b>3-Responder</b>	<b>4-MoverPasta</b>	<b>5-Imprimir</b>
				<b>6-LerCorpo</b>

## Anexo 2 Arquivo *ascii* com as Respostas Tabuladas

*Este arquivo contém os dados das respostas do usuário-B. O arquivo apresenta nove colunas, com os seguintes valores: coluna-1 contém os identificadores das mensagens, coluna-2 contém as iniciais do nome do usuário-B, coluna-3 a coluna-8 contém as respostas das ações do usuário-B aplicadas sobre a mensagem e a coluna-9 contém o peso que representa o grau de interesse do usuário-B sobre a mensagem.*

```

bi01 fa 0 0 0 1 0 0 3
bi02 fa 1 0 0 0 0 0 1
bi03 fa 1 0 0 0 0 0 1
bi04 fa 1 0 0 0 0 0 1
bi04 fa 1 0 0 0 0 0 3
bi06 fa 1 0 0 0 0 0 3
bi07 fa 0 0 0 0 1 1 7
bi08 fa 0 0 0 1 0 1 7
bi09 fa 1 0 0 0 0 0 1
bi10 fa 1 0 0 0 0 1 5
bi11 fa 1 0 0 0 0 0 1
bi12 fa 0 0 0 1 1 0 7
bi13 fa 1 0 0 0 0 0 1
bi14 fa 0 0 0 0 1 0 5
bi15 fa 0 0 0 1 0 0 7
bi16 fa 1 0 0 0 0 0 1
bi17 fa 1 0 0 0 0 0 1
bi18 fa 0 0 0 0 1 0 5
bi19 fa 1 0 0 0 0 0 1
bi20 fa 1 0 0 0 0 0 1
bi21 fa 0 0 0 1 0 1 3
bi22 fa 0 0 0 1 1 1 9
bi23 fa 1 0 0 0 0 0 1
bi24 fa 1 0 0 0 0 0 1
bi25 fa 1 0 0 0 0 0 1
bi26 fa 0 0 0 1 0 1 3
bi27 fa 1 0 0 0 0 0 1
bi28 fa 0 1 0 0 0 1 5
bi29 fa 1 0 0 0 0 1 5
bi30 fa 1 0 0 0 0 1 3
cm01 fa 1 0 0 0 0 1 5
cm02 fa 0 1 0 1 0 1 9
cm03 fa 0 0 0 1 0 1 7
cm04 fa 0 0 0 1 1 1 9
cm05 fa 0 0 0 1 1 1 9
cm06 fa 1 0 0 0 0 1 7
cm07 fa 0 0 0 1 0 1 9
cm08 fa 0 0 0 1 0 1 9
cm09 fa 1 0 0 0 0 1 5
cm10 fa 0 0 0 1 0 1 9
cm11 fa 1 0 0 0 0 1 5

```

### Anexo 3 Código Fonte do Método *geraParametro*

*// gera os parametros usando um conjunto de casos pré-classificados gravados no historico*

```
public void geraParametro()
{
    int cont = 1;
    String peso[] = {"9.0", "7.0", "5.0", "3.0", "1.0"};
    String array[] = {" ", " ", " ", " ", " ", " ", " ", " ", " ", " ", " "};
    for (int i=0; i<13; i++){
        for (int j=0; j<5; j++){
            mat_param[i][j]=0.0;
        }
    }
    leRegistroHistorico(array, cont);
    while (array[0].compareTo(" ") != 0) {
        for (int i=0; i<5; i++){
            if (array[8].trim().compareTo(peso[i]) == 0){
                mat_param[0][i] = mat_param[0][i] + 1;
                for (int j=1; j<7; j++){
                    if (array[j+1].compareTo("1") == 0){
                        mat_param[j][i] = mat_param[j][i] + 1;
                    }
                    else
                        mat_param[j+6][i] = mat_param[j+6][i] + 1;
                }
            }
        }
        cont++;
        leRegistroHistorico(array, cont);
    }
    for (int i=1; i<13; i++){
        for (int j=0; j<5; j++){
            mat_param[i][j] = (double)mat_param[i][j] / mat_param[0][j];
        }
    }
    double soma=0;
    for (int j=0; j<5; j++){
        soma = mat_param[0][j] + soma;
    }
    for (int j=0; j<5; j++){
        mat_param[0][j] = (double)mat_param[0][j] / soma;
    }
}
```

## Anexo 4 Código Fonte da Classe *StopList*

```

public class StopList {
    private Hashtable TabList = new Hashtable();
    public int lin;
    public int col;
    public int cont;
    public String Lista1[] = {
//3706 palavras não relevantes que serão excluídas da mensagem//
};
//*****
// Método que inclui termos a excluir em uma tabela hash
public void incluiTableHash(String lista[]) {
    for (int i=0; i<lista.length;i++)
    {
        palavra li = new palavra(lista[i]);
        String cod = new String( Integer.toString(lista[i].hashCode()));
        TabList.put(cod, li);
    }
}
//*****
// Método que captura termos interessantes e inclui no vetor Termos[]

public void tokening (String linha){
    StringTokenizer st = new StringTokenizer (linha, " ");
    while (st.hasMoreTokens())
    {
        String aux=st.nextToken();
        if (aux.compareTo("inicioMensagem")==0) {
            aux=st.nextToken();
            idmensagem=aux;
            System.out.println("Idenficador "+idmensagem);
        }
        else if (aux.compareTo("fimMensagem")==0) {
            buscaResposta(idmensagem);
            for (int i=0; i<tam_termos;i++) {
                System.out.println(termos[i]);}
            for (int i=0; i<8;i++) {
                System.out.println(evento[i]);}
            trataTermos(termos, tam_termos);
            trataMensagem(termos,evento,idmensagem);
        }
        else {
            String prim = new String( Integer.toString(aux.hashCode()));
            Object val= TabList.get(prim);
            if (val==null){
                termos[tam_termos]=aux;
                tam_termos++;
                System.out.println(aux);
            }
        }
    }
}

```



```

    }
    }

}

//*****
//Método principal da classe StopList que chama os demais métodos para extração dos
termos, também exclui caracteres especiais do texto

public void StopList()
{
try
{
String line_str;
String line_fim;
int line_number;
incluirTableHash(Lista1);
int contador=0;
String output=" ";
FileReader is = new FileReader("Mensagens");
BufferedReader ds = new BufferedReader(is);
while ((line_str = ds.readLine()) != null)
{
StringTokenizer ast = new StringTokenizer (line_str, " ");
if (ast.hasMoreTokens()) {
String aux_str=ast.nextToken();
contador++;
output= " "+ contador;
System.out.println(output);
System.out.println(aux_str);
System.out.println(line_str);

if ((aux_str.compareTo("xyz")!=0) &&
(aux_str.compareTo("fimMensagem")!=0)) {
line_fim=line_str.replace(',',' ');
line_fim=line_fim.replace(';',' ');
line_fim=line_fim.replace("'",' ');
line_fim=line_fim.replace('[',' ');
line_fim=line_fim.replace(']',' ');
line_fim=line_fim.replace('(',' ');
line_fim=line_fim.replace(')',' ');
line_fim=line_fim.replace(':', ' ');
line_fim=line_fim.replace('.', ' ');
line_fim=line_fim.replace('!', ' ');
line_fim=line_fim.replace('#',' ');
line_fim=line_fim.replace('$',' ');
line_fim=line_fim.replace('%',' ');
line_fim=line_fim.replace('^',' ');

```

```

line_fim=line_fim.replace('&',' ');
line_fim=line_fim.replace('*',' ');
line_fim=line_fim.replace('+',' ');
line_fim=line_fim.replace('/',' ');
line_fim=line_fim.replace('0',' ');
line_fim=line_fim.replace('1',' ');
line_fim=line_fim.replace('2',' ');
line_fim=line_fim.replace('3',' ');
line_fim=line_fim.replace('4',' ');
line_fim=line_fim.replace('5',' ');
line_fim=line_fim.replace('6',' ');
line_fim=line_fim.replace('7',' ');
line_fim=line_fim.replace('8',' ');
line_fim=line_fim.replace('9',' ');
line_fim=line_fim.replace('=',' ');
line_fim=line_fim.replace('_', ' ');
line_fim=line_fim.replace('>',' ');
line_fim=line_fim.replace('<',' ');
line_fim=line_fim.replace('?',' ');
line_fim=line_fim.replace('}',' ');
line_fim=line_fim.replace('{',' ');
line_fim=line_fim.replace('^',' ');
line_fim=line_fim.replace('-', ' ');
line_fim=line_fim.replace('°',' ');
line_fim=line_fim.replace('ª',' ');
line_fim=line_fim.replace('\\',' ');
line_fim=line_fim.toLowerCase();
tokening(line_fim);
System.out.println(line_fim);
}
else {
    tokening(line_str);
}
}
}
}
catch (IOException e)
{
    System.out.println("File error: " + e);
}

}
}

```

## Anexo 5 Código Fonte do Método *classifica*

*//Método que aplica o classificador Naive Bayes em cada registro do historico*

```
public void classifica()
{
    int cont = 1;
    double peso[] = {9.0,7.0,5.0,3.0,1.0};
    leRegistroHistorico(array,cont);
    while (array[0].compareTo(" ")!=0) {
        double numerador[]={1.0,1.0,1.0,1.0,1.0};
        double valor_classe[]={0.0,0.0,0.0,0.0,0.0};
        for (int i=0;i<5;i++){
            numerador[i]=(double)mat_param[0][i]*numerador[i];
            for (int j=1;j<7;j++){
                if (array[j+1].compareTo("1")==0){
                    numerador[i]=(double)mat_param[j][i]*numerador[i];
                }
                else
                    numerador[i]=(double)mat_param[j+6][i]*numerador[i];
            }
        }
        for (int i=0;i<5;i++){
            valor_classe[i]=(double)numerador[i];
        }
        int pos_maior=0;
        double maior=valor_classe[0];
        for (int i=1;i<5;i++){
            if (valor_classe[i]>maior) {
                maior=valor_classe[i];
                pos_maior=i;
            }
        }
        array[8]=String.valueOf(peso[pos_maior]);
        adicionaRegistroHistorico(array,cont);
        tot_termo_hist=cont - 1;
        cont++;
        leRegistroHistorico(array,cont);
    }
}
```

## Anexo 6 Arquivo Historico\_classificado

*Impressão parcial do arquivo Historico\_classificado contendo termos classificados do usuário B;*

bi01	comissao	1000001.0
bi02	comissao	1100015.0
bi03	comissao	1000001.0
bi03	terminal	1000001.0
bi04	comissao	0101015.0
bi04	onibus	0101015.0
bi05	biblioteca	1000017.0
bi05	acesso	1000017.0
bi05	estudo	1000017.0
bi06	biblioteca	1000017.0
bi07	linux	0001017.0
bi07	programming	0001017.0
bi07	digital	0001017.0
bi08	ufrgs	1000001.0
bi09	biblioteca	1000017.0
bi09	emprestimo	1000017.0
bi10	biblioteca	1000017.0
bi10	emprestimo	1000017.0
bi10	livro	1000017.0
bi11	biblioteca	0001017.0
bi11	bolsista	0001017.0
bi11	feira	0001017.0
bi11	horario	0001017.0
bi11	servicos	0001017.0
bi12	matlab	1000017.0
bi12	livros	1000017.0
bi12	computer	1000017.0
bi12	signal	1000017.0
bi12	biblioteca	1000017.0
bi12	doacao	1000017.0
bi12	motorola	1000017.0
bi12	pct	1000017.0
bi12	language	1000017.0
bi12	ufrgs	1000017.0
bi13	biblioteca	0001017.0
bi13	horario	0001017.0
bi14	capes	1000001.0
bi15	capes	1000001.0
bi16	biblioteca	1000001.0
bi16	ferias	1000001.0
bi17	biblioteca	0001017.0
bi17	bolsista	0001017.0

## Bibliografia

- [AAM 95] AAMODT, A. **Knowledge Acquisition and Learning by Experience – The Role of Case-Specific Knowledge**. Dragvoll, Norway: University of Trondheim, Department of Informatics, 1995.
- [AGN 94] AGNAR, A; ENRIC, P. Case-Based Reasoning: Foundational Issues, Methodological Variations, and System Approaches. **AI Communications**, Catalonia, v.7, p.39-59, 1994.
- [AHA 92] AHA, D.W. Tolerating Noisy, Irrelevant and Novel Attributes in Instance-Based Learning Algorithms. **International Journal of Man-Machine Studies**, [S.l.], v.36, p.267-287, 1992.
- [AKO 97] AKOULCHINA, I.; GANASCIA, J.G. SATELIT-Agent : An Adaptive Interface Based on Learning Interface Agents Technology. In: INTERNATIONAL CONFERENCE USER MODELING,UM, 6., 1997. **User Modeling**: proceedings. Sardinia: Springer-Verlag, 1997. p.21-32.
- [AND 2000] ANDROUTSOPOULOS, I. et al. **Learning to Filter Spam E-Mail**: Athens, Greece: Institute of Informatics and Telecommunications National Centre for Scientific Research “Demokritos” , 2000.
- [ASN 97] ASNICAR, F.A.; TASSO, C. ifWeb: a Prototype of User Model-Based Intelligent Agent for Document Filtering and Navigation in the World Wide Web. In: INTERNATIONAL CONFERENCE ON USER MODELING, UM, 6., 1997. **User Modeling**: proceeding. Sardinia: Springer-Verlag, 1997. Disponível em: <<http://fit.gmd.de/UM97/Tasso/Tasso.html>>. Acesso em: 20 fev. 2001.
- [BAL 2001] BALINSKI, R. **Filtragem de Informação no Ambiente Direto**. 2001. Dissertação (Mestrado em Ciência da Computação) – Instituto de Informática, UFRGS, Porto Alegre.
- [BEN 97] BENAKI, E.; KARKALETSIS, V. A.; SPYROPOULOS, C. D. User Modeling in WWW: the UMIE Prototype. In: INTERNATIONAL CONFERENCE ON USER MODELING, UM, 6., 1997. **User Modeling**: proceeding Sardinia: Springer-Verlag, 1997. Disponível em: <<http://fit.gmd.de/UM97/Benaki/Benaki.html>> Acesso em: 20 fev. 2001.
- [BIL 96] BILLSUS, D.; PAZZANI, M. **Revising User Profiles**: The Search for Interesting Web Sites. Irvine: Department of Information and Computer Science. University of California, 1996. Disponível em: <<http://www.ics.uci.edu/~mlearn/MLPapers.html>>. Acesso em: 20 Fev. 2001

- [BIL 98] BILLSUS, D.; PAZZANI, M. **A Personal News Agent that Talks, Learns and Explains**. Irvine: Department of Information and Computer Science. University of California, 1998. Disponível em: <<http://www.ics.uci.edu/~mlearn/MLPapers.html>>. Acesso em: 02 fev. 2001.
- [BLU 97] BLUM, A; LANGLEY, P. **Selection of Relevant Features and Examples in Machine Learning**. Pennsylvania: School of Computer Science, Carnegie Mellon University Pittsburgh, 1997.
- [BOO 98] BOONE, G. **Concept Features in Re: Agent**, an Intelligent Email Agent. Atlanta, Georgia: Georgia Institute of Technology, 1998.
- [BOR 99] BORGELT, C.; GEBHARDT, J. **A Naive Bayes Style Possibilitic Classifier**. Magdeburg, Germany:Universitätsplatz , 1999. Disponível em: <[fuzzy.cs.uni-magdeburg.de/~borgelt/papers/eufit\\_99.ps.gz](http://fuzzy.cs.uni-magdeburg.de/~borgelt/papers/eufit_99.ps.gz)>. Acesso em: 22 jan. 2001.
- [CAR 99a] CARNEIRO, A. L.; SILVA, W.T. **Introdução a Redes Bayesianas**. Brasília: 1999. (Relatório de Pesquisa CIC/UnB – 09/99). Disponível em: <<ftp://ftp.cic.unb.br/pub/publications/report/rr.99-09.ps.Z>>. Acesso em: 13 set. 2001.
- [CAR 99b] CARDOSO, J.C. **Considerações ao Projeto de Interfaces de Bibliotecas Digitais**. 1999. Trabalho Individual (Mestrado em Ciência da Computação) – Pontifícia Universidade Católica do Rio Grande do Sul, PUCRG, Porto Alegre.
- [CHA 99] CHANG, Y.H. **Email Filtering: Machine learning Techniques and na Implementation for the UNIX Pine Mail System**. Cambridge, MA: M.I.T. A.I. Lab & L.C.S., 1999.
- [CHE 97] CHEN, L.; SYCARA, K. **WebMate: A Personal Agent for Browsing and Searching**. Pittsburgh: Carnegie Mellon University, 1997.
- [CHE 99a] CHENG, J.; GREINER, R. **Learning Bayesian Belief Network Classifiers: Algorithms and System**. Alberta, Canada: Department of Computing Science, University Alberta, 1999.
- [CHE 99b] CHENG, J.; GREINER, R. Comparing Bayesian Networks classifiers. In: UAI, 1999. **Proceedings ...** Alberta, Canadá: Morgan Kaufman, 1999.
- [DEI 2001] DEITEL, H.; DEITEL, P. **JAVA como programar**. 3 ed. Porto Alegre: Bookman, 2001. 1201p.
- [DIR 2001] DIRETO. **Software de Correio, Catálogo e Agenda Corporativo**. Disponível em: <<http://www.direto.org.br>>. Acesso em: 16 set. 2001.

- [DOM 96] DOMINGOS, P.; PAZZANI, M. Beyond Independence: Conditions for the Optimality of the Simple Bayesian Classifier. In: EUROPEAN CONFERENCE MACHINE LEARNING, ML, 6., 1996. **Machine Learning: proceedings**. San Francisco, CA: L. Saitta, 1996. p105-112.
- [ENG 2000] ENGEL, P. M. **Redes Neurais**. Apostila da disciplina Inteligência Artificial II. Mestrado em Ciência da Computação – Convênio da FACCAR/UFRGS, 2000.
- [FAO 2000] FAORO, Marcelo. **Um Estudo Sobre Agentes Inteligentes na Web**. 2000. Trabalho Individual (Mestrado em Ciência da Computação) – Instituto de Informática, UFRGS, Porto Alegre.
- [FAP 2000] FAPERGS. **Projeto Sistemas Avançados para Comunicação Eletrônica – Software Aberto de Correio, Agenda e Catálogo**. Disponível em: <<http://www.inf.ufrgs.br/proctpar/direto/ProjetoFapergsEdital062000.html>> Acesso em: 20 jul. 2000.
- [FEL 98] FÉLIX, L. et al. **MLC++ Biblioteca de Aprendizado de Máquina em C++**. São Carlos: Instituto de Ciências Matemáticas de São Carlos, 1998. Relatório Técnico.
- [FIN 89] FININ, T . A general User Modeling Shell. In: KOBASA, A.; WAHLSTER, E.W. (Ed.). **User Models in Dialog Systems**. New York: Springer Verlag, 1989. 471 p. p.411-430.
- [FRE 2000] FREUND, J. E.; GARY, A. S. **Estatística Aplicada: Economia, Administração e Contabilidade**. 9 ed. Porto Alegre: Bookman, 2000.
- [FRI 97] FRIEDMAN, N.; GEIGER, D.; GOLDSZMIDT, M. **Bayesian Network Classifiers**. Boston: Kluwer Academic, 1997.
- [GNU 2000] GNU. **General Public License**. Disponível em: <<http://www.gnu.org/copyleft/gpl.html>> Acesso em: 12. Set. 2001.
- [GOL 96] GOLDMAN, C.; LANGER, A.; ROSENSCHEIN, S. **Musag: an Agent that Learns What You Mean**. Jerusalém, Israel: Institute of Computer Science, University of Hebrew, 1996.
- [HEC 95] HECKERMAN, D. **A Tutorial on Learning Bayesian Networks**. Redmond, WA: Microsoft Corporation, 1995.
- [HEC 98] HECKERMAN, D.; HORVITZ, E. Inferring Informational Goals from Free-Text Queries: A Bayesian Approach. In: UAI, 1998. **Proceedings...** Madison: Morgan Kaufmann Publishers, 1997. p.230-237.
- [HEN 94] HENERY, R. J. Classification. In: MICHIE, D.; SPIEGELHALTER D.J.; TAYLOR, C.C. (Ed.). **Classification, Machine Learning, Neural and Statistical Classification**. Glasgow: Ellis Horwood, 1994.

- [HRU 97] HRUSCHKA, J.; SILVA, E. R.; TEIXEIRA, W. **Propagação de Crença em Redes Bayesianas**. Brasília: UnB, 1997. (Relatório de Pesquisa CIC/UNB – 02/97).
- [ITS 2001] ITSKEVITCH, J. **Automatic Hierarchical E-mail Classification Using Association Rules**. Tese submetida para requerimento do grau de mestre pela escola de Ciência da Computação de Simon Fraser University, 2001.
- [JOA 98] JOACHIMS, T.; FREITAG, D.; MITCHELL, T. **WebWatcher: A Tour Guide for the World Wide Web**. Pittsburg: School of Computer Science, Carnegie Mellon University, 1998.
- [KIM 2000] KIM, J. **Information Filtering Resources**. Disponível em: <<http://www.enee.umd.edu/medlab/filter/filter.html>> Acesso em: 20 set. 2001.
- [KOH 97] KOHAVI, R.; JOHN, G. Wrappers for Feature Subset Selection. **Artificial Intelligence**, Amsterdam, v. 97, p. 273-324, 1997.
- [KON 90] KONONENKO, P. Comparison of Inductive and Naive Bayesian Learning Approaches to Automatic knowledge Acquisition. In: WIELINGA, B. et al. **Current trends in Knowledge acquisition** Amsterdam: IOS Press, 1990.
- [KON 91] KONONENKO, I. Semi-Naive Bayesian classifier. In: EUROPEAN WORKING SESSION ON LEARNING, 6., 1991. **Machine Learning: Proceedings**. Sardinia: Springer-Verlag, 1991. p.206-219.
- [KON 97] KONTKANEN, P. et al. Comparing Predictive Inference Methods for Discrete Domains. In: WORKSHOP ON ARTIFICIAL INTELLIGENCE AND STATISTICS, 6., 1997. **Artificial Intelligence and Statistics**. Reading: Addison-Wesley, 1997. p.311-318.
- [KRU 97] KRULWICH, B.; BURKEY, C. The InfoFinder Agent: Learning User Interest Through Heuristic Phrase Extration. **IEEE Expert**, Los Alamitos, v.12, p.22-27, 1997.
- [LAD 2000] LADEIRA, M. **Diagrama de Influências Múltiplo Secionado**. 2000. Tese (Doutorado em Ciência da Computação) - Instituto de Informática, UFRGS, Porto Alegre.
- [LAN 92] LANGLEY, P.; IBA, W.; THOMPSON, K. **An Analysis of Bayesian Classifiers**. San José, CA: AI Research Branch, NASA Ames Research Center, 1992.
- [LAN 93] LANGLEY, P.; IBA, W. Average-case Analysis of a Nearest Neighbor algorithm. In: INTERNATIONAL JOINT CONFERENCE ON ARTIFICIAL INTELLIGENCE, 1993. **Proceedings...** Chambéry, France: [s.n.], 1993. p.889-894.



- [LAN 94] LANGLEY, P.; IBA, W.; SAGE, S. Induction of Selective Bayesian Classifiers. In: UAI, 1994. **Proceedings...** San Francisco, CA: [s.n], Morgan Kaufmann, 1994. p.309-406.
- [LIE 95] LIEBERMAN, H. **Autonomous Interface Agents**. Cambridge: Media Laboratory, Massachusetts Institute of Technology, 1995.
- [LOP 99] LOPES, P. **Probabilidades e Estatísticas**. Rio de Janeiro: Reichmann & Affonso Ed., 1999.
- [MER 2000] MERETAKIS, D.; LU, H.; WUTHRICH, B. **A Study on the Performance of Large Bayes Classifiers**. Hong Kong: Computer Science Department, Hong Kong University of Science and Tecnology, 2000.
- [MEY83] MEYER, P. L. **Probabilidade: Aplicações à Estatística**. 2 ed. Rio de Janeiro: Livros Técnicos e Científicos, 1983.
- [MIN 96] MINIO, M.; TASSO, C. User Modeling for Information Filtering on Internet Services: Exploiting na Extended Version of the UMT Shell. In: INTERNATIONAL CONFERENCE ON USER MODELING, UM, 6., 1996. **User Modeling: proceeding**. Hawaii:[s.n.], 1996. Disponível em: <<http://www.cs.ju.oz.au/bob/um96-workshop.html>>. Acesso em: 22 jan. 2001.
- [NES 98] NESBITT, S.; VEL, O. **A Collaborative Filtering Agent System for Dynamic Virtual Communities on the Web**. Townsville, Australia: Department of Computer Science, 1998.
- [NIG 98] NIGAM, K. **Text Classification from Labeled and Unlabeled Documents using EM**. Boston: Kluwer Academic, 1998.
- [PAT 94] PATTIE, M. Agents that Reduce Work and Information Overload. **AI Communications**, Catalonia, v.7, p.31-40, 1994.
- [PAY 95a] PAYNE, R. T.; EDWARDS, P. **Interface Agents that Learn: An Investigation of Learning Issues in a Mail Agent Interface**. Scotland: Department of Computing Science – Universidade of Aberdeen, King’s College, Aberdeen, 1995. Disponível em: <<http://www.csd.abdn.ac.uk/~pedwards/research.html>>. Acesso em: 21 set. 2001.
- [PAY 95b] PAYNE, R. T. **Instance-Based Prototypical Learning of Set Valued Attributes**. Aberdeen, Scotland, UK: Department of Computing Science – Universidade of Aberdeen, King’s College, 1995.
- [PAZ 2000] PAZZANI, M. **Representation of Eletronic Mail Filtering Profiles: A User Study**. Irvine: Department of Information and Computer Science, University of California, 2000.

- [PAZ 95] PAZZANI, M. **Algorithms for Learning Augmented Bayesian Classifiers**. Irvine: Department of Information and Computer Science, University of California, 1995.
- [PAZ 96] PAZZANI, M.; MURAMATSU, J.; BILLSUS, D. **Syskill & Webert: Identifying Web Sites**. Irvine: Department of Information and Computer Science. University of California, 1996. Disponível em: <<http://www.ics.uci.edu/~mlearn/MLPapers.html>>. Acesso em: 20 fev. 2001.
- [PAZ 97] PAZZANI, M.; NGUYEN, L.; MANTIK, S. **Learning from Hotlists and Coldlists: Towards a WWW Information Filtering and Seeking Agent**. Irvine: Department of Information and Computer Science, University of California, 1997.
- [PER 88] PEARL, J. **Probabilistic Reasoning in Intelligent Systems**. San Mateo, California: Morgan Kaufman, 1988.
- [POZ 2000] POZO, A. T. R. **Uma Visão de Máquinas de Aprendizado**. Disponível em: <<http://www.inf.ufrgs.br/~aurora/mlc/captotal>>. Acesso em: 22 jan. 2001.
- [PRO 2000] PROCERGS. **Software Direto**. Disponível em: <<http://www.procergs.com.br>>. Acesso em: 12 fev. 2001.
- [PRO 2001] PROCERGS. **Tutorial de Instalação do Direto**. Disponível em: <[www.direto.org.br](http://www.direto.org.br)>. Acesso em: 15 abr. 2001.
- [QUI 86] QUINLAN, J.R. Induction of Decision Trees, Machine Learning. **Machine Learning**, Dordrecht, v.1, p.81-106, 1986.
- [QUI 93] QUINLAN, J.R. **C4.5: Programs for Machine Learning**. San Mateo, California: Morgan Kaufmann, 1993.
- [REN 2000] RENNIE, J.D.M. **Ifile: Application of Machine Learning to E-Mail Filtering**. Cambridge, MA: [s.n.], 2000.
- [SAH 98] SAHAMI, M.et al. A Bayesian Approach to Filtering Junk E-mail, in Learning For Text Categorization. In: AAAI, 1998, Madison. **Proceedings...** Menlo Park: AAAI, 1998.
- [SEB 99] SEBASTIANI, F. **A Tutorial on Automated Text Categorisation**. Pisa, Itália: Instituto di Elaborazione dell'Informazione Consiglio Nazionale delle Ricerche, 1999.
- [SEG 99] SEGAL, B. R.; KEPHART, J.O. **Mailcat: An Inteligent Assistant for Organization E-Mail**. Yortown Heights, NY: IBM Thomas J. Watson Research Center, 1999.

- [TIR 99] TIRRI, H.; SILANDER, T. **Bayesian Classification. Projeto Bayda.** Finlandia: Complex Systems Computation Group (CosCo), Department of Computer Science, University of Helsinki, 1999. Disponível em: <<http://www.cs.helsinki.fi/research/cosco/Projects/NONE/html/doc/bayda4.htm>>. Acesso em: 12 abr. 2001.
- [VEL 98] VEL, O.; NESBITT, S. **A Collaborative Filtering Agent System for Dynamic Virtual Communities on the Web.** Townsville, Austrália: Institute of Computer Science, 1998.