UNIVERSIDADE DO VALE DO RIO DOS SINOS
CIÊNCIAS EXATAS E TECNOLÓGICAS
PROGRAMA INTERDISCIPLINAR DE PÓS-GRADUAÇÃO EM
COMPUTAÇÃO APLICADA

Gustavo Bisognin

Utilização de Máquinas de Suporte Vetorial Para Predição de Estruturas Terciárias de Proteínas.

São Leopoldo
2007

Gustavo Bisognin

Utilização de Máquinas de Suporte Vetorial Para Predição de Estruturas Terciárias de Proteínas.

Dissertação apresentada à Universidade do Vale do Rio dos Sinos como requisito parcial para obtenção do título de Mestre em Computação Aplicada.

Orientador: Prof. Dr. Adelmo Luís Cechin

São Leopoldo
2007

CIP — CATALOGAÇÃO NA PUBLICAÇÃO

Bisognin, Gustavo

Utilização de Máquinas de Suporte Vetorial Para Predição de Estruturas Terciárias de Proteínas. / por Gustavo Bisognin. — São Leopoldo: Ciências Exatas e Tecnológicas da UNISINOS, 2007.

101 f.: il.

Dissertação (mestrado) — Universidade do Vale do Rio dos Sinos. Ciências Exatas e Tecnológicas. Programa Interdisciplinar de Pós-Graduação em Computação Aplicada, São Leopoldo, BR–RS, 2007. Orientador: Cechin, Adelmo Luis.

1. Máquina de Suporte Vetorial. 2. Aprendizado de Máquina. 3. Estruturas Terciárias de Proteínas. I. Cechin, Adelmo Luis. II. Título.

"Dedico esta dissertação as minhas tias Vanilde Bisognin e Eleni Bisognin."

Agradecimentos

Agradeço em primeiro lugar a Deus, por sempre iluminar meus caminhos e guiar os meus passos durante o período de estudo que direcionaram a obtenção deste título.

Agradeço a toda minha família, em especial aos meus pais Lidio Bisognin e Ignês S. S. Bisognin, e a minha irmã Luciane Bisognin Ceretta por todo o incentivo e apoio prestados neste difícil período.

Agradeço ao meu orientador Adelmo Luis Cechin por me aceitar como seu pupilo e acreditar no meu trabalho, compartilhando toda a sua experiência e conhecimento.

Agradeço aos amigos que costumo chamar de "irmãos", Cícero Raupp Rolim, Cristiano Galina, Fernando Cercatto e Fabiano Bisognin Franco por todo apoio, ajuda, incentivo e companheirismo.

Agradeço aos amigos de longa data João, Nicolau, Eduardo, Alison, Everton e Jaganata pelo apoio e companheirismo nos momentos em que precisei.

Agradeço aos companheiros de mestrado Igor Lorenzatto, Rogério Martins, André Tochetto e Darci Levis pelo apoio e incentivo mútuo.

Agradeço as amigas Denise, Marialva e Daniela pelos conselhos e ajuda prestada durante toda a pesquisa.

Agradeço ao professor José Carlos Merino Mombach por ter me orientado e concedido a bolsa de estudos.

Agradeço aos funcionários e professores do PIPCA, por todo o conhecimento compartilhado durante o período de estudos.

Finalmente, agradeço a Hewlett-Packard computadores pelo apoio financeiro concedido através da bolsa de estudos.

Resumo

A estrutura tridimensional de uma proteína está diretamente ligada a sua função. Diversos projetos de seqüenciamento genéticos acumulam um grande número de seqüências de proteínas cujas estruturas primárias e secundárias são conhecidas. Entretanto, as informações sobre suas estruturas tridimensionais estão disponíveis somente para uma pequena fração destas proteínas. Este fato evidencia a necessidade da criação de métodos automáticos para a predição de estruturas terciárias de proteínas a partir de suas estruturas primárias. Conseqüentemente, ferramentas computacionais são utilizadas para o tratamento, seleção e análise destes dados. Atualmente, um novo método de aprendizado de máquina denominado Máquina de Suporte Vetorial (MSV) tem superado métodos tradicionais como as Redes Neurais Artificiais (RNA) no tratamento de problemas de classificação. Nesta dissertação utilizamos as MSV para a classificação automática de proteínas. A principal contribuição deste trabalho foi a metodologia proposta para o tratamento do problema. Esta metodologia consiste em compor os vetores de suporte com os valores do alinhamento das estruturas secundárias preditas do conjunto de proteínas selecionadas. Neste trabalho, analisamos um conjunto de dados referentes aos 27 *folds* mais populares descritos na hierarquia SCOP. Os dados foram treinados com diferentes atributos, atingindo uma taxa de 57% de exemplos classificados corretamente para o melhor modelo.

Palavras-chave: Máquina de Suporte Vetorial, Aprendizado de Máquina, Estruturas Terciárias de Proteínas.

TITLE: "SUPPORT VECTOR MACHINE FOR TERTIARY STRUCTURE PREDICTION"

Abstract

The three-dimensional structure of a protein is directly related to its function. Many projects of genetic sequence analysis accumulate a great number of protein sequences whose primary and secondary structures are known. However, the information on its three-dimensional structures are available only for a small fraction of these proteins. This fact evidences the necessity of creation of automatic methods for the prediction of tertiary protein structures from its primary structures. Consequently, computational tools are used for the treatment, election and analysis of these data. Currently, a new method of machine learning called Support Vector Machine (SVM) has surpassed traditional methods as Artificial Neural Networks (ANN) in the treatment of classification problems. In this master thesis we use the SVM for the automatic protein classification. The main contribution of this work was the methodology proposal for the treatment of the problem. This methodology consists in composing the support vectors with the values of the predicted secondary structures alignment of the selected proteins set. In this work, we analyze a set of data related to the 27 more popular *folds* described in the SCOP hierarchy. The data had been trained with different attributes, reaching a tax of 57% of examples classified correctly for the optimum model.

Keywords: Support Vector Machine, Machine Learning, Tertiary Protein Structures.

Sumário

Lista de Abreviaturas

AM Aprendizado de Máquina

RNA Redes Neurais Artificiais

MSV Máquina de Suporte Vetorial

HSO Hiperplano de Separação Ótimo

SCOP *Structural Classification of Proteins*

PDB *Protein Data Bank*

KKT Karush Kuhn Tucker

HSSP *Homology-Derived Structures of Proteins*

AAC *Amino Acid Composition*

CATH *Protein Structure Classification*

Lista de Figuras

Lista de Tabelas

Lista de Símbolos

$\|x\|$	Norma euclidiana do vetor x
$\|x\|$	Valor absoluto de x
$\in$	Símbolo para "pertence"
$\cup$	Símbolo para "união"
$\sum$	Símbolo para "somatório"
∂	Símbolo para "derivada parcial"
$\subseteq$	Símbolo para "esta contido ou igual"
$P(A\|B)$	Probabilidade de A dado B
$\langle\ \mathbf{w} \cdot \mathbf{x}\ \rangle$	Produto escalar de $\mathbf{w}$ com $\mathbf{x}$

Capítulo 1

Introdução

Uma vasta quantidade de dados biológicos vem sendo disponibilizados a uma taxa de atualização elevada, fazendo com que os bancos de dados atuais cresçam exponencialmente. Esse fato tem sido causado pela utilização de novas e eficientes técnicas aplicadas na análise das seqüências de genoma e proteoma. A manipulação e análise dos dados dispostos nessas bases de dados tornou-se um dos maiores desafios da Biologia Computacional [BRUNAK and BALDI 2001].

A Biologia Computacional diz respeito a utilização de técnicas e ferramentas de computação para a resolução de problemas biológicos [BRUNAK and BALDI 2001]. Dentre as diversas áreas da Biologia, a biologia molecular destaca-se como a área mais promissora para a aplicação de técnicas computacionais [SETÚBAL and MEIDANIS 1997]. Nesse contexto, a computação, pode ser aplicada na resolução de problemas como comparação de seqüências (DNA, RNA e proteínas), montagem de fragmentos, reconhecimento de genes, identificação e análise da expressão de genes e determinação da estrutura de proteínas [SETÚBAL and MEIDANIS 1997, BRUNAK and BALDI 2001].

A aplicação de técnicas computacionais para a resolução de problemas biológicos teve início na década de 1980, quando um grupo de pesquisadores envolvendo biólogos, cientistas da computação, matemáticos e físicos resolveram aplicar seus esforços no desenvolvimento de métodos para a modelagem de sistemas biológicos. Com isso Técnicas de Aprendizado de Máquina (AM) [MITCHELL 1997] estão sendo cada vez mais empregadas para o tratamento de problemas biológicos, por possuírem capacidade de aprender automaticamente a partir de grandes quantidades de dados e produzir hipóteses úteis [BRUNAK and BALDI 2001].

É importante ressaltar a grande quantidade de problemas existentes na Biologia Molecular que podem ser abordados com técnicas de AM, como por exemplo, a predição de estrutura de proteínas [BRUNAK and BALDI 2001].

De fato, a determinação da seqüência de uma proteína é uma tarefa mais fácil de ser realizada do que a determinação de sua estrutura, o que explica a grande diferença entre o número de seqüências e o número de estruturas conhecidas. A fim de diminuir esta diferença, a Biologia Computacional trata o problema baseando-se no fato de que a função de uma proteína está diretamente relacionada com a seqüência de aminoácidos que a compõem [GUIMARÃES and MELO 2003, LEHNINGER 2000].

Uma das formas de tratamento mais utilizadas para a determinação da função de uma proteína a partir de sua estrutura primária é a análise seqüencial, onde a seqüência desejada é comparada com outra de função conhecida procurando alinhar resíduos de aminoácidos idênticos ou similares, de modo a evidenciar semelhanças locais ou globais.

A compreensão da relação existente entre a seqüência de aminoácidos de uma proteína e sua estrutura, se constituem um dos maiores objetivos dos pesquisadores desta área, uma vez estabelecida esta relação, a estrutura secundária da proteína pode ser determinada com um alto nível de confiança. Entretanto, a compreensão da relação entre seqüência e estrutura não é um problema trivial de ser resolvido. Durante os últimos anos, vários pesquisadores obtiveram um sucesso considerável com a predição de estruturas de proteínas baseados na sua seqüência. Este tipo de informação possui uma grande importância na modelagem de proteínas.

Na busca de melhores resultados para a predição das estruturas de proteínas, diversos estudos foram feitos combinando a seqüência de aminoácidos das proteínas com atributos físicos e químicos de cada aminoácido da cadeia. Por exemplo, atributos como hidrofobicidade, massa, volume entre outros têm sido testados e combinados entre si [DING and DUBCHAK 2001, WANG 2002].

A análise das semelhanças entre as conformações das proteínas auxilia na compreensão do relacionamento entre seqüência, estrutura e função. Estas semelhanças podem ser evidenciadas através da busca de padrões podendo caracterizar famílias específicas de proteínas, ou seja, proteínas podem estar relacionadas estrutural ou funcionalmente. Essa abordagem busca entre outras coisas, identificar padrões de dobramento visando a rapidez na obtenção de informações em bancos de dados de estruturas e a determinação da relação entre as topologias das proteínas dispostas nestes bancos.

Dada a vasta quantidade de dados existentes atualmente, torna-se necessário a aplicação de uma abordagem analítica robusta para catalogar e representar a seqüência de genes respeitando seu significado biológico. Conseqüentemente, existe a necessidade da utilização de ferramentas computacionais para realização de uma análise eficiente dos dados coletados.

Existem vários métodos de aprendizado de máquina que são aplicados na predição de estruturas biológicas. Dentre eles, destacam-se as Máquinas de Suporte Vetorial (MSV).

As MSV apresentam-se como uma técnica nova e promissora para a classificação e regressão. Este método foi desenvolvido nos últimos anos com a colaboração de universidades, laboratórios e empresas, sendo um importante tópico no aprendizado de máquina tanto do ponto de vista teórico quanto prático.

Pode-se definir uma Máquina de Suporte Vetorial como sendo uma técnica de aprendizado de máquina baseada na estatística e fundamentada nos princípios de Minimização do Risco Estrutural e na teoria da dimensão Vapnik-Chervonenkis (VC). Entre as principais características que popularizaram seu uso em problemas biológicos estão sua boa capacidade de generalização e robustez diante de dados de alta dimensão, como os encontrados em grande parte das aplicações envolvendo o reconhecimento de genes, análise de dados de expressão gênica e classificação de proteínas [HAYKIN 2001].

Esta técnica foi inicialmente proposta por Vapnik e colaboradores, no seu livro "*The Nature of Statistical Learning Theory*", e atualmente tem sido aplicada com sucesso em diversos problemas de classificação e regressão [MERCIER and LENNON 2003].

Basicamente uma Máquina de Suporte Vetorial é uma máquina linear cuja idéia principal consiste em construir um hiperplano como superfície de decisão onde a margem de separação entre os exemplos rotulados como positivos e negativos tem de ser máxima. Uma noção central à construção do algoritmo de aprendizado por vetor de suporte é o núcleo do produto interno entre um *vetor de suporte* x_i e o vetor x retirado do espaço de entrada. Os vetores de suporte consistem num pequeno subconjunto de treinamento extraído pelo algoritmo. Dependendo de como este núcleo é gerado, podemos construir diferentes máquinas de aprendizado, caracterizadas por superfície de decisão própria [HAYKIN 2001].

O objetivo da classificação Vetor Suporte é elaborar uma forma computacionalmente eficiente capaz de tratar grandes conjuntos de amostras e aprender hiperplanos que otimizem os limites de generalização de separação em um espaço de características de alta dimensão. A teoria da generalização fornece uma orientação clara sobre como controlar a capacidade e logo como prevenir modelos ruins controlando as medidas das margens dos hiperplanos, enquanto a teoria da otimização fornece as técnicas matemáticas necessárias para encontrar hiperplanos otimizando essas medidas [HAYKIN 2001].

Um conceito chave das MSV é a implementação do mapeamento não-linear dos dados de entrada para um espaço característico de alta dimensão, onde um

hiperplano ótimo é construído para separar os dados linearmente em duas classes.

Quando os dados de treinamento são separáveis, o hiperplano ótimo no espaço característico apresenta a máxima margem de separação [SEMOLINI 2003], já no caso destes dados serem linearmente inseparáveis, é necessário a aplicação de uma função de *kernel* com o objetivo de aumentar a dimensão destes dados tornando-os separáveis.

1.1 Objetivo Geral

Este trabalho tem como objetivo geral a utilização Máquinas de Suporte Vetorial para a classificação automática de estruturas terciárias de proteínas baseando-se nas informações estruturais dispostas pela base de dados SCOP.

1.2 Objetivos Específicos

Como objetivo específico, destaca-se a utilização de MSV para classificação automática de proteínas compostas por múltiplas classes. Visto que as MSV apresentam-se como classificadores essencialmente binários, torna-se necessário a aplicação de um método multi-classe para a resolução deste problema. Como por exemplo o método Um-Contra-Um, onde uma determinada base de dados é treinada atribuindo a uma classe específica um valor positivo $(+1)$ e à classe seguinte o valor negativo (-1), sendo cada uma das classes treinada com a outra, formando assim um conjunto de $n(n-1)/2$ classificadores binários.

Para a composição dos dados de treinamento, são utilizadas informações referentes à estrutura primária das proteínas em questão obtidas da base de dados PDB. Além disso, outro objetivo deste trabalho é a predição e análise das estruturas secundárias destas proteínas além da composição dos vetores de suporte com os dados de alinhamento das seqüências realizadas com o *software* de alinhamento de seqüências *ClustalW*.

Capítulo 2

Conceitos Básicos em Biologia Molecular

A Biologia molecular pode ser definida como sendo o estudo da química e física da vida. Os seres vivos são constituídos de moléculas desprovidas de vida. Estas moléculas quando examinadas isoladamente, comportam-se de acordo com as leis químicas e físicas que descrevem o comportamento da matéria inanimada.

Os organismos vivos possuem atributos extraordinários que não são exibidos por uma coleção de moléculas escolhidas ao acaso. Por isso, torna-se de fundamental importância o conhecimento dos aspectos básicos da química dos organismos para melhor compreender o fenômeno da vida [LEHNINGER 2000].

Neste capítulo serão descritas as principais terminologias e conceitos básicos comuns ao problema que abordamos neste estudo.

2.1 Aminoácidos

Os aminoácidos são ácidos orgânicos formados por átomos de carbono, hidrogênio, oxigênio e nitrogênio. Alguns tipos de aminoácidos também podem conter átomos de enxofre.

Existem vinte aminoácidos diferentes na natureza, que fazem parte das proteínas e peptídeos. A nomenclatura utilizada para estes aminoácidos pode ser observada na Tabela 2.1.

Tabela 2.1 – *Nomenclatura dos aminoácidos.*

Letra	Abreviação	Identificação	Letra	Abreviação	Identificação
A	ALA	Alanina	C	CIS	Cisteina
D	ASP	Asparato	E	GLU	Glutamato
F	FEN	Fenilalanina	G	GLI	Glicina
H	HIS	Histidina	I	ILE	Isoleucina
K	LIS	Lisina	L	LEU	Leucina
M	MET	Metionina	N	ASN	Asparagina
P	PRO	Prolina	Q	GLN	Glutamina
R	ARG	Argenina	S	SER	Serinaina
T	TRE	Treonina	V	VAL	Valina
W	TRP	Triptofano	Y	TIR	Tirosina

Estes aminoácidos são freqüentemente referidos como sendo aminoácidos padrão, primários ou normais, para distingui-los dos aminoácidos que são modificados no interior das proteínas [LEHNINGER 2000].

Os aminoácidos padrão, por convenção internacional, são designados por abreviações de três letras, derivadas de seus nomes vindos da língua inglesa ou por uma única letra, como pode ser observado na Tabela (2.1) [LEHNINGER 2000].

2.1.1 Estrutura dos Aminoácidos

Todos os vinte aminoácidos encontrados nas proteínas possuem quatro grupos de átomos, sendo um hidrogênio, um grupo amina e um grupo carboxila idênticos para todos os aminoácidos. Estes grupos encontram-se ligados a um átomo de carbono, o qual ocupa uma posição central na molécula, (Figura 2.1) [WANNMACHER and DIAS 1976].

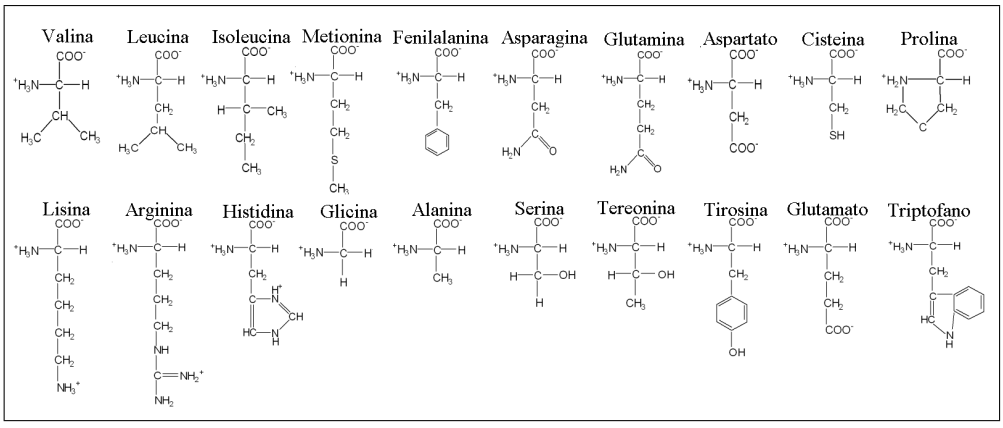


Figura 2.1 – *Estrutura dos 20 aminoácidos padrões .*

Os aminoácidos diferem uns dos outros através de suas cadeias laterais ou grupos R, que faz a quarta ligação com o carbono central. No aminoácido glicina por exemplo, R é o átomo de hidrogênio (H), já na Alanina, o R é um grupamento com quatro átomos (CH3).

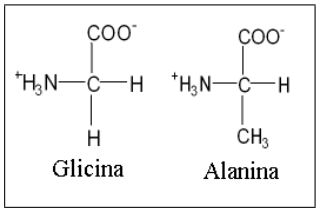


Figura 2.2 – *Fórmula de dois aminoácidos, apresentando as diferenças no grupo R.*

2.1.2 Propriedades Físico-químicas dos Aminoácidos

Segundo Lehninger [LEHNINGER 2000] os aminoácidos possuem propriedades físicas e químicas que combinadas interferem diretamente nas interações que formam e estabilizam a estrutura tridimensional das proteínas. Dentre estas propriedades, fazem parte:

- **A hidrofobicidade;**
- **O raio de van der Waals;**
- **A massa;**
- **A refratividade;**
- **A área de superfície.**

Uma lista mais completa e com os respectivos valores das propriedades para cada aminoácido pode ser encontrada em: http://expasy.org/cgi-bin/protscale.pl.

2.2 Proteínas

Quando olhamos uma célula ao microscópio ou analisarmos sua atividade elétrica ou bioquímica, estamos na verdade, observando proteínas. As proteínas constituem a maior parte da massa celular seca. Não são meramente os blocos que constituem as células, elas também executam praticamente todas as funções celulares. O papel central ocupado pelas proteínas é evidenciado no fato de que a informação genética é, em ultima instância, expressa como proteínas [LEHNINGER 2000].

Mais de 33% das proteínas que compõem o corpo humano encontram-se nos músculos. A miosina é a proteína responsável pela formação das fibras, que são elementos contráteis fundamentais para o movimento muscular. Os ossos e cartilagens possuem outros 20% [AMABIS and MEIDANIS 1998].

Muitas proteínas servem como filamentos de suporte para fornecer proteção ou resistência para estruturas biológicas. O principal componente das cartilagens e dos tendões, e que também assegura a estabilidade estrutural do esqueleto é a proteína fibrosa colágeno, que apresenta alta resistência a tensão. Já a queratina é o principal componente estrutural do cabelo, unhas, chifres, lã, escamas e penas e é responsável por proteger os tecidos internos do ambiente externo [LEHNINGER 2000].

Uma molécula de proteína é formada por uma longa cadeia de aminoácidos, cada uma ligada ao seu vizinho por uma ligação peptídica covalente. Por essa razão, as proteínas são também chamadas de polipeptídeos. Cada tipo de proteína tem uma seqüência de aminoácidos que lhe é característica. Milhares de proteínas diferentes são conhecidas, cada uma com sua própria seqüência de aminoácidos.

A seqüência de aminoácidos repetitiva dos átomos ao longo da cadeia principal do polipeptídeo é chamada de cadeia polipeptídica. Ligados a esta cadeia repetitiva estão as cadeias laterais dos diferentes aminoácidos, que não estão envolvidos na formação da ligação peptídica e que conferem a cada aminoácido as suas propriedades únicas: as 20 diferentes cadeias laterais (Figura 2.1). Algumas destas cadeias são apolares e hidrofóbicas (aversão à água), outras são negativas ou positivamente carregadas, algumas são mais negativas e assim por diante [ALBERTS et al. 2004].

Como resultado de todas as interações, cada tipo de proteína tem uma estrutura tridimensional particular, que é determinada pela seqüência dos aminoácidos na sua cadeia.

Sugere-se que a seqüência de aminoácidos representa um papel fundamental na determinação da estrutura tridimensional da proteína que reflete na sua função [LEHNINGER 2000].

2.3 Estrutura das Proteínas

O conhecimento da função das proteínas é de fundamental importância para o entendimento de suas possíveis formas estruturais.

A complexidade da estrutura das proteínas é inerente ao seu tamanho. São considerados quatro níveis de organização estrutural que se denominam:

- **Estrutura primária;**
- **Estrutura secundária;**
- **Estrutura terciária;**
- **Estrutura quaternária.**

O termo específico estrutura primária de uma proteína refere-se à seqüência de resíduos de aminoácidos da cadeia polipeptídica. Vinte aminoácidos diferentes são comumente encontrados nas proteínas. Portanto, os grupos R são variáveis de acordo com cada estrutura de cada resíduo de aminoácido e a estrutura primária varia de acordo com a seqüência que estes se encontram [WANNMACHER and DIAS 1976].

Em geral a substituição de um único aminoácido basta para prejudicar o funcionamento da proteína, podendo causar sérias conseqüências para o organismo. Um exemplo disso é a anemia falsiforme ou siclemia, que é um tipo de anemia hereditária, causada pela troca de um único aminoácido em uma das cadeias polipeptídicas da hemoglobina.

A estrutura secundária refere-se aos arranjos regulares e recorrentes no espaço de resíduos de aminoácidos adjacentes em uma cadeia polipeptídica. Estes elementos regulares podem ser distribuídos em três classes:

- **Hélices α (alpha);**
- **Fitas β (beta);**
- ***Coils*.**

A hélice α é uma conformação da cadeia peptídica que apresenta pontes de hidrogênio entre os grupos (- NH - e - C = O) das ligações peptídicas dentro de uma mesma cadeia [WANNMACHER and DIAS 1976]. Esta estrutura é o arranjo mais simples que uma cadeia polipeptídica pode assumir com as suas ligações rígidas, porém com outras ligações livres para girar, formando uma estrutura em hélice como pode ser observado na Figura (2.3) [LEHNINGER 2000].

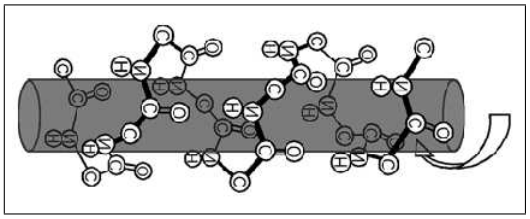


Figura 2.3 – *Representação esquemática de uma α hélice*

Ela é a estrutura predominante na *alpha* queratina. Nas proteínas globulares, perto de $(\frac{1}{4})$ de todos os resíduos de aminoácidos são encontrados em hélice α, variando fortemente o valor desta fração de uma proteína para outra.

Na estrutura em folha β, os aminoácidos assumem a configuração semelhante a de uma folha de papel pregueado, onde as dobras ou pregas, são representadas por pontes de hidrogênio.

Esta estrutura está estabilizada por pontes de hidrogênio entre os grupos amino e carboxila de diferentes cadeias polipeptídicas combinadas no sentido paralelo ou antiparalelo como pode ser observado na Figura 2.4 [ROBERTIS and ROBERTIS-Jr. 1993].

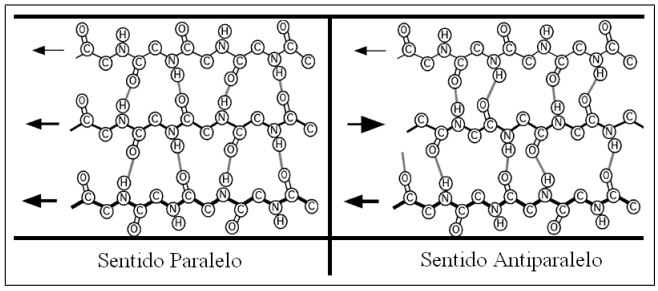


Figura 2.4 – *Representação esquemática de uma folha* β

Na classe *coil*, os segmentos da proteína que não apresentam ligações transversais assumem uma configuração ao acaso. Isto se deve, em parte, ao fato de certos aminoácidos, como a prolina, romperem a estrutura helicoidal [ROBERTIS and ROBERTIS-Jr. 1993].

A estrutura terciária é a forma pela qual as regiões helicoidais e de *coil* se dispõem entre si. Diz respeito à forma tridimensional específica assumida pela proteína como resultado do enovelamento global de toda a cadeia polipeptídica [ROBERTIS and ROBERTIS-Jr. 1993]. A Figura 2.5 ilustra a conformação espacial de uma proteína.

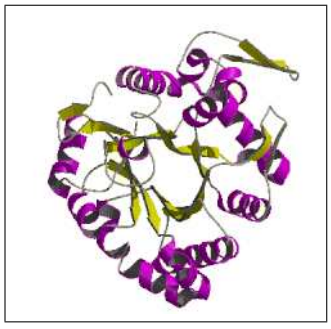


Figura 2.5 – *Representação esquemática da estrutura terciária da uma proteína*

Uma estrutura quaternária é uma associação estável de múltiplas cadeias de polipeptídeos que resultam em uma unidade ativa. Nem todas as proteínas exibem uma estrutura quaternária.

2.4 Classificação das Proteínas

A determinação da estrutura tridimensional das proteínas é um dos problemas centrais da biologia molecular estrutural.

Inúmeros projetos de seqüenciamento de genes acumulam um extenso grupo de seqüências de proteínas. Entretanto, as informações sobre as suas estruturas tridimensionais estão disponíveis somente para uma pequena fração deste grupo [HUBBARD et al. 2000].

Um importante recurso para a determinação da função de novas estruturas de proteínas, é a sua classificação em famílias. Estimativas têm mostrado que o número de famílias existentes na natureza é limitado, dado este que vem sendo comprovado à medida que as estruturas determinadas mais recentemente recaem, na sua maioria, em famílias previamente determinadas [HIGGINS and TAYLOR 2000].

Um considerável avanço foi alcançado ao atribuir uma proteína a uma classe de acordo com o seu *fold*. As proteínas estão definidas como tendo um *fold* comum se suas estruturas secundárias principais apresentarem o mesmo arranjo.

A fim de facilitar o acesso a estas informações, foi construída a base de dados SCOP (*Structural Classification of Proteins*). A base de dados SCOP armazena conjuntos de proteínas que foram classificadas, manualmente, em uma estrutura hierárquica baseado na comparação de suas estruturas [HUBBARD et al. 2000]. Esta base de dados é pública e encontra-se disponível para acesso na Internet em: http://scop.berkeley.edu/ [MURZIM et al. 1995].

A base de dados SCOP tem como objetivo principal, prover uma relação detalhada e inclusiva das relações estruturais e evolutivas das proteínas de estrutura conhecida [HUBBARD et al. 2000]. A Tabela 2.2 descreve os vários níveis que compõem a hierarquia SCOP.

Para a classificação de uma proteína de acordo com a hierarquia SCOP, é realizada manualmente uma inspeção visual e uma comparação das estruturas das proteínas em questão.

A base de dados SCOP é subdividida em níveis hierárquicos, que são: A classe, o *fold*, a superfamília, a família, o domínio e a referência / PDB .

Tabela 2.2 – *Níveis que compõem a hierarquia SCOP.*

Classe	***Folds***	**Superfamílias**	**Famílias**
α	218	376	608
β	144	290	560
α/β	136	222	629
$\alpha+\beta$	279	409	717
Proteínas Multi-domínio	46	46	61
Membranas	47	88	99
Pequenas Proteínas	75	108	171
Total	**945**	**1539**	**2845**

A seqüência ou referência PDB, pode ser considerada o mais baixo nível desta hierarquia. A Figura 2.6, ilustra os principais níveis da hierarquia SCOP.

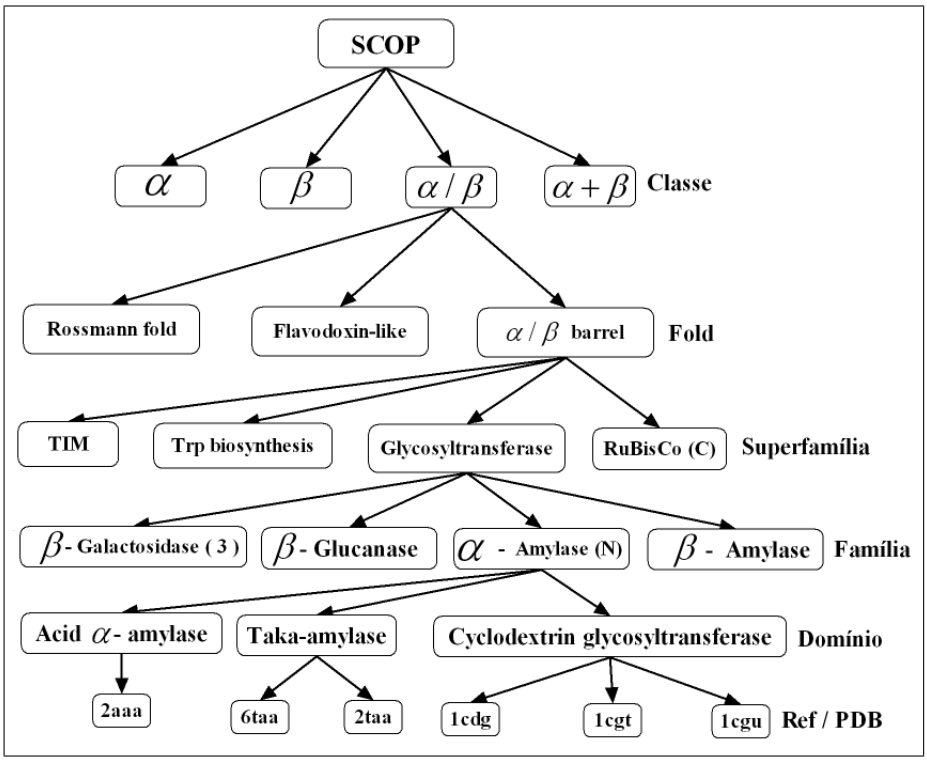


Figura 2.6 – *Hierarquia SCOP, exibindo os níveis principais.*

Podemos observar que o primeiro nível da hierarquia SCOP é a classe. Diferentes *folds* são agrupados em classes, e na sua maioria, são associados a uma das quatro classes estruturais definidas pela hierarquia SCOP.

As proteínas que pertencem a classe estrutural α são formadas quase exclusivamente por hélices α, com as eventuais folhas β localizadas na periferia da proteína. Já as proteínas que são classificadas como β, são constituídas quase

exclusivamente por folhas β, principalmente antiparalelas, com as eventuais hélices α localizadas na periferia.

As proteínas que pertencem a classe α/β, por sua vez, apresentam uma alternância acentuada de hélices α e folhas β, tipicamente paralelas, e formam um aglomerado central rodeado por hélices α. Em contrapartida, a classe $\alpha + \beta$ inclui as proteínas formadas por um número significativo de hélices α e folhas β. Por sua vez, esta classe não é denominada por nenhum dos elementos, nem apresentam a alternância observada na classe α/β.

Na Figura 2.7 podemos observar as quatro classes principais da hierarquia SCOP. Outras classes foram atribuídas para peptídeos, pequenas proteínas, modelos teóricos e hidratos de carbono.

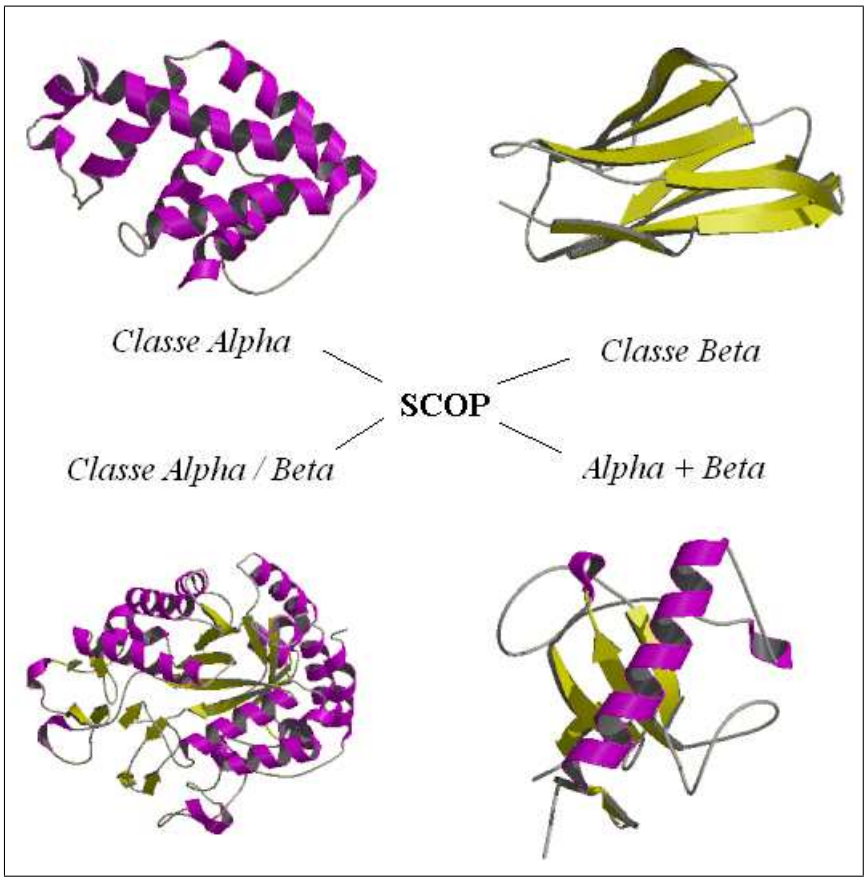


Figura 2.7 – *Representação das quatro classes principais da hierarquia SCOP.*

Posteriormente temos o segundo nível da hierarquia SCOP, o*fold*. As famílias e superfamílias são definidas como tendo um *fold* em comum se suas proteínas possuírem as mesmas estruturas secundárias principais no mesmo arranjo.

Acredita-se que as semelhanças estruturais das proteínas que estão classificadas no mesmo *fold* originam-se de suas propriedades físicas e químicas que favorecem

determinadas combinações estruturais.

Como descrição, os *folds* podem ser avaliados como sendo a representação da arquitetura das proteínas. Duas proteínas são classificadas como tendo um *fold* em comum, se possuem elementos estruturais comparáveis da estrutura secundária, com a mesma topologia das conexões.

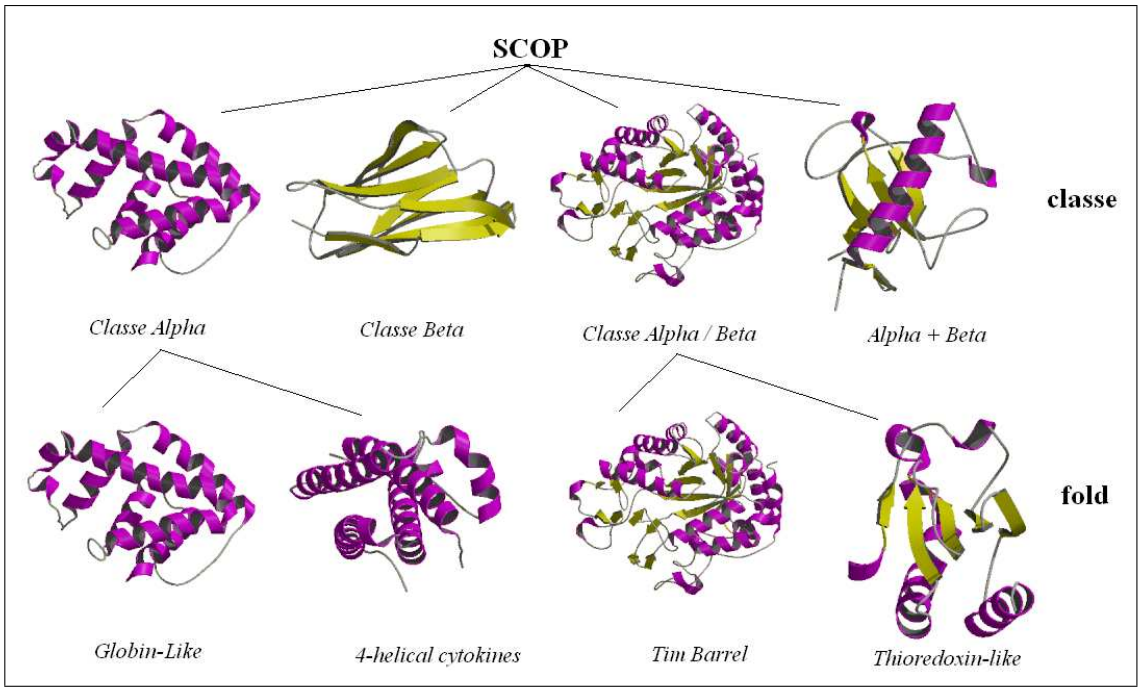


Figura 2.8 – *Representação de quatro folds da hierarquia SCOP.*

Tomando como exemplo o *fold* TIM barrel, visualizado na Figura 2.8, podemos observar que ele se distingue por possuir um núcleo de oito folhas β paralelas que se encontram juntamente fechadas como se fossem aduelas de um barril. O exterior é rodeado por hélices α as quais são encontradas em toda parte externa deste barril.

Para que as proteínas possam ser agrupadas em famílias com um relacionamento evolucionário provável, elas devem atender a um dos seguintes critérios:

- Todas as proteínas possuem similaridades significativas na seqüência;
- As proteínas cujas funções e características estruturais são bastante similares.

As superfamílias podem ser definidas como sendo a composição das famílias com um relacionamento evolucionário provável baseado em características estruturais e funcionais comuns. Além disso, as superfamílias também agrupam as famílias que possuem proteínas com um baixo grau de identidade na seqüência, porém com características funcionais que sugerem uma origem evolucionária provavelmente comum.

O último nível da hierarquia SCOP é o domínio da categoria. Tem como objetivo diferenciar possíveis regiões independentes encontradas em grandes proteínas.

Capítulo 3

Teoria da Otimização Matemática

O processo de treinamento das Máquinas de Suporte Vetorial envolve a resolução de um problema de otimização matemática. Neste capítulo, são descritas algumas definições da teoria de otimização necessárias para fundamentar a formulação e a tarefa de treinamento das Máquinas de Suporte Vetorial a serem apresentadas no Capítulo 4.

3.1 Introdução

Uma abordagem clássica das Máquinas de Suporte Vetorial é o processo de transformação do problema de otimização primal em sua representação dual, através da teoria de Lagrange. Entre os vários benefícios que esta abordagem proporciona, destaca - se a redução dos problemas relacionados com a alta dimensionalidade dos dados.

A teoria da otimização matemática é uma subárea da matemática que envolve a determinação de soluções para problemas modelados por funções e que devem ser determinadas de forma a minimizar ou maximizar uma certa função que é sujeita a restrições. A otimização matemática é responsável pelo desenvolvimento de algoritmos que buscam encontrar tais soluções. Ela não somente provê técnicas algorítmicas, como também é responsável pela definição de condições necessárias para uma função dada ser uma solução [CRISTIANINI and TAYLOR 2000].

Na maioria dos casos, questões envolvendo a teoria de otimização matemática iniciam-se com um conjunto de variáveis ou parâmetros independentes e incluem condições ou restrições que limitam os valores destas variáveis. A solução para um problema de otimização, é o conjunto destas variáveis que satisfazem as restrições e maximizam ou minimizam a função de custo [BURGES 1998].

3.1.1 Problema Primordial

De uma forma geral para o problema a ser considerado, deve-se determinar o máximo ou o mínimo de uma função sujeito a algumas restrições. A forma geral em que se apresenta o problema de otimização a seguir:

Definição 3.1.1 *Problema da otimização primordial dadas as funções* f, g_i *e* h_i, *definidas no domínio* $\Omega \subseteq \Re^n$,

$$\begin{array}{lll} \text{Minimizar} & f(\mathbf{w}) & \mathbf{w} \in \Omega \\ \text{Sujeito as restrições} \begin{cases} & \\ & \end{cases} & \begin{array}{ll} g_i(\mathbf{w}) \leq 0 & \text{i=1,2,...,k;} \\ h_i(\mathbf{w}) = 0 & \text{i=1,2,...,m.} \end{array} \end{array}$$

Sendo que $f(\mathbf{w})$ é denominada de *função objetivo*, e as demais relações, de restrições de desigualdade $g_i(\mathbf{w})$ e de restrições de igualdade $h_i(\mathbf{w})$, respectivamente [LUEMBERGUER 1973].

O valor ótimo da função objetivo é denominado *valor do problema de otimização*. A região de domínio onde a função objetiva esta definida e onde todas as restrições são satisfeitas é denominada *região factível*. Um outro conceito importante é a *restrição ativa*, uma restrição de desigualdade $g_i(\mathbf{w}) \leq 0$ é denominada ativa para um ponto factível $\mathbf{w}$ se $g_i(\mathbf{w}) = 0$ e inativa se $g_i(\mathbf{w}) < 0$. Por convenção, qualquer restrição de igualdade $h_i(\mathbf{w}) = 0$, é classificada como *ativa*, para qualquer ponto factível $\mathbf{w}$.

A restrição ativa, para um ponto factível $\mathbf{w}$, restringe a região de factibilidade nas vizinhanças de $\mathbf{w}$, enquanto as outras regiões inativas, não têm influência nas vizinhanças de $\mathbf{w}$.

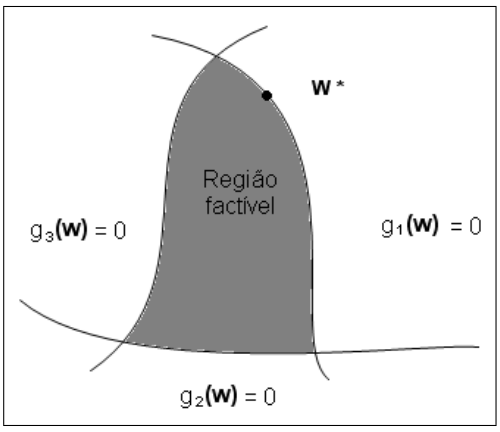


Figura 3.1 – *Representação da dependência entre a solução ótima e as restrições ativas e inativas.*

A Figura 3.1 demonstra a dependência entre a solução ótima e as restrições ativas e inativas. A solução ótima $\mathbf{w}^*$ não depende das restrições inativas

$g_2(\mathbf{w})$ e $g_3(\mathbf{w})$. Se forem conhecidas quais as restrições que são ativas para o problema de otimização, a solução torna-se um ponto de mínimo local, definido ignorando as restrições inativas e tratando as restrições ativas como restrições de igualdade [LUEMBERGUER 1973].

Definição 3.1.2 *Convexidade Uma função f definida em um conjunto convexo Ω é denominada convexa se, para cada w_1 e $w_2 \in \Omega$ e cada α, $0 \leq \alpha \leq 1$, for satisfeita a Relação(3.1)*

$$f(\alpha w_1 + (1-\alpha)w_2) \leq \alpha f(w_1) + (1-\alpha)f(w_2). \tag{3.1}$$

Se, para cada $0 < \alpha < 1$ e $w_1 \neq w_2$, for satisfeita a Relação (3.2) então f é denominada *estritamente convexa*.

$$f(\alpha w_1 + (1-\alpha)w_2) < \alpha f(w_1) + (1-\alpha)f(w_2). \tag{3.2}$$

Tendo em vista que, para o treinamento das Máquinas de Suporte Vetorial, as restrições serão lineares e a função objetivo será convexa, o problema de otimização envolvido neste treinamento também será convexo [CRISTIANINI and TAYLOR 2000].

Para a resolução de um problema de otimização deste tipo, é fundamental o estudo da teoria de Lagrange.

3.2 Teoria de Lagrange

Para a resolução de problemas de otimização que não apresentam restrições de desigualdade a minimização da função-objetivo pode ser caracterizada na forma proposta por Fermat em 1629 e generalizada por Lagrange em 1797 [CRISTIANINI and TAYLOR 2000]. Em 1951, Kuhn e Tucker estenderam o método e permitiram restrições de desigualdade [SEMOLINI 2003].

Teorema 3.2.1 (Fermat) *A condição necessária para que $\boldsymbol{w}^*$ seja um mínimo de $f(\boldsymbol{w})$, sendo f uma função quadrática convexa, é $\frac{\partial f(\boldsymbol{w}^*)}{\partial \boldsymbol{w}} = 0$. Esta condição, junto com a de convexidade de f, é uma condição suficiente.*

Para a resolução de problemas de otimização envolvendo restrições, é necessário a definição de uma função, conhecida como função Lagrangeana expressa pela soma da função objetivo e uma combinação linear da função de restrição, onde os coeficientes α e β são denominados de multiplicadores de Lagrange.

Definição 3.2.1 *Dado um problema de otimização que possui uma função objetivo* $f(\boldsymbol{w})$ *e restrições dadas por igualdades* $h_i(\boldsymbol{w}) = 0, i = 1, ..., m$, *a função Lagrangeana é definida como:*

$$L(\mathbf{w}, \beta) = f(\mathbf{w}) + \sum_{i=1}^{m} \beta_i h_i(\mathbf{w}) \tag{3.3}$$

A teoria de Lagrange em problemas convexos, permite a descrição Dual. Esta descrição é mais simples de solucionar computacionalmente pois o manuseio direto das restrições de desigualdade é difícil de ser solucionado [CRISTIANINI and TAYLOR 2000].

No problema Dual, as variáveis duais são representas pelos multiplicadores de Lagrange (α e β). Neste método, as variáveis duais são desconhecidas e precisam ser encontradas para a resolução do problema. A definição 3.2.2 descreve esta idéia.

Definição 3.2.2 (Generalização de Lagrange) *Dado o problema de otimização com domínio convexo* $\Omega \subseteq \Re^n$.

$$\begin{array}{lll} \text{Minimizar} & f(\mathbf{w}) & \mathbf{w} \in \Omega \\ \text{Sujeito ás restrições} \left\{ \begin{array}{l} \\ \end{array} \right. & \begin{array}{l} g_i(\mathbf{w}) \leq 0 \\ h_i(\mathbf{w}) = 0 \end{array} & \begin{array}{l} i=1,2,...,k; \\ i=1,2,...,m. \end{array} \end{array}$$

A função Lagrangeana generalizada pode ser definida como:

$$L(\mathbf{w}, \alpha, \beta) = f(\mathbf{w}) + \sum_{i=1}^{k} \alpha_i g_i(\mathbf{w}) + \sum_{i=1}^{m} \beta_i h_i(\mathbf{w}). \tag{3.4}$$

As condições para uma solução ótima de um problema de otimização geral, são apresentadas pelas condições de Karush-Kuhn-Tucker (**KKT**). Estas condições são utilizadas no Capítulo 4.

3.3 Condições de Karush-Kuhn-Tucker

Definição 3.3.1 (Karush-Kuhn-Tucker) *Dado o problema de otimização com domínio convexo* $\Omega \subseteq \Re^n$,

$$\begin{array}{lll} \text{Minimizar} & f(\mathbf{w}) & \mathbf{w} \in \Omega \\ \text{Sujeito ás restrições} \left\{ \begin{array}{l} \\ \end{array} \right. & \begin{array}{l} g_i(\mathbf{w}) \leq 0 \\ h_i(\mathbf{w}) = 0 \end{array} & \begin{array}{l} i=1,2,...,k, \\ i=1,2,...,m, \end{array} \end{array}$$

com $f \in C^1$ convexa e, g_i, h_i funções de restrições lineares. As condições necessárias e suficientes para que um ponto $\mathbf{w}^*$ seja ótimo, é que devem existir valores α^* e β^*, tais que:

$$\frac{\partial L(\mathbf{w}^*, \alpha^*, \beta^*)}{\partial \mathbf{w}} = 0; \tag{3.5}$$

$$\frac{\partial L(\mathbf{w}^*, \alpha^*, \beta^*)}{\partial \beta} = 0; \tag{3.6}$$

$$\alpha_i^* g_i(\mathbf{w}^*) = 0, \quad i = 1, ..., k; \tag{3.7}$$

$$g_i(w^*) \leq 0, \quad i = 1, ..., k; \tag{3.8}$$

$$\alpha_i^* \geq 0, \quad i = 1, ..., k. \tag{3.9}$$

A Relação (3.9) é denominada de condição preliminar de Karush-Kuhn-Tucker, a qual implica que para restrições ativas $\alpha_i^* \geq 0$, enquanto que, para restrições inativas, $\alpha_i^* = 0$ [CRISTIANINI and TAYLOR 2000].

O problema de otimização convexo através da teoria de Lagrange proporciona uma alternativa de descrição dual, que pode ser resolvida mais facilmente do que uma descrição primal, pois esta possui restrições de desigualdade que não são triviais de serem atendidas.

A teoria que envolve as Máquinas de Suporte Vetorial tradicionalmente utiliza esta estratégia, pois a representação dual permite trabalhar com um espaço de alta dimensão. Tal fato que ocorre devido ao número de parâmetros ajustados não depender da dimensão dos dados de entrada.

Capítulo 4

Conceitos Básicos em Máquinas de Suporte Vetorial

Neste capítulo serão abordados alguns aspectos teóricos envolvendo MSV. Inicialmente será apresentada uma introdução sobre o assunto, logo após, serão descritos os conceitos de classificação linearmente separável onde conceitua-se o hiperplano de separação ótimo e os vetores de suporte. Em seguida serão apresentados os conceitos de classificação não linearmente separável e superfícies não-lineares. E por fim, são descritos o espaço de características, o mapeamento implícito e as funções de *kernel*.

4.1 Introdução

As Máquinas de Suporte Vetorial (MSV) foram recentemente introduzidas como sendo uma técnica de reconhecimento de padrões [PONTIL and VERRI 1997]. Inicialmente proposta por Vapnik e colaboradores, esta técnica de aprendizagem mostrou-se ser um método muito poderoso, o qual em poucos anos de sua utilização já superou uma vasta quantidade de sistemas em uma ampla variedade de aplicações [CRISTIANINI and TAYLOR 2000].

Diversas técnicas tradicionais de reconhecimento de padrões são baseadas na minimização do risco empírico, isto é, tenta-se otimizar o desempenho sobre o conjunto de treinamento. As MSV minimizam o risco estrutural, ou seja, a probabilidade de classificar de forma errada padrões ainda não vistos por uma distribuição de probabilidade dos dados fixa e desconhecida [CRISTIANINI and TAYLOR 2000].

O objetivo deste tipo de classificação é elaborar uma forma eficiente do ponto de vista computacional de modo a maximizar a margem entre os dados e,

conseqüentemente, melhorar a sua capacidade de generalização.

A teoria de generalização fornece uma orientação clara sobre como controlar a capacidade e prevenir modelos 'ruins', controlando as medidas das margens dos hiperplanos, enquanto a teoria da otimização fornece algumas técnicas matemáticas que são necessárias para encontrar hiperplanos otimizando essas medidas. Existem vários limites de generalização, como por exemplo podemos querer otimizar a margem máxima, a distribuição das margens ou o número de vetores de suporte. A abordagem mais usual é tratar o problema minimizando a norma do vetor peso [CRISTIANINI and TAYLOR 2000].

A idéia de máquinas de suporte vetorial no caso linearmente separável, pode ser explicada de forma simples. Dado um conjunto de treinamento X que possui pontos de duas classes, uma MSV separa estas classes através de um hiperplano que é determinado por alguns pontos de X os quais são denominados vetores de suporte. No caso separável, este hiperplano maximiza a margem, ou duas vezes a distância mínima de cada classe ao hiperplano. Todos os vetores de suporte caem na mesma distância mínima a partir do hiperplano, recebendo o nome de vetores margem. Nos casos reais, as duas classes podem não ser separáveis fazendo com que tanto o hiperplano quanto os vetores de suporte sejam obtidos da solução de um problema de otimização com restrições [PONTIL and VERRI 1997].

4.2 Classificação Linearmente Separável

A classificação linear é usualmente descrita utilizando uma função real dada por $f : \mathbf{X} \subseteq \Re^n \rightarrow \Re$ na seguinte forma: a entrada $\mathbf{x} = (x_1, ..., x_n)^T$ é atribuída a uma classe positiva se $f(\mathbf{x}) \geq 0$, e a uma classe negativa caso contrário. Considera-se o caso onde $f(\mathbf{x})$ é uma função linear de $\mathbf{x} \in \mathbf{X}$, então ela pode ser escrita como:

$$\begin{aligned} f(\mathbf{x}) &= \langle \mathbf{w} \cdot \mathbf{x} \rangle + b \\ &= \sum_{i=1}^{n} w_i x_i + b. \end{aligned}$$

onde $(\mathbf{w}, b) \in \Re^n \times \Re^n$ são parâmetros que controlam a função e a regra de decisão é dada por sinal$(f(\mathbf{x}))$, onde é usada a convenção que sinal$(0) = 1$. Estes parâmetros devem ser aprendidos a partir dos dados [CRISTIANINI and TAYLOR 2000].

A interpretação geométrica deste tipo de hipótese é que o espaço de entrada X é dividido em duas partes pelo hiperplano definido pela equação $\langle \mathbf{w} \cdot \mathbf{x} \rangle + b = 0$ como pode ser observado na Figura (4.1) [CRISTIANINI and TAYLOR 2000]. Um hiperplano é um subespaço afim de dimensão $n - 1$ que divide o espaço em duas metades, as quais são formadas pelas entradas das duas classes distintas.

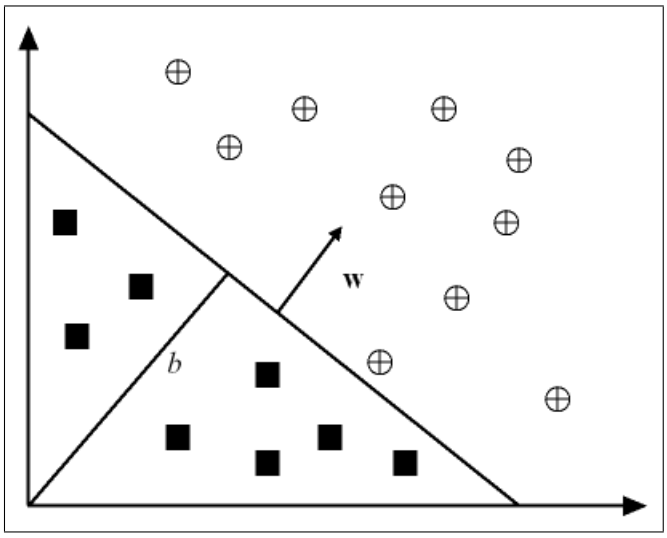


Figura 4.1 – *Hiperplano separando* $\mathbf{w}$,b *para um conjunto de treinamento de duas dimensões.*

Como exemplo podemos observar a Figura 4.1 onde a linha diagonal corresponde ao hiperplano com a classe positiva acima representada por um circulo cruzado e a classe negativa abaixo denotada por um quadrado. O vetor $\mathbf{w}$ define uma direção perpendicular ao hiperplano, enquanto variar o valor de b o hiperplano move-se paralelamente a ele mesmo. As quantidades $\mathbf{w}$ e b serão referenciadas como *vetor peso* e *tendência*, termos bastante utilizados na literatura de redes neurais [CRISTIANINI and TAYLOR 2000].

Pelo fato das MSV utilizarem aprendizado supervisionado a partir de exemplos, será necessária a introdução de alguma notação para referenciar entradas, saídas, conjuntos de treinamento, etc.

Usualmente utiliza-se $\boldsymbol{X}$ para denotar o espaço de entrada e $\boldsymbol{Y}$ para denotar o domínio da saída. Tipicamente tem-se $\boldsymbol{X} \subseteq \Re^n$, enquanto para a classificação binária $\boldsymbol{Y} = \{-1, +1\}$ e para classificação m-classes $\boldsymbol{Y} = \{1, 2, ..., m\}$. O conjunto de treinamento é um grupo de exemplos selecionados de acordo com um determinado contexto contendo características específicas capazes de identificar determinadas classes. Também podem ser chamados de dados de treinamento, e é geralmente denotado por:

$$S = ((x_1, y_1), ..., (x_l, y_l)) \subseteq (\boldsymbol{X} \times \boldsymbol{Y})^l,$$

onde l é o número de exemplos.

Trata-se x_i como exemplos ou instâncias e y_i como rótulos. O conjunto de treinamento S pode ser considerado trivial se todos os exemplos possuírem rótulos iguais.

4.2.1 Hiperplano de Separação Ótimo - HSO

Basicamente pode-se colocar a idéia de MSV como sendo encontrar o hiperplano h com uma margem máxima de separação entre os dados de classes distintas representados no espaço vetorial $\boldsymbol{X}$ [SCHÖLKOPF and SMOLA 2002].

Um hiperplano é denominado de margem máxima ou Separação Ótima caso separe um conjunto de vetores de classes distintas e a distância entre os vetores mais próximos ao hiperplano seja máxima [VAPNIK 1998].

Na Figura 4.2, em (a) observamos um hiperplano com margem não-máxima e em (b) um hiperplano com margem máxima.

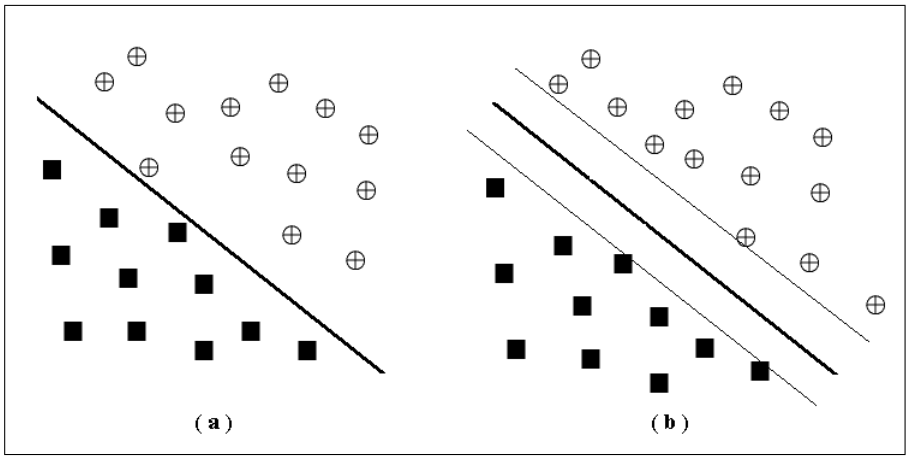


Figura 4.2 – *(a) Hiperplano com margem não-máxima e (b) Hiperplano com margem máxima (adaptado de [PONTIL and VERRI 1997]).*

Para o problema de classificação utilizando o aprendizado supervisionado, as amostras de treinamento são formadas pelo conjunto de dados de entrada associados às suas correspondentes respostas pré-classificadas, indicadas através de rótulos ou dados de saída. O objetivo após o treinamento é classificar novas amostras ainda não rotuladas.

No caso linearmente separável, o algoritmo de vetor suporte procura o hiperplano de separação com a maior margem, e pode ser representado da seguinte forma: Tomamos por base que os dados de treinamento satisfazem as seguintes restrições:

$$\langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b \geq +1, \text{ para } y_i = +1 \tag{4.1}$$

$$\langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b \leq -1, \text{ para } y_i = -1 \tag{4.2}$$

As equações podem ser combinadas na seguinte inequação:

$$y_i(\langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b) - 1 \geq 0, \text{ para } i = 1, ..., l \tag{4.3}$$

Sem perda de generalidade, pode-se considerar os hiperplanos canônicos $(\mathbf{w}, \mathbf{x}_i)$, onde os hiperplanos satisfazem a Equação (4.4) e onde os parâmetros $\mathbf{w}, b$ são restringidos por,

$$min_{i\ =\ 1,...,l} |\langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b| = +1. \tag{4.4}$$

Esta restrição aplicada à parametrização é uma alternativa preferível para simplificar a formulação do problema. Trocando em palavras, o que ela define é: *a norma do vetor peso* $\|\boldsymbol{w}\|$ *deve ser igual ao inverso da distância do ponto mais próximo no conjunto de dados ao hiperplano.*

Um hiperplano de separação em forma canônica deve satisfazer as seguintes restrições,

$$y_i(\langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b) \geq 1 \text{ para } i = 1, ..., l. \tag{4.5}$$

O hiperplano representado pelo par $(\mathbf{w}, b)$ define a equação

$$f(x) = \langle \mathbf{w} \cdot \mathbf{x} \rangle + b = 0. \tag{4.6}$$

chamada de hiperplano de separação, onde $\mathbf{w}$ é normal ao hiperplano, $\frac{|b|}{\|\mathbf{w}\|}$ é a distância perpendicular do hiperplano à origem e $\|\mathbf{w}\| = \sqrt{\langle \mathbf{w} \cdot \mathbf{w} \rangle} = \sqrt{\sum_{i=1}^{l} \mathbf{x}_i^2}$ é a norma euclidiana de $\mathbf{w}$. Seja $d_+(d_-)$ a menor distância a partir do hiperplano ao exemplo positivo (negativo) mais próximo. Define-se a margemρ de um hiperplano de separação como sendo a maior margem geométrica entre todos os hiperplanos, ou seja, $\rho = d_+ + d_-$. A distância $d_i(\mathbf{w}, b; \mathbf{x}_i)$ de um ponto $\mathbf{x}_i$ ao hiperplano $(\mathbf{w}, b)$, ou seja, sua margem é:

$$d_i(\mathbf{w}, b; \mathbf{x}_i) = \frac{|\langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b|}{\| \mathbf{w} \|} = \frac{y_i(\langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b)}{\|\mathbf{w}\|}. \tag{4.7}$$

Combinando as Equações (4.3) e (4.7), para todo $\mathbf{x}_i \in S$, temos:

$$d_i(\mathbf{w}, b; x_i) \geq \frac{1}{\|\mathbf{w}\|}. \tag{4.8}$$

Portanto, $\frac{1}{\|\mathbf{w}\|}$ é o limite inferior da distância entre os pontos $\mathbf{x}_i$ e o hiperplano de separação $(\mathbf{w}, b)$. As distâncias d_+ e d_- ficam

$$d_+ = d_- = \frac{1}{\|\mathbf{w}\|}. \tag{4.9}$$

Como a margem é dada por $\rho = (d_+ + d_-)$, então

$$\rho = \frac{2}{\| \mathbf{w} \|}. \tag{4.10}$$

O hiperplano de separação ótimo é dado pela minimização da margem ρ, sujeita às restrições da Equação (4.5). Logo o hiperplano que divide otimamente os dados é aquele que minimiza

$$\Phi(\mathbf{w}) = \frac{1}{2}\langle \mathbf{w} \cdot \mathbf{w} \rangle = \frac{1}{2} \| \mathbf{w} \|^2 . \tag{4.11}$$

Formalmente temos,

Problema **T1**

Minimize $\frac{1}{2} \parallel \mathbf{w} \parallel^2$

Sujeito a $y_i(\langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b) \geq 1,$ para $i = 1, ..., l.$

Desta formulação, pode-se observar que se os parâmetros$(\mathbf{w}, b)$ resolvem o problema **T1**, então para pelo menos um $\mathbf{x}_i \in S$ tem-se $y_i(\langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b) = 1$. Isso implica que a solução de **T1** é sempre um hiperplano de separação na representação canônica. Ainda, o parâmetro b entra nas restrições mas não na função objetivo a ser minimizada.

A Figura (4.3) apresenta de forma gráfica a margem geométrica de um ponto$\mathbf{x}_i$ e a margem ρ do hiperplano de separação ótimo. Os círculos cruzados representam os exemplos positivos e os quadrados representam os exemplos negativos. Os exemplos que caem sobre as margens, representadas por linhas tracejadas, são os vetores suporte para esse conjunto de treinamento.

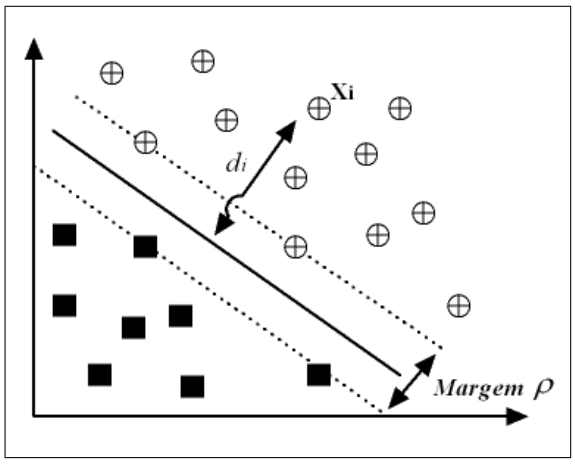


Figura 4.3 – *Margem geométrica de um ponto* $\boldsymbol{x}_i$ *e a margem* ρ *do hiperplano de separação ótimo*

4.2.2 Vetores de Suporte

O problema **T1** pode ser resolvido utilizando o método clássico de multiplicadores de Lagrange que foram introduzidos no Capítulo 3. Inicialmente as restrições (4.5) serão substituídas por restrições de multiplicadores de Lagrange. Nesta reformularão do problema, os dados de treinamento somente aparecerão (nos algoritmos de treinamento e teste) na forma de produtos internos entre vetores. Esta é uma propriedade crucial que permitirá generalizar o procedimento para o caso não linear [BURGES 1998].

Assim, seja $\alpha = (\alpha_1, ..., \alpha_l)$ os l multiplicadores de Lagrange não negativos associados com às restrições (4.5), a solução do problema **T1** equivale a determinar

o ponto de sela da função,

$$L(\mathbf{w}, b, \alpha) = \frac{1}{2} \parallel \mathbf{w} \parallel^2 - \sum_{i=1}^{l} \alpha_i \{y_i(\langle \mathbf{x}_i \cdot \mathbf{w} \rangle + b) - 1\} \tag{4.12}$$

sujeita a $\alpha_i > 0$ para $i = 1, ..., l$. No ponto de sela, L tem um valor mínimo para $\mathbf{w} = \mathbf{w}^*$ e $b = b^*$, e um valor máximo para $\alpha = \alpha^*$, assim podemos escrever:

$$\frac{\partial L}{\partial b} = \sum_{i=1}^{l} y_i \alpha_i = 0, \tag{4.13}$$

$$\frac{\partial L}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^{l} y_i \alpha_i \mathbf{x}_i = 0, \tag{4.14}$$

$$\text{com} \quad \frac{\partial L}{\partial \mathbf{w}} = \left(\frac{\partial L}{\partial w_1}, \frac{\partial L}{\partial w_2}, ..., \frac{\partial L}{\partial w_l} \right). \tag{4.15}$$

Substituindo as Equações (4.13) e (4.14) no lado direito de (4.12), observa-se que o problema **T1** reduz-se a maximização da função

$$L(\alpha) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i \alpha_j y_i y_j \langle \mathbf{x}_i \cdot \mathbf{x}_j \rangle, \tag{4.16}$$

sujeito a restrição (4.13) com $\alpha \geq 0$. Este novo problema é denominado problema dual, e pode ser formulado da seguinte maneira:

Problema	**T2**
Maximize	$\sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i \alpha_j y_i y_j \langle \mathbf{x}_i \cdot \mathbf{x}_j \rangle$
Sujeito a	$\sum_{i=1}^{l} \alpha_i y_i = 0$
	$\alpha \geq 0$

Para o par $(\mathbf{w}^*, b^*)$, da Equação (4.14) segue que

$$\mathbf{w}^* = \sum_{i=1}^{l} \alpha_i^* y_i \mathbf{x}_i \tag{4.17}$$

enquanto b^* pode ser determinado pelas equações de **Karush-Kuhn-Tucker** (KKT) que foram introduzidas no Capítulo 3. Temos então

$$\alpha_i^* (y_i(\langle \mathbf{w}^* \cdot \mathbf{x}_i \rangle + b^*) - 1) = 0, \text{ com } i = 1, 2, ..., l. \tag{4.18}$$

Podemos observar na Equação (4.18) que somente os α_i^*'s podem ser diferentes de zero, são aqueles para os quais as restrições (4.5) são satisfeitas com o sinal de igualdade. Os pontos $\mathbf{x}_i$ correspondentes, chamados de vetor suporte, são os pontos de S mais próximos do **HSO**.

Dado o vetor suporte $\mathbf{x}_j$, o parâmetro b^* pode ser obtido da condição de **KKT** por

$$b^* = y_j - \langle \mathbf{w}^* \cdot \mathbf{x}_j \rangle. \tag{4.19}$$

O problema da classificação de um novo ponto dado $\mathbf{x}$ é agora resolvido calculando

$$\text{sinal}(\langle \mathbf{w}^* \cdot \mathbf{x}_i \rangle + b^*). \tag{4.20}$$

Por fim, os vetores suporte agrupam toda a informação contida no conjunto de dados de treinamento S que é necessário para a classificação de novos pontos de dados.

4.3 Classes Linearmente Inseparáveis

Em alguns casos, as classes não são linearmente separáveis, causando uma difícil classificação. Nestes casos, não é possível construir um hiperplano separando os dados de treinamento sem erros de classificação. Entretanto, é possível encontrar um hiperplano que minimiza o erro de classificação junto às amostras de treinamento.

Com o objetivo de tornar o método descrito na seção anterior capaz de manipular dados não linearmente separáveis, é necessário "relaxar" as restrições do problema.

Este método pode ser generalizado introduzindo l variáveis não negativas, $\xi = (\xi_1, \xi_2, ..., \xi_l)$, tais que:

$$y_i(\langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b) \geq 1 - \xi_i, \text{ para } i = 1, ..., l. \tag{4.21}$$

Os escalares ξ são denominados *variáveis de folga*, e medem os desvios dos pontos de treinamento para a condição ideal de separação das classes. Se o ponto $\mathbf{x}_i$ satisfaz a inequação (4.5), então ξ_i é nulo e (4.21) reduz-se a (4.5). Caso contrário, se o ponto $\mathbf{x}_i$ não satisfaz a equação (4.21), o termo $-\xi_i$ é adicionado ao lado direito de (4.5) para obter a equação (4.21). O **HSO** generalizado é então considerado como a solução para

Problema	**T3**
Minimize	$\frac{1}{2} \parallel \mathbf{w} \parallel^2 + C\sum_{i=1}^{l} \xi_i$
Sujeito a	$y_i(\langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b) \geq 1 - \xi_i, \text{ para } i = 1, ..., l$
	$\xi \geq 0.$

O termo $C\sum_{i=1}^{l} \xi_i$ pode ser considerado como sendo uma medida de erro de classificação. Ele faz o **HSO** menos sensível à presença de exemplos 'mal comportados' no conjunto de treinamento. O parâmetro C pode ser considerado como um parâmetro de regularização. O **HSO** tende a maximizar a distância mínima $\frac{1}{\|\mathbf{w}\|}$ para um C pequeno e minimizar o número de pontos classificados errados para um C grande. Para valores intermediários de C a solução do problema **T3** compensa o erro para uma margem grande.

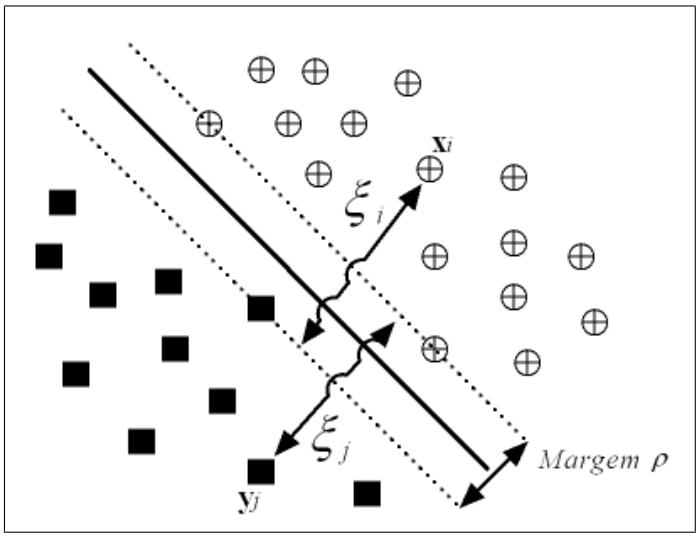


Figura 4.4 – *Exemplos de variáveis de folga* ξ_i *e* ξ_j.

A Figura 4.4 apresenta um exemplo das variáveis de folga ξ_i e ξ_j para o problema de classificação. Elas medem (informalmente) quanto um ponto falhou em ter uma margem de $\frac{\rho}{2}$ a partir do hiperplano. Se $\xi_i > \frac{\rho}{2}$, então $\mathbf{x}_i$ é classificado de forma errada por $(\mathbf{w}, b)$.

Comparando com o caso linearmente separável, o problema **T3** pode ser transformado no dual

Problema	**T4**
Maximize	$\sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i \alpha_j y_i y_j \langle \mathbf{x}_i \cdot \mathbf{x}_j \rangle$
Sujeito a	$\sum_{i=1}^{l} y_i \alpha_i = 0$
	$0 \leq \alpha_i \leq C,$ para $i = 1, ..., l$

Das restrições do problema **T4** segue que se C é suficientemente grande e o conjunto S é linearmente separável e o problema **T4** reduz-se a **T2**.

Para o par $(\mathbf{w}^*, b^*)$, encontramos que

$$\mathbf{w}^* = \sum_{i=1}^{l} \alpha_i^* y_i \mathbf{x}_i, \tag{4.22}$$

enquanto b^* pode novamente ser determinado a partir de α, solução do problema **T4**, e pelas novas condições de **Karush-Kuhn-Tucher**

$$\alpha_i^* (y_i \langle \mathbf{w}^* \cdot \mathbf{x}_i \rangle + b^*) - 1 + \xi_i^* = 0, \text{ para } i = 1, 2, ..., l, \tag{4.23}$$

$$(C - \alpha_i^*) \xi_i^* = 0, \text{ para } i = 1, 2, ..., l, \tag{4.24}$$

onde ξ_i^* são os valores dos ξ_i's no ponto de sela. Analogamente ao caso separável, os pontos $\mathbf{x}_i$ onde $\alpha_i^* > 0$ são chamados de *vetores suporte*. Neste caso, a principal diferença é a distinção feita entre vetores de suporte para os casos onde $\xi_i^* > C$ e

$\xi_i^* = C$. No caso da condição (4.24) segue $\xi_i^* = 0$, e logo, os vetores suporte caem a uma distância $\frac{1}{\|\mathbf{W}^*\|}$ do **HSO**. Estes vetores suporte são denominados vetores margem.

Os vetores suporte para os $\alpha_i^* = C$, são pontos classificados errados se $\xi_i > 1$. Se $0 < \xi \leq 1$, então são pontos corretamente classificados, entretanto mais próximo que $\frac{1}{\|\mathbf{W}^*\|}$ do **HSO** ou, em alguns casos, pontos caem sobre a margem $\xi_i = 0$.

4.4 Superfícies Não Lineares

Quando trabalhamos com aplicações do mundo real, os dados geralmente não são linearmente separáveis. Uma das características mais interessantes das MSV é a sua capacidade de aprender em espaços não lineares. A idéia básica é fazer um mapeamento dos dados para um espaço de características onde eles possam ser linearmente separáveis.

A forma de separação dos dados neste caso pode ser diferente a de um **HSO**, como podemos observar na Figura 4.5. Nesta representação, os dois conjuntos de dados são linearmente inseparáveis. A linha sólida é o hiperplano de separação ótimo, as linhas tracejadas são as margens, os pontos posicionamos sob a linha tracejada são denominados vetores margem ($\alpha_i^* < C$) e os exemplos envolvidos por um circulo são os erros ($\alpha_i^* = C$).

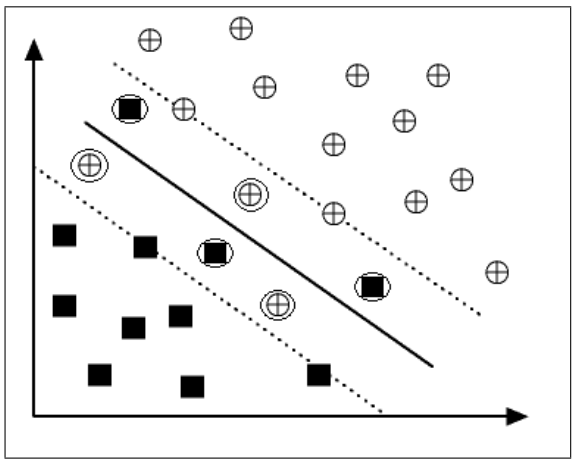


Figura 4.5 – *Hiperplano de separação ótimo generalizado.*

4.4.1 Espaço de Características

As técnicas de aprendizagem de máquina sofrem influencia direta dos dados, mais especificamente de seus atributos. Por exemplo, se o número de atributos nos dados for muito grande, o desempenho computacional do algoritmo de

aprendizagem pode ser degradado, ou sua precisão, no caso do número de atributos ser muito pequeno ou insignificante (redundante ou impuros). De qualquer forma, a complexidade da função objetivo a ser aprendida depende da forma como é representada e a dificuldade da tarefa de aprendizagem também pode variar de acordo com esta forma de representação. Portanto deve-se escolher uma representação que melhor se adapta ao problema específico a ser aprendido. Uma técnica comum em aprendizagem de máquina é mudar a representação dos dados [CRISTIANINI and TAYLOR 2000]:

$$\mathbf{x} = (x_1, ..., x_n) \longmapsto \phi(\mathbf{x}) = (\phi_1(\mathbf{x}), ..., \phi_m(\mathbf{x}))$$

Isso é equivalente a mapear o espaço de entrada $\boldsymbol{X}$ em un novo espaço,

$$\boldsymbol{F} = \{\phi(\mathbf{x}) | \, \mathbf{x} \in \boldsymbol{X}\}$$

chamado de espaço de características.

Por exemplo, um algoritmo de separação linear não consegue aprender uma função não linear como

$$f(x_1, x_2, x_3) = C\frac{x_1 x_2}{x_3^2}.$$

Um simples mapeamento do tipo

$$(x_1, x_2, x_3) \longmapsto (y_1, y_2, y_3) = (\ \ln\ x_1,\ \ln\ x_2,\ \ln\ x_3)$$

resulta na seguinte representação:

$$g(y_1, y_2, y_3) = \ \ln\ f(x_1, x_2, x_3) = \ \ln\ C + \ \ln\ x_1 + \ \ln\ x_2 - 2\ \ln\ x_3 = C + y_1 + y_2 - 2y_3,$$

o que pode ser aprendido por um algoritmo linear. Isso pode simplificar uma tarefa bastante utilizada em aprendizagem de máquina chamada seleção de características.

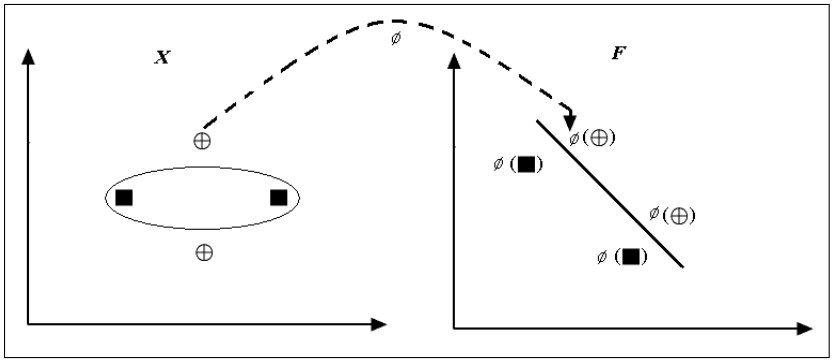


Figura 4.6 – *Exemplo de mapeamento de características de um espaço de entrada bidimensional para um espaço de características bidimensional.*

A Figura 4.6 representa um exemplo do mapeamento de características de um espaço de entrada bidimensional, onde os dados não podem ser separados por uma

função linear no espaço de entrada, mas podem ser no espaço de característica. A Máquina de Suporte Vetorial mapeia os dados para espaços de dimensões muito alta onde a separação linear torna-se fácil.

4.4.2 Mapeamento Implícito

Para que um algoritmo linear aprenda funções não lineares, é necessário selecionar um conjunto de características não lineares e reescreve-las (mapeá-las) em outra representação. Isto equivale a aplicar um mapeamento não linear e fixo dos dados para um espaço de características, no qual um algoritmo linear pode ser usado [CRISTIANINI and TAYLOR 2000]. O problema **T4**

$$\begin{array}{ll} \text{Problema} & \mathbf{T4} \\ \text{Maximize} & \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i \alpha_j y_i y_j \langle \mathbf{x}_i \cdot \mathbf{x}_j \rangle \\ \text{Sujeito a} & \sum_{i=1}^{l} y_i \alpha_i = 0 \\ & 0 \leq \alpha_i \leq C, \text{ para } i = 1, ..., l \end{array}$$

usa o produto interno $\langle \mathbf{x}_i \cdot \mathbf{x}_j \rangle$ dos dados de treinamento para encontrar o hiperplano de separação ótimo. Uma vez que conseguimos fazer o mapeamento do espaço de entrada no espaço de características utilizando a transformação ϕ, então o problema **T4** pode ser transformado no problema **T5**

$$\begin{array}{ll} \text{Problema} & \mathbf{T5} \\ \text{Maximize} & \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i \alpha_j y_i y_j \langle \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j) \rangle \\ \text{Sujeito a} & \sum_{i=1}^{l} y_i \alpha_i = 0 \\ & 0 \leq \alpha_i \leq C, \text{ para } i = 1, ..., l \end{array}$$

onde $\phi : X \longmapsto F$ é um mapeamento não linear do espaço de entrada no espaço de características.

Entretanto, calcular o produto interno $\langle \phi(\mathbf{x}_i) \cdot \phi(\mathbf{x}_j) \rangle$ diretamente no espaço de características pode ser inviável computacionalmente. Por exemplo, supondo que desejamos mapear atributos em $\Re^2$ para o espaço de características formado por todos os produtos possíveis entre os atributos, ou seja:

$$\phi : \Re^2 \rightarrow F = \Re^3$$
$$(x_1, x_2) \longmapsto (x_1^2, x_2^2, x_1 x_2).$$

Com isso, pode-se coletar todas as características de monômios de grau 2 nesse mapeamento não linear. Essa abordagem é adequada para poucos atributos, porém torna-se inviável para problemas de tamanhos reais [SCHÖLKOPF and SMOLA 2002].

Para resolver este problema, utilizamos uma função chamada de *kernel*. Esta

função implicitamente faz o mapeamento no espaço de características e logo após utiliza um algoritmo linear para classificar tal espaço.

4.4.3 Funções de *kernel*

Um *kernel* pode ser representado como sendo uma função K, tal que $\mathbf{x}, \mathbf{z} \in X$

$$K(\mathbf{x}, \mathbf{z}) = \langle \phi(\mathbf{x}) \cdot \phi(\mathbf{z}) \rangle.$$

onde ϕ é um mapeamento de X em um espaço de características produto interno F. Então o problema **T5** pode ser transformado em

Problema **T6**
Maximize $\sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i \alpha_j y_i y_j K(\mathbf{x}_i, \mathbf{x}_j)$
Sujeito a $\sum_{i=1}^{l} y_i \alpha_i = 0$
$0 \leq \alpha_i \leq C,$ para $i = 1, ..., l$

Por exemplo, para um mapeamento

$$\phi : (x_1, x_2) \longmapsto (x_1^2, x_2^2, x_1 x_2, x_2 x_1),$$

o produto interno em F tem a forma de

$$\langle \phi(\mathbf{x}) \cdot \phi(\mathbf{y}) \rangle = (x_1^2 y_1^2, x_2^2 y_2^2, 2 x_1 x_2 y_1 y_2) = \langle \mathbf{x} \cdot \mathbf{y} \rangle^2.$$

Isso significa que a função *kernel* K que substitui o produto interno do mapeamento ϕ é simplesmente o quadrado do produto interno do espaço de entrada. Alguns benefícios diretos podem ser obtidos com sua utilização:

- Não é preciso conhecer diretamente o mapeamento ϕ, pois este é computado implicitamente;
- A dimensão do espaço de características não necessariamente afeta o desempenho computacional;
- As propriedades de representação de *kernel* podem ser utilizadas com diferentes teorias de aprendizagem.

Com isso, todo algoritmo linear que utiliza somente produtos escalares pode implicitamente ser executado em um espaço de características F (de dimensão potencialmente alta), usando *kernels* [CRISTIANINI and TAYLOR 2000].

4.4.4 Exemplos de Funções *kernel*

Anteriormente vimos que as Máquinas de Suporte Vetorial podem aprender a separar dados do tipo linearmente separáveis quanto inseparáveis através de funções

kernel. Quando aplicamos MSV à dados reais, necessitamos encontrar a melhor configuração de parâmetros que melhor generaliza os dados. Basicamente estes parâmetros são:

- C: o peso que o erro exerce na função objetivo e que também limita os multiplicadores de Lagrange;
- *kernel*: formula à superfície de melhor separação do dados;
- *Parâmetros do kernel*: o *kernel* possui parâmetros que influenciam o poder de generalização da superfície, como por exemplo o grau d no *kernel* (polinômio de grau d).

A Tabela (4.1) apresenta alguns tipos de *kernels* mais utilizados em SVM.

Tabela 4.1 – *Exemplo dos kernels mais populares.*

Função *kernel*	**Tipo de Classificador**
$K(\mathbf{x},\mathbf{y}) = (\langle \mathbf{x}\cdot\mathbf{y}\rangle + 1)^d$	Polinômio de Grau d
$K(\mathbf{x},\mathbf{y}) = \exp(-\frac{\|\langle \mathbf{x}-\mathbf{y}\rangle\|^2}{2\sigma^2})$	Gaussiano - RBF
$K(\mathbf{x},\mathbf{y}) = \tanh(\langle \mathbf{x}\cdot\mathbf{y}\rangle) - \theta$	Perceptron Multi Camadas
$K(\mathbf{x},\mathbf{y}) = 1 + \langle \mathbf{x}\cdot\mathbf{y}\rangle + \frac{1}{2}\langle \mathbf{x}\cdot\mathbf{y}\rangle \min(\mathbf{x}*\mathbf{y}) - \frac{1}{6}\min(\mathbf{x}*\mathbf{y})^3$	Splines Linear

4.5 Métodos Multiclasses

Diversos métodos de classificação são muito eficientes quando estão trabalhando apenas com duas classes. Para a resolução de problemas que possuem um grande número de classes, são utilizados métodos de alto nível que reduzem o problema a uma classificação binária [DING and DUBCHAK 2001, SOUTO et al. 2003]. Neste capítulo serão descritos os dois principais métodos aplicados na resolução de problemas com múltiplas classes.

4.5.1 Método Um-Contra-Um

Este método é muito simples e eficiente para a resolução de problems multiclasses. Supondo que existam em nosso problema n classes, para cada par destas n classes é construído um classificador binário. Cada classificador é construído utilizando como conjunto de treinamento apenas exemplos das duas classes envolvidas, obtendo assim um total de $n(n-1)/2$ classificadores

[DING and DUBCHAK 2001]. A Figura 4.7 ilustra esquematicamente o método, onde as formas quadrangulares representam as classes e as ligações entre elas ilustram a disposição em que os classificadores são compostos.

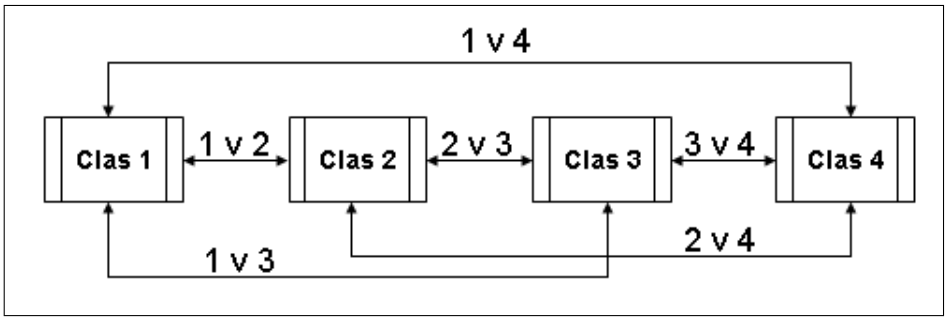


Figura 4.7 – *Representação do método um contra um. Cada ligação entre duas classes representa um classificador binário*

A divisão de um problema multiclasses em múltiplos subproblemas binários apresenta algumas vantagens. Uma delas é a criação de diferentes limites de decisão para cada par de classes, deste modo mesmo se um exemplo for mal classificado em um classificador binário, existe ainda a possibilidade de ser corretamente classificado, pois existem $n-1$ classificadores binários para cada classe.

Uma desvantagem importante deste método, é que se o problema possuir muitas classes, o número de classificadores binários requeridos será da ordem den^2 explodirá. Além disso o elevado número de classificadores torna a classificação lenta pois existe a necessidade de avaliar todos os classificadores antes de uma decisão ser tomada [SANTOS 2002, DING and DUBCHAK 2001].

4.5.2 Método Um-Contra-Todos

Este método é um dos mais utilizados para a classificação multiclasses em MSV. Supondo que existam n classes em nosso problema, o método consiste em particionar estas n classes em dois grupos: Um grupo é formado por uma classe e o outro é formado por todas as outras classes restantes. Um classificador binário é treinado com estes dois grupos, e este procedimento é repetido para cada uma das n classes. Conseqüentemente teremos n classificadores [SANTOS 2002, DING and DUBCHAK 2001], a Figura 4.8 representa este método.

A notável vantagem deste método é a utilização de poucos classificadores, pois necessita de somente n classificadores, o que torna a classificação mais rápida em relação ao método um-contra-um que utiliza $n(n-1)/2$ classificadores binários.

Entretanto, o pequeno número de classificadores utilizados pode não ser suficientes para impor um limite de decisão aceitável entre as classes. Além disso

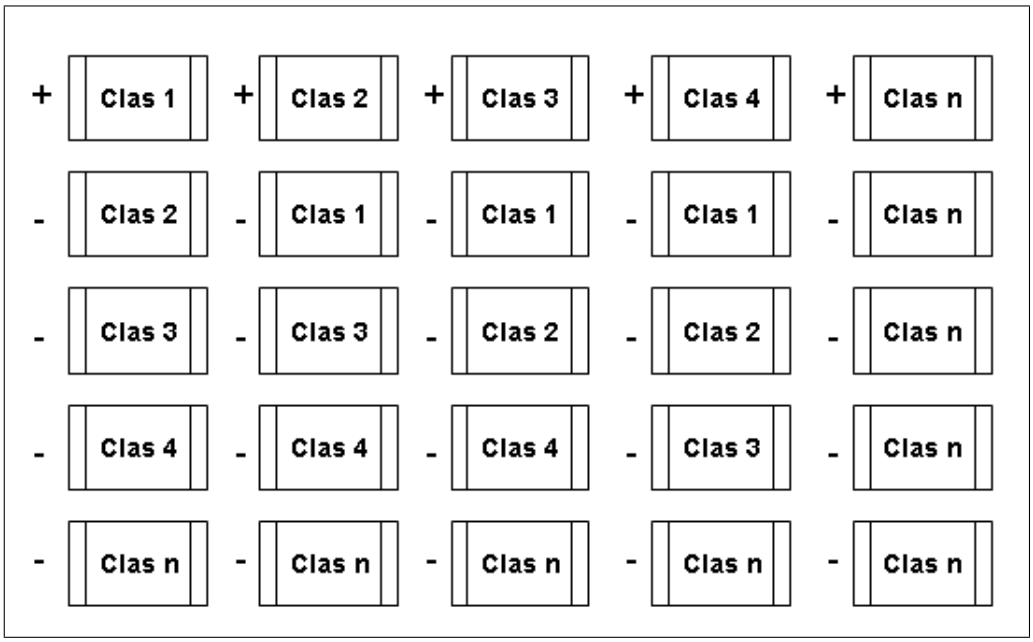


Figura 4.8 – *Representação do método Um Contra Todos.*

neste método, todas as classes são envolvidas em cada classificador, sendo assim treinar estes classificadores pode consumir muito tempo.

Uma outra desvantagem é o desbalancemento que o método causa nos dados de treinamento, esta distribuição desigual entre os exemplos da classe isolada e as classes restantes torna mais difícil a classificação.

Capítulo 5

Revisão Bibliográfica

Este capítulo tem como objetivo apresentar uma revisão bibliográfica sobre a técnica de aprendizado de máquina MSV, a qual é aplicada dentro do contexto de classificação de proteínas, apresentando alguns dos principais trabalhos relacionados.

As seções são formadas da seguinte maneira: primeiramente são descritos alguns trabalhos relacionadas com a predição da estrutura secundária das proteínas. Posteriormente são apresentadas algumas aplicações consideradas bem representativas de MSV aplicadas na resolução de problemas biológicos.

5.1 Técnicas Computacionais Aplicadas na Biologia

Dentro da vasta quantidade de dados produzidos diariamente no campo da biologia molecular, uma grande quantidade refere-se aos aspectos estruturais e funcionais dos genes e das proteínas.

Atualmente as pesquisas em biologia molecular dependem diretamente do auxílio de técnicas computacionais. O volume de dados é grande, tornando a análise manual dos aspectos estruturais e funcionais dos genes cada vez mais impraticável [CAI et al. 2001, SCHLICK 2002, SOUTO et al. 2003].

A determinação de estruturas de proteínas através de métodos empíricos como cristalografia por difração de raio-X e ressonância magnética não é trivial e apresenta caro custo temporal e financeiro. Por este motivo, os métodos computacionais para predição de estruturas protéicas vem sendo rigorosamente estudados e utilizados com eficiência na análise de dados. [WANG et al. 2004, METFESSEL and SAURUGGER 1993, QIAN and SEJNOWSKI 1988].

5.1.1 Predição da Estrutura Secundária de Proteínas

A predição de estruturas protéicas a partir da sua seqüência de aminoácidos, é provavelmente um dos mais importantes problemas em processamento de informação genética, com imensa significação científica e largas aplicações na engenharia genética. Recentemente, tem-se observado com atenção aos métodos para a predição de estruturas terciárias de proteínas, como por exemplo os métodos '*Homology Modeling*'[MAY and BLUNDELL 1994] e '*Inverse Folding*'[WODAK and ROOMAN 1993]. Estes métodos são baseados em pontuação do alinhamento entre as seqüências de teste e as seqüências com estruturas conhecidas, e não apresenta bons resultados para as seqüências que possuem menos de 25% de similaridade com as seqüências de treinamento [SANDER and SCHNEIDER 1991].

O problema clássico da predição de estruturas secundárias de proteínas, consiste em classificar cada resíduo de uma seqüência de aminoácidos em uma das subestruturas recorrentes da conformação tridimensional de uma proteína, que são: hélices α, fitas β ou *coils* [QIAN and SEJNOWSKI 1988, ROST and SANDER 1996, COST and SALZBERG 1993, BARTON 1995].

Existem vários trabalhos que utilizam métodos de aprendizado de máquina para a predição de regiões de α hélices. Alguns destes métodos alcançaram um sucesso moderado, com uma taxa de predição variando entre 70% e 80% dependendo das condições dos dados utilizados na fase de experimentação [MAMITSUKA and YAMANISCHI 1995, MUGGLETON et al. 1993, KNELLER et al. 1990].

A utilização do método de aprendizado de máquina RNA para o tratamento do problema da predição de estruturas de proteínas tem se mostrado bem eficiente, e é utilizado em alguns dos principais classificadores que estão disponíveis atualmente [POLLASTRI et al. 2002, JONSSON et al. 2000, GUIMARÃES and MELO 2003].

Um exemplo clássico desta técnica de IA aplicada na resolução deste problema, é o trabalho proposto por Qian e Sejnowisk [QIAN and SEJNOWSKI 1988], no qual é apresentado um estudo preliminar para a escolha da melhor configuração de uma rede neural para predição de estrutura secundária de proteínas.

As redes utilizadas por Quian e Sejnowski são MLPs com uma camada escondida. Foram realizados experimentos variando os tamanhos de janela e o número de nós na camada escondida, sendo o número de saída sempre igual a três: um para Hélice α, um para Fitas β e outro para *coil* [GUIMARÃES and MELO 2003].

A Figura 5.1 representa um diagrama de arquitetura de rede representando, a camada de entrada, a camada oculta e a saída, utilizado no trabalho de Qian e Sejnowski.

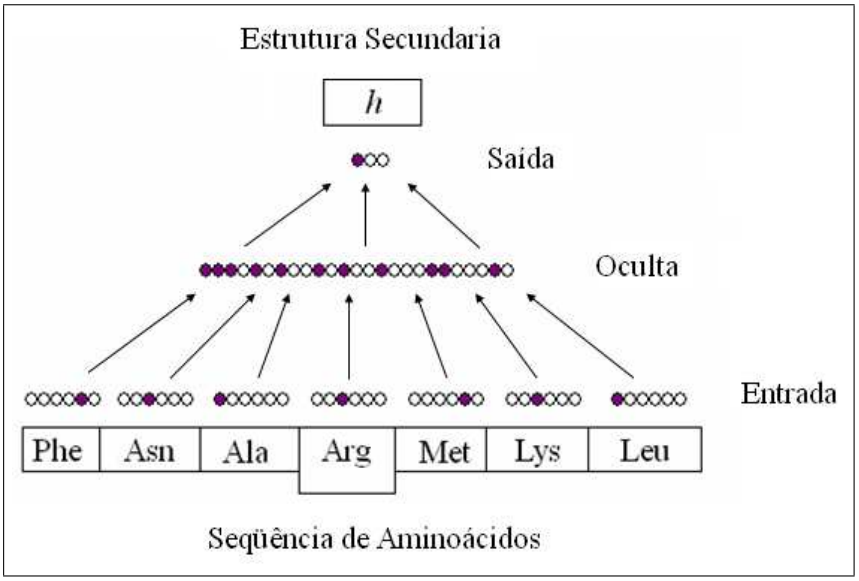


Figura 5.1 – *Diagrama de arquitetura de rede utilizada por Qian e Sejnowski.*

A rede original utilizada no trabalho de Quian e Sejnowski possui 13 grupos de entrada, com 21 unidades por grupo, representando uma extensão de 13 aminoácidos contíguos. Na Figura 5.1 são ilustrados somente 7 grupos de entrada com 7 unidades por grupo. As informações da camada de entrada foram transformadas na camada intermediária para produzir as 3 saídas, representando a predição da estrutura secundária para o aminoácido central [QIAN and SEJNOWSKI 1988].

As redes foram treinadas com o tradicional algoritmo *Backpropagation* utilizando para isso um subconjunto de um banco de dados com 106 seqüências de proteínas globulares não similares. Para os testes foram utilizados dois subconjuntos diferentes do que foi empregado no treinamento. Um destes subconjuntos era formado por 15 seqüências e outro por 6. A performance da rede é avaliada através de uma medida de desempenho, bastante conhecida denominada de Q_3 [SCHULZ and SCHIRMER 1979]. Esta medida fornece a porcentagem dos resultados classificados corretamente. Seu cálculo pode ser observado na equação 5.1, onde $NRCC$ representa o número de resíduos classificados corretamente eNTR o número total de resíduos.

$$Q_3 = \frac{NRCC}{NTR} \times 100 \tag{5.1}$$

Neste experimento, o melhor desempenho Q_3, obtido pelas redes de Quian e Sejnowski foi de $64,3\%$, utilizando uma janela de tamanho 13, e 40 nós na camada

escondida. Este resultado foi superior a qualquer outro obtido através de métodos estatísticos, mais utilizados na época.

A partir do sucesso obtido por esta abordagem, diversos outros trabalhos foram propostos experimentando diferentes arquiteturas, algoritmos e tratamento de entrada de dados [BRUNAK and BALDI 2001, ROST 2001]. Um avanço significativo nas predições foi obtido através da introdução de mais informação biológica para as redes, mais especificamente informações evolucionárias, por meio do uso de perfis de seqüências de proteínas como dados de entrada [GUIMARÃES and MELO 2003].

Um outro aspecto importante a ser considerado pelos classificadores, é o tamanho dos dados de entrada. Através da utilização de perfis, cerca de 20 nós são necessários na camada de entrada para a codificação de cada aminoácido de uma janela. Uma abordagem interessante para este problema foi proposta por Riis e Krogh [RIIS and KROGH 1996].

Para o treinamento e teste das redes, Riis e Krogh utilizaram o banco de dados RS126, desenvolvido por Rost e Sander [ROST and SANDER 1994]. O RS126 consiste em 126 proteínas globulares não homólogas, obtidas do HSSP [SANDER and SCHNEIDER 1991]. Todas as proteínas selecionadas para compor este banco de dados possuem um percentual máximo de 25% de similaridade entre os pares, para seqüências maiores que 80 resíduos.

O modelo de validação utilizado foi o *cross-validation*. Este método foi aplicado da seguinte maneira: um conjunto de seqüências é dividido em sete partes de tamanhos aproximadamente iguais, seis partes são utilizadas para treinamento e uma para teste. Este processo é repetido ciclicamente até que todas as partes tenham sido testadas [GUIMARÃES and MELO 2003].

A codificação local dos aminoácidos é feita aplicando um caso particular do método *Weight Sharing*, onde as redes escolhem a melhor representação para os aminoácidos, os quais são inicialmente codificados ortogonalmente [GUIMARÃES and MELO 2003].

Após a codificação, janelas de 15 aminoácidos são apresentadas como dados de entrada para as redes que foram desenvolvidas para predizer, separadamente, cada uma das três classes (Hélice, *Strand*, *Coil*). Estas redes apresentam apenas uma saída, onde a decisão é baseada em um limite de 0.5. Se a saída for maior que este limite então a entrada correspondente é classificada como sendo da estrutura em consideração. Para a predição de cada estrutura secundária, é utilizado uma combinação de cinco destas redes, cada uma contendo diferentes números de nós na camada escondida. O número de pesos ajustáveis de todas as redes juntas após o emprego da codificação é inferior a 600. As saídas de cada grupo de cinco redes

são combinadas por uma outra rede estrutura a estrutura cuja predição do resíduo central é escolhida como sendo a maior das três saídas que sao normalizadas com a função *Softmax* [BRIDLE 1990, DUDA et al. 2001].

Os resultados obtidos recentemente com a utilização de RNAs tem alcançado taxas de acerto superiores a 73%, confirmando a eficácia do método [PETERSEM et al. 2000].

Outro método de aprendizado de máquina que vem sendo aplicado na predição de estrutura de proteínas são as Máquinas de Suporte Vetorial (MSV). Esta técnica tem alcançado resultados comparáveis aos obtidos pelas RNAs. Em [GUO et al. 2004] as MSV são aplicadas na predição de estruturas secundárias de proteínas.

5.2 Aplicações de Máquinas de Suporte Vetorial na Biologia

Máquina de Suporte Vetorial pode ser descrito como sendo um algoritmo de aprendizado supervisionado, muito útil para o reconhecimento de padrões sutis em bases de dados complexas [PAVLIDIS et al. 2004]. O algoritmo aprende com uma base de dados de treinamento, e com isso é capaz de classificar dados desconhecidos. Esta nova tecnologia de aprendizado de máquina tem obtido grande sucesso na resolução de problemas biológicos. Esta seção apresenta alguns dos principais trabalhos que utilizam esta técnica.

Esta nova tecnologia, vem sendo aplicada com sucesso na resolução de problemas biológicos. As MSVs tem apresentado uma performance superior as técnicas tradicionais em diversas situações, tais como: predição de características estruturais e funcionais de proteínas, categorização automática de expressão gênica de microarrays de DNA, reconhecimento de padrões de estruturas celulares, seqüências de DNA etc. A seguir são descritas algumas aplicações de MSV nos problemas citados.

O trabalho de Ding e Dubchak [DING and DUBCHAK 2001], utiliza Máquinas de Suporte Vetorial e Redes Neurais Artificiais para a classificação de proteínas segundo suas características estruturais. Os dados utilizados para os experimentos, foram obtidos da base de dados SCOP.

Os autores utilizaram as estruturas secundárias como estruturas primitivas, e as proteínas foram classificaram em *folds* SCOP. A base de dados SCOP possui mais de 800 *folds* de proteínas classificados manualmente em uma organização hierárquica baseado na compreensão de suas estruturas.

Apesar de existir este grande número de *folds*, apenas os 27 mais populosos

foram analisados nos experimentos de Ding e Dubchak, sendo 6 pertencentes a classe $all-\alpha$, 9 a classe $all-\beta$, 9 a classe α/β e 3 a classe $\alpha+\beta$.

Na classificação dos dados, Ding e Dubchak utilizaram métodos multi-classe bem conhecidos como o Um-Contra-Todos. Além disso, propuseram um novo método que foi denominado Um-Único-Versus-Outros.

Na comparação das RNA com as MSV, observou-se que as RNA apresentaram um número elevado de falsos positivos. Em conseqüência disto, os classificadores que utilizaram esta técnica obtiveram um desempenho inferior. Outro importante aspecto apresentado neste trabalho foi a eficiência em termos computacionais, enquanto as RNA utilizavam um longo tempo para convergir, as MSV mostraram rapidez na apresentação dos resultados. Desta forma muitos experimentos foram realizados somente com os classificadores que utilizavam MSV.

Zaki e coautores [ZAKI et al. 2005] apresentam um trabalho similar ao proposto por Ding e Dubchak, onde utilizam o método de aproximação de kernel juntamente com MSV para a classificação de proteínas. Este trabalho apresenta resultados estruturais de 3 familias selecionadas da base de dados SCOP.

Isik e coautores [ISIK et al. 2004] aplicam Máquinas de Suporte Vetorial para predição da classe estrutural de uma proteínas. As MSV São aplicadas a uma base de dados extraídas do banco de dados de proteína (PDB). As proteínas estão classificadas em uma das seguintes classes estruturais: $all-\alpha, all-\beta, \alpha/\beta$ e $\alpha+\beta$.

Neste trabalho foram utilizados dois diferentes tipos de características como vetores de entrada da MSV, denominados AAC e Trio AAC. Para o AAC, os vetores de entrada da MSV foram compostos pelas porcentagens de cada aminoácido da proteína. Neste caso os vetores de suporte possuem dimensão 20 pelo fato de haver 20 diferentes tipos de aminoácidos envolvidos na composição de uma proteína. No Trio AAC, a seqüência de aminoácidos é dividida em grupos de três, e os vetores de suporte são compostos pela porcentagem em que cada aminoácido ocorre dentro destes grupos.

Para diminuir a dimensionalidade do Trio AAC, foram utilizados diversos *clusters* como entrada para os vetores de suporte. Estes *clusters* de aminoácidos são compostos de acordo com a hidrofobicidade e informações de carga dos aminoácidos dadas por Thomas e Dill [THOMAS and DILL 1996].

Para ambos os testes foi utilizado uma base de dados composta por 117 proteínas de treinamento (29 α, 30 β, 29 α/β, 29 $\alpha+\beta$) e 63 proteínas de teste (8 α, 22 β, 9 α/β, 24 $\alpha+\beta$)[CHOU 1995]. Para aplicação da MSV foi utilizado o software LIBSVM [CHANG and LIN 2001].

A taxa de classificação do AAC foi de $74,3\%$ de acerto enquanto que o Trio AAC obteve $84,6\%$. Para o experimento em questão o classificador Trio AAC obteve

melhor desempenho.

Outro trabalho que aplica Máquinas de Suporte Vetorial para a predição de características estruturais de proteínas, é o proposto por Cai [CAI et al. 2001]. As MSV são aplicadas a um grupo de dados extraídos da base de dados SCOP. Nesta base de dados, as proteínas são classificadas em uma das seguintes classes estruturais: $all-\alpha, all-\beta, \alpha/\beta, \alpha\beta$.

Este trabalho utilizou duas séries de dados: Uma com 277 proteínas distribuídas em 70 α, 61 β, 81 α/β, 65 $\alpha+\beta$. A outra série de dados conta com 498 exemplos proteínas sendo 107 α, 126 β, 136 α/β, 129 $\alpha+\beta$. Para o treinamento e classificação das MSVs foi utilizado o *software* SVM^{light}[JOACHIMS et al. 1999].

Em nosso trabalho, utilizamos a mesma base de dados selecionada por Ding e Dubchack [DING and DUBCHAK 2001]. Foram analisamos somente os 27 principais *folds* dos 971 existentes na hierarquia SCOP atualmente.

Com base nesta pesquisa, propomos uma nova metodologia para o tratamento dos problemas envolvendo a classificação automática de estruturas terciárias de proteínas.

Capítulo 6

Metodologia

Neste capítulo é descrita a metodologia utilizada para a predição da estrutura terciária de proteínas a partir das suas características estruturais. Basicamente, pode ser apresentada da seguinte forma:

- **Definição das Ferramentas Utilizadas:** Especificação de quais ferramentas foram utilizadas para o tratamento e armazenamento dos dados, treinamento das MSVs, e análise dos resultados.
- **Definição dos Atributos Utilizados:** Determinação de quais atributos referentes aos dados serão utilizados para composição da base de treinamento.
- **Seleção dos Dados:** Seleção dos dados desejados, dentro das diversas bases de dados existentes.
- **Composição da Base de Dados de Treinamento:** Normalização dos dados brutos selecionados para a composição da base de treinamento.
- **Treinamento dos Classificadores:** Treinamento dos classificadores com os dados normalizados da base de treinamento.
- **Medidas de Desempenho:** Medidas utilizadas para avaliação dos resultados. Neste trabalho será utilizado o método de validação *Leave-One-Out*.
- **Reconstrução dos Vetores de Suporte:** A partir das medidas de avaliação, reconstruir os vetores de suporte.

A Figura 6.1 representa uma visão global da metodologia. A seguir são expostas, de forma detalhada, cada uma das etapas propostas.

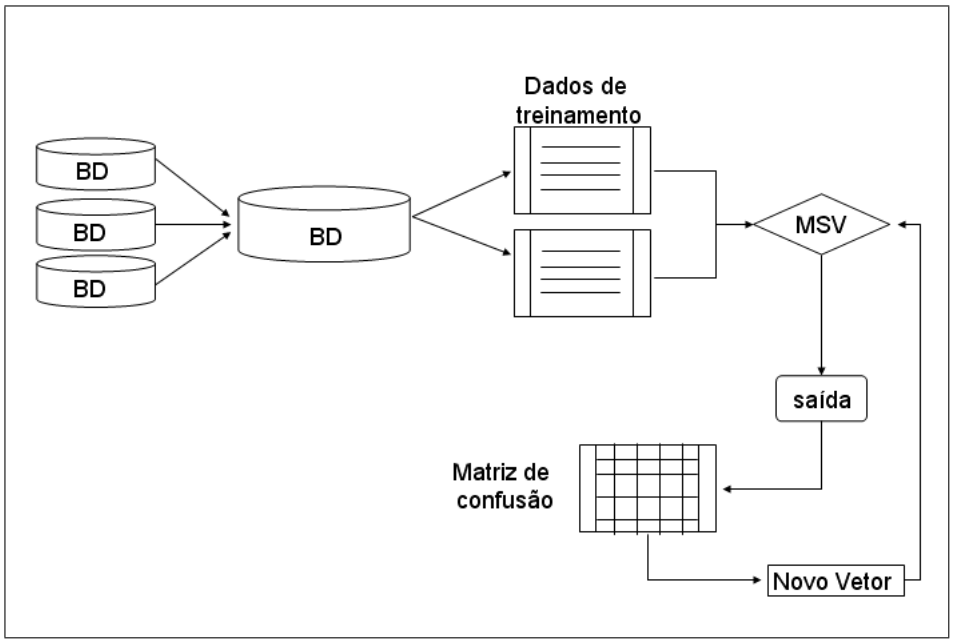


Figura 6.1 – *Representação Global da Metodologia.*

6.1 Definição das Ferramentas Utilizadas

Uma das tarefas de fundamental importância para o bom andamento de qualquer trabalho, é a correta definição das ferramentas utilizadas para o seu desenvolvimento. Atualmente, existem várias ferramentas disponíveis com uma série de funcionalidades aplicadas no tratamento de classificação de dados.

Algumas destas ferramentas são utilizadas no desenvolvimento deste trabalho. Elas foram selecionadas por possuírem funcionalidades que facilitam a resolução do problema proposto.

Inicialmente foi necessário definir os dados a serem analisados. Estes dados foram adquiridos de alguns bancos de dados públicos, tais como CATH, PDB e SCOP. Estes dados foram pré-processados e armazenados em uma nova base de dados padronizada.

Para a realização do pré-processamento, é feito o uso da linguagem AWK do *Linux*. Ela é utilizada por ser de fácil compreensão e possuir funções específicas para a manipulação deste tipo de dados. Além disso, AWK é um *software* livre e pode ser encontrada em qualquer distribuição *Linux*.

Para predição da estrutura secundária das proteínas, foi utilizado o *software* Jnet. Este *software* possui distribuição livre e é bastante simples de ser compreendido, além disso, as estruturas secundárias por ele preditas são muito próximas das estruturas reais.

Um exemplo disso pode ser observado na Figura 6.2, onde é feita a comparação

entre as estruturas real e predita da proteína 1fcda1. Podemos notar a semelhança estrutural quanto a distribuição das α hélices e folhas β existente entre as duas estruturas.

Esta semelhança torna viável a utilização da estrutura secundária predita, facilitando a automatização deste processo.

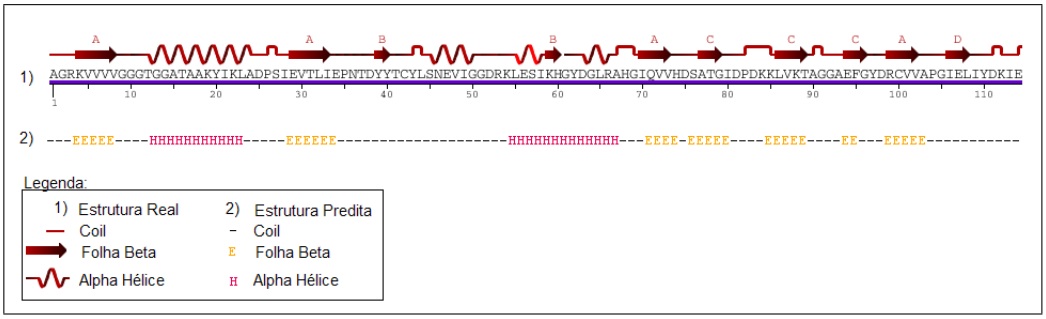


Figura 6.2 – *Comparação entre as estruturas real e predita da proteína 1fcda1.*

Após a predição da estrutura secundária, estas seqüências necessitam ser alinhadas de forma automática para a obtenção dos atributos desejados. Para o alinhamento destas estruturas, foi utilizado o *software* de alinhamento de seqüências *ClustalW*. Além de possuir farta bibliografia, ele também é um dos programas mais respeitados na área para este fim. *ClustalW* também possui distribuição livre.

A Figura 6.3 apresenta um exemplo de alinhamento de duas estruturas feito pelo *software ClustalW*. Como entrada, fornecemos os dados a serem alinhados, e na saída são apresentados os resultados deste alinhamento.

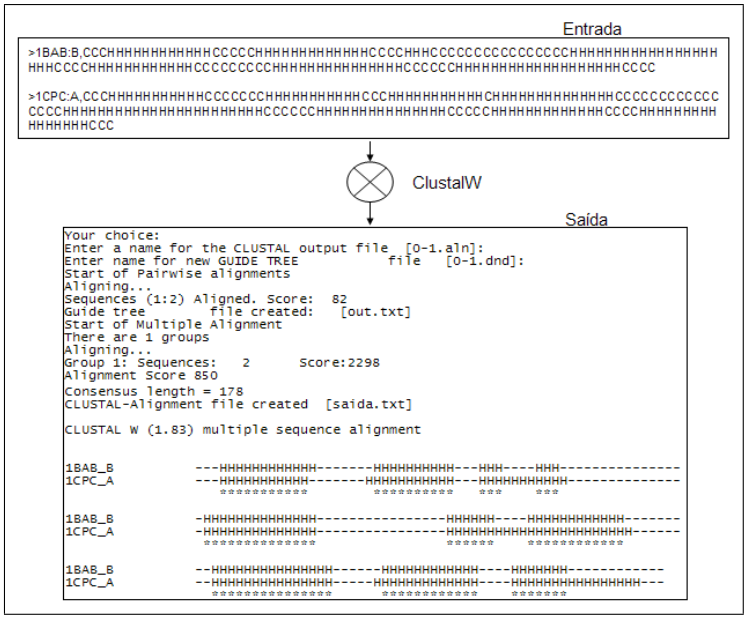


Figura 6.3 – *Exemplo de alinhamento de duas estruturas utilizando o software ClustalW.*

Os dados obtidos com as etapas de pré-processamento, predição e alinhamento foram armazenados em tabelas, para isso foi utilizado o SGBD (Sistema Gerenciador de Banco de Dados) *PostgreSQL*. Este SGBD também é um *software* livre e possui uma série de manuais explicando seu funcionamento, além disso, permite o uso da linguagem SQL que facilita a manipulação dos dados armazenados.

O último passo é a determinação da ferramenta utilizada para a manipulação das MSV. Em nosso trabalho, utilizamos duas ferramentas bastante referenciadas na bibliografia. Uma delas é o *software* SVM^{light} [VAPNIK 1998], por possuir flexibilidade no tratamento de problemas com múltiplas classes e permitir o uso de diferentes métodos multiclasse.

A outra é a biblioteca *LibSVM*, desenvolvida por [CHANG and LIN 2001]. Esta biblioteca tem a vantagem de possuir integração com outros *softwares* como a ferramenta estatística R-*project*. Esta ferramenta é muito utilizada em nossos experimentos para a manipulação, treinamento e predição dos dados selecionados.

6.2 Descrição do Conjunto de Dados

Os dados manipulados neste trabalho são compostos por um conjunto de proteínas encontradas em alguns bancos de dados disponíveis publicamente, dentre eles os mais utilizados são o CATH, PDB e SCOP.

Cada banco possui uma forma diferente de armazenamento e disposição de seus dados, como por exemplo, na base de dados CATH a seqüência de aminoácidos da proteína encontra-se separada da sua referência, tamanho e *fold*. Já a base de dados PDB armazena estas informações em tabelas separadas onde a seqüência de aminoácidos, o tamanho da proteína e os dados de estrutura secundária real são obtidos através da pesquisa pela referência da proteína.

Com isso, é necessário aplicar um pré-processamento nestes dados a fim de armazená-los de acordo com o padrão desejado para os experimentos. Este padrão pode ser observado na Figura 6.4.

```
>1DVH   72      ADGAALYKSCIGCHGADGSKAAMGSAKPVKGQGAEELYKKMKGYADGSYGGERKAMMTNAVKKYSDEELKALADYMSKL 3
```

Figura 6.4 – *Exemplo do padrão de armazenamento dos dados.*

A primeira coluna armazena a referência PDB da proteína antecedia do sinal »". Este sinal é exigido pelo *software* de predição de estrutura secundária *Jnet*. A segunda, terceira e quarta colunas referem-se respectivamente ao tamanho da seqüência, composição dos aminoácidos da proteína e o *fold* a qual pertence.

6.2.1 Definição dos Atributos Utilizados

Diversos atributos podem ser considerados para o conjunto de proteínas, como por exemplo: a composição dos aminoácidos, a estrutura secundária predita, a hidrofobicidade, o raio de Van der Waals, a polaridade, entre outros. Algumas bases de dados, como CATH (*Protein Structure Classification*) e PDB (*Protein Data Bank*), disponibilizam dados deste tipo.

Em nossos experimentos, inicialmente consideramos os dados referentes a composição dos aminoácidos de cada proteína, posteriormente utilizamos os dados da estrutura secundária predita para a composição dos vetores de suporte. Além disso, também necessitamos saber a qual *fold* cada proteína pertence.

Para obter-se estas informações, normalmente utiliza-se a base de dados SCOP (*Structural Classification of Proteins*), onde cada proteína é classificada de acordo com um *fold* baseado nas informações evolucionárias de outras estruturas conhecidas.

6.2.2 Seleção dos Dados

Primeiramente é necessário determinar qual será o conjunto de proteínas a ser analisado. Após tomada esta decisão, é criada uma nova base de dados contendo o nome das proteínas desejadas, o tamanho, a seqüência de aminoácidos que as compõem, a estrutura secundária predita e o *fold* específico ao qual pertencem. Uma visão geral deste processo pode ser observada na Figura 6.5.

```
>1BAB:B   30  GLSAAQRQVIAATWKDIAGADNGAGVGKKCLIKFLSAHPQMAAVFGF   CCCHHHHHHHHHHHCCCCCHHHHHHHHHHHHCCCCHHHCCCCCCC  1
>1CPC:A   30  VLSEGEWQLVLHVWAKVEADVAGHGQDILIRLFKSHPETLEKFDRFK  CCCHHHHHHHHHHHCCCCCCCHHHHHHHHHHHCCCHHHHHHHHHHC  1
>1CPC:B   30  SLSAAEADLAGKSWAPVFANKNANGLDFLVALFEKFPDSANFFADFK  CCCHHHHHHHHHHHCCCCCCCCHHHHHHHHHHHHHHHHHHHHHHHHC  1
```

Figura 6.5 – *Visão geral do processo de armazenamento dos dados selecionados.*

Após a coleta, os dados são submetidos a um pré-processamento onde são obtidas algumas informações, as quais serão utilizadas para a composição dos vetores de suporte.

Como primeiro experimento, utilizamos os dados de estrutura primária referentes a composição dos aminoácidos de cada proteína para a construção dos classificadores. Neste caso os dados são submetidos a um pré-processamento afim de se obter o cálculo da porcentagem com que cada aminoácido ocorre na proteína.

Estes dados são utilizados na composição dos vetores de suporte do conjunto de dados de treinamento. A Figura 6.6 apresenta uma visão geral do processo de automatização do cálculo da porcentagem dos aminoácidos para cada proteína selecionada da base de dados padronizada.

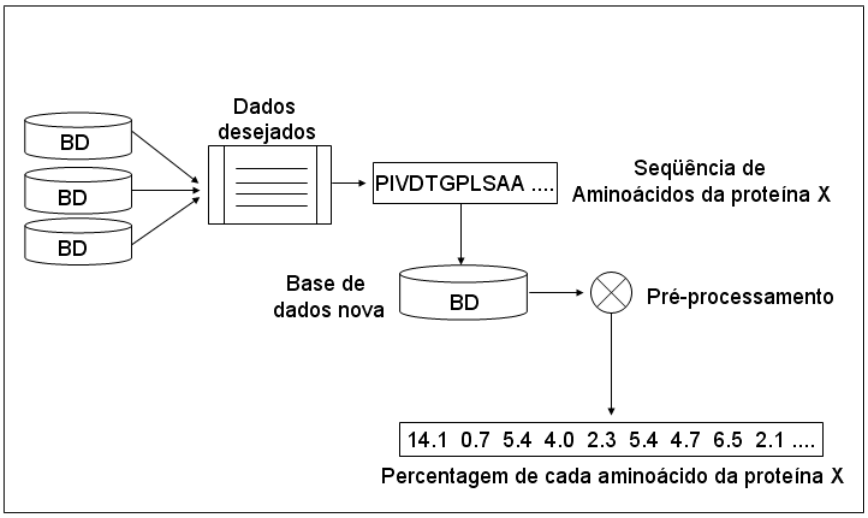


Figura 6.6 – *Visão geral do processo de seleção e tratamento dos dados.*

O segundo experimento envolve a utilização dos dados referentes a estrutura secundária das proteínas selecionadas para composição dos classificadores.

Pelo fato de estarmos utilizando dados de estrutura secundária predita, é necessário submeter os dados de estrutura primária como entrada para um *software* de predição de estrutura obtendo como saída as estruturas secundárias preditas. Além disso, é necessário realizar o alinhamento destas estruturas afim de compor uma nova base de dados contendo informações como: o *score* de alinhamento, o *score* bruto, o tamanho das estruturas alinhadas e o *fold* a qual pertencem. Com base nestas informações são construídos os vetores de suporte. A Figura6.7 ilustra o funcionamento deste processo.

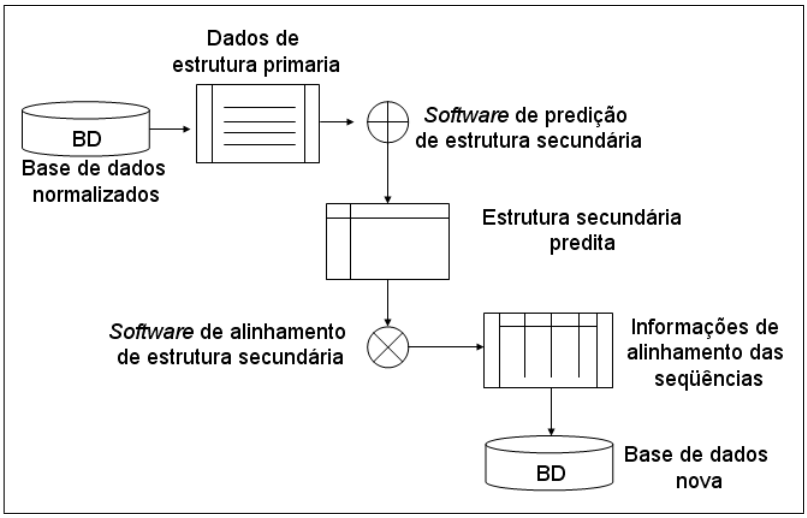


Figura 6.7 – *Visão geral do processo de tratamento dos dados de estrutura secundária.*

6.2.3 Conjunto de Dados de Treinamento

O conjunto de treinamento é dividido em duas bases dados. A primeira é chamada de Base1, e é composta por um conjunto de 296 proteínas classificadas em 27 diferentes *folds*.

A segunda, denominada Base2, contém 665 exemplos de proteínas também agrupadas nos 27 *folds*. Uma descrição detalhada deste conjunto de proteínas e seus respectivos *folds* pode ser observada na Tabela 6.1.

No caso de seqüências muito parecidas, a predição das MSV são influenciadas aumentando o valor do desempenho da classificação. Afim de evitar este problema, as duas bases de dados utilizadas contêm somente exemplos onde duas proteínas não possuam mais de 35% de similaridade na seqüência.

Neste conjunto de dados, não existem interações entre as classes, ou seja, uma mesma proteína não pode pertencer a mais de um *fold*.

6.3 Implementação dos Classificadores

Esta seção está dividida em três partes. Inicialmente é descrita uma metodologia para tratar problemas com múltiplas classes. A seguir é apresentada a metodologia utilizada no treinamento das MSV. Por fim são apresentadas as medidas de desempenho aplicadas neste estudo.

Para a implementação dos classificadores, são consideradas duas etapas:

- **Composição dos vetores de suporte dos classificadores:** Nesta etapa é feita a composição dos vetores de suporte para o primeiro e segundo experimento.

  Para o primeiro experimento, é criado um método para automatização do processo de composição dos vetores de suporte com a porcentagem dos aminoácidos de cada proteína.

  No segundo experimento, é determinada uma metodologia para a composição dos vetores de suporte utilizando dados referentes a estrutura secundária predita das proteínas selecionadas. Nesta etapa são criados alguns processos de automatização para a predição e alinhamento da estrutura secundária das proteínas bem como a composição dos vetores de suporte com os dados obtidos deste alinhamento.

- **Treinamento dos classificadores:** Nesta etapa, os classificadores obtidos anteriormente são treinados, e é definido o tipo de *kernel* que apresenta melhor desempenho para a distribuição dos dados analisados. Para os nossos

experimentos, testamos três tipos de *kernels*, o polinomial, o Gaussiano-RBF e o Linear. Nos casos testados, o Linear obteve melhor desempenho. Além disso, nesta etapa também definimos o método multiclasse a ser utilizado.

Neste trabalho abordamos dois métodos, o Um-Contra-Um e o Um-Contra-Todos. Em nossos testes, o método Um-Contra-Um obteve uma taxa de 23 % de acerto, contra 17 % do outro método, sendo este o mais eficiente para a classificação do tipo de dados proposto.

6.3.1 Abordagens para Múltiplas Classes

O problema proposto consiste em analisar 27 classes. Visto que as Máquinas de Suporte Vetorial se apresentam como classificadores essencialmente binários, é necessária a definição de métodos para o tratamento de problemas com múltiplas classes. Duas abordagens distintas são aplicadas neste problema:

A primeira propõe a utilização do método Um-Contra-Todos. Neste método, n classificadores são construídos e treinados independentemente. Cada classificador representa uma das classes evolutivas, desta forma a classe em questão é denominada classe positiva (+1) e todas as outras classes formam a classe negativa (-1). Os dados devem ser rotulados novamente para cada classificador.

A segunda abordagem propõe a utilização do método Um-Contra-Um, onde são construídos $n(n-1)/2$ classificadores que são treinados independentemente. Neste método, um classificador é construído para cada par de classes possíveis. Os classificadores são construídos utilizando apenas os dados das duas classes envolvidas. Novamente existe a necessidade de rotular os dados de forma específica para cada classificador.

6.3.2 Treinamento e Validação dos Classificadores

O treinamento e validação dos classificadores do problema em questão é realizado em duas etapas.

A primeira, consiste em utilizar duas bases de dados para treinamento, a Base1 e a Base2. O conjunto de dados Base2 foi composto com a intenção de aumentar a base de dados de treinamento tornando o sistema mais genérico. A distribuição dos dados pode ser observada na Tabela 6.1.

A segunda etapa consiste em utilizar o método de validação *Leave-One-Out*. Neste método utiliza-se $(m-1)$ exemplos de uma amostra de tamanho m para o treinamento do classificador, que é testado no único exemplo remanescente. Este processo é repetido m vezes, cada vez deixando de considerar um único exemplo. O erro é estimado através da soma total dos erros em cada teste, dividido por$m$.

Tabela 6.1 – *Dados de treinamento.*

Fold	**Seqüência**	**Índice**	**Base1**	**Base2**
α				
Globin-like	1	1	12	18
Cytocrome c	2	3	7	16
DNA-binding 3-helical bundle	3	4	12	32
4-helical cytokines	4	7	7	15
4-helical up-and-down bundle	5	9	8	17
Alpha; EF-hand	6	11	7	16
β				
Immunoglobulin-like beta-sandwich	7	20	30	73
Cupredoxins	8	23	9	20
Viral coat and capsid proteins	9	26	5	18
ConA-like lectins/glucanases	10	30	7	13
SH3-like barrel	11	31	8	16
OB-*fold*	12	32	12	32
Trefoil	13	33	9	12
Trypsin-like serine proteases	14	35	9	13
Lipocalins	15	39	8	15
α/β				
(TIM)-barrel	16	46	30	71
FAD (also NAD)-binding motif	17	47	11	23
Flavodoxin-like	18	48	10	23
NAD(p)-binding Rossmann-*fold*	19	51	13	39
P-loop containing nucleotide	20	54	10	22
Thioredoxin-like	21	57	9	16
Ribonuclease H-like motif	22	59	10	22
Hydrolases	23	62	11	18
Periplasmic binding protein-like	24	69	11	15
$\alpha + \beta$				
Beta-Grasp	25	72	7	14
Ferrodoxin-like	26	87	12	37
Small inhibitors, toxins, lectins	27	110	13	39

6.3.3 Medidas de Desempenho

A medida de desempenho utilizada foi o padrão Q [BRUNAK and BALDI 2001], e pode ser representada pela seguinte equação:

$$Q = \sum_{i=1}^{k} w_i Q_i. \tag{6.1}$$

A variável Q representa a medida de precisão total do modelo, e é calculada pelo somatório das variáveis w_i multiplicadas por Q_i, onde $w_i = (TC_i + TE_i)/T$, e $Q_i = TC_i/(TC_i + TE_i)$. A variável TC_i é a quantidade de exemplos classificados corretamente por classe, TE_i é a quantidade de exemplos classificados errados por classe, e T é o total de amostras da validação.

A apresentação destes resultados é feita através de uma matriz de confusão. A matriz de confusão de um classificador c oferece uma medida efetiva do modelo mostrando o número de classificações corretas *versus* o resultado do classificador para cada classe, sobre um conjunto de exemplos T.

Desta maneira, os resultados classificados podem ser observados em duas dimensões: classes verdadeiras e classes resultantes, para cada k classes distintas.

Tomamos como exemplo uma classificação binária, com as classes rotuladas de (+1) e (-1). Neste exemplo, os dois erros possíveis são denominados *falso positivo* F_P e *falso negativo* F_N. A Tabela 6.2 apresenta a matriz de confusão para este problema, onde T_P é o número de exemplos positivos classificados corretamente e T_N é o número de exemplos negativos classificados corretamente do total de $T = (T_P + T_N + F_P + F_N)$ exemplos.

Tabela 6.2 – *Matriz de confusão para um classificador binário.*

classe	Preditos como (+1)	Preditos como (-1)
+1	Verdadeiros Positivos (T_P)	Falsos Negativos (F_N)
-1	Falsos Positivos (F_P)	Verdadeiros Negativos (T_N)

Como o problema em questão possui múltiplas classes, a matriz de confusão é construída da seguinte forma: as proteínas selecionadas para a composição dos vetores de suporte são rotuladas de acordo com o seu *fold*. Esta informação é passada para a MSV como sendo a última posição de cada vetor de suporte da base de dados de treinamento.

A Figura 6.8 demonstra os vetores de suporte para três proteínas cujos *folds* estão na última posição do vetor. Estes vetores de suporte representam os valores dos atributos utilizados e variam o seu tamanho de acordo com os experimentos propostos.

```
5.59 6.99 11.19 1.40 1.40 6.99  6.29  3.50 4.20 2.10 3.50 0.70 8.39 4.20 4.90 2.80 0.70 1
4.40 4.40 3.14  4.40 6.29 10.69 3.77  3.14 1.89 6.92 5.03 7.55 5.03 6.92 1.26 0.63 1.89 1
1.41 6.34 9.15  2.82 4.23 7.75  10.56 2.11 2.82 4.23 4.23 0.70 4.23 4.93 7.04 1.41 3.52 1
```

Figura 6.8 – *Exemplo de Vetores de Suporte.*

Após a composição destes vetores, eles são utilizados como entrada da MSV para o treinamento do modelo. O modelo é treinado utilizando o método *Leave-One-Out* no qual cada proteína é retirada do conjunto de dados de treinamento e testada para cada uma das classes. O resultado do teste irá compor uma matriz de confusão na qual cada proteína testada individualmente gerará um ponto para a classe predita, sendo assim o total de pontos igual ao tamanho da amostra.

Esta matriz de confusão será de dimensão 27 linhas por 27 colunas, onde as linhas representam as 27 classes SCOP conhecidas e as colunas representam a classe apontada pelo simulador.

Para a composição desta matriz, utiliza-se o seguinte critério: toma-se o primeiro vetor de suporte e analisa-se a qual classe SCOP ele pertence. Tendo esta informação é necessário saber qual classe foi predita pelo simulador, então esta classe receberá um ponto. Este processo é repetido m vezes, onde m é o tamanho total da amostra.

A partir destes dados é composta a matriz de confusão, como pode ser observado na Figura 6.9.

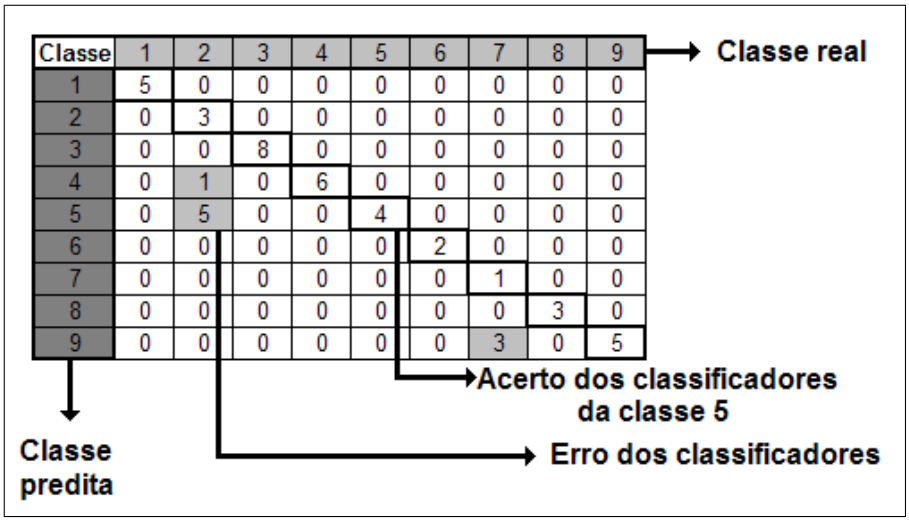


Figura 6.9 – *Representação genérica de uma matriz de confusão para problemas multiclasse.*

O processo de treinamento, classificação e composição da matriz de confusão é abordado de forma diferente para cada conjunto de dados analisados. Neste trabalho abordamos duas características, a primeira refere-se a estrutura primária

das proteínas selecionadas, e a segunda abordagem trata da análise dos dados de alinhamento de estruturas secundárias preditas a partir das estruturas primárias das proteínas selecionadas.

Para os dados referentes a estrutura primária, o processo ocorre da seguinte maneira: primeiramente é necessário pré-processar os dados para obter a porcentagem de cada aminoácido da proteína, logo após, uma base de dados é composta com estes dados e submetida a MSV para o treinamento do modelo.

O conjunto de "teste" é formado por uma única proteína distinta do grupo de treinamento, e esta é predita pela MSV baseando-se no modelo treinado.

Uma visão geral de todo este processo pode ser observada na Figura 6.10, onde a saída é composta pelos valores de predição de cada proteína dentro das classes analisadas e a partir destas informações compõe-se a matriz de confusão.

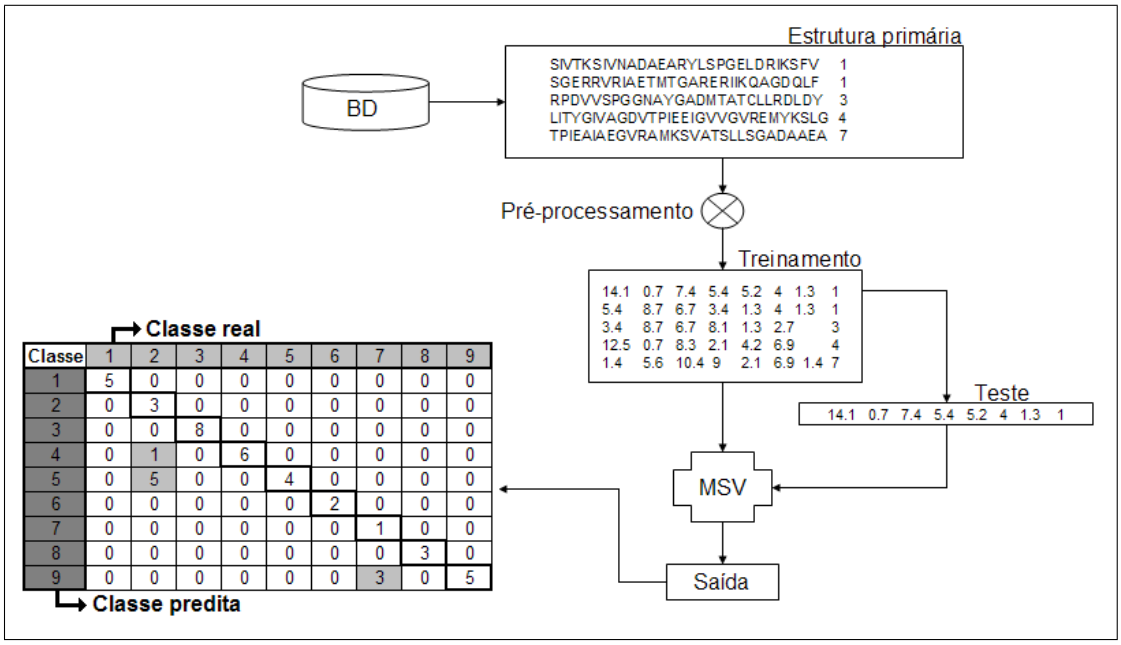


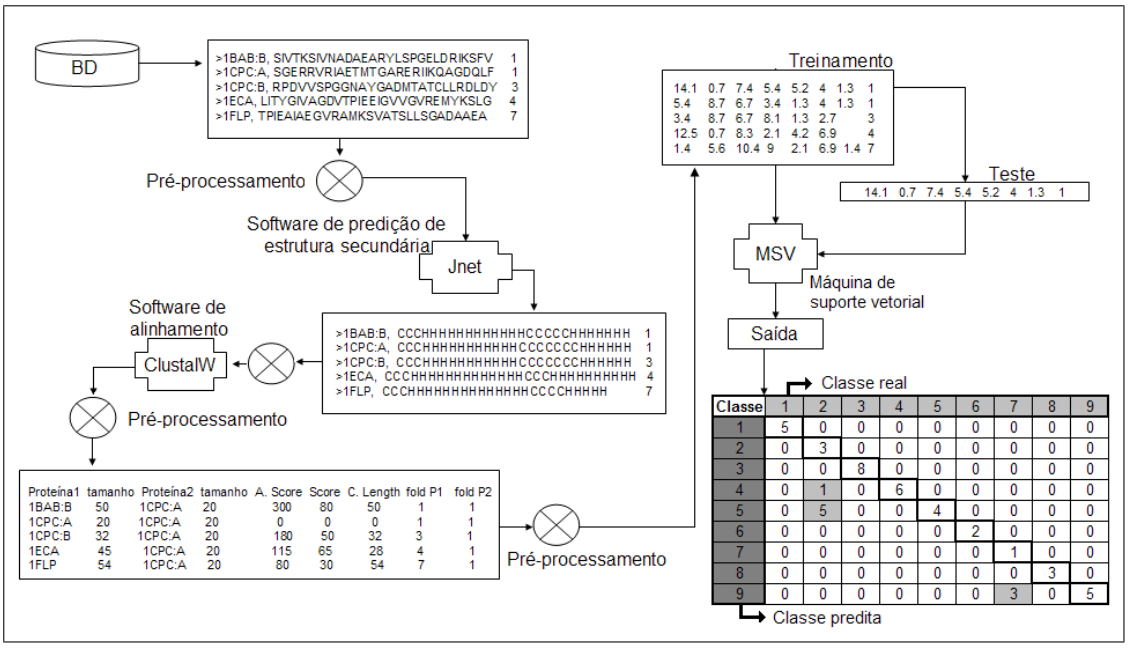
Figura 6.10 – *Visão geral do processo de construção da matriz de confusão para a base de dados de estrutura primária.*

A segunda abordagem sugere a utilização de informações extraídas do alinhamento das estruturas secundárias preditas das proteínas selecionadas, sendo que este processo é mais complexo que o primeiro e envolve uma série de etapas.

A primeira etapa consiste em submeter os dados de estrutura primária para um *software* de predição de estrutura secundária. Logo após, estas estruturas preditas são alinhadas de forma que cada estrutura seja alinhada com todas as outras da amostra, e os resultados deste alinhamento são guardados em uma nova base de dados.

A partir desta base de dados são analisadas quais informações irão compor os vetores de suporte. Estes vetores serão utilizados no treinamento do modelo e predição das estruturas individuais retiradas da base de dados pelo método *Leave-One-Out*. Por fim, baseando-se nestas informações é construída uma nova matriz de confusão.

Uma visão geral deste processo pode ser observada na Figura 6.11, a qual demonstra em detalhes o tratamento aplicado aos dados, desde a seleção das proteínas até a composição do modelo de treinamento e exibição dos resultados através da matriz de confusão.



Classe	1	2	3	4	5	6	7	8	9
1	5	0	0	0	0	0	0	0	0
2	0	3	0	0	0	0	0	0	0
3	0	0	8	0	0	0	0	0	0
4	0	1	0	6	0	0	0	0	0
5	0	5	0	0	4	0	0	0	0
6	0	0	0	0	0	2	0	0	0
7	0	0	0	0	0	0	1	0	0
8	0	0	0	0	0	0	0	3	0
9	0	0	0	0	0	0	3	0	5

Figura 6.11 – *Visão geral do processo de construção da matriz de confusão para a base de dados de estrutura secundária.*

6.3.4 Reconstrução dos Vetores de Suporte

Após a construção da matriz de confusão é feita a análise dos resultados obtidos. Geralmente, apenas com dados de estrutura primária como composição dos aminoácidos, a taxa de erro entre as classes é grande. No intuito de diminuir esta taxa, os vetores de suporte são reconstruídos combinando dados obtidos do alinhamento das estrutura secundárias preditas das proteínas selecionadas. Com isso, a confusão entre as classes será menor e conseqüentemente haverá um aumento da precisão do modelo.

Capítulo 7

Resultados

Este capítulo tem como objetivo apresentar os experimentos realizados no decorrer deste trabalho e os resultados com eles obtidos. Basicamente, pode ser dividido da seguinte forma:

- **Experimentos Preliminares:** Definição de experimentos preliminares com o objetivo de conhecer o comportamento das MSV aplicadas na classificação automática de proteínas.
- **Especificação dos parâmetros utilizados para as MSV:** Determinação dos parâmetros da MSV a serem utilizados no treinamento dos dados.
- **Especificação do conjunto de atributos:** Determinação dos atributos utilizados na composição dos vetores de suporte do conjunto de treinamento.
- **Resultados individuais dos atributos utilizados:** Apresentação dos resultados obtidos com o treinamento dos dados referentes a cada um dos atributos considerados.
- **Análise dos resultados obtidos:** Avaliação e quantificação da taxa de acerto para os resultados de cada atributo analisado.

Uma visão geral destas etapas pode ser visualizada na Figura 7.1, a qual demonstra graficamente todo o processo de tomada de decisão e apresentação dos resultados.

O primeiro item refere-se a análise preliminar das MSV. Este passo é de fundamental importância, pois com ele adquirimos o conhecimento necessário para decidir quais parâmetros ajustar e o tipo de atributo a ser analisado. O segundo trata da definição dos parâmetros das MSV que melhor se adaptam aos dados selecionados. O terceiro, quarto e quinto itens representam respectivamente a definição dos atributos, apresentação e análise dos resultados.

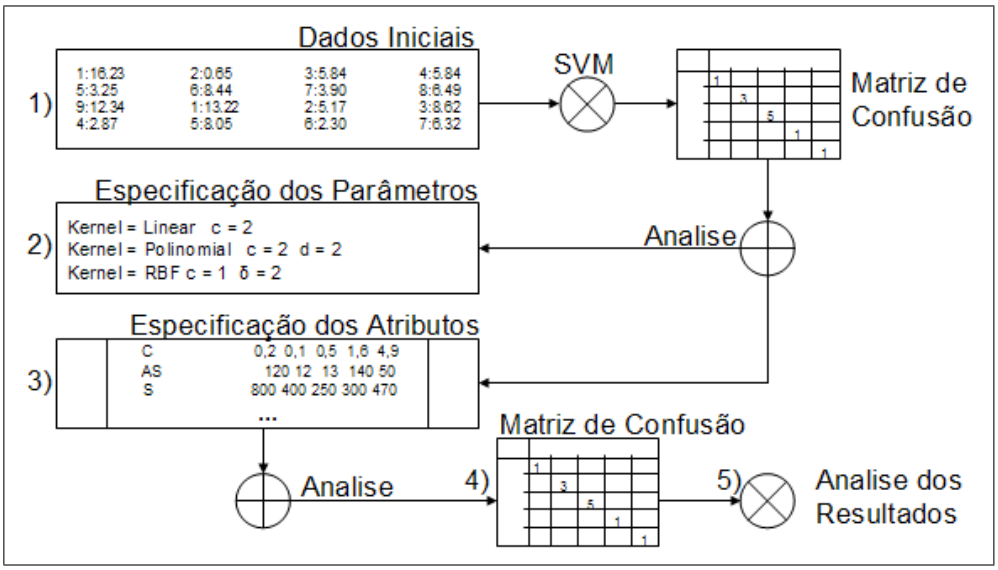


Figura 7.1 – *Visão geral do processo de geração e apresentação dos resultados.*

7.1 Experimentos Preliminares

Os experimentos preliminares têm o objetivo de avaliar o comportamento das MSV quando aplicadas na classificação de proteínas segundo a hierarquia SCOP.

Para a realização destes experimentos, criou-se um conjunto de dados de proteínas obtidas da base de dados PDB. O único atributo utilizado para compor os vetores de suporte foi a composição dos aminoácidos, onde é calculada a porcentagem que cada um dos 20 aminoácidos ocorre dentro da cadeia polipeptídica das proteínas em questão.

A partir deste conjunto de treinamento, aplicou-se o método Um-Contra-Todos descrito anteriormente na seção 4.5.2. Na aplicação deste método, foram criados 27 novos conjuntos de dados onde cada um deles é representado dentro de uma das classes da Tabela 6.1.

Para cada conjunto de dados, foi atribuído o rótulo (+1) para a classe em questão e (-1) para o restante das classes, como pode ser observado na Figura 7.2. Neste estágio inicial de treinamento, foram utilizados os parâmetros *Default* oferecidos pelo *software* SVM^{Light}.

```
+1 1:14.8 2:1.2 3:4.9 4:3.7 5:3.7 6:8 7:0.6 8:4.9 9:4.9 10:8 11:1.2 12:4.3 13:3.7 14:3.1 15:3.7 16:9.9 17:7.4
+1 1:18 2:1.7 3:7.6 4:2.9 5:2.3 6:8.1 7:0 8:5.2 9:3.5 10:8.7 11:3.5 12:4.1 13:1.7 14:2.9 15:5.8 16:8.1 17:4.7
-1 1:14.1 2:1.3 3:5.1 4:14.1 5:2.6 6:5.1 7:2.6 8:9 9:7.7 10:7.7 11:1.3 12:2.6 13:2.6 14:5.1 15:9 16:2.6 17:2.6
-1 1:12.7 2:0 3:2.7 4:16.4 5:0.9 6:3.6 7:0.9 8:1.8 9:9.1 10:16.4 11:0.9 12:0.9 13:8.2 14:5.5 15:10.9 16:0 17:0.9
```

Figura 7.2 – *Exemplo do conjunto de dados de treinamento do software* SVM^{Light}.

Os resultados são descritos através de uma matriz de confusão para cada classificador.

Com a análise dos resultados, concluímos que somente o atributo composição dos aminoácidos não o suficientemente capaz de separar as classes de forma correta.

O maior índice de confusão ocorreu entre os *folds* 46 e 47. Neste caso ambos são classificados dentro da hierarquia SCOP como pertencendo a mesma classe estrutural α/β. Para tratar este problema, propõe-se a análise da estrutura secundária das seqüências em questão, e através dela, a composição de novos vetores capazes de separar estas informações.

Como exemplo tomamos as estruturas secundárias das proteínas "1fcda1" e "1fcda2" do *fold* 46 e "3lada1" e "3lada2" do *fold* 47, ambas preditas pelo sistema *Jpred* (Figura 7.3 e 7.4).

Figura 7.3 – *Exemplo de estrutura secundária do fold 46.*

Figura 7.4 – *Exemplo de estrutura secundária do fold 47.*

Analisando estas estruturas, percebeu-se que ambas possuem grande semelhança quanto a disposição de suas hélices α e folhas β, porém uma das seqüências referentes ao *fold* 46 possui somente 2 hélices α enquanto as demais apresentam 3. Isto nos leva a crer que acrescentando estas informações como novos atributos aos vetores de suporte, o erro de classificação será diminuído.

7.1.1 Especificação dos Parâmetros Utilizados para as MSV

Uma característica bastante relatada na literatura de MSV é a pouca quantidade de parâmetros livres informados pelo usuário para a realização da tarefa de treinamento dos dados.

Tomamos como exemplo os tipos de produtos internos *kernels*, utilizados pelos *softwares* SVM^{light} e *LibSVM*. No caso do *kernel* linear, o único parâmetro a ser informado pelo usuário é a penalização C, já no *kernel* polinomial é necessário definir também o grau d do polinômio. Para o *kernel* radial (RBF), além do parâmetro de penalização C também é necessário definir o parâmetro σ.

Antes de serem iniciados os treinamentos, foram realizados alguns testes a fim de se estimar os valores mais indicados para as variáveis dos produtos internos *kernel*. Para o parâmetro de penalização C, foram testados valores de 1 a 1000 no intuito de se estimar o valor ótimo dos parâmetros para o tipo da amostra analisada [JOACHIMS et al. 1999].

As MSV foram treinadas para cada atributo utilizando os valores dos parâmetros apresentados na Tabela 7.1. Uma vez testados, estes valores foram assumidos como sendo os mais adequados para o tipo de distribuição dos dados analisados.

Tabela 7.1 – *Tabela dos parâmetros de penalização e dos produtos internos kernel.*

kernel	**C**	**Parâmetro Interno**
Linear	2	-
Polinomial	2	d = 2
RBF	1	$\sigma = 1$

7.2 Especificação do Conjunto de Atributos

Conforme descrito na Seção (6.2.2), este trabalho considerou atributos referentes a duas características estruturais das proteínas. A primeira delas foi a estrutura primária, onde o atributo analisado foi a composição dos aminoácidos (C). Para a construção da base de dados de treinamento com atributo (C), foi calculado a porcentagem em que cada aminoácido ocorre dentro da seqüência. A Figura (7.5) ilustra a determinação deste atributo.

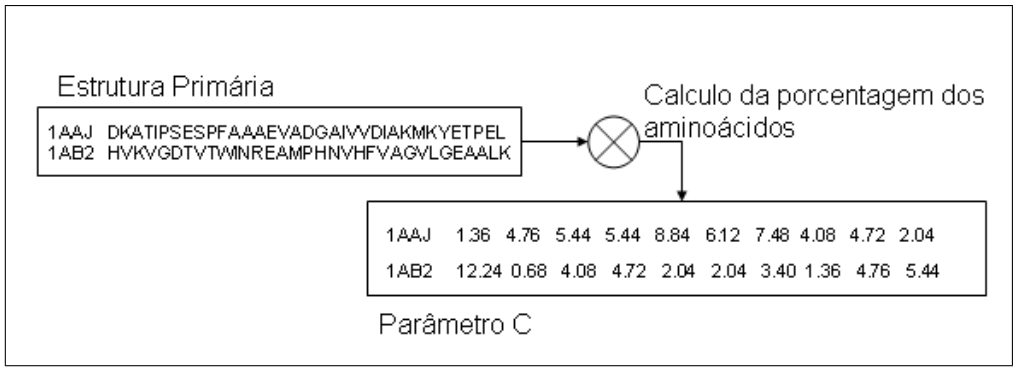


Figura 7.5 – *Visão geral do processo de determinação do atributo C.*

A segunda característica estudada foi a estrutura secundária predita. As estruturas preditas foram alinhadas duas a duas onde cada proteína foi alinhada com todas as outras da amostra.

A saída do *software* de alinhamento de seqüências contém diversas informações que se apresentam misturadas com os atributos desejados. Para que possamos trabalhar com estes atributos de forma automática, foi feito um processamento nos dados de alinhamento, selecionando somente os dados por nós julgados relevantes de serem analisados. Estes dados foram armazenados em uma nova base de dados, a qual permite fácil acesso e manipulação dos mesmos.

Um exemplo do alinhamento de duas proteínas pode ser observado na Figura 7.6. Ela demonstra graficamente o processo de mineração e tratamento dos atributos desejados a partir dos dados de alinhamento brutos. Neste exemplo, podemos observar a disposição dos dados no arquivo de saída do *software* de alinhamento. O cabeçalho apresenta as estruturas a serem alinhadas e seus respectivos tamanhos, o restante dos dados desejados encontram-se misturados com informações não relevantes para o nosso estudo. A extração destes dados é feita de forma automática, e este processo possui um alto custo computacional e temporal.

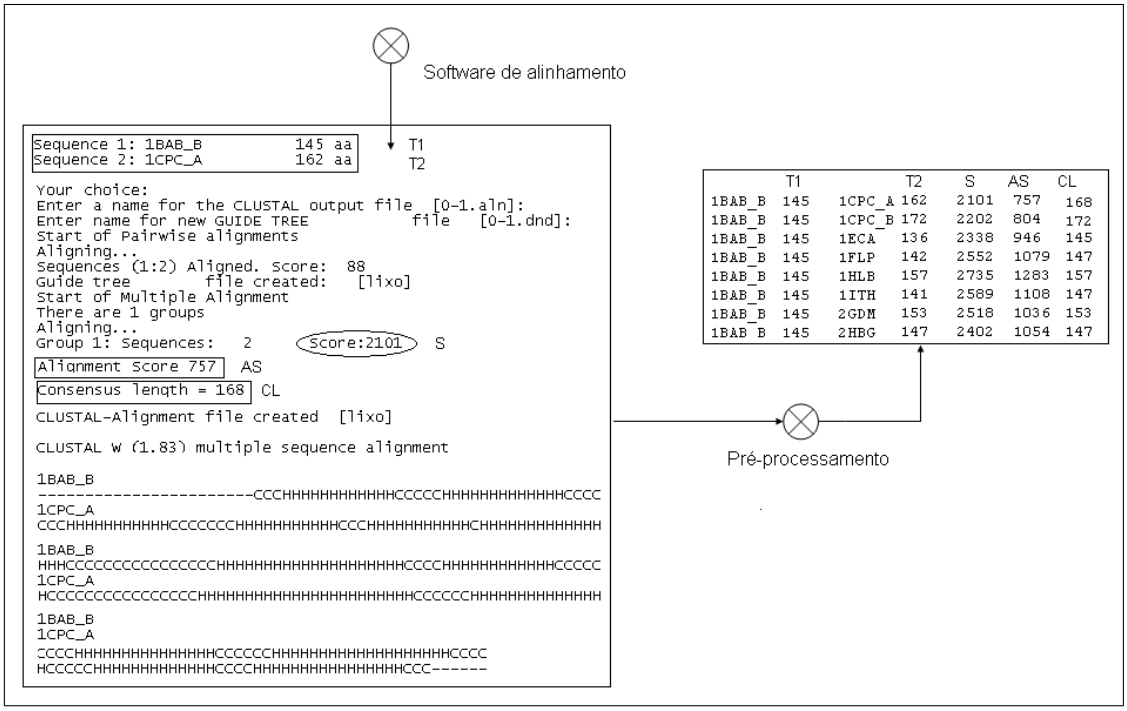


Figura 7.6 – *Visão geral do processo de determinação dos atributos do alinhamento das seqüências.*

Uma descrição completa dos nomes e significados de cada atributo utilizado para o treinamento do modelo proposto pode ser visualizada na Tabela 7.2.

Estes atributos foram utilizados para compor os vetores de suporte da base de dados de treinamento. Primeiramente foram analisados individualmente, logo após foram combinados entre si afim de aumentar o tamanho dos vetores de suporte, procurando-se elevar a precisão do modelo.

Tabela 7.2 – *Tabela dos atributos analisados.*

Estrutura	**Nome do atributo**	**Atributo**
Primária	Composição dos aminoácidos	C
Secundária	*Score* de alinhamento	AS
	Score bruto	S
	Tamanho das seqüências alinhadas	CL
	Tamanho da primeira seqüência alinhada	T1
	Tamanho da segunda seqüência alinhada	T2

A Tabela 7.4 descreve os atributos utilizados para o treinamento do modelo. A primeira coluna refere-se as combinações dos atributos treinados e como são determinados para compor os vetores de suporte. Na segunda coluna é apresentada a descrição de cada experimento.

Tabela 7.3 – *Tabela dos atributos utilizados no treinamento do modelo.*

Atributos	**Descrição**
C	Composição dos aminoácidos
Max (AS)	Máximo valor de AS por classe
Max (S)	Máximo valor de S por classe
Média (AS)	Média do AS por classe
Média (S)	Média do S por classe
Max [AS/(T1+T2)]	Máximo valor de AS dividido por (T1+T2) por classe
Max [S/(T1+T2)]	Máximo valor de S dividido por (T1+T2) por classe
Max [AS/(CL)]	Máximo valor de AS dividido pelo CL correspondente por classe
Max [S/(CL)]	Máximo valor de S dividido pelo CL correspondente por classe
Max (AS) | Max (S)	Máximo valor de AS combinado com o máximo valor de S
C | Max (AS)	C combinado com o máximo valor de AS

No primeiro experimento realizado foram treinados somente os dados referentes ao atributo C. Este atributo é obtido de forma automática, e é calculado a partir da estrutura primária das proteínas em questão.

O conjunto de dados de treinamento referentes ao atributo C possui vetores na ordem de 20 elementos. Isso se dá pelo fato de existirem 20 aminoácidos envolvidos na composição da estrutura primária da proteína. Neste caso, as posições do vetor representam o percentual de cada aminoácido da proteína.

No segundo experimento foram utilizados atributos referentes ao alinhamento da estrutura secundária predita das proteínas selecionadas. Estes atributos foram analisados individualmente e depois combinados a fim de aumentar a precisão do

modelo treinado.

Os vetores de suporte compostos pelos parâmetros referentes aos dados de alinhamento possuem cerca de 27 elementos. Cada elemento do vetor corresponde ao valor do alinhamento das estruturas pertencentes a uma determinada classe, portanto o tamanho dos vetores de suporte para os atributos referentes a estrutura secundária são proporcionais ao número de classes trabalhadas.

O atributo *score* de alinhamento (AS) é calculado através de uma matriz denominada *Blosum 45*. Para cada aminoácido alinhado corretamente é incrementado ao AS o valor determinado pela matriz. Quando duas estruturas parecidas são alinhadas, o AS recebe um valor alto, o qual indica uma possível classe em comum. Para estruturas pertencentes a classes distintas o valor do AS diminui, pois o *software* de alinhamento necessita inserir os chamados *Gaps*, os quais penalizam negativamente o valor do AS.

Em contrapartida ao AS, o atributo *score* bruto (S) é calculado levando em consideração somente os dados alinhados incrementando o valor de S caso haja um alinhamento perfeito.

Os atributos T1 e T2 referem-se respectivamente ao tamanho da primeira seqüência alinhada e ao tamanho da segunda seqüência alinhada. O atributo CL representa a quantidade de estruturas perfeitamente alinhadas, e é calculado simplesmente pela soma destes alinhamentos.

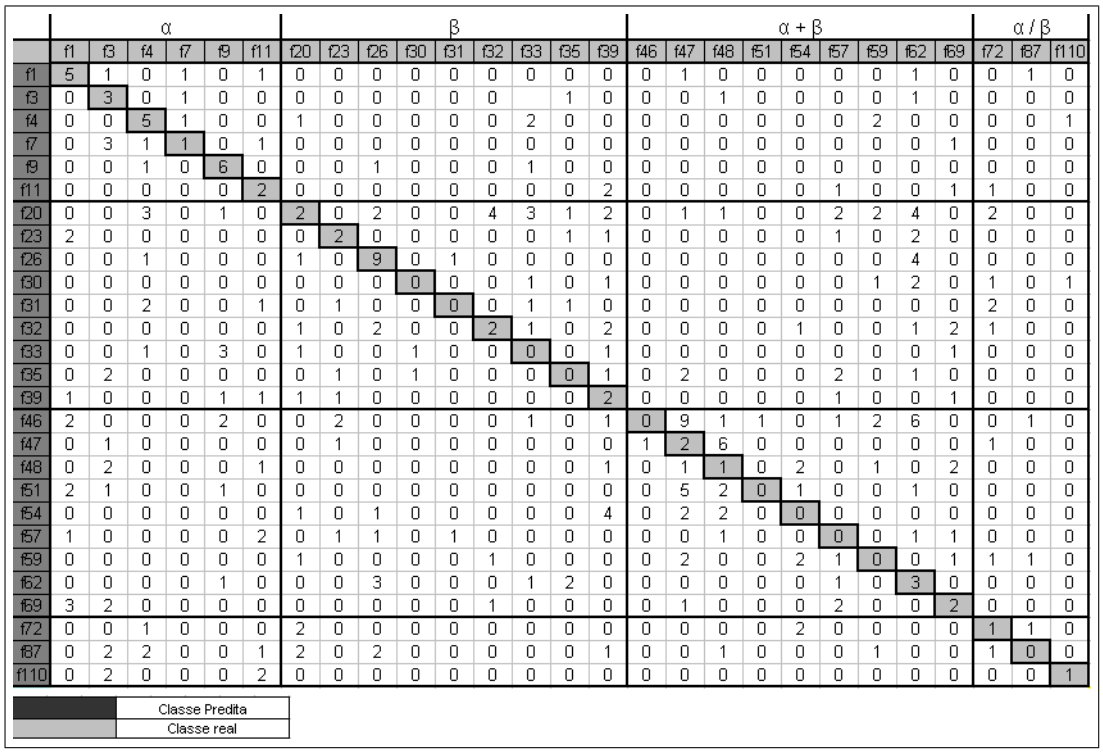
7.2.1 Resultados Individuais dos Atributos Utilizados

A primeira etapa do treinamento dos dados envolve a decisão do método multiclasse a ser utilizado. Para a determinação deste método, foram testados dados referentes a composição dos aminoácidos configurados nos dois métodos multiclasse propostos.

Inicialmente foram analisados os resultados obtidos do treinamento do atributo C para o método Um-Conta-Um. Estes resultados são apresentados através de uma matriz de confusão, a qual utiliza um esquema de pontuação baseado no método de validação *Leave-One-Out*.

A base de dados é treinada retirando a cada passo um dos vetores, o qual é predito baseado no modelo treinado. Este vetor predito irá gerar um ponto para a classe a qual foi atribuído. A matriz de confusão é composta pela soma destes pontos para cada classe, e isso proporciona uma visão geral da quantidade de classes corretas e mal classificadas.

Os resultados apresentados com o método Um-Contra-Um para o atributo C podem ser observados na matriz de confusão apresentada pela Figura 7.7.



	α						β									α + β									α / β		
	f1	f3	f4	f7	f9	f11	f20	f23	f26	f30	f31	f32	f33	f35	f39	f46	f47	f48	f51	f54	f57	f59	f62	f69	f72	f87	f110
f1	5	1	0	1	0	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	1	0
f3	0	3	0	1	0	0	0	0	0	0	0	0		1	0	0	0	1	0	0	0	0	1	0	0	0	0
f4	0	0	5	1	0	0	1	0	0	0	0	0	2	0	0	0	0	0	0	0	0	2	0	0	0	0	1
f7	0	3	1	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
f9	0	0	1	0	6	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
f11	0	0	0	0	0	2	0	0	0	0	0	0	0	0	2	0	0	0	0	0	1	0	0	1	1	0	0
f20	0	0	3	0	1	0	2	0	2	0	0	4	3	1	2	0	1	1	0	0	2	2	4	0	2	0	0
f23	2	0	0	0	0	0	0	2	0	0	0	0	0	1	1	0	0	0	0	0	1	0	2	0	0	0	0
f26	0	0	1	0	0	0	1	0	9	0	1	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0
f30	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	1	2	0	1	0	1
f31	0	0	2	0	0	1	0	1	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	2	0	0
f32	0	0	0	0	0	0	1	0	2	0	0	2	1	0	2	0	0	0	0	1	0	0	1	2	1	0	0
f33	0	0	1	0	3	0	1	0	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0
f35	0	2	0	0	0	0	0	1	0	1	0	0	0	0	1	0	2	0	0	0	2	0	1	0	0	0	0
f39	1	0	0	0	1	1	1	1	0	0	0	0	0	0	2	0	0	0	0	0	1	0	0	1	0	0	0
f46	2	0	0	0	2	0	0	2	0	0	0	0	1	0	1	0	9	1	1	0	1	2	6	0	0	1	0
f47	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	1	2	6	0	0	0	0	0	0	1	0	0
f48	0	2	0	0	0	1	0	0	0	0	0	0	0	0	1	0	1	1	0	2	0	1	0	2	0	0	0
f51	2	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	5	2	0	1	0	0	1	0	0	0	0
f54	0	0	0	0	0	0	1	0	1	0	0	0	0	0	4	0	2	2	0	0	0	0	0	0	0	0	0
f57	1	0	0	0	0	2	0	1	1	0	1	0	0	0	0	0	0	1	0	0	0	0	1	1	0	0	0
f59	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	0	2	0	0	2	1	0	0	1	1	1	0
f62	0	0	0	0	1	0	0	0	3	0	0	0	1	2	0	0	0	0	0	0	1	0	3	0	0	0	0
f69	3	2	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	0	0	0	2	0	0	2	0	0	0
f72	0	0	1	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	1	1	0
f87	0	2	2	0	0	1	2	0	2	0	0	0	0	0	1	0	0	1	0	0	0	1	0	0	1	0	0
f110	0	2	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1

Classe Predita
Classe real

Figura 7.7 – *Resultado do treinamento do atributo C com o método Um-Contra-Um.*

A diagonal principal da matriz de confusão apresenta o número de exemplos corretamente classificados. O restante dela apresenta o número de exemplos mal classificados para cada classe.

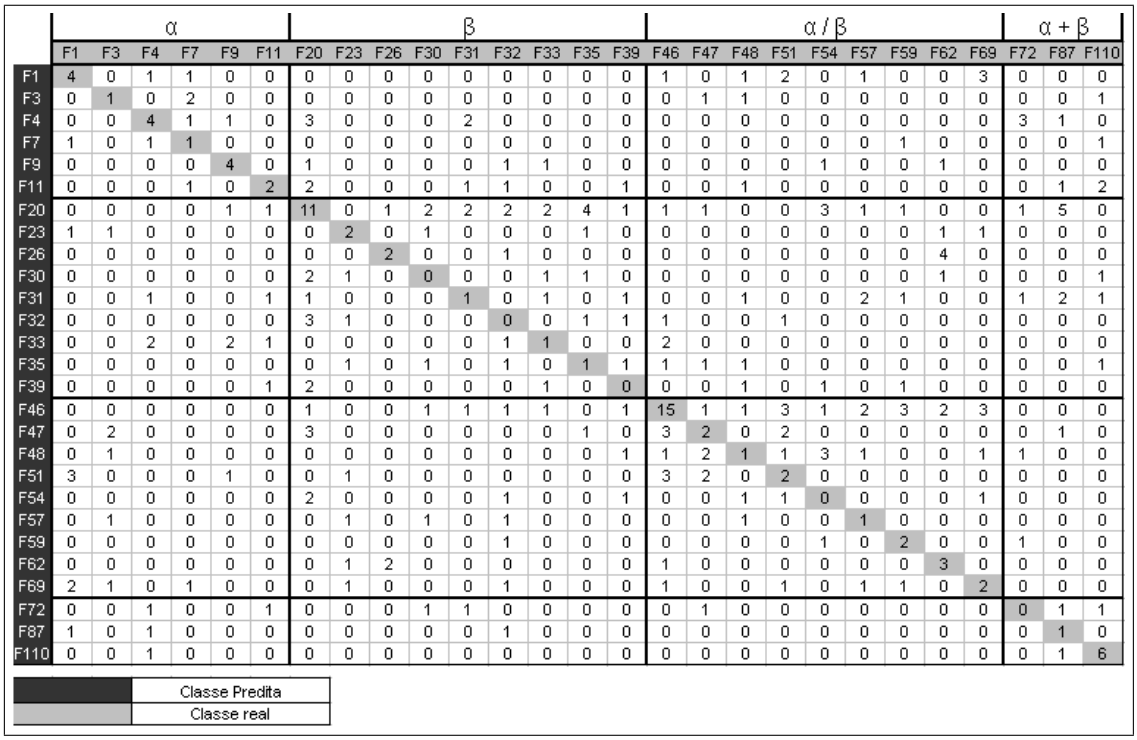
No caso em questão, observamos um grade número de exemplos mal classificados, ou seja, a taxa de confusão entre as classes é grande. Além disso, observando os exemplos referentes aos *folds* 20 e 46 notamos que estes se confundem com quase todos os outros *folds*, mesmo possuindo estruturas diferentes das preditas.

A taxa de acerto deste atributo para o método analisado foi de 17 %. Este valor de precisão é baixo, indicando a ineficiência do método para a distribuição dos dados analisados.

O segundo teste foi realizado com os dados de treinamento do atributo C utilizando o método multiclasse Um-Contra-Um. Assim como no primeiro teste, os dados foram treinados e classificados utilizando o esquema de validação *Leave-One-Out*, e os resultados da pontuação das classes são apresentados por uma matriz de confusão.

Estes resultados podem ser observados na matriz de confusão apresentada na Figura 7.8, a qual demonstra uma melhora significativa (23%) na taxa de acerto para a classificação dos dados analisados com este método. Além disso, proporciona a análise da taxa de acerto do treinamento dos dados referentes ao atributo C para

a classificação das proteínas em questão.



	α						β									α / β									α + β		
	F1	F3	F4	F7	F9	F11	F20	F23	F26	F30	F31	F32	F33	F35	F39	F46	F47	F48	F51	F54	F57	F59	F62	F69	F72	F87	F110
F1	4	0	1	1	0	0	0	0	0	0	0	0	0	0	0	1	0	1	2	0	1	0	0	3	0	0	0
F3	0	1	0	2	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	1
F4	0	0	4	1	1	0	3	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	3	1	0
F7	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1
F9	0	0	0	0	4	0	1	0	0	0	0	1	1	0	0	0	0	0	0	1	0	0	1	0	0	0	0
F11	0	0	0	1	0	2	2	0	0	0	1	1	0	0	1	0	0	1	0	0	0	0	0	0	0	1	2
F20	0	0	0	0	1	1	11	0	1	2	2	2	2	4	1	1	1	0	0	3	1	1	0	0	1	5	0
F23	1	1	0	0	0	0	0	2	0	1	0	0	0	1	0	0	0	0	0	0	0	0	1	1	0	0	0
F26	0	0	0	0	0	0	0	0	2	0	0	1	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0
F30	0	0	0	0	0	0	2	1	0	0	0	0	1	1	0	0	0	0	0	0	0	0	1	0	0	0	1
F31	0	0	1	0	0	1	1	0	0	0	1	0	1	0	1	0	0	1	0	0	2	1	0	0	1	2	1
F32	0	0	0	0	0	0	3	1	0	0	0	0	0	1	1	1	0	0	1	0	0	0	0	0	0	0	0
F33	0	0	2	0	2	1	0	0	0	0	0	1	1	0	0	2	0	0	0	0	0	0	0	0	0	0	0
F35	0	0	0	0	0	0	0	1	0	1	0	1	0	1	1	1	1	1	0	0	0	0	0	0	0	0	1
F39	0	0	0	0	0	1	2	0	0	0	0	0	1	0	0	0	0	1	0	1	0	1	0	0	0	0	0
F46	0	0	0	0	0	0	1	0	0	1	1	1	1	0	1	15	1	1	3	1	2	3	2	3	0	0	0
F47	0	2	0	0	0	0	3	0	0	0	0	0	0	1	0	3	2	0	2	0	0	0	0	0	0	1	0
F48	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	1	2	1	1	3	1	0	0	1	1	0	0
F51	3	0	0	0	1	0	0	1	0	0	0	0	0	0	0	3	2	0	2	0	0	0	0	0	0	0	0
F54	0	0	0	0	0	0	2	0	0	0	0	1	0	0	1	0	0	1	1	0	0	0	0	1	0	0	0
F57	0	1	0	0	0	0	0	1	0	1	0	1	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0
F59	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	1	0	2	0	0	1	0	0
F62	0	0	0	0	0	0	0	1	2	0	0	0	0	0	0	1	0	0	0	0	0	0	3	0	0	0	0
F69	2	1	0	1	0	0	0	1	0	0	0	1	0	0	0	1	0	0	1	0	1	1	0	2	0	0	0
F72	0	0	1	0	0	1	0	0	0	1	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	1
F87	1	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
F110	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	6

Figura 7.8 – *Resultado do treinamento do atributo C com o método Um-Contra-Todos.*

Após a analise dos resultados obtidos deste treinamento, notamos o aumento da taxa de acerto de 17 para 23 %. Os treinamentos referentes aos *folds* 20 e 46 apresentaram uma melhora significativa no número de exemplos classificados corretamente dentro das 27 classes analisadas.

Em nossos experimentos utilizamos o método multiclasse Um-Contra-Todos, pelo fato deste apresentar melhor desempenho (23%) em comparação ao outro método testado (17%).

Apesar da melhora obtida com o método Um-Contra-Todos ser significante, ela ainda não é satisfatória. O desempenho de 23 % obtido com o treinamento dos dados referentes ao atributo composição dos aminoácidos (C) não é suficientemente capaz de identificar as classes da maioria dos vetores classificados, tornando assim o atributo pouco confiável.

Para melhorar este desempenho, passamos a analisar os atributos referentes ao alinhamento das estruturas secundárias preditas. Estes atributos são utilizados na composição dos vetores de suporte para o treinamento de novos modelos.

O primeiro atributo analisado foi o *score* de alinhamento (AS), o qual apresentou um comportamento bastante promisor para os *folds* referentes as estruturas pertencentes a classe *all* α.

A Figura 7.9 descreve de forma gráfica o comportamento dos vetores de suporte das proteínas pertencentes aos *folds* 1, 3 e 4 da base1, referentes à classe estrutural *all* α.

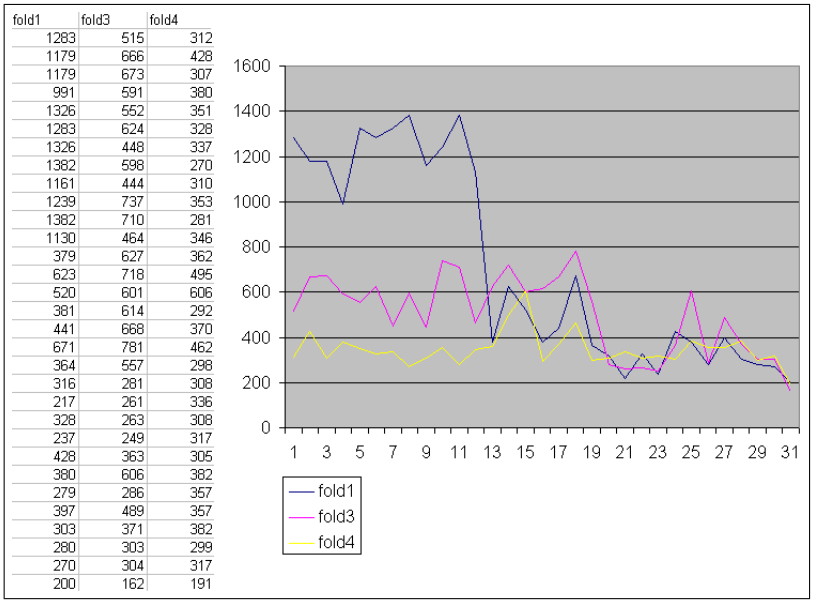


Figura 7.9 – *Gráfico do comportamento dos dados do AS para classe* α.

Analisando o comportamento deste gráfico, podemos observar uma distinção clara entre os *folds* 1 e 3. Os valores de AS das proteínas pertencentes a estes *folds* se diferenciam. Já no caso do *fold* 4 observamos uma pequena confusão com os valores dos *folds* 1 e 3, podendo afetar a taxa de acerto dos classificadores para este atributo.

Com base neste estudo, treinamos os dados individuais do atributo AS como primeiro passo desta etapa. Os vetores de suporte das proteínas referentes a este atributo foram compostos da seguinte forma: cada proteína alinhada gera um valor de AS, tendo assim uma lista de valores de AS para cada *fold*. Com base nesta lista, é calculado o maior valor de AS dentre os valores pertencentes ao mesmo *fold*, tendo assim um vetor de 27 posições onde cada posição corresponde ao maior valor do AS para cada *fold*.

A Figura 7.10 apresenta os dados de alinhamento da proteína 1BAB:B com as demais pertencentes ao *fold* 1. Na primeira coluna, é apresentada a referência PDB da proteína analisada, a segunda coluna refere-se ao tamanho desta proteína, a terceira e quarta colunas representam respectivamente a referência da proteína alinhada e o seu tamanho. Na quinta coluna encontram-se os valores de AS que desejamos analisar.

O maior valor de AS neste caso é 1283. Este valor irá compor a primeira posição do vetor de suporte da proteína 1BAB:B. As demais posições serão

compostas pelos maiores valores de AS do alinhamento da proteína 1BAB:B para cada *fold*. Estes valores devem ser menores para as proteínas alinhadas pertencentes a *folds* distintos, tornando assim os vetores de suporte capazes de classificar corretamente as proteínas selecionadas dentro dos 27 *folds* analisados.

	T1		T2	AS	fold
1BAB_B	145	1CPC_A	162	757	1
1BAB_B	145	1CPC_B	172	804	1
1BAB_B	145	1ECA	136	946	1
1BAB_B	145	1FLP	142	1079	1
1BAB_B	145	1HLB	157	**1283**	1
1BAB_B	145	1ITH	141	1108	1
1BAB_B	145	2GDM	153	1036	1
1BAB_B	145	2HBG	147	1054	1
1BAB_B	145	2LHB	149	966	1
1BAB_B	145	2MGE	154	1195	1
1BAB_B	145	3SDH_A	146	1018	1

Figura 7.10 – *Valores do AS para o alinhamento da proteína 1BAB:B do fold 1.*

Após a composição dos vetores de suporte com os dados do AS, foi feito o treinamento do modelo e a classificação com o método de validação *Leave-One-Out*. O resultado dos classificadores para este atributo são demonstrados pela matriz de confusão na Figura 7.11.

	α						β									α / β									α + β		
	F1	F3	F4	F7	F9	F11	F20	F23	F26	F30	F31	F32	F33	F35	F39	F46	F47	F48	F51	F54	F57	F59	F62	F69	F72	F87	F110
F1	12	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
F3	0	6	1	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
F4	0	0	8	0	0	2	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
F7	0	0	0	4	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
F9	0	1	0	1	3	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	1	0
F11	0	0	2	0	1	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
F20	0	0	0	0	0	0	26	6	1	5	0	2	1	2	0	0	0	0	0	0	0	0	0	0	0	0	0
F23	0	0	0	0	0	0	2	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
F26	0	0	0	0	0	0	0	1	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
F30	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
F31	0	0	0	0	0	0	0	0	0	1	5	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	2
F32	0	0	0	0	0	0	2	0	0	0	0	1	1	2	1	1	0	0	1	0	2	0	0	0	1	4	1
F33	0	0	0	0	0	0	0	0	0	0	0	1	5	2	1	0	0	0	0	0	0	0	0	0	0	0	0
F35	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1
F39	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0
F46	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	23	2	1	1	3	0	0	1	2	0	0	0
F47	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	2	1	1	1	2	2	0	0	0
F48	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	2	0	0	0	1	0
F51	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	1	5	0	0	0	0	0	0	0	0
F54	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	3	2	1	1	2	2	0	0	0
F57	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	4	0	1	0	0	0	0
F59	0	0	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	2	0	1	0	3	0	0	0	0	0
F62	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	1	0	0	1	0	2	4	2	0	0	0
F69	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	1	3	0	0	0
F72	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	4	1	0
F87	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	1	3	0
F110	0	0	0	0	0	0	0	0	0	0	3	2	0	1	0	0	0	0	0	0	0	0	0	0	0	2	9

Classe Real
Classe Predita

Figura 7.11 – *Matriz de confusão para treinamento do atributo AS.*

Analisando estes dados, podemos perceber uma diminuição da confusão das proteínas analisadas. A taxa de acerto do modelo treinado com os valores

deste atributo foi de 52 %, indicando uma melhora significativa na predição em comparação aos resultados obtidos anteriormente.

Apesar desta melhora ser significativa, a taxa de acerto dos classificadores ainda não é satisfatória. Analisando os resultados da matriz de confusão percebemos uma grande concentração de exemplos mal classificados em determinados *folds*.

Por exemplo, no caso dos *folds* 23 e 30 correspondentes a classe estrutural *all* β, existe uma grande confusão com o *fold* 20 também correspondente a esta classe. Outro exemplo de confusão é o apresentado pelos *folds* 54 e 47 correspondentes a classe estrutural α/β, onde a confusão maior é com o *fold* 46 da mesma classe.

Voltando aos dados de alinhamento, verificamos que o atributo *score* bruto (S) apresenta um comportamento semelhante ao AS, indicando uma possível melhora nos valores para a composição dos vetores de suporte. Este comportamento pode ser observado na Figura 7.12.

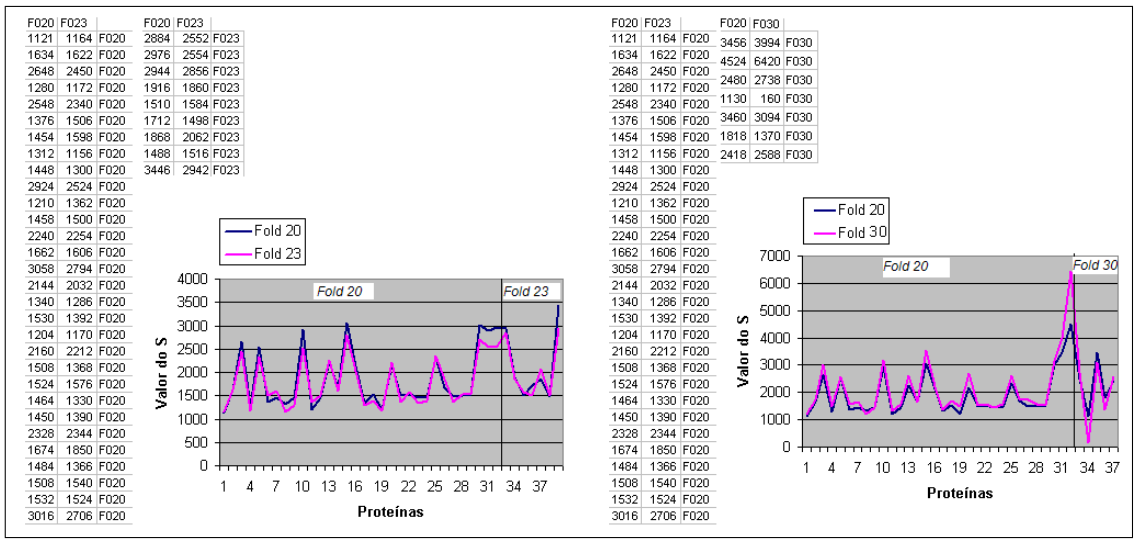


Figura 7.12 – *Análise dos valores do atributo S*

Os gráficos apresentados na Figura 7.12 são referentes aos dados do atributo S. Da mesma forma que o AS, os vetores de suporte referentes a este atributo também são compostos calculando o maior valor de S para cada *fold*.

Os *folds* 23 e 30 apresentaram uma alta taxa de confusão com o *fold* 20 para os dados do atributo AS. Por isso, resolvemos analisar estes casos separadamente para os dados do S.

Podemos observar que a linha referente ao *fold* 20 sofre uma queda quando entram os valores do S para as proteínas dos *folds* 23 e 30. Este fato faz com que os vetores de suporte sejam capazes de melhor separar os valores dos dados para estes *folds*.

O objetivo é obter o aumento da precisão para a classificação das proteínas pertencentes aos *folds* de maior confusão, e com isso, melhorar o desempenho do modelo.

Com base na análise destes dados, foram compostos os vetores de suporte utilizando os valores referentes ao atributo S. Estes vetores foram treinados e classificados utilizando o esquema *Leave-One-Out* compondo uma nova matriz de confusão, a qual apresenta o resultado da classificação para este atributo.

A matriz de confusão contendo os resultados da classificação para o atributo S pode ser visualizada na Figura 7.13.

	α						β									α + β							α / β				
	F1	F3	F4	F7	F9	F11	F20	F23	F26	F30	F31	F32	F33	F35	F39	F46	F47	F48	F51	F54	F57	F59	F62	F69	F72	F87	F110
F1	10	0	0	2	2	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
F3	0	3	2	0	0	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	1
F4	0	0	8	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
F7	1	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
F9	0	1	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	1	0	0	0	1	0
F11	1	0	0	0	1	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
F20	0	0	0	0	0	0	25	7	1	5	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
F23	0	0	0	0	0	0	2	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
F26	0	0	0	0	0	0	1	0	3	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
F30	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
F31	0	0	0	0	0	0	0	0	0	1	5	2	0	1	0	0	0	0	0	0	0	0	0	0	1	0	2
F32	0	0	0	0	0	0	0	0	0	0	0	3	1	0	1	1	1	0	0	0	0	0	0	0	2	4	0
F33	0	0	0	0	0	0	0	0	0	0	0	0	7	2	1	0	0	0	0	0	0	0	0	0	0	0	0
F35	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0
F39	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0
F46	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	18	3	2	2	3	0	1	3	6	0	0	0
F47	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	1	0	0	0	3	0	0	1	0
F48	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	1	3	4	0	0	3	0	1	0	0	0
F51	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	2	0	1	0	3	0	0	0	0	0
F54	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	2	4	1	2	0	0	0	0	0
F57	0	2	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0	0	0	1	0	0
F59	0	0	0	1	0	0	0	0	0	0	0	1	0	0	0	1	1	0	4	2	1	0	0	0	0	0	0
F62	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	2	0	0	0	0	0	3	1	0	0	0
F69	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	4	0	0	0	0	0	0	2	3	0	0	0
F72	0	0	0	0	0	0	0	0	0	0	1	3	1	0	0	0	0	0	0	0	1	0	0	0	2	0	0
F87	0	1	1	0	1	0	1	0	0	0	0	2	0	0	0	0	1	0	0	0	1	0	0	0	0	2	0
F110	0	0	1	0	0	1	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	2	10

Classe Predita
Classe Real

Figura 7.13 – *Matriz de confusão para treinamento do atributo S.*

Apesar dos resultados obtidos com o treinamento dos dados referentes ao atributo S terem apresentado uma diminuição da confusão entre a maioria dos *folds* da classe estrutural α/β, eles ainda não conseguiram resolver o problema da confusão dos *folds* 23 e 30 com o *fold* 20 desta mesma classe. Além disso, os *folds* da classe estrutural *all* α apresentaram uma maior confusão comparados aos resultados obtidos com o treinamento dos dados referentes ao atributo AS.

O aumento na confusão dos *folds* desta classe causou uma queda no valor do desempenho para o modelo treinado. A taxa de acerto desta classificação não passou de 46 %, indicando uma piora da precisão dos exemplos treinados com o atributo S em comparação ao AS.

Analisando o conjunto de proteínas correspondente ao *fold* 20, percebemos uma diferença grande do tamanho destas seqüências comparadas com as demais

analisadas. Isso nos leva a crer que este tamanho tem influência sobre o valor dos dados referentes aos atributos S e AS.

Esta influência pode ser a causa da confusão entre os *folds*, pois estes atributos geralmente apresentam um valor alto quando alinhados com seqüências de tamanho grande.

Uma forma encontrada para diminuir a influência do tamanho das proteínas sobre o valor dos atributos S e AS foi a composição dos vetores de suporte com a média dos valores destes atributos para cada classe.

Como exemplo, tomamos novamente os dados de alinhamento da proteína 1BAB:B com as demais pertencentes ao *fold* 1. O vetor de suporte para esta proteína é composto pelo cálculo da média dos valores referentes aos atributos analisados, observado na Figura 7.14.

	T1		T2	S	AS
1BAB_B	145	1CPC_A	162	2101	757
1BAB_B	145	1CPC_B	172	2202	804
1BAB_B	145	1ECA	136	2338	946
1BAB_B	145	1FLP	142	2552	1079
1BAB_B	145	1HLB	157	2735	1283
1BAB_B	145	1ITH	141	2589	1108
1BAB_B	145	2GDM	153	2518	1036
1BAB_B	145	2HBG	147	2402	1054
1BAB_B	145	2LHB	149	2378	966
1BAB_B	145	2MGE	154	2735	1195
1BAB_B	145	3SDH_A	146	2454	1018
1BAB_B	145	1C53	79	1073	276
			Somatorio	28077	11522
			Média	2339,75	960,1667

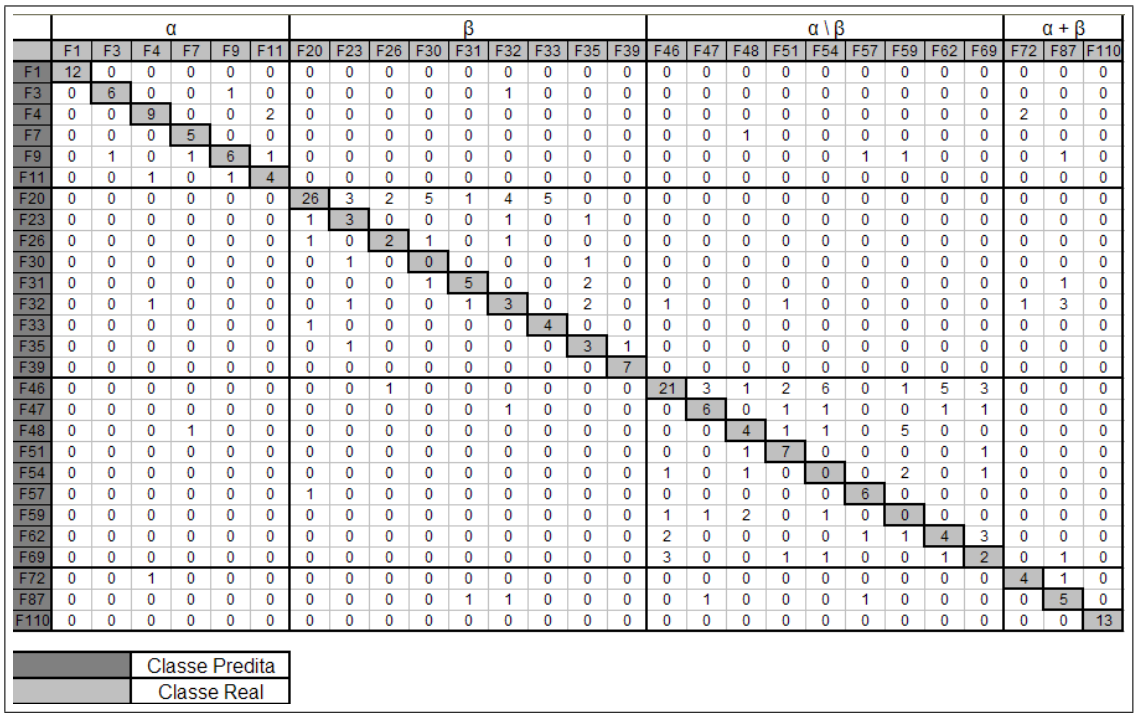
Figura 7.14 – *Exemplo do cálculo da média dos valores de S e AS para a proteína 1BAB:B.*

Após a composição dos vetores de suporte com a média dos valores de S e AS foram treinados dois novos modelos, um para cada atributo.

Os resultados do treinamento e classificação para estes atributos demonstraram uma melhora na taxa de acerto da classificação com a utilização dos dados da média em comparação aos modelos analisados anteriormente. Esse fato demonstra a real existência da influência do tamanho das seqüências sobre os atributos treinados.

As matrizes de confusão mostrando os resultados dos treinamentos para os valores da média destes atributos podem ser observadas respectivamente nas Figuras 7.15 e 7.16.

Analisando a matriz de confusão da classificação dos dados do atributo AS, percebemos a diminuição do erro dos classificadores para os *folds* pertencentes a classe estrutural *all* α, proporcionando o desempenho de 56% para o modelo treinado.

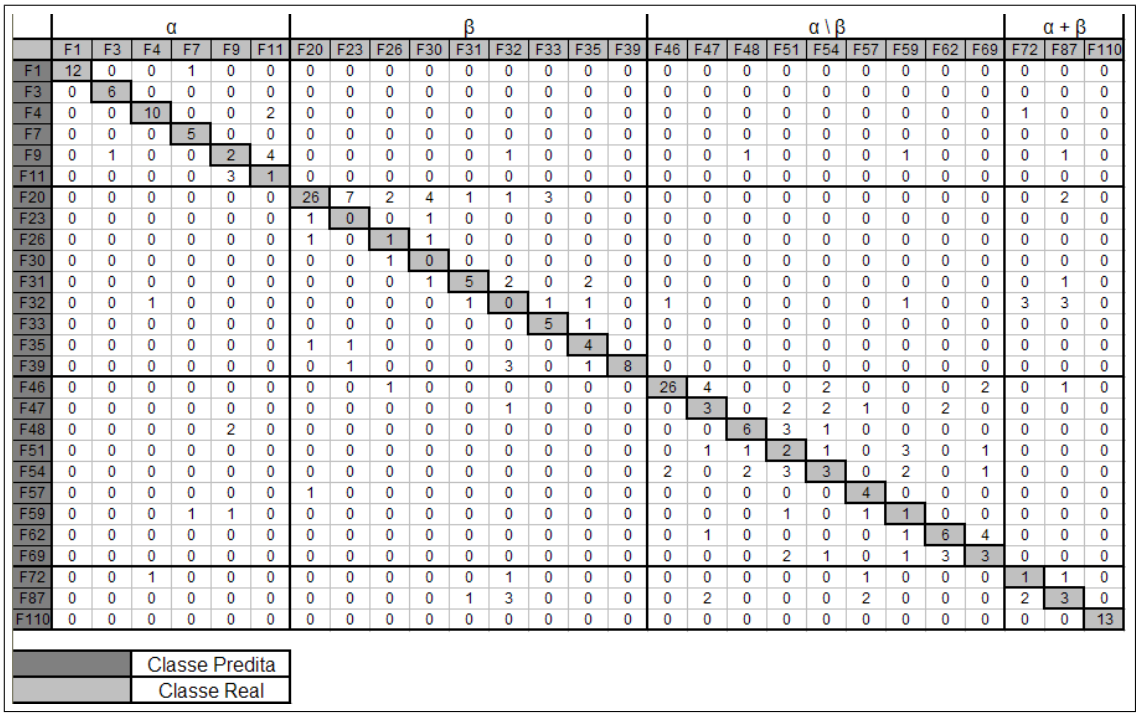


	α						β									α \ β									α + β		
	F1	F3	F4	F7	F9	F11	F20	F23	F26	F30	F31	F32	F33	F35	F39	F46	F47	F48	F51	F54	F57	F59	F62	F69	F72	F87	F110
F1	12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
F3	0	6	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
F4	0	0	9	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0
F7	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
F9	0	1	0	1	6	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	1	0
F11	0	0	1	0	1	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
F20	0	0	0	0	0	0	26	3	2	5	1	4	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0
F23	0	0	0	0	0	0	1	3	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
F26	0	0	0	0	0	0	1	0	2	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
F30	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
F31	0	0	0	0	0	0	0	0	0	1	5	0	0	2	0	0	0	0	0	0	0	0	0	0	0	1	0
F32	0	0	1	0	0	0	0	1	0	0	1	3	0	2	0	1	0	0	1	0	0	0	0	0	1	3	0
F33	0	0	0	0	0	0	1	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0
F35	0	0	0	0	0	0	0	1	0	0	0	0	0	3	1	0	0	0	0	0	0	0	0	0	0	0	0
F39	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7	0	0	0	0	0	0	0	0	0	0	0	0
F46	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	21	3	1	2	6	0	1	5	3	0	0	0
F47	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	6	0	1	1	0	0	1	1	0	0	0
F48	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	4	1	1	0	5	0	0	0	0	0
F51	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	7	0	0	0	0	1	0	0	0
F54	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	2	0	1	0	0	0
F57	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	0
F59	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	2	0	1	0	0	0	0	0	0	0
F62	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	1	1	4	3	0	0	0
F69	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	1	1	0	0	1	2	0	1	0
F72	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	1	0
F87	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	0	1	0	0	0	1	0	0	0	0	5	0
F110	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	13

Classe Predita
Classe Real

Figura 7.15 – *Matriz de confusão para os valores da média do AS.*

O desempenho do treinamento deste atributo é bastante satisfatório (56%), sendo o melhor obtido até o momento. Porém, para os *folds* 20 e 46 considerados mais problemáticos, a quantidade de confusão com os demais *folds* ainda continua alta.



	α						β									α \ β									α + β		
	F1	F3	F4	F7	F9	F11	F20	F23	F26	F30	F31	F32	F33	F35	F39	F46	F47	F48	F51	F54	F57	F59	F62	F69	F72	F87	F110
F1	12	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
F3	0	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
F4	0	0	10	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
F7	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
F9	0	1	0	0	2	4	0	0	0	0	0	1	0	0	0	0	0	1	0	0	0	1	0	0	0	1	0
F11	0	0	0	0	3	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
F20	0	0	0	0	0	0	26	7	2	4	1	1	3	0	0	0	0	0	0	0	0	0	0	0	0	2	0
F23	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
F26	0	0	0	0	0	0	1	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
F30	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
F31	0	0	0	0	0	0	0	0	0	1	5	2	0	2	0	0	0	0	0	0	0	0	0	0	0	1	0
F32	0	0	1	0	0	0	0	0	0	0	1	0	1	1	0	1	0	0	0	0	0	1	0	0	3	3	0
F33	0	0	0	0	0	0	0	0	0	0	0	0	5	1	0	0	0	0	0	0	0	0	0	0	0	0	0
F35	0	0	0	0	0	0	1	1	0	0	0	0	0	4	0	0	0	0	0	0	0	0	0	0	0	0	0
F39	0	0	0	0	0	0	0	1	0	0	0	3	0	1	8	0	0	0	0	0	0	0	0	0	0	0	0
F46	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	26	4	0	0	2	0	0	0	2	0	1	0
F47	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	3	0	2	2	1	0	2	0	0	0	0
F48	0	0	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	6	3	1	0	0	0	0	0	0	0
F51	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	2	1	0	3	0	1	0	0	0
F54	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	2	3	3	0	2	0	1	0	0	0
F57	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	4	0	0	0	0	0	0
F59	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	0	0	0	0	0
F62	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	6	4	0	0	0
F69	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	1	0	1	3	3	0	0	0
F72	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0	1	1	0
F87	0	0	0	0	0	0	0	0	0	0	1	3	0	0	0	0	2	0	0	0	2	0	0	0	2	3	0
F110	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	13

Classe Predita
Classe Real

Figura 7.16 – *Matriz de confusão para os valores da média do S.*

O modelo treinado com os dados do atributo S apresentou uma taxa de acerto de 53 %, persistindo a confusão dos *folds* 20 e 46 com os demais analisados.

Apesar da taxa de 56 % de acerto dos classificadores obtida com o treinamento dos vetores de suporte compostos com a média dos valores do atributo AS ser bastante satisfatória, ainda busca-se aumentar este valor.

Analisando os resultados obtidos, observamos a elevada ocorrência de exemplos mal classificados entre os *folds* 20 e 46. A confusão destes *folds* específicos persiste em praticamente todos os experimentos realizados até o momento. Este fato faz com que nossos esforços sejam concentrados na tentativa de compor vetores de suporte capazes de melhor separar os dados, e conseqüentemente aumentar a precisão do modelo.

Baseado nos experimentos realizados, observamos que a influência do tamanho das seqüências sobre os atributos analisados é um caso a ser analisado. Além disso, também podemos perceber o bom desempenho do treinamento dos vetores de suporte com os dados do AS.

Com base nisso, foram compostos novos vetores de suporte com dados referentes ao atributo AS. Os vetores são formados pelos maiores valores da divisão do AS pela soma dos tamanhos das proteínas alinhadas. Esta divisão foi outra forma encontrada para diminuir a influência do tamanho das seqüências sobre o atributo analisado, buscando melhorar a precisão do modelo treinado.

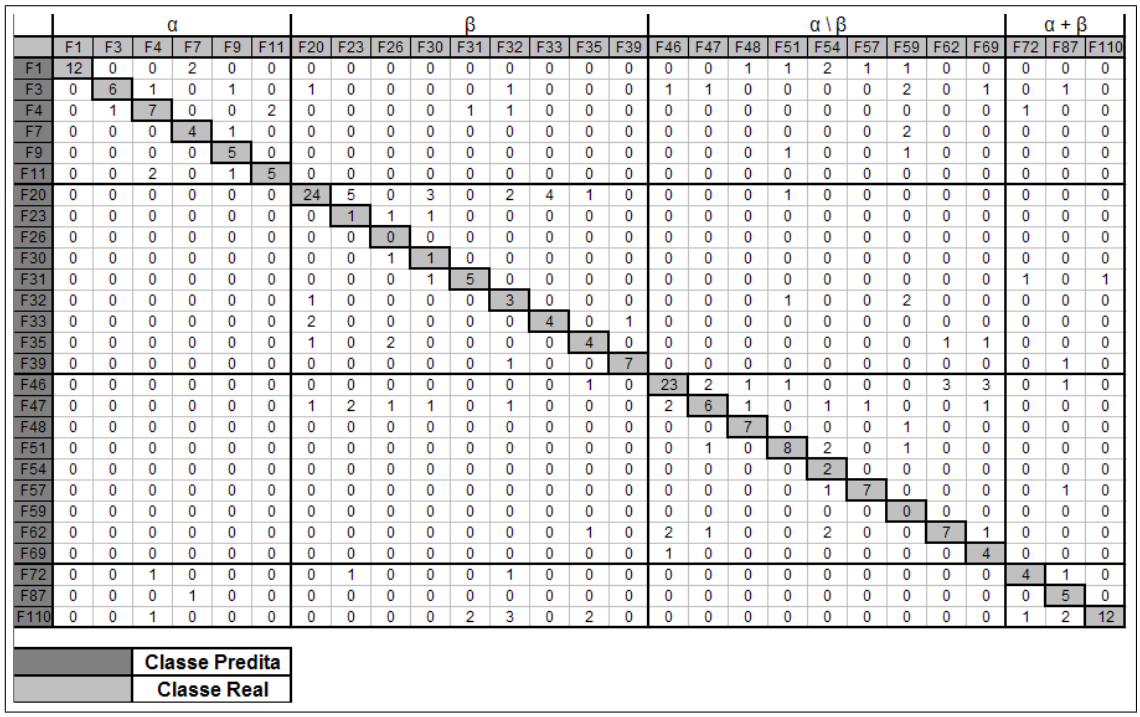
A Figura 7.17 demonstra o funcionamento deste processo. Novamente utilizamos os dados de alinhamento da proteína 1BAB:B como exemplo para o cálculo dos valores que irão compor os novos vetores de suporte.

	T1		T2	AS	AS/(T1+T2)
1BAB_B	145	1CPC_A	162	757	2,5
1BAB_B	145	1CPC_B	172	804	2,5
1BAB_B	145	1ECA	136	946	3,4
1BAB_B	145	1FLP	142	1079	3,8
1BAB_B	145	1HLB	157	1283	4,2
1BAB_B	145	1ITH	141	1108	3,9
1BAB_B	145	2GDM	153	1036	3,5
1BAB_B	145	2HBG	147	1054	3,6
1BAB_B	145	2LHB	149	966	3,3
1BAB_B	145	2MGE	154	1195	4,0
1BAB_B	145	3SDH_A	146	1018	3,5
1BAB_B	145	1C53	79	276	1,2
			4,2	Maior valor do AS/(T1 + T2)	

Figura 7.17 – *Exemplo do cálculo da divisão dos valores de AS pela soma dos tamanhos das seqüências alinhadas.*

Os vetores de suporte compostos com estes valores foram treinados e classificados utilizando o método de validação *Leave-One-Out*.

Os resultados da classificação para este atributo são apresentados na matriz de confusão apresentados na Figura 7.18.



	α						β									α \ β									α + β		
	F1	F3	F4	F7	F9	F11	F20	F23	F26	F30	F31	F32	F33	F35	F39	F46	F47	F48	F51	F54	F57	F59	F62	F69	F72	F87	F110
F1	12	0	0	2	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	2	1	1	0	0	0	0	0
F3	0	6	1	0	1	0	1	0	0	0	0	1	0	0	0	1	1	0	0	0	0	2	0	1	0	1	0
F4	0	1	7	0	0	2	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
F7	0	0	0	4	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0
F9	0	0	0	0	5	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	0
F11	0	0	2	0	1	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
F20	0	0	0	0	0	0	24	5	0	3	0	2	4	1	0	0	0	0	1	0	0	0	0	0	0	0	0
F23	0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
F26	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
F30	0	0	0	0	0	0	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
F31	0	0	0	0	0	0	0	0	0	1	5	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1
F32	0	0	0	0	0	0	1	0	0	0	0	3	0	0	0	0	0	0	1	0	0	2	0	0	0	0	0
F33	0	0	0	0	0	0	2	0	0	0	0	0	4	0	1	0	0	0	0	0	0	0	0	0	0	0	0
F35	0	0	0	0	0	0	1	0	2	0	0	0	0	4	0	0	0	0	0	0	0	0	1	1	0	0	0
F39	0	0	0	0	0	0	0	0	0	0	0	1	0	0	7	0	0	0	0	0	0	0	0	0	0	1	0
F46	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	23	2	1	1	0	0	0	3	3	0	1	0
F47	0	0	0	0	0	0	1	2	1	1	0	1	0	0	0	2	6	1	0	1	1	0	0	1	0	0	0
F48	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	7	0	0	0	1	0	0	0	0	0
F51	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	8	2	0	1	0	0	0	0	0
F54	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0	0
F57	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	7	0	0	0	0	1	0
F59	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
F62	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	2	1	0	0	2	0	0	7	1	0	0	0
F69	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	4	0	0	0
F72	0	0	1	0	0	0	0	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	4	1	0
F87	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	5	0
F110	0	0	1	0	0	0	0	0	0	0	2	3	0	2	0	0	0	0	0	0	0	0	0	0	1	2	12

Figura 7.18 – *Matriz de confusão para os valores da divisão do atributo AS pela soma do tamanho das seqüências alinhadas da Base 1.*

Analisando esta matriz, observamos uma melhora na classificação dos exemplos para os *folds* que apresentaram maior confusão nos modelos anteriormente treinados.

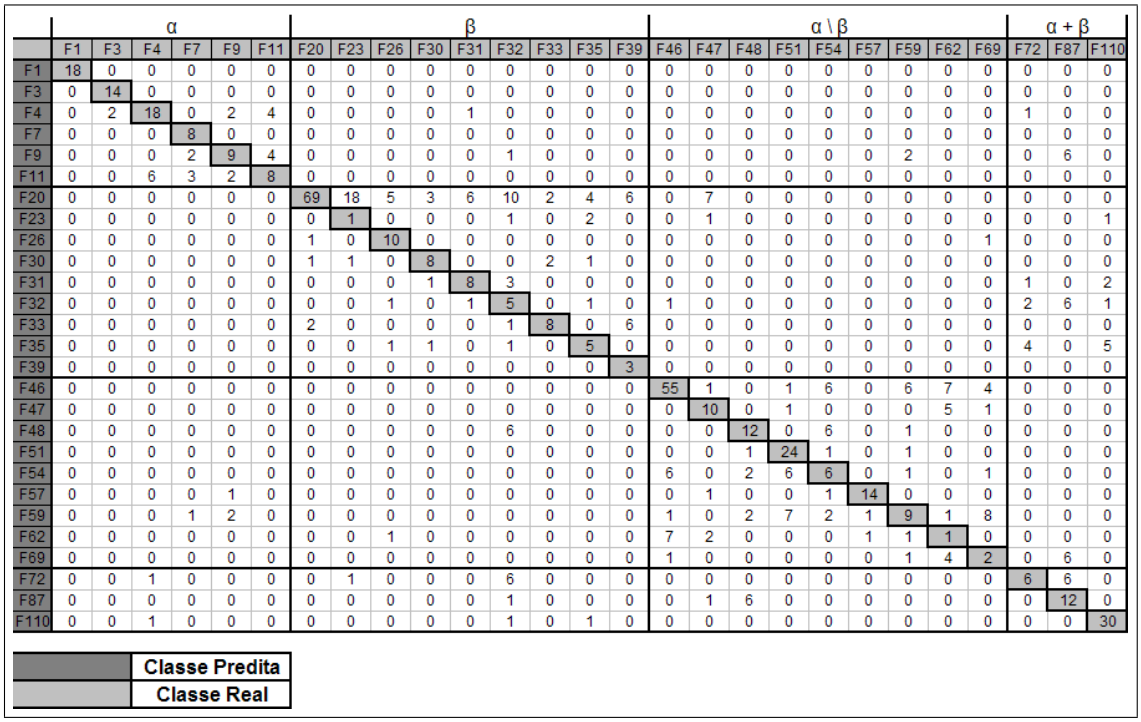
A taxa de acertos para este atributo foi de 57%, superando todos os outros modelos treinados até o momento. Isso indica uma possível metodologia a ser seguida para o tratamento dos problemas de classificação deste tipo.

Apesar do desempenho da predição dos dados com este atributo ser bastante satisfatório, ainda buscamos melhorar este valor. Com isso, outros modelos foram treinados e analisados utilizando diferentes bases de dados, as quais foram descritas anteriormente na Tabela 6.1.

Observando esta tabela, podemos perceber uma diferença no tamanho das bases de dados selecionadas. A Base 2 foi composta para tornar o modelo mais genérico, abordando um número maior de tipos de estruturas conhecidas. Porém, os resultados obtidos para o treinamento dos vetores de suporte com os dados referentes aos atributos desta base não superaram os valores de precisão obtidos com os dados da Base 1.

A Figura 7.19 descreve a matriz de confuzão com os resultados obtidos para o treinamento do melhor modelo avaliado referente aos dados da Base 2. Este modelo possui uma taxa de desempenho de 56,3%, sendo inferior aos 57,1% obtidos com o

treinamento do mesmo atributo para a Base 1.



	α						β									α \ β									α + β		
	F1	F3	F4	F7	F9	F11	F20	F23	F26	F30	F31	F32	F33	F35	F39	F46	F47	F48	F51	F54	F57	F59	F62	F69	F72	F87	F110
F1	18	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
F3	0	14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
F4	0	2	18	0	2	4	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
F7	0	0	0	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
F9	0	0	0	2	9	4	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	2	0	0	0	6	0
F11	0	0	6	3	2	8	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
F20	0	0	0	0	0	0	69	18	5	3	6	10	2	4	6	0	7	0	0	0	0	0	0	0	0	0	0
F23	0	0	0	0	0	0	0	1	0	0	0	1	0	2	0	0	1	0	0	0	0	0	0	0	0	0	1
F26	0	0	0	0	0	0	1	0	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
F30	0	0	0	0	0	0	1	1	0	8	0	0	2	1	0	0	0	0	0	0	0	0	0	0	0	0	0
F31	0	0	0	0	0	0	0	0	0	1	8	3	0	0	0	0	0	0	0	0	0	0	0	0	1	0	2
F32	0	0	0	0	0	0	0	0	1	0	1	5	0	1	0	1	0	0	0	0	0	0	0	0	2	6	1
F33	0	0	0	0	0	0	2	0	0	0	0	1	8	0	6	0	0	0	0	0	0	0	0	0	0	0	0
F35	0	0	0	0	0	0	0	0	1	1	0	1	0	5	0	0	0	0	0	0	0	0	0	0	4	0	5
F39	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0	0	0	0	0	0
F46	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	55	1	0	1	6	0	6	7	4	0	0	0
F47	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	10	0	1	0	0	0	5	1	0	0	0
F48	0	0	0	0	0	0	0	0	0	0	0	6	0	0	0	0	0	12	0	6	0	1	0	0	0	0	0
F51	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	24	1	0	1	0	0	0	0	0
F54	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	6	0	2	6	6	0	1	0	1	0	0	0
F57	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	14	0	0	0	0	0	0
F59	0	0	0	1	2	0	0	0	0	0	0	0	0	0	0	1	0	2	7	2	1	9	1	8	0	0	0
F62	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	7	2	0	0	0	1	1	1	0	0	0	0
F69	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	4	2	0	6	0
F72	0	0	1	0	0	0	0	1	0	0	0	6	0	0	0	0	0	0	0	0	0	0	0	0	6	6	0
F87	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	1	6	0	0	0	0	0	0	0	12	0
F110	0	0	1	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	0	0	0	30

Figura 7.19 – *Matriz de confusão para os valores da divisão do atributo AS pela soma do tamanho das seqüências alinhadas da Base 2.*

A Tabela 7.4 descreve todos os experimentos realizados com ambas as bases, bem como a taxa de precisão obtida para cada um deles.

Tabela 7.4 – *Tabela com os valores dos experimentos realizados.*

Atributos	**Base 1**	**Base 2**
C	23%	25%
Max (AS)	52%	51%
Max (S)	45%	41%
Média (AS)	56%	54%
Média (S)	52%	50%
Max [AS/(T1+T2)]	57,1%	56,3%
Max [S/(T1+T2)]	55%	53%
Max [AS/(CL)]	46%	43%
Max [S/(CL)]	48%	48%
Max (AS) | Max (S)	52%	51%
C | Max (AS)	46%	45%

Analisando os dados desta tabela, observamos que a maior taxa de acertos foi de 57% obtida com o treinamento do modelo composto com os dados referentes e

56,3% ao atributo Max [AS/(T1+T2)], em ambas as bases. Estes resultados são melhor analisados na Seção (7.3).

7.3 Análise dos Resultados Obtidos

Os resultados obtidos com o treinamento dos atributos propostos foram muito satisfatórios. As melhores taxas de acerto foram obtidas com o treinamento dos dados referentes ao alinhamento das estruturas secundárias preditas. Isso demonstra o bom funcionamento da metodologia proposta para tratar problemas envolvendo a classificação automática de proteínas.

Além disso, esta taxa supera os valores de precisão apresentados em trabalhos semelhantes encontrados na literatura, como no caso do trabalho proposto por Ding e Dubchak [DING and DUBCHAK 2001].

Este trabalho propôs a utilização de RNA e MSV para a classificação automática de proteínas. Os autores utilizaram os atributos composição dos aminoácidos (C), estrutura secundária (S), hidrofobicidade (H), polarizabilidade (Z), polaridade (P) e volume de *van der Waals*(V). O melhor resultado obtido foi de 56,5 % com a combinação dos atributos CSHP utilizando o método Um-Contra-Um.

O esquema de validação do modelo utilizado pelos autores deste trabalho foi o 10*fold Cross Validation*. Este método proporciona uma avaliação aproximada do modelo treinado.

A metodologia proposta em nosso trabalho obteve 57,1% de precisão do modelo treinado, superando os 56.5% obtidos por Ding e Dubchak para a mesma base de dados analisada. Além disso, o método de validação utilizado em nosso trabalho foi o *Leave-One-Out*, o qual proporciona a medida efetiva da precisão da classificação. Este método é considerado superior ao utilizado pelos autores do trabalho descrito anteriormente, pois com ele conseguimos obter uma medida precisa do modelo treinado ao contrario do método por eles utilizado que faz uma aproximação da medida do desempenho para os treinamentos realizados.

Outra vantagem importante da metodologia adotada em nosso trabalho é a ordem dos vetores de suporte de melhor desempenho não superar 27 elementos, ao contrário dos descritos no trabalho anterior, os quais apresentam ordem de 83 elementos para os melhores resultados.

Os resultados obtidos com a metodologia proposta mostram que a previsão do *fold* SCOP associado com o alinhamento da estrutura secundária predita obteve resultados bastante satisfatórios. Desta forma, uma maior precisão poderia ser alcançada em um sistema no qual a influência do tamanho das seqüências no cálculo

dos valores dos atributos utilizados para a composição dos vetores de suporte fosse minimizada.

Capítulo 8

Conclusões

Esta dissertação apresentou uma nova metodologia que propõe a utilização dos dados de alinhamento de estruturas secundárias preditas para a composição dos vetores de suporte que são treinados utilizando a técnica de aprendizado de máquina denominada Máquinas de Suporte Vetorial (MSV).

No Capítulo 2 foram apresentados os principais conceitos e terminologias utilizadas pela Biologia Molecular para descrever o problema abordado neste estudo. Estes conceitos possuem fundamental importância para o entendimento e tratamento do problema proposto neste trabalho. Dentre os tópicos abordados, destacam-se os aminoácidos, as proteínas e suas conformações estruturais.

Neste capítulo, podemos perceber a vasta complexidade que envolve as estruturas de proteínas. Além disso, nos fornece uma visão clara do problema analisado bem como sua relevância do ponto de vista biológico.

A tarefa de treinamento das MSV envolve a resolução de um problema de Otimização Matemática. O Capítulo 3 apresenta algumas definições desta teoria com o objetivo de fundamentar a utilização desta técnica de aprendizado de máquina para o treinamento dos dados.

Além disso, esta teoria é de fundamental importância para o entendimento dos conceitos básicos da técnica MSV abordada em nosso estudo.

Nest capítulo observamos a complexidade matemática que envolve a técnica de aprendizado de maquina utilizada para o treinamento dos modelos.

O Capítulo 4 apresenta de forma detalhada os aspectos teóricos associados com a teoria matemática que descreve o funcionamento das MSV. Esta teoria não é trivial de ser compreendida, por isso, os processos matemáticos envolvidos nesta técnica foram apresentados de forma seqüencial com o objetivo de simplificar o entendimento dos mesmos. Além disso, foram utilizados gráficos, os quais representam a idéia dos principais processos matemáticos utilizados.

Este capítulo possui ainda uma seção que descreve os dois principais métodos multiclasse aplicados para a classificação de problemas deste tipo. Um deles é o método Um-Contra-Um e o outro é o método Um-Contra-Todos. O objetivo foi testar qual deles melhor se aplica na classificação do tipo de distribuição de dados utilizados em nossos experimentos.

Após o estudo dos aspectos básicos e teóricos envolvidos no desenvolvimento da pesquisa abordada neste trabalho, foram selecionados alguns dos principais trabalhos relacionados com o problema que estamos analisando. O Capítulo 5 descreve alguns destes trabalhos, abordando temas como: técnicas computacionais aplicadas na biologia, predição de estruturas secundárias de proteínas e aplicação de MSV no tratamento de problemas biológicos.

Esta etapa foi uma das principais desenvolvidas neste estudo, pois nos proporcionou o embasamento teórico necessário para a aplicação da técnica proposta. Além disso, destacamos os valores dos experimentos realizados por Ding e Dubchack [DING and DUBCHAK 2001] como parâmetros serem superados.

A metodologia proposta para tratar o problema analisado foi descrita no Capítulo 6. Ela foi dividida da seguinte forma: inicialmente foram apresentadas a descrição do conjunto de dados, depois foram definidos os atributos utilizados no treinamento dos modelos, a seleção dos dados e a composição do conjunto de dados de treinamento. Neste capítulo também foi descrito a implementação dos classificadores e o método utilizado para medir o desempenho dos modelos treinados.

Após estas definições, foi realizado o primeiro treinamento e classificação dos dados. Este experimento teve o objetivo de avaliar o comportamento das MSV quando aplicadas na classificação de proteínas segundo a hierarquia SCOP. Além disso, também serviu para validar os conceitos básicos apresentados nos capítulos anteriores.

Nesta etapa, foi possível avaliar o conjunto de dados adotado, identificando o nível de confusão para cada *fold* classificado. Este experimento contribuiu para a compreensão prática da técnica, reforçando o conhecimento teórico e avaliando a aplicabilidade das MSV para o tratamento de problemas de classificação de proteínas.

Os experimentos realizados no decorrer do trabalho foram apresentados no Capítulo 7. A metodologia utilizada para a seleção, composição dos vetores, treinamento e predição dos dados teve o objetivo de aprimorar o desempenho dos modelos encontrados na literatura como no trabalho desenvolvido por Ding e Dubchak [DING and DUBCHAK 2001]. Para isso, foram compostos diversos conjuntos de dados utilizando valores de diferentes atributos.

A cada treinamento os valores foram analisados, e a partir dos resultados

foram desenvolvidas novas estratégias para o tratamento do problema. Além disso, estes experimentos permitiram a aplicação de diferentes tipos de *kernel*, bem como a utilização do método *Leave-One-Out* para a composição das matrizes de confusão, fornecendo uma medida efetiva do modelo treinado.

Três tipos de *kernel* foram comparados: linear, polinomial e RBF. Dentre os três, o *kernel* linear apresentou o melhor desempenho para ambos os experimentos realizados.

O *kernel* polinomial apresentou uma capacidade razoável de generalização dos dados, porém não supera os valores obtidos com o *kernel* linear. O último analisado foi o RBF, o qual apresentou as piores taxas de acertos para os modelos treinados.

Após a definição do melhor tipo de *kernel* para os dados selecionados, foram realizados experimentos afim de comparar o desempenho de dois métodos multiclasse, o Um-Contra-Um e o Um-Contra-Todos.

A maior vantagem da utilização do método Um-Contra-Todos é o fato de este envolver poucos classificadores no treinamento do modelo, tornando a classificação mais rápida em relação ao outro método proposto. Entretanto, percebemos a falta de balanceamento nos dados, o que prejudica o seu desempenho.

O método que obteve melhor desempenho foi o Um-Contra-Um, sendo capaz de diminuir o índice de confusão dos dados classificados em ambos experimentos realizados. Além disso, este método possui fácil manipulação, uma vez que é internamente implementado no pacote *LibSVM* para o *software R-project*.

A maior contribuição deste trabalho foi a metodologia proposta para o tratamento do problema de classificação automática de proteínas.

Os vetores de suporte foram compostos com valores de atributos gerados pelo alinhamento de estruturas secundárias preditas das proteínas selecionadas. Estes vetores foram treinados e avaliados gerando taxas de acertos bastante satisfatórias, as quais superaram os valores obtidos por trabalhos semelhantes encontrados na literatura.

O principal desafio enfrentado com a aplicação desta metodologia foi a influência exercida pelo tamanho das seqüências sobre os valores dos atributos utilizados na composição dos vetores de suporte treinados.

Algumas estratégias foram aplicadas na tentativa de diminuir esta influência, sendo que a melhor delas foi a utilização do maior valor da divisão do atributo AS pelo tamanho das seqüências alinhadas, a qual gerou a taxa de 57% de acertos para os exemplos treinados.

Com isso, concluímos que a metodologia proposta para o tratamento do problema de classificação automática de proteínas é eficiente, podendo ser utilizada como base para a classificação de novas estruturas ainda desconhecidas.

8.1 Perspectivas de Trabalhos Futuros

Os experimentos realizados no decorrer deste trabalho provaram a eficiência da metodologia proposta e da técnica de aprendizado de máquina MSV aplicada na classificação automática de proteínas. Porém, cabem algumas considerações e sugestões para trabalhos futuros, a quais são apresentadas a seguir:

- Realizar um estudo mais aprofundado das características estruturais das proteínas, bem como o aumento das bases de dados de treinamento abrangendo um número maior de parâmetros e *folds* analisados.

- Desenvolver métodos automáticos capazes de selecionar informações como a ordem em que as estruturas de α hélice e folhas β se apresentam na estrutura secundária das proteínas. Estas informações combinadas com outros atributos irão compor um vetor de suporte de tamanho variável que possivelmente melhoraria a separação dos dados de treinamento diminuindo a confusão entre os classificadores, o que aumentaria a eficiência do método.

- Realizar uma análise crítica de cada proteína que compõe o conjunto de dados de treinamento a fim de obter somente a informação biológica que caracteriza o *fold* desejado, uma vez que se constatou a existência de informação irrelevante nos dados treinados. Isso causa uma confusão da classificação dos dados diminuindo a precisão do modelo.

- Realizar experimentos utilizando os dados de alinhamento das estruturas reais das proteínas para o treinamento do modelo, e a partir deste classificar os dados com as informações da estrutura secundária predita.

Bibliografia

[ALBERTS et al. 2004] ALBERTS, B., JOHNSON, A., LEWIS, J., RAFF, M., ROBERTS, K., and WALTER, P. (2004). *Biologia Molecular da Célula*. Porto Alegre: Artmed:4.ed.

[AMABIS and MEIDANIS 1998] AMABIS, M. and MEIDANIS, J. (1998). *Fundamento da Biologia Moderna*. Moderna: 2.ed., São Paulo.

[BARTON 1995] BARTON, G. J. (1995). Protein secondary structure prediction. *Curr. Opin. Struct. Biol*, 5:372–376.

[BRIDLE 1990] BRIDLE, J. (1990). Training stochastic model recognition algorithms as networks cam leadto maximum mutual information estimation of parameters. *Advances in neural information processing systems 2 table of contents*, pages 211–217.

[BRUNAK and BALDI 2001] BRUNAK, S. and BALDI, P. (2001). *Bioinformatics: The Machine Learning Approach*. Bradford Book.

[BURGES 1998] BURGES, C. (1998). A tutorial on support vector machines for pattern re-cognition. *Data Mining and Knowledge Discovery*, 2(2):121–167.

[CAI et al. 2001] CAI, Y.-D., LIU, X.-J., XU, X.-B., and ZHOU, G.-P. (2001). Support vector machines for predicting protein structural class. *BMC Bioinformatics*, 2(3):3.

[CHANG and LIN 2001] CHANG, C. and LIN, C. (2001). Libsvm: a library for support vector machines. *Software available at http://www. csie. ntu. edu. tw/cjlin/libsvm*, 80:604–611.

[CHOU 1995] CHOU, K. (1995). A novel approach to predicting protein structural classes in a amino acid composition space. *PROTEINS: Structure, Function, and Genetics*, 21:319–344.

[COST and SALZBERG 1993] COST, S. and SALZBERG, S. (1993). A weighted nearest neighbor algorithm for learning with symbolic features. *Machine Learning*, 10(1):57–78.

[CRISTIANINI and TAYLOR 2000] CRISTIANINI, N. and TAYLOR, J. S. (2000). *An introduction to Support Vector Machines and other Kernel-based learning methods*. Cambridge University Press.

[DING and DUBCHAK 2001] DING, C. and DUBCHAK, I. (2001). Multi-class protein fold recognition using support vector machines and other kernel-based learning methods. *Cambridge University Press*, 17(4):349–358.

[DUDA et al. 2001] DUDA, R. O., HART, P. E., and STORK, D. G. (2001). *Pattern Classification*. Wiley Interscience:2.ed.

[GUIMARÃES and MELO 2003] GUIMARÃES, K. and MELO, J. (2003). Uma introdução à análise de seqüências e estruturas biológicas. *Pernambuco*.

[GUO et al. 2004] GUO, J., CHEN, H., SUN, Z., and LIN, Y. (2004). A novel method for protein secondary structure prediction using dual-layer svm and profiles. *Proteins Structure Function and Bioinformatics*, 54(4):738–743.

[HAYKIN 2001] HAYKIN, S. (2001). *Redes Neurais:Princípios e Prática*. Bookman:2.ed.

[HIGGINS and TAYLOR 2000] HIGGINS, D. and TAYLOR, W. (2000). *Bioinformatics: Sequence, Structure, and Databanks: a Practical Approach* Oxford University Press.

[HUBBARD et al. 2000] HUBBARD, T., MURZIN, A., BRENER, S., and C.CHOTIA (2000). Scop:a structural classification of protein data base.*Nucleic Acids Research*, pages 52 236–239.

[ISIK et al. 2004] ISIK, Z., YANIKOGLU, B., and SEZERMAN, U. (2004). Protein structural class determination using support vector machines. *Lecture notes in computer science*, 3280:82–89.

[JOACHIMS et al. 1999] JOACHIMS, T., SCHÖLKOPF, B., BURGES, C., and SMOLA, A. (1999). Making large-scale svm learning practical. advances in kernel methods - support vector learning. *Advances in kernel methods: support vector learning table of contents*, pages 169–184.

[JONSSON et al. 2000] JONSSON, K., KITTLER, J., and MATAS, J. (2000). Learning support vectors for face authentication:sensitivity to mis-registrations. *Fourth IEEE International Conference*.

[KNELLER et al. 1990] KNELLER, D., COHEN, F., and LANGRIDGE, R. (1990). Improvements in protein secondary structure prediction by an enhanced neural network. *J Mol Biol*, 214(1):171–82.

[LEHNINGER 2000] LEHNINGER, A. (2000). *Principles of Biochemistry*. M.Cox.worth.

[LUEMBERGUER 1973] LUEMBERGUER, D. (1973). *Intruduction to Linear and nonlinear programming*. Addison-Wesley Publishing Company.

[MAMITSUKA and YAMANISCHI 1995] MAMITSUKA, H. and YAMANISCHI, K. (1995). α-helix region prediction with stochastic rule learning. *Bioinformatics*, 11:399–411.

[MAY and BLUNDELL 1994] MAY, A. and BLUNDELL, T. (1994). Automated comparative modelling of protein structures. *Curr Opin Biotechnol*, 5(4):355–60.

[MERCIER and LENNON 2003] MERCIER, G. and LENNON, M. (2003). Support vector machines for hyperspectral image classification with spectral-based kernels. *Geoscience and Remote Sensing Symposium, 2003. IGARSS'03. Proceedings. 2003 IEEE International*, 1:288–290.

[METFESSEL and SAURUGGER 1993] METFESSEL, B. and SAURUGGER, P. (1993). Pattern recognition in the prediction of protein structural class. *26th Hawaii Intternational Conference on System Sciences*, 1:679–688.

[MITCHELL 1997] MITCHELL, T. (1997). *Machine Learning*. McGraw Hill:1.ed., New York.

[MUGGLETON et al. 1993] MUGGLETON, S., KING, R., and STERNBERG, M. (1993). Protein secondary structure prediction using logic-based machine learning. *Protein Engineering Design and Selection*, 6(5):549.

[MURZIM et al. 1995] MURZIM, A., BRENNER, S., HUBBARD, T., and CHOTHIA, C. (1995). Scop: a structural classification of proteins database for the investigation of sequences and structures. *J. Mol. Biol*, 247(4):536–540.

[PAVLIDIS et al. 2004] PAVLIDIS, P., WAPINSKI, I., and NOBLE, W. S. (2004). Support vector machine classification on the web. *Bioinformatics*, 20:4.

[PETERSEM et al. 2000] PETERSEM, T., LUNDGAARD, C., NIELSEN, M., and BOHR, H. (2000). Prediction of protein secondary structure at 80 percent accuracy. *Proteins Structure Function and Genetics*, 41(1):17–20.

[POLLASTRI et al. 2002] POLLASTRI, G., PRZYBYLSKI, D., ROST, B., and BALDI, P. (2002). Improving the prediction of protein secondary structure in three and eight classes using recurrent neural networks and profiles. *Proteins Structure Function and Genetics*, 47(2):228–235.

[PONTIL and VERRI 1997] PONTIL, M. and VERRI, A. (1997). Properties of support vector machines. *ai-publications*.

[QIAN and SEJNOWSKI 1988] QIAN, N. and SEJNOWSKI, T. J. (1988). Predicting the secondary structure of globular proteins using neural network models. *J. Mol. Biol*, 202(4):865–884.

[RIIS and KROGH 1996] RIIS, S. K. and KROGH, A. (1996). Improving prediction of protein secondary structureusing structured neural networks and multiple sequence alignments. *Journal of Computational Biology*, 3(1):163–184.

[ROBERTIS and ROBERTIS-Jr. 1993] ROBERTIS, E. and ROBERTIS-Jr., E. (1993). *Bases da Biologia Celular e Molecular*. Guanabara Koogan s.a:2.ed., Rio de Janeiro.

[ROST 2001] ROST, B. (2001). Review:protein secondary structure prediction continues to rise. *J. Struct. Biol*, 134:204–218.

[ROST and SANDER 1994] ROST, B. and SANDER, C. (1994). Combining evolutionary information and neural networks to predict protein secondary structure. *Proteins: Struct. Funct. Genet*, 19:55–72.

[ROST and SANDER 1996] ROST, B. and SANDER, C. (1996). Bridging the protein sequence-structure gap by structure predictions. *Annual Review of Biophysics and Biomolecular Structure*, 25(1):113–136.

[SANDER and SCHNEIDER 1991] SANDER, C. and SCHNEIDER, R. (1991). Database of homology-derived structures and the structural meaning of sequence alignment. *Proteins: Structure, Function & Genetics*, 9(1):56–68.

[SANTOS 2002] SANTOS, E. (2002). Teoria e aplicação de support vector machines à aprendizagem e reconhecimento de objetos baseados na aparência. Master's thesis, Campina Grande.

[SCHLICK 2002] SCHLICK, T. (2002). *Molecular Modeling and Simulation: An Interdisciplinary Guide*. Springer-Verlag New York, Inc. Secaucus, NJ, USA.

[SCHÖLKOPF and SMOLA 2002] SCHÖLKOPF, B. and SMOLA, A. J. (2002). *Learning With Kernels*. MIT press, Cambridge:1.ed., Massachusetts, London, England.

[SCHULZ and SCHIRMER 1979] SCHULZ, G. and SCHIRMER, R. (1979). *Principles of Proteins Structure*. Springer-Verlag New York.

[SEMOLINI 2003] SEMOLINI, R. (2003). Support vector machines,inferência transdutiva e o problema de classificação. Master's thesis, Universidade Estadual de Campinas, Campinas,São Paulo.

[SETÚBAL and MEIDANIS 1997] SETÚBAL, J. and MEIDANIS, J. (1997). *Introduction to Computational Molecular Biology*. PWS Publishing Company.

[SOUTO et al. 2003] SOUTO, M., LORENA, A., DELBEM, A., and CARVALHO, A. (2003). Técnicas de aprendizado de máquina para problemas de biologia molecular. *Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo-São Carlos*.

[THOMAS and DILL 1996] THOMAS, P. and DILL, K. (1996). An iterative method for extracting energy-like quantities from protein structures. *Proc Natl Acad Sci*, 93:11628–11633.

[VAPNIK 1998] VAPNIK, V. (1998). *Statistical Learning Theory*. Springer Werlang:2.ed., New York.

[WANG et al. 2004] WANG, L.-H., LI, Y.-F., LIU, J., and ZHOU, H.-B. (2004). Predicting protein secondary structure by a support vector machine based on a new coding scheme. *Genome Inform Ser Workshop Genome Inform*, 15(2):181–90.

[WANG 2002] WANG, Y. (2002). Aplication of support vector machines in bioinformatics. Master's thesis, National Taiwan University, Taiwan.

[WANNMACHER and DIAS 1976] WANNMACHER, C. and DIAS, R. (1976). *Bioquímica Fundamental*. Graphé:3.ed., Porto Alegre.

[WODAK and ROOMAN 1993] WODAK, S. and ROOMAN, M. (1993). Generating and testing protein folds. *Curr. Opin. Struct. Biol*, 3:247–259.

[ZAKI et al. 2005] ZAKI, N. M., DERIS, S., and ILLIAS, R. (2005). Application of string kernels in protein sequence classification. *Appl Bioinformatics*, 4(1):45–52.