

TESE

1184

UNIVERSIDADE FEDERAL DE ITAJUBÁ

TESE DE DOUTORADO

MAQUINAS DE APRENDIZADO E EXTRAÇÃO DE ATRIBUTOS
PARA A ANÁLISE DE ESTABILIDADE TRANSITÓRIA
DE SISTEMAS DE POTÊNCIA

LUCIANO DE SOUZA MOULIN

ITAJUBÁ - 2002

CLASS. 621.316 (043.2)

CUTTER. M 926 m

TOMBO. 1184

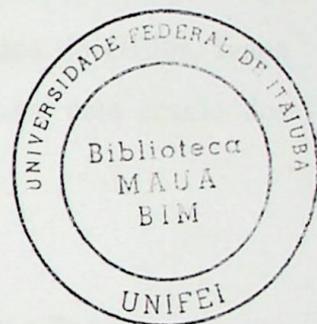
UNIVERSIDADE FEDERAL DE ITAJUBÁ

INSTITUTO DE ENGENHARIA ELÉTRICA

Tese de Doutorado:

**Máquinas de Aprendizado e Extração de
Atributos para a Análise de Estabilidade
Transitória de Sistemas de Potência**

Luciano de Souza Moulin



Orientador: Alexandre P. Alves da Silva

Co-Orientador: Mohamed A. El-Sharkawi

Itajubá - MG

Julho / 2002

Agradecimentos

Agradeço aos meus orientadores, Prof. Alexandre P. Alves da Silva e Prof. Mohamed A. El-Sharkawi, sem os quais a realização deste trabalho não teria sido possível.

Agradeço aos professores Mohamed A. El-Sharkawi e Robert J. Marks II, da Universidade de Washington, EUA, por terem me recebido como aluno visitante no CIA Lab (*Computational Intelligence Applications Lab*).

Agradeço aos professores do GESis (Grupo de Engenharia de Sistemas, UNIFED), J. W. Marangon Lima, Pedro Paulo C. Mendes, Cláudio Ferreira, e aos alunos do GESis, Ivana Campos Honório, Júlio César Lemes Pinto e Leonardo Vinícius Gomes, pela cessão dos resultados dos estudos de estabilidade do projeto Priba / ONS.

Agradeço aos professores Armando M. Leite da Silva e Germano Lambert Torres, por terem me apoiado em momentos importantes do curso de doutorado.

Agradeço aos membros da banca examinadora, Prof. Djalma M. Falcão, Prof. Carlos E. Pedreira, Prof. Armando M. Leite da Silva, Prof. Antônio C. Zambroni de Souza e Prof. Alexandre P. Alves da Silva, pelas críticas e sugestões.

Agradeço à minha família.

Resumo

A abordagem que se baseia em Redes Neurais aplicadas à Análise de Estabilidade Transitória de Sistemas de Potência tem sido apresentada como uma ferramenta bastante promissora para aplicações *on-line*. As grandes dimensionalidades decorrentes da própria natureza dos Sistemas de Potência tornam necessária a implementação de técnicas de Extração de Atributos para que a aplicação seja factível. Ao mesmo tempo, a Extração de Atributos oferece informações de sensibilidade que podem ser utilizadas para identificar os atributos de entrada mais apropriados para ações de controle.

Esta tese apresenta algumas técnicas de Extração de Atributos, além de uma novo tipo de máquina de aprendizado não-linear, as Máquinas de Vetores Suporte, mostrando sua aplicabilidade à Análise de Estabilidade Transitória de Sistemas de Potências. Esse novo modelo neural pode ser visto como uma abordagem diferente ao problema de altas dimensionalidades, devido a seu treinamento relativamente rápido, que pode ser combinado com as técnicas de Extração de Atributos.

As motivações teóricas a respeito das Máquinas de Vetores Suporte são explicadas conceitualmente neste trabalho. Também são utilizados modelos *Perceptrons* de Múltiplas Camadas e os dois modelos são aplicados à Análise de Estabilidade Transitória de dois Sistemas de Potência, o IEEE 50 Máquinas (145 barras) e o subsistema brasileiro Priba (2684 barras).

É demonstrada a aplicabilidade de tais idéias aos sistemas de potência, notadamente de grande porte (Sistema Priba) e são discutidos aspectos de adequação dos modelos, tempo de treinamento e precisão dos resultados.

Abstract

The Neural Networks based Power System Transient Stability Analysis has been presented as a promising tool for on-line applications. The large dimensionalities of the Power Systems turn it necessary to implement Feature Extraction techniques to make the application feasible in practice. At the same time, Feature Extraction offers sensitivity information which can help to identify the input features best suited for control action.

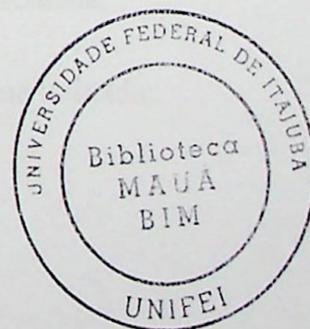
This thesis presents some Feature Extraction techniques, and a new type of learning machine, the Support Vector Machines, showing its suitability for the Power Systems Transient Stability Analysis. This new Neural Network model can be seen as a different approach to cope with the high dimensionality, due to its fast training capability, which can be combined with the Feature Extraction.

The theoretical motivation about Support Vector Machines is conceptually explained in this work. Multilayer Perceptrons are also used and both models are applied to the Transient Stability Analysis of two Power Systems, the IEEE 50 Machines (145 buses) system and the Brazilian Priba (2484 buses).

The feasibility of such ideas for Power Systems is demonstrated, notably for large scale ones (Priba System), and aspects of model adequacy, training time and classification accuracy are discussed.

SIMBOLOGIA	I
LISTA DE SIGLAS	IV
LISTA DE FIGURAS.....	VI
LISTA DE TABELAS	VIII
CAPÍTULO 1 – INTRODUÇÃO.....	1
1.1 – ANÁLISE DE ESTABILIDADE TRANSITÓRIA DE SISTEMAS DE POTÊNCIA.....	1
1.2 – REDES NEURAIS PARA A ANÁLISE DE ESTABILIDADE TRANSITÓRIA.....	3
1.3 – CONTRIBUIÇÕES / OBJETIVOS	8
CAPÍTULO 2 – ANÁLISE DE ESTABILIDADE TRANSITÓRIA DE SISTEMAS DE POTÊNCIA VIA REDES NEURAIS	10
2.1 – GERAÇÃO DE CONJUNTOS DE TREINAMENTO	13
2.2 – ATRIBUTOS DE ENTRADA E SAÍDA.....	16
2.3 – CONSIDERAÇÕES FINAIS	19
CAPÍTULO 3 – EXTRAÇÃO DE ATRIBUTOS.....	22
3.1 – EXTRAÇÃO DE COMPONENTES PRINCIPAIS OU TRANSFORMAÇÃO DE KARHUNEN-LOÈVE.....	24
3.2 – FUNÇÕES DE AVALIAÇÃO	27
3.3 – MÉTODOS DE BUSCA	35
3.4 – OUTROS MÉTODOS DE EXTRAÇÃO DE ATRIBUTOS	41
3.5 – CONSIDERAÇÕES FINAIS	42
CAPÍTULO 4 – MÁQUINAS DE VETORES SUPORTE	44
4.1 – TEORIA DO APRENDIZADO ESTATÍSTICO (<i>STATISTICAL LEARNING THEORY</i>)	46
4.2 – MÁQUINAS DE VETORES SUPORTE LINEARES – CASO SEPARÁVEL.....	52
4.3 – MÁQUINAS DE VETORES SUPORTE LINEARES – CASO NÃO SEPARÁVEL.....	57
4.4 – MÁQUINAS DE VETORES SUPORTE NÃO LINEARES	59
4.5 – CONSIDERAÇÕES FINAIS	64

CAPÍTULO 5 – METODOLOGIA E APLICAÇÃO	66
5.1 – MODELOS NEURAIIS	66
5.1.1 – <i>Perceptrons Multicamadas</i>	66
5.1.2 - <i>Redes de Vetores Suporte</i>	68
5.2 – TÉCNICAS AUTOMÁTICAS DE EXTRAÇÃO DE ATRIBUTOS.....	69
5.2.1 - <i>Busca Sequencial</i>	70
5.2.2 - <i>Extração de Componentes Principais</i>	70
5.2.3 - <i>Algoritmos Genéticos</i>	71
5.3 – RESULTADOS	73
5.3.1. - <i>Sistema IEEE 50 Máquinas</i>	73
5.3.2 - <i>Sistema Priba</i>	81
CAPÍTULO 6 – CONCLUSÕES.....	89
REFERÊNCIAS.....	93
ANEXO I – REDES NEURAIIS ARTIFICIAIS	104
I.1 – PERCEPTRONS MULTICAMADAS	105
I.2 – ALGORITMO RETROPROPAGAÇÃO DE ERROS.....	108
ANEXO II – ANÁLISE DE ESTABILIDADE DINÂMICA DE SISTEMAS ELÉTRICOS DE POTÊNCIA	114
II.1 – ANÁLISE DE ESTABILIDADE DE PEQUENAS PERTURBAÇÕES	117
II.2 – ANÁLISE DE ESTABILIDADE DE TENSÃO	119
II.3 – ANÁLISE DE ESTABILIDADE TRANSITÓRIA	122
II.3.1 – <i>Representação do Sistema de Potência</i>	123
II.3.2 – <i>Análise de Estabilidade Transitória Através de Simulação no Domínio do Tempo</i>	125
II.3.3 – <i>Análise de Estabilidade Transitória Através do Método Direto</i>	128
ANEXO III – TABELAS DE RESULTADOS DO SISTEMA PRIBA	134



Simbologia

- H → entropia;
- \mathbf{x} → vetor de entrada;
- n → dimensão do espaço que contém o vetor de entrada;
- $p(\mathbf{x})$ → função densidade de probabilidade do vetor de entrada;
- T → conjunto de treinamento;
- R^n → espaço real de dimensão n ;
- y → medida de estabilidade referente a \mathbf{x} ;
- $F(A, \mathbf{x})$ → função de transformação da Extração de Atributos;
- f → vetor resultante da Extração de Atributos;
- d → dimensão do espaço que contém f ou dimensão do espaço reduzido;
- β → vetor de parâmetros de uma função classificadora ou máquina de aprendizado;
- $f(\mathbf{x}) = f(\beta, \mathbf{x})$ → função classificadora ou máquina de aprendizado;
- $R(\beta)$ → risco esperado ou risco verdadeiro de uma máquina treinada;
- $R_{emp}(\beta)$ → risco empírico ou erro de treinamento de uma máquina treinada;
- N → tamanho do conjunto de treinamento;
- η → medida de confiança do limite do risco verdadeiro;
- h → dimensão VC;

- w → vetor do hiperplano classificador da MVS;
- b → escalar do hiperplano classificador da MVS;
- ρ → margem de separação do hiperplano classificador da MVS;
- D → bola que contém os vetores de entrada de treinamento;
- α → vetor de multiplicadores de Lagrange;
- α_i → multiplicadores de Lagrange ou componentes de α ;
- $L(w, b, \alpha)$ → Lagrangiano;
- $\tau(w)$ → função objetivo da otimização da MVS, caso separável;
- $V(w)$ → função objetivo da otimização da MVS, caso não separável;
- $W(\alpha)$ → função objetivo da otimização dual da MVS;
- C → fator de penalização da função objetivo da otimização quadrática da MVS, caso não separável;
- ε_i → variáveis de relaxação;
- Φ → vetor resultante da transformação não-linear para o espaço expandido;
- ϕ_i → componentes de Φ ;
- $K(a, b)$ → função *Kernel* de Mercer;
- p → parâmetro da função *Kernel* polinomial;
- σ → parâmetro da função *Kernel* RBF Gaussiana;
- t → operador transposição de matriz ou vetor;
- v → vetor de projeção da função Discriminante de Fisher;
- I → informação mútua;
- FD → função Discriminante de Fisher;

$p(\mathbf{x}|C_i)$ → função densidade de probabilidade do vetor de entrada, \mathbf{x} , condicional à classe C_i ;

$\|\bullet\|$ → norma ou módulo de um vetor;

Siglas

AFT → Análise de Fatiabilidade Transitória;

AG(s) → Algoritmo(s) Genético(s);

BRS → Branch and Bound;

BS → Busca Sequencial;

CCT → Control Clearing Time;

EA → Extração de Atributos;

ECP → Extração de Componentes Principais;

EEF → Função de Energia Transiente;

IM → Informação Mútua;

LTs → Listas de Transições;

MRE → Minimização do Risco Estrutural;

MVS(s) → Método(s) de Vetores Superiores;

MLP(s) → Perceptron(s) Multicamadas;

PrGA → Perceptron Genético;

RN → Rede Neural;

RNs → Redes Neurais;

RVS(s) → Rede(s) de Vetores Superiores;

SA → Solução de Atributos;

SDT → Simulação no Domínio do Tempo;

AET	– Análise de Estabilidade Transitória;
AG(s)	– Algoritmo(s) Genético(s);
B&B	– <i>Branch and Bound</i> ;
BS	– Busca Sequencial;
CCT	– <i>Critical Clearing Time</i> ;
EA	– Extração de Atributos;
ECP	– Extração de Componentes Principais;
FET	– Função de Energia Transitória;
IM	– Informação Mútua;
LTs	– Linhas de Transmissão;
MRE	– Minimização do Risco Estrutural;
MVS(s)	– Máquina(s) de Vetores Suporte;
MLP(s)	– <i>Perceptron(s)</i> Multicamadas;
Priba	– Paranaíba-Grande;
RN	– Rede Neural;
RNs	– Redes Neurais;
RVS(s)	– Rede(s) de Vetores Suporte;
SA	– Seleção de Atributos;
SDT	– Simulação no Domínio do Tempo;

- SF – Sequencial para Frente;
- ST – Sequencial para Trás;
- VC – Vapnik-Chervonenski.

Lista de Figuras

- 3.1 – Exemplo de observações de um sistema conjunto de dados, através de observações de um processo Markov
- 3.2 – Exemplo de uma distribuição sobre um espaço com uma matriz de covariância
- 3.3 – Linha geométrica comparativa entre os métodos de Bayes e de Cárter
- 3.4 – Exemplo de um teste de Bayes e de Cárter
- 3.5 – Organização hierárquica de uma máquina de Bayes para seleção de atributos
- 3.6 – Organização da estrutura de um teste de Bayes e de Cárter
- 4.1 – Conteúdo do treinamento
- 4.2 – Exemplo de uma máquina de Bayes e de Cárter
- 4.3 – 4.4 Exemplos de treinamento organizado em duas classes
- 4.4 – Comparação VC e C
- 4.5 – Limites superiores e valores verdadeiros de $R(\hat{f})$
- 4.6 – Comparação de treinamento separável e hipotético de separação
- 4.7 – Máquina de Vectors Suportadores – caso separável
- 4.8 – Exemplos de treinamento não separável
- 4.9 – Comparação de treinamento não separável
- 4.10 – Máquina de Vectors Suportadores não linear
- 4.11 – MVS não linear com ponto de apoio

Lista de Figuras

- 3.1 – DUAS REPRESENTAÇÕES DE UM MESMO CONJUNTO DE DADOS, ATRAVÉS DOS ATRIBUTOS
(A) (X_1, X_2) E (B) (X_1, X_3)
- 3.2 – EXEMPLO DE DUAS DISTRIBUIÇÕES COM MESMA MATRIZ DE COVARIÂNCIA
- 3.3 – LUGAR GEOMÉTRICO COMPARATIVO ENTRE DISTÂNCIAS EUCLIDIANA E DE CHEBYSHEV
- 3.4 – EXEMPLO DO MÉTODO *BRANCH & BOUND*
- 3.5 – CROMOSSOMO REPRESENTATIVO DE UMA SOLUÇÃO CANDIDATA PARA SELEÇÃO DE ATRIBUTOS
- 3.6 – CRUZAMENTO EM UM PONTO: (A) ANTES E (B) DEPOIS DO CRUZAMENTO
- 4.1 – CONJUNTO DE TREINAMENTO
- 4.2 – EXEMPLO DE RETAS ORIENTADAS CLASSIFICADORAS
- 4.3 – 4 EXEMPLOS DE TREINAMENTO ORGANIZADOS EM DUAS CLASSES
- 4.4 – CONFIANÇA $VC \propto 1/N$
- 4.5 – LIMITES SUPERIORES E VALORES VERDADEIROS DE $R(\beta)$
- 4.6 – CONJUNTO DE TREINAMENTO SEPARÁVEL E HIPERPLANO DE SEPARAÇÃO
- 4.7 – MÁQUINA DE VETORES SUPORTE LINEAR – CASO SEPARÁVEL
- 4.8 – EXEMPLOS DE TREINAMENTO NÃO SEPARÁVEIS
- 4.9 – CONJUNTO DE TREINAMENTO NÃO SEPARÁVEL
- 4.10 – MÁQUINA DE VETORES SUPORTE NÃO LINEAR
- 4.11 – MVS NÃO LINEAR COM FUNÇÃO *KERNEL*

4.12 – AVALIAÇÃO DO LIMITE DO RISCO DE MVS

5.1 – CURVA ROC DA MVS RBF GAUSSIANO – IEEE 50 MÁQUINAS

5.2 – CURVA ROC DA MVS POLINOMIAL – IEEE 50 MÁQUINAS

5.3 – CURVA ROC DA MVS RBF GAUSSIANO – PRIBA

5.4 – CURVA ROC DA MVS POLINOMIAL – PRIBA

5.1 – CLASSIFICAÇÃO DE ESTABILIDADE TRANSITÓRIA DO SISTEMA IEEE 50 MÁQUINAS
CONJUNTO COMPLETO DE ATRIBUTOS

5.2 – CLASSIFICAÇÃO DE ESTABILIDADE TRANSITÓRIA DO SISTEMA IEEE 50 MÁQUINAS
CONJUNTO DE ESTABILIDADE DE ATRIBUTOS

5.3 – CLASSIFICAÇÃO DE ESTABILIDADE TRANSITÓRIA DO SISTEMA PRIBA – CONJUNTO
COMPLETO DE ATRIBUTOS

5.4 – MELHORES RESULTADOS DO SISTEMA PRIBA EM TERMOS DE DISPENSAS PALHAS

5.5 – MELHORES RESULTADOS DO SISTEMA PRIBA EM TERMOS DE TARA DE SERRA

5.6 – MELHORES RESULTADOS DO SISTEMA PRIBA EM TERMOS DE TEMPO DE TRABALHEIRO

III.1 – BUSCA SEQUENCIAL 1

III.2 – BUSCA SEQUENCIAL 2

III.3 – EXTRAÇÃO DE COMPONENTES PRINCIPAIS 1

III.4 – EXTRAÇÃO DE COMPONENTES PRINCIPAIS 2

III.5 – ALGORITMOS GENÉTICOS 1

III.6 – ALGORITMOS GENÉTICOS 2

Lista de Tabelas

- 3.1 – EXEMPLO ILUSTRATIVO DO OPERADOR SELEÇÃO
- 5.1 – CLASSIFICAÇÕES DE ESTABILIDADE TRANSITÓRIA DO SISTEMA IEEE 50 MÁQUINAS – CONJUNTO COMPLETO DE ATRIBUTOS
- 5.2 – CLASSIFICAÇÕES DE ESTABILIDADE TRANSITÓRIA DO SISTEMA IEEE 50 MÁQUINAS – CONJUNTOS REDUZIDOS DE ATRIBUTOS
- 5.3 – CLASSIFICAÇÕES DE ESTABILIDADE TRANSITÓRIA DO SISTEMA PRIBA – CONJUNTO COMPLETO DE ATRIBUTOS
- 5.4 – MELHORES RESULTADOS DO SISTEMA PRIBA EM TERMOS DE DISPENSAS FALSAS
- 5.5 – MELHORES RESULTADOS DO SISTEMA PRIBA EM TERMOS DE TAXA DE ERRO
- 5.6 – MELHORES RESULTADOS DO SISTEMA PRIBA EM TERMOS DE TEMPO DE TREINAMENTO
- III.1 – BUSCA SEQUENCIAL 1
- III.2 – BUSCA SEQUENCIAL 2
- III.3 – EXTRAÇÃO DE COMPONENTES PRINCIPAIS 1
- III.4 – EXTRAÇÃO DE COMPONENTES PRINCIPAIS 2
- III.5 – ALGORITMOS GENÉTICOS 1
- III.6 – ALGORITMOS GENÉTICOS 2

1.1 – Análise de Estabilidade Transitória de Sistemas de Potência

Os sistemas de potência de países industrializados como o Brasil são considerados de alta prioridade para a segurança da população e para a atividade econômica em geral. Essa preocupação é aumentada pela tendência de desregulamentação do setor, que deve aumentar a liberdade e acessibilidade dos agentes, ao mesmo tempo diminuindo a possibilidade de controle centralizado. A Análise de Estabilidade Transitória (AET) estuda a capacidade do sistema de potência em fornecer serviço e manter sincronismo das máquinas em caso de ocorrência de alguma contingência. Pelas condições citadas, uma AET mais exata tem se tornado imprescindível. No passado, as companhias elétricas podiam justificar melhorias na operação do sistema ou na infra-estrutura baseado em métodos aproximados de análise de segurança. No ambiente desregulamentado, essa situação tende a mudar.

A avaliação do estado de segurança do sistema exigiria a simulação de todas as contingências em tempo real, mas isso obviamente torna-se proibitivo. Por isso, métodos rápidos e aproximados são utilizados pelas companhias, assumindo um subconjunto limitado das informações mais críticas.

O método mais tradicionalmente utilizado para AET é a Simulação no Domínio do Tempo (SDT), em que as equações diferenciais não lineares do sistema, que representam a dinâmica das máquinas síncronas e de outros equipamentos de interesse, e as equações algébricas do fluxo de potência, que representam a rede, são resolvidas passo a passo e de maneira alternada através de métodos numéricos. Os valores das variáveis de interesse do sistema podem ser calculados em cada passo.

Uma outra abordagem que tenta resolver o problema do elevado tempo computacional necessário para se calcular o estado de segurança do sistema, é a dos métodos diretos, baseados no critério de estabilidade de Lyapunov [Fouad, 1992], [Pai, 1989]. Essa abordagem começou

a mostrar-se como alternativa promissora em meados da década de 70, indicando a possibilidade de aplicação para sistemas de grande porte [El-Abiad, 1966], [Saito, 1975]. Ela utiliza uma Função de Energia Transitória (FET) como função de Lyapunov para avaliar a condição de segurança do sistema após a ocorrência de algum distúrbio. Isso é feito pela comparação entre o valor da FET no momento do isolamento do distúrbio (energia transitória), V_{cl} , e o valor crítico da mesma, V_{cr} . A energia transitória é a quantidade total de energia injetada no sistema pelo distúrbio. O valor crítico corresponde à máxima quantidade de energia que o sistema é capaz de absorver, para determinado ponto de operação, e ainda manter a estabilidade. A chamada margem de energia transitória é calculada pela diferença entre esses dois valores, $V_{cr} - V_{cl}$. Se for positiva, significa que o sistema irá suportar o impacto e voltará ao regime permanente em estado seguro. Se for negativa, medidas corretivas/preventivas deverão ser tomadas para que o sistema não entre em colapso. Além disso, uma estimativa do grau de estabilidade (instabilidade) é obtido com a magnitude da margem de energia transitória.

Apesar dos grandes avanços conseguidos no desenvolvimento dos métodos diretos, os problemas ainda existentes incluem a dificuldade de modelagem detalhada do sistema e a vulnerabilidade a problemas numéricos. Consequentemente, as aproximações inerentes levam a resultados pouco confiáveis e inadequados para o uso prático. A solução dos problemas numéricos pode exigir um esforço computacional tão grande, que o tempo de processamento torna-se equivalente ao das Simulações no Domínio do Tempo. Os métodos diretos podem ser usados para o ranqueamento de contingências, onde a necessidade de exatidão nos resultados é menos crítica.

A necessidade de realizar a análise de estabilidade transitória de maneira *on-line* ou em tempo quase real é largamente reconhecida. Essa necessidade torna-se premente também devido à pressão econômica, que tende a adiar ainda mais a construção de novas plantas e itens de infra-estrutura da rede. O sistema é forçado a operar mais perto dos limites de segurança, e métodos conservadores que antes podiam ser utilizados em regiões distantes dos limites de estabilidade tornam-se inaceitáveis [Mansour, 2000], [Sterngold, 2001]. Por isso, muitas pesquisas têm sido realizadas em todo o mundo para sintetizar a análise *on-line* de maneira prática. Não há ainda consenso quanto a qual esquema seria o melhor, mas em geral

pode-se vislumbrar algumas tendências nas propostas até então realizadas para sistemas reais de grande porte:

a) Utilizar funções de energia ou sistemas inteligentes para ranquear inicialmente as contingências e/ou identificar as menos severas. Esses métodos são tomados como avaliadores aproximados, que podem ser utilizados numa estimativa prévia. Logo em seguida, simulações detalhadas são executadas para avaliar as contingências consideradas mais severas [Demaree, 1994], [Chadalavada,1997], [Chiang,1999], [Brandwajn,1997], [Morison,1999], [Aresi,1999], [Mansour,1997a], [Mansour,1997b], [Rovnyak,1994], [Wehenkel,1994], [Wehenkel,1989];

b) Utilizar funções de energia ou sistemas inteligentes para avaliar e interromper as simulações dos casos considerados estáveis durante o processo de simulação no domínio do tempo. Neste caso as técnicas de funções de energia ou sistemas inteligentes são executados simultaneamente às simulações de estabilidade, e as simulações dos casos considerados estáveis são descartadas para economizar esforço computacional [Scala,1996], [Kamwa,2001];

c) Aproveitar a alta capacidade de sistemas computacionais em paralelo e fazer simulações detalhadas de todos os casos [Chadalava,1997], [Scala,1996], [Alvarado,1979], [Decker,1992], [Taoka,1992];

Algumas combinações entre os quatro itens anteriores também são possíveis. O item (c) pode ser utilizado em qualquer esquema híbrido que utilize as simulações no domínio do tempo.

1.2 – Redes Neurais para a Análise de Estabilidade Transitória

Especificamente nos itens (a) e (b) da Seção 1.1 em que sistemas inteligentes são considerados como ferramenta de AET, as Redes Neurais (RNs) possuem uma forte participação: A primeira aplicação de RNs para a AET foi proposta em 1989 [Sobajic,1989]. Desde então, os classificadores neurais têm sido estudados como proposta alternativa a ser integrada à AET de sistemas de potência de grande porte e em tempo quase real, em esquemas

como indicado nos itens (a) e (b) anteriores [Mansour,1997a], [Mansour,1997b], [El-Sharkawi,1996a], [El-Sharkawi,1996b], [Kumar,1991], [Omata,1993].

A utilização de uma RN exige uma fase de treinamento em que dados de contingências são aproveitados para geração de conjuntos de treinamento e teste. Podem ser considerados dados históricos com situações plausíveis de operação, com variados níveis de carga, diferentes alternativas de comissionamento e despacho de potência.

O objetivo final de um classificador neural é estimar a estabilidade de qualquer novo padrão ou ponto de operação corretamente sem precisar executar avaliações de contingência demoradas. É possível também obter uma estimativa de quão estável (instável) é o caso. Cabe mencionar que essa abordagem não impõe restrições de topologia do sistema, nem restrições operacionais [Sobajic,1989].

As principais vantagens da utilização dessa abordagem para a análise de estabilidade são:

- Necessidade de poucos dados para a análise, comparado às técnicas tradicionais;
- Possibilidade de controle em tempo real;
- O esforço computacional é transferido para a etapa de treinamento. Depois de treinadas, as RNs podem oferecer respostas em tempo desprezível;
- Evita-se a análise através de condições de operação aproximadas. Com a capacidade de generalização inerente às RNs, é possível obter estimativas de estabilidade em tempo real, referentes aos pontos de operação realizados de fato;
- Respostas simples são geradas, como: estável ou instável, com a possibilidade de estimativa de margens de estabilidade e regiões vulneráveis de operação.

A intensa atividade de pesquisa na área de Reconhecimento de Padrões, e especificamente em RNs, também contribuiu para revelar impedimentos que dificultam a maior aceitação de tais modelos como sistemas eficazes e confiáveis para as diversas aplicações possíveis da atualidade. Considerando as RNs como modelos não-lineares, apesar de serem capazes de produzir uma representação compacta da função a ser aproximada, eles

sofrem de duas sérias dificuldades: (1) inexistência de medidas de complexidade simples de se calcular e (2) inexistência de solução global ótima [Cherkassky,1998]. A importância da dificuldade (1) parte da necessidade de obter-se uma maneira de avaliar a capacidade de generalização desses estimadores. Problemas clássicos como a Maldição da Dimensionalidade¹ (*Curse of Dimensionality*), lentidão do treinamento, sensibilidade à inicialização e à parada do treinamento, convergência para mínimos locais, característica de caixa preta e a elevada carga de empirismo inerente ao desenvolvimento de uma RN têm motivado críticas e prejudicado a expansão desta área que apresenta resultados tão promissores.

Portanto, um desafio da aplicação de RNs como máquinas de aprendizado é o esforço necessário para se “acoplar” os dados disponíveis com o tipo e a estrutura da RN (casamento de complexidade). Isso acarreta um grande consumo de tempo em tarefas como análise dos dados, definição da estrutura da RN e o treinamento repetido para variadas estruturas, com diferentes amostras dos dados disponíveis.

O “acoplamento” de uma RN com os dados de treinamento pode ser feito de três maneiras: (1) adaptando-se a RN para que ela ajuste-se ao conjunto de treinamento, (2) modificando-se os dados para que se ajustem à RN e (3) gerando-se dados dinamicamente de uma maneira controlada, em função e ao longo do treinamento da RN. Combinações desses três casos podem ser utilizadas para se “casar” os dados de treinamento com a RN. Os casos (1) e (2) assumem que o conjunto de treinamento é fixo, enquanto o caso (3) assume que os dados de treinamento podem ser gerados à medida que sejam necessários. Nesse último caso, em geral no treinamento supervisionado, há algum custo inerente ao processo de geração de dados, caso contrário seria mais vantajoso utilizar essa fonte de dados diretamente em lugar da RN. Um exemplo do caso (3) é o aprendizado baseado em questionamento (*query-based learning*) [Jensen,1999].

¹ A Maldição da Dimensionalidade determina, como regra aproximada, que o tamanho do conjunto de treinamento para RNs deve crescer exponencialmente com o número de variáveis de entrada.

A Extração de Atributos (EA) é uma abordagem que procura ajustar os dados à RN, sendo os dados transformados em uma forma que seja mais apropriada ao treinamento. Ela aplica transformações lineares ou não-lineares ao vetor de entradas (de dimensão n), podendo levá-lo a um novo espaço vetorial que pode ter dimensão d maior ($d > n$) ou menor ($d < n$) que o espaço original. Normalmente, deseja-se realizar a transformação para um espaço de menor dimensão. Pode-se considerar a Seleção de Atributos (SA) como um caso particular da EA, quando a transformação é realizada escolhendo-se componentes do vetor de entradas original por meio de algum critério de avaliação. Neste caso, o objetivo é escolher o melhor subconjunto de componentes (também chamadas de variáveis ou atributos) capaz de executar a tarefa de classificação ou regressão com uma precisão satisfatória. Na Seleção de Atributos, a dimensão do vetor de entradas resultante é sempre menor ou igual à dimensão do vetor original ($m \leq n$).

Portanto, a Extração de Atributos é capaz de levar à redução da dimensionalidade dos dados de entrada, atacando a Maldição da Dimensionalidade e a lentidão no treinamento de máquinas de aprendizado. A EA talvez seja a parte mais difícil e ao mesmo tempo a menos entendida dos problemas de reconhecimento de padrões. No contexto da análise de estabilidade de sistemas de potência, há a necessidade de se considerar uma grande quantidade de informações contidas em variáveis como: condições de carga/geração pré-distúrbio; dados dinâmicos de máquina; ações de controle pós-distúrbio; tipo e duração do distúrbio; condições de operação.

Um número ilimitado de dados poderia ser utilizado num sistema de potência qualquer. Os avanços da aplicação de RNs ao problema de AET e as dificuldades encontradas levam a crer que a chave para o sucesso dos classificadores neurais é o processo de Extração de Atributos. Para a análise de estabilidade em tempo quase real de sistemas de grande porte, em esquema unificados como mostrado em [Jardim, 2000b] e [Jardim, 2000a], ou esquemas de identificação da borda de segurança como mostrado em [Kassabalidis], que consideram as RNs como imprescindíveis a sua viabilidade, torna-se necessário o desenvolvimento de técnicas de EA automáticas.

Uma nova visão sobre a implementação de métodos e máquinas de aprendizado surgiu com a Teoria do Aprendizado Estatístico (*Statistical Learning Theory*) e com o conceito de

Minimização do Risco Estrutural (*Structural Risk Minimization*) [Vapnik,1998], que consideram a capacidade de generalização das máquinas como função da dimensão VC (*Vapnik-Chervonenski*) das mesmas. A dimensão VC proporciona uma noção sobre a complexidade da máquina e pode ser usada em conjunto com o erro de treinamento empírico (erro de treinamento calculado para um dado conjunto de treinamento) para se avaliar o valor esperado do erro de teste (capacidade de generalização ou valor médio do erro de teste calculado com todos os conjuntos de teste possíveis, obtidos a partir de uma dada função não-linear). Porém, para modelos não-lineares que existem atualmente, dois problemas impedem uma aplicação rigorosa do formalismo inerente à Minimização do Risco Estrutural (MRE). Primeiro, a estimativa da dimensão VC é muito difícil. Segundo, a minimização do erro de treinamento empírico desses modelos leva a uma solução local. Sob essa perspectiva, em aplicações práticas surgiu a necessidade do uso de técnicas heurísticas baseadas na experiência empírica para controlar a complexidade e a capacidade de generalização dos modelos não lineares, em particular das RNs.

A busca de uma estrutura para a qual seja possível estimar a dimensão VC e ao mesmo tempo aplicar a MRE fez surgir e desenvolver-se recentemente um tipo de máquina de aprendizado não-linear chamada de Máquinas de Vetores Suporte (*Support Vector Machines*) [Vapnik,1998]. As Máquinas de Vetores Suporte (MVS) aplicam transformações não-lineares ao vetor de entrada, aumentando a sua dimensão, e em seguida executam uma classificação ou uma regressão linear. Na classificação, por exemplo, classes que normalmente não poderiam ser facilmente separadas no espaço original podem tornar-se separáveis ou mais fáceis de serem discriminadas no novo espaço transformado, de maior dimensão. A classificação linear que se segue usa propriedades geométricas para calcular as superfícies de separação diretamente a partir dos dados de treinamento. Os classificadores MVS têm a sua capacidade de generalização maximizada durante o treinamento, e o seu desenvolvimento teórico não exige a premissa de se conhecer a função densidade de probabilidade da população da qual é tomado o conjunto de treinamento.

As redes neurais do tipo MVS, utilizadas juntamente com a Extração de Atributos, constituem uma proposta de se atacar alguns dos problemas inerentes ao treinamento de RNs e a suas aplicações na AET:

- A sensibilidade à inicialização e a convergência para mínimos locais não existe porque o treinamento de MVS consiste na solução de um problema quadrático, com solução única;
- O entendimento da teoria das MVS permitiu a elaboração de algoritmos de treinamento relativamente rápidos, mesmo com conjuntos de treinamento grandes e altas dimensionalidades de entrada;
- A carga de empirismo é minorada. Há a necessidade da escolha de parâmetros cujo papel pode ser entendido de maneira mais precisa, a partir da teoria que deu origem às MVS.

1.3 – Contribuições / Objetivos

Esta tese procura possibilitar a integração das RNs às ferramentas tradicionais de análise de estabilidade de sistemas de potência, em aplicações práticas. Algumas características dos problemas de estabilidade transitória de sistemas de grande porte, juntamente com características intrínsecas das RNs, acarretam dificuldades e impedimentos para sua aplicação à análise de estabilidade transitória. Esta tese ataca essas dificuldades/impedimentos, apresentando como contribuições e objetivos:

- A proposta de processos automáticos de Extração de Atributos para redes neurais através de técnicas novas ou que generalizam técnicas já existentes, e que permitam resultados sem interferência humana direta. O objetivo é atacar a Maldição da Dimensionalidade e o treinamento lento.
- Introduzir a aplicação de um novo paradigma de máquinas de aprendizado, a Teoria do Aprendizado Estatístico, à AET de sistemas elétricos através de classificadores do tipo MVS. As vantagens apresentadas por esses classificadores suprem algumas dificuldades inerentes à abordagem de AET de sistemas de potência via redes neurais, mais especificamente devido à: (a) inexistência da sensibilidade à inicialização, parada de treinamento e convergência para mínimos locais, (b) utilização de algoritmos de treinamento relativamente rápidos, mesmo com

conjuntos de treinamento grandes e altas dimensionalidades de entrada, (c) maior entendimento do funcionamento teórico de tais classificadores, com redução da carga de empirismo em sua aplicação. A aplicação de MVS à AET permite esclarecer aspectos de sua aplicação prática. As expectativas são de que a redução da dimensionalidade dos dados de entrada seja de menor importância para as MVS do que para RNs do tipo *perceptron* multicamadas, por exemplo, devido ao treinamento rápido. Mas os efeitos da esparsidade na representação dos dados devem ser testados, através da apreciação sobre a exatidão dos resultados em conjuntos completos e reduzidos de atributos.

- Demonstrar a aplicabilidade das técnicas e máquinas de aprendizado aqui descritas à AET de sistemas de grande porte através de simulações e testes com os sistemas IEEE 50 Máquinas e o subsistema brasileiro da bacia Paranaíba-Grande, com 2684 barras. A aplicação a sistemas de grande porte possibilita a inserção das redes neurais em esquemas de AET. O desenvolvimento de processos de Extração de Atributos automáticos que promovam maior compatibilidade entre dados e máquina, e a possibilidade da aplicação das idéias aqui desenvolvidas para a AET de sistemas de potência via RNs proporcionará a solução de um entrave que tem impedido a expansão e consolidação dessa tecnologia nessa importante aplicação.

Análise de Estabilidade Transitória de Sistemas de Potência via Redes Neurais

No processo de Análise de Estabilidade Transitória, cujos detalhes são apresentados no Anexo II, existe uma série de passos padrão, que se inicia com a determinação de um caso base. Este corresponde a uma determinada configuração e condições de operação do sistema para as quais não existam estudos de estabilidade que permitam uma operação segura. Por isso é necessário executar uma análise de estabilidade, que pode ocorrer também antes da aprovação de manutenções agendadas de equipamentos de grande porte.

A partir do caso base é realizado um ranqueamento e uma seleção de contingências utilizadas em simulações detalhadas, geralmente no domínio do tempo, e calcular a evolução das variáveis dinâmicas após a ocorrência do distúrbio. Tais variáveis incluem ângulos dos rotores das máquinas, velocidades dos geradores, amplitudes das variações de potência dos geradores, amplitudes das variações de fluxos, etc. Com esse estudo, é possível calcular limites de operação para o caso base, que serão válidos também para outras condições de operação semelhantes. A partir daí são formuladas ações de controle preventivas e corretivas, como rearranjo de geração e intercâmbios, corte de carga, etc., como resultado dos estudos realizados.

Apesar dos grandes avanços na eficiência do processamento computacional, os métodos tradicionais de Análise de Estabilidade Transitória ainda requerem esforços elevados para calcular índices ou margens de estabilidade de maneira exata e confiável. Isso deve-se à complexidade estrutural e à vasta faixa de operação de sistemas elétricos modernos.

Com a expansão dos sistemas de potência e o aumento de suas complexidades, a análise de estabilidade tornou-se ainda mais crítica e difícil. A atual tendência de desregulamentação do setor e a participação de muitos agentes no mercado de energia podem contribuir para diminuir a margem de estabilidade [Mansour, 2000], [Sterngold, 2001]. Isso

tem levado à investigação de técnicas rápidas e exatas que permitam a AET em tempo quase real. Devido à natureza inflexível da análise de estabilidade *off-line* os limites operacionais geralmente são determinados de maneira bastante conservativa. Essa é uma grande motivação para o desenvolvimento de um procedimento em tempo quase real, de forma que a disponibilidade dos dados em intervalos de tempo curtos possa remover as incertezas referentes às condições de operação, tornando a avaliação mais precisa e confiável [Jardim, 2000b]. Para isso a AET em tempo quase real deve ser rápida o suficiente tal que o ponto de operação possa ser considerado "constante" durante a avaliação. Um ciclo de 3-5 minutos poderia ser considerado satisfatório [Jardim, 2000b].

Em [Kumar, 1991] é apresentada uma visão geral sobre as possibilidades de aplicação de RNs num processo como este e são apontadas como principais potencialidades dessa aplicação, os subproblemas da Análise de Estabilidade que exigem generalização dos resultados, isto é, generalização de casos estudados explicitamente para outras condições operacionais que não podem ser estudadas.

A primeira proposta de aplicação de RNs para a AET foi lançada em 1989 [Sobajic, 1989], quando uma rede neural *perceptron* multicamadas (MLP) treinada via Retropropagação de Erros foi aplicada para prever o Tempo Crítico de Remoção da Falta (*Critical Clearing Time*, CCT) em um sistema de quatro máquinas, para um dado distúrbio. Os autores mostraram que era possível estimar corretamente o CCT com uma mesma RN, mesmo quando diferentes configurações de topologia eram consideradas. Outros fatores foram mantidos constantes, como o procedimento de remoção das faltas.

Desde então, as RNs têm sido usadas para prever índices de estabilidade, bem como a classificação do sistema em estável/instável, dada a ocorrência de distúrbios.

Várias publicações do final da década de 80 descrevem o uso de RNs para sistemas pequenos, com no máximo 4 máquinas. Em [Pao, 1992] é proposto o uso combinado de um procedimento de aprendizado supervisionado e não supervisionado com uma arquitetura de RN do tipo *functional link net*, para obter o CCT.

Em [Omata, 1993] é proposta a utilização de uma variável de saída contínua para melhorar a performance da AET via RN. Uma RN é aplicada a um sistema de 16 barras e 6

máquinas. Os resultados mostraram-se melhores do que aqueles obtidos em trabalhos semelhantes, quando saídas binárias foram utilizadas para indicar os estados estáveis/instáveis.

Em [McCalley, 1997] um sistema de visualização da borda de segurança é proposto, baseado na geração de gráficos bidimensionais (*nomograms*), e utilizando a capacidade computacional das RNs para reduzir o longo tempo normalmente necessário para a geração dos gráficos. A substituição das simulações pelas saídas proporcionadas pela RN permite a geração de um número maior de pontos devido a sua rapidez de processamento, produzindo gráficos mais precisos.

Em [Aboytes, 1996] é apresentada a aplicação de RNs para a AET em um sistema do México com 53 máquinas e 227 barras. São investigados novos tipos de atributos de entrada que incluem a representação da mudança de topologia, níveis de carga e geração, e tipos de contingência. É utilizada como saída uma variável binária representando os estados estável/instável do sistema. Uma organização especial de padrões de treinamento com uma separação por tipo de contingência é proposta para reduzir os erros de classificação. Nesse trabalho é constatado que o uso de variáveis dinâmicas como entradas das RNs não contribui para a melhoria do desempenho, assim como também notado em [Miranda, 1994].

Em [Mansour, 1997a] e [Mansour, 1997b] é apresentada a aplicação de RNs para ranqueamento de contingências em dois sistemas do Canadá, o da B. C. Hydro, com 1393 barras e 209 máquinas, e o da Hydro Quebec, com 963 barras e 87 máquinas. São utilizadas 57 entradas para o primeiro sistema e 61 para o segundo. Dois tipos de saídas contínuas, a margem de energia transitória e o CCT são usados no treinamento da RN. Vários testes são realizados considerando uma série de alternativas: divisão do sistema em três partes separadas e estudo de cada uma delas; uso de uma RN com duas saídas correspondentes aos índices de estabilidade; uso de duas RNs desacopladas, cada uma com um índice de estabilidade diferente, para as mesmas entradas; redução do número de atributos de entrada, baseado em experiência dos especialistas. Os resultados foram analisados em termos das amplitudes dos erros, dos valores médios, valores máximos e mínimos, número de alarmes falsos (falsos positivos) e número de falsos negativos. Os resultados indicaram que o uso de uma única RN para cada índice de estabilidade teve desempenho levemente melhor que uma RN com duas

saídas; que a divisão do sistema em subsistemas menores teve um grande impacto na melhoria dos resultados. Também nesse trabalho é concluído que o uso de variáveis dinâmicas não contribui para a melhoria do desempenho das RNs.

Em [Aresi, 1999] é utilizado um sistema híbrido com um sistema especialista e RNs, sendo estas utilizadas para definir a classificação preliminar do sistema em estável/instável. Como sistema de teste, foi utilizado o sistema de Marrocos, com 28 máquinas e 123 linhas de transmissão. Uma RN é treinada para cada uma das 28 máquinas, e como variáveis de entrada são utilizados o ângulo e a energia cinética de cada gerador no instante da falta. Cada RN tem uma saída binária indicando se aquela máquina específica torna-se estável ou instável após o distúrbio.

Em [Demaree, 1994] é descrita a implementação de um sistema de análise de estabilidade *on-line* para o centro de controle da empresa B. C. Hydro. Em tal trabalho as RNs são consideradas como ferramenta para o ranqueamento inicial dos casos, e também é afirmado que mais pesquisas estão sendo realizadas no sentido da utilização das RNs para esse fim.

Em [Kamwa, 2001] é proposto um sistema de ranqueamento de contingências aplicado ao longo da simulação de estabilidade dos casos. Nessa proposta, à medida que as simulações avançam, um sistema inteligente híbrido composto de RNs e um sistema *fuzzy* são utilizados para indicar os casos estáveis. A simulação desses casos é então interrompida para a economia de esforço computacional. São mostrados resultados promissores de simulações realizadas utilizando dois sistemas da empresa Hydro-Québec: um sistema fictício com 67 barras, 23 máquinas geradoras, e um sistema com 783 barras, 1072 linhas, 130 geradores, que é normalmente utilizado no departamento de planejamento da operação da empresa.

2.1 – Geração de Conjuntos de Treinamento

A aplicação de RNs para a AET depende da verificação de que há algumas características pré-distúrbio do sistema que o levam a uma situação estável ou instável após o distúrbio. Portanto, há a necessidade da formação de um conjunto de treinamento, seja através da coleta

de dados históricos de contingências ou através da simulação com métodos tradicionais. Esse estágio corresponde à geração de um grande número de condições de operação prováveis. Comumente, pode-se utilizar como ponto de partida um ou mais casos base e adicionar variações aos mesmos, como mudanças de carga, padrões de geração e topologia. Cada cenário deve ser submetido à execução de um programa de fluxo de potência para testar a factibilidade do mesmo.

Em seguida, uma ou mais contingências são simuladas no domínio do tempo para obter a resposta de estabilidade de cada caso, seja como informação binária estável/instável ou como valor real correspondente a uma margem de estabilidade. A formação do conjunto de treinamento pode ser muito lenta porque, dependendo do número de entradas da RN, uma grande quantidade de dados de treinamento pode ser necessária para um bom desempenho da mesma. E, necessariamente, esses dados devem ser gerados através de simulação com métodos convencionais. Porém, uma vez treinada, a RN apresenta respostas dentro de um tempo ínfimo, necessário para a realização de apenas algumas operações de soma e multiplicação. É o poder de generalização da RN que lhe confere grande potencialidade para a aplicação em AET porque quando um novo cenário de operação surge, ela é capaz de produzir respostas satisfatórias em curtíssimo intervalo de tempo.

Em [Sobajic, 1989] a simulação no domínio do tempo é executada para avaliar o critério *first swing*. As cargas são representadas como impedâncias constantes incluídas nos termos diagonais da matriz de admitâncias do sistema. Curtos-circuitos trifásicos são simulados próximos aos terminais de linhas de transmissão através da adição de admitâncias de valores elevados próximas aos barramentos correspondentes. Para gerar o conjunto de treinamento, duas topologias distintas foram consideradas: a primeira, com todas as 7 linhas em serviço e a segunda, com uma linha removida. Para cada topologia, 15 níveis de carga diferentes foram selecionados na faixa de 0,6 a duas vezes a carga do ponto de operação nominal. O conjunto de treinamento consistia de 30 padrões com os respectivos valores de CCT como valores de saída.

Em [Fischl, 1989] foram calculados os fluxos de potência para 9 níveis de carga diferentes e para 9 configurações diferentes de topologia. Cada uma destas configurações corresponde a contingências diferentes, que representam a perda de uma linha. No total, há 81

combinações diferentes de configurações e níveis de carga. O resultado de saída foi avaliado como limites de fluxos nas linhas de transmissão, sendo que os casos que violassem esses limites eram considerados inseguros, e os que estivessem abaixo dos limites, considerados seguros.

Em [Omata, 1993] foi treinada uma única RN, sendo o conjunto de treinamento com 6 níveis de carga e geração diferentes, variando de 100% a 150%, e 8 contingências diferentes ocorridas em diferentes partes do sistema. Foi simulado um mesmo tipo de contingência simples, faltas trifásicas, com duração de 200 ms, mas ocorridas em diferentes partes do sistema. No total foram coletados 48 casos de cálculos de estabilidade transitória, combinando 6 condições de carga com 8 contingências.

Em [Aboytes, 1996] foi utilizado um total de 2700 padrões de treinamento através de simulações no domínio do tempo considerando as seguintes condições de operação:

- 6 tipos de distúrbios, falta fase-terra e falta trifásica em três barramentos críticos;
- 2 níveis de carga, máximo e mínimo;
- 3 configurações do sistema, nenhuma linha de transmissão fora de serviço, e duas linhas fora de serviço (alternativamente);
- 3 configurações de geração; e
- 25 combinações de geração.

Em [Mansour, 1997a] e [Mansour, 1997b] são gerados conjuntos de treinamento para dois subsistemas do Canadá, o B.C. Hydro e o Hydro Quebec. Para o primeiro foram gerados inicialmente 36 casos base. Modificando os padrões de geração desses casos iniciais criaram-se mais 72 casos, num total de 108. Vinte e cinco contingências diferentes foram selecionadas para os estudos e, quando combinadas com os casos base, produziram um total de 1158 simulações diferentes. No segundo subsistema, foram desenvolvidos 57 casos base para uma parte do sistema denominada corredor oeste e 32 para outra denominada corredor leste. Utilizando um número de contingências e de localizações de faltas, um total de 266 simulações diferentes foram executadas para gerar os exemplos.

Em [Aresi, 1999], a geração do conjunto de treinamento considera o estado inicial do sistema, o tipo de falta ocorrido e a duração da mesma. O número de combinações é reduzido, agindo na duração da falta (considerando tempos de remoção da falta entre 0,1 e 0,5 s, por exemplo), no tipo da falta (considerando somente faltas trifásicas) e no estado inicial do sistema (tal como carga e topologia). A representação da topologia leva em conta que em condições normais de operação as companhias satisfazem a demanda com um número limitado de configurações da rede. As simulações para geração do conjunto de treinamento foram realizadas com 3 níveis de carga diferentes e ocorrência de faltas em cada linha do sistema.

2.2 – Atributos de Entrada e Saída

A quantidade de fatores envolvidos na descrição do estado operacional do sistema de potência permite a formulação de um número muito grande de atributos que podem servir como variáveis de entrada na tarefa de classificação executada por uma RN. Também, inúmeras formas de se quantificar a estabilidade do sistema frente à ocorrência de algum distúrbio permitem várias maneiras de se representar essa informação como saída da RN.

Muitos atributos de entrada e saída já foram propostos e utilizados na aplicação de RNs à AET. Através da observação das possibilidades apresentadas na literatura, pode-se obter alguma inspiração sobre esses fatores. Note-se que a formulação de atributos pode ser feita em uma quantidade aumentada, após a qual se realiza a Extração de Atributos, para redução da dimensionalidade ou, como é prática mais usual, a sugestão de um especialista é utilizada como escolha final dos atributos de entrada, baseada em sua experiência no assunto.

Em [Sobajic, 1989] são utilizadas como entradas os ângulos dos rotores de cada máquina com relação ao centro de inércia, sendo os parâmetros de aceleração de cada gerador definidos como

$$I_i = \frac{P_{mi} - P_{fi}}{M_i}, \quad i = 1, \dots, N_G \quad (2.1)$$

sendo

P_{mi} → potência mecânica fornecida ao gerador i ;

P_{fi} → potência reduzida de saída do gerador i durante a falta;

M_i → momento de inércia do gerador i .

Também são utilizados como entradas a seguinte função de (2.1):

$$I_i = \frac{(P_{mi} - P_{fi})^2}{M_i}, \quad i = 1, \dots, N_G \quad (2.2)$$

No sistema de 4 máquinas testado em [Sobajic, 1989] há um total de 12 atributos de entrada e um atributo de saída, o CCT para uma dada contingência.

Em [Aboytes, 1996] foram escolhidas 12 variáveis de entrada, em função de:

- Potências ativa e reativa geradas em cada máquina;
- Posição angular relativa de cada máquina;
- Número de unidades sincronizadas em cada nível de tensão;
- Nível de carga; e
- Topologia.

Nesse trabalho ([Aboytes, 1996]) foram realizados testes com variáveis estáticas e dinâmicas. Segundo os resultados obtidos, não houve vantagens em se incluírem as variáveis dinâmicas.

Em [Mansour, 1997a] e [Mansour, 1997b], três tipos de informações são consideradas como importantes para caracterização da condição de operação do sistema: (a) condições iniciais, (b) configuração da rede pré-distúrbio e (c) contingência. Uma lista de atributos é formulada com base na experiência dos especialistas em estudos de estabilidade, em função de:

- Potências ativa e reativa injetadas nos barramentos;
- Cargas ativa e reativa totais;
- Potências ativa e reativa gerada;
- Magnitude de tensão nos barramentos;
- Ângulo ou separação angular dos barramentos;
- Ângulo interno da máquina;
- Fluxo de potência ativa e reativa nos ramos;
- Geração total do sistema;
- Exportações de potência ativa e reativa do sistema;
- Fluxos de potência ativa de intercâmbios;
- Reservas de potência reativa dos geradores;
- Reserva de potência reativa do sistema;
- Velocidade do gerador no momento de remoção da falta;
- Energia cinética no momento de remoção da falta;
- Energia cinética corrigida no momento de remoção da falta;
- Impedância de transferência;
- Acelerações dos geradores no momento da remoção da falta.

Como atributos de saída foram utilizados a margem de energia transitória e o CCT. Nesses trabalhos ([Mansour, 1997a] e [Mansour, 1997b]), os resultados indicam que a inclusão de variáveis dinâmicas acarreta um aumento muito pequeno na precisão e, portanto, a sua eliminação pode diminuir significativamente o esforço computacional sem piorar o desempenho.

Em [Pao, 1992] foram utilizadas as potências ativa e reativa geradas durante a ocorrência da falta, potência mecânica posições angulares dos rotores e inércia das máquinas. Potência ativa gerada, tensões internas e inércia das máquinas foram utilizadas como entradas em [Miranda, 1994]. Ângulos dos Pontos de Equilíbrio Instáveis e margem de energia transitória foram utilizados em [Zhou, 1994].

Em [Kamwa, 2001] foram utilizadas grandezas no domínio da frequência para registrar a evolução da estabilidade do sistema, com uma taxa de amostragem de uma amostra por ciclo, desde o início da simulação da ocorrência de uma falta. Os sistemas elétricos de 783 barras e 67 barras foram divididos em 8 áreas, cada um, nas quais foi instalado um número de unidades de medição de fasores, para que o módulo e ângulo das tensões em dados barramentos fossem obtidos. Os atributos de entrada são obtidos a partir de índices de severidade calculados ao longo do tempo e associados com ângulo, frequência, aceleração e produto interno. A motivação para usar tais atributos é que através do cálculo da densidade espectral de potência de tais grandezas, pode-se monitorar as oscilações mecânicas e obter informações sobre as características transitórias do sistema. A razão principal para essa abordagem é que a densidade espectral do sinal de oscilação tem relação estreita com o excesso de energia cinética injetada no sistema pela falta. Nesse trabalho ([Kamwa, 2001]) foram utilizados 28 atributos de entradas, cada um deles contendo informações combinadas de todos os barramentos monitorados pelos medidores de fase.

2.3 – Considerações Finais

Apesar do seu relativo sucesso na aplicação à AET, a utilização de muitos atributos de entrada para RNs demanda um extenso processo de treinamento, que ainda pode acarretar resultados de generalizações ruins, devido à Maldição da Dimensionalidade. Em geral, essa é a maior desvantagem para a aplicação de RNs em sistemas de potência de grande porte com centenas (até milhares) de geradores, porque tal problema exige um grande número de variáveis. Isso pode tornar-se uma tarefa proibitiva. Portanto, algum método de Extração de Atributos é necessário para reduzir a dimensionalidade do espaço de entrada da RN. O maior objetivo é

obter o melhor desempenho possível, ao mesmo tempo tentando reduzir o tempo de treinamento da RN.

Como pode ser observado com diversos métodos estatísticos de reconhecimento de padrões, à medida que eles começaram a ser testados em sistemas de maior porte, saindo do domínio puramente investigativo e tornando suas potencialidades de aplicação prática, as altas dimensionalidades envolvidas tornaram-se um impedimento para a sua expansão. Sistemas de tamanho real acarretam uma grande quantidade de atributos e levam à Maldição da Dimensionalidade.

Especificamente nas aplicações de RNs à AET, em quase todos os trabalhos realizados até agora, a redução da dimensionalidade é aplicada através da escolha humana de atributos, quando um especialista julga previamente que atributos de entrada são mais apropriados para utilização na RN. Faltam métodos automáticos de Extração de Atributos capazes de moldar os dados de treinamento de maneira apropriada para uso no problema de AET via RNs, ou ainda outras alternativas de modo a ajustar a RN aos dados existentes.

Os capítulos seguintes apresentarão uma revisão das técnicas estatísticas de Extração de Atributos para o processamento de sinais. Também apresentará algumas propostas de Extração de Atributos, baseadas nessas técnicas, de maneira a generalizar sua aplicação, ou técnicas novas que possam ser aplicadas em dados de treinamento para classificação por *Perceptron Multicamadas* ou Máquinas de Vetores Suporte.

Das aplicações em sistemas reais de grande porte citadas na subseção 2.2, em [Kamwa, 2001] são utilizadas atributos de entrada oriundos de variáveis dinâmicas e da análise no domínio da frequência. Em [Aboytes, 1996], [Mansour, 1997a] e [Mansour, 1997b] são utilizados atributos oriundos de variáveis estáticas e dinâmicas. Os testes desses últimos trabalhos indicam que a introdução de variáveis dinâmicas não acarreta ganhos significativos na performance das RNs.

Não há evidências suficientes para decidir se as variáveis estáticas são mais adequadas do que as variáveis dinâmicas como atributos de RNs aplicadas à AET de sistemas de potência. Nesta tese são utilizadas somente variáveis estáticas que descrevem condições de operação anteriores à ocorrência da contingência, a topologia do sistema e a localização da

contingência. Futuramente serão testadas também variáveis de natureza dinâmica, adquiridas durante a ocorrência do distúrbio, incluindo a combinação com variáveis de natureza estática.

Extração de Atributos

Extração de Atributos

Grande parte das aplicações de classificação em problemas do mundo real utiliza treinamento supervisionado, quando cada exemplo de treinamento é associado a um rótulo de classificação. Normalmente, as funções densidade de probabilidade das classes existentes são desconhecidas. Os atributos de entrada relevantes quase sempre também são desconhecidos a princípio. Portanto, muitos atributos candidatos são inicialmente introduzidos para melhor representar o sistema modelado. Infelizmente muitos desses atributos são parcialmente ou completamente irrelevantes, ou ainda redundantes. Assim, o tamanho do conjunto de treinamento necessário para se obter um bom classificador pode ser muito grande. Tão grande que o treinamento poderia não ocorrer de maneira satisfatória caso não se realizasse uma transformação a partir desses atributos originais.

A Extração de Atributos (EA) consiste de um processo em que os atributos são transformados com o objetivo de se obter redução na representação dos dados, em termos de dimensionalidade, ao mesmo tempo preservando ou melhorando a capacidade de classificação. Utilizando um conjunto de treinamento, T , de pares ordenados (x_i, y_i) , $T = \{x_i, y_i\}_{i=1}^N$, sendo que x_i é um vetor n -dimensional de variáveis ou atributos reais ($x_i \in R^n$) e $y_i \in \{+1, -1\}$ representa a classe à qual x_i pertence, o objetivo da EA é determinar uma transformação

$$\begin{aligned} F: R^n &\rightarrow R^d \\ x &\rightarrow f = F(A, x) \end{aligned}$$

do espaço original R^n para um outro R^d de dimensão d ($d < n$), a partir do conjunto de treinamento T . Então, o conjunto de treinamento passa a ser representado por um novo conjunto de treinamento, $T' = \{f_i, y_i\}_{i=1}^N$. O símbolo A representa uma matriz de transformação a princípio desconhecida, e $f \in R^d$. A Extração de Atributos será considerada como satisfatória se um ponto de R^d , obtido pela transformação citada, puder ser classificado em

uma das classes existentes com mínimo erro, isto é, o número de classificações incorretas de um conjunto de teste deve ser tão pequeno quanto possível, qualquer que seja o conjunto de teste tomado, desde que ele tenha sido gerado pela mesma distribuição que gerou o conjunto de treinamento.

Um processo de EA requer um procedimento de *geração* de transformações candidatas e um critério de *avaliação* capaz de comparar as transformações assim geradas. Caso a EA seja feita de forma analítica, a geração da transformação é feita em um só passo e o critério de avaliação encontra-se embutido no desenvolvimento analítico. Por outro lado, a geração de soluções candidatas pode ser realizada por algum procedimento iterativo ou *método de busca*, guiado pelo critério ou função de *avaliação*.

A Extração de Atributos linear é realizada quando a transformação é dada por $f = A \cdot x$, e um caso particular acontece quando A é uma matriz $d \times n$ formada por “zeros” e “uns”, com apenas um valor diferente de zero em cada linha. Nesse caso, a EA é chamada de Seleção de Atributos (SA). Em outras palavras, vetores de atributos candidatos, $f \in R^d$, são gerados selecionando-se d componentes (também chamados atributos ou variáveis de entrada) do vetor de entrada original, $x \in R^n$. Uma busca exaustiva para encontrar o melhor vetor de atributos exigiria a avaliação de todos os subconjuntos possíveis formados por d componentes. O número de possibilidades é igual a $\binom{n}{d} = \frac{n!}{(n-d)!d!}$, o que torna a busca exaustiva algo intratável porque facilmente esse número toma valores muito altos. Então, é necessário um método heurístico de otimização para realizar a busca no espaço de possibilidades, minimizando a função de avaliação.

Este capítulo inicia-se com um método de EA muito conhecido, a Extração de Componentes Principais ou Transformação de Karhunen-Loève, na Seção 3.1. Em seguida, Funções de Avaliação e Métodos de Busca são apresentados nas Seções 3.2 e 3.3, respectivamente. Alguns outros métodos de EA encontrados na literatura recente e considerações finais são apresentados nas Seções 3.4 e 3.5, respectivamente.

3.1 – Extração de Componentes Principais ou Transformação Karhunen-Loève

Seja x um vetor aleatório de dimensão m contendo os atributos de entrada descritivos das condições de operação do sistema elétrico. Assume-se que esse vetor aleatório x tenha média igual a zero [Haykin, 1999]:

$$E[x] = 0 \quad (3.1)$$

sendo E o operador Esperança Matemática ou Esperança Estatística. Se x tiver uma média diferente de zero, basta subtrair a média antes de prosseguir com a análise. Seja q um vetor unitário, também de dimensão m , no qual o vetor x será projetado. Essa projeção é definida por um produto interno entre os vetores x e q :

$$P = x' \cdot q = q' \cdot x$$

sujeito à condição

$$\|q\| = (q' \cdot q)^{1/2} = 1 \quad (3.2)$$

A projeção P é uma variável aleatória com média e variância relacionada às estatísticas do vetor aleatório x . Com a premissa de que o vetor aleatório x tenha média igual a zero, segue que o valor médio da projeção P também é zero:

$$E[P] = q' E[x] = 0 \quad (3.3)$$

A variância de P é, portanto, o mesmo que seu valor médio quadrático. Nesse caso, pode-se escrever:

$$\sigma^2 = E[P^2] = E[(q' \cdot x)(x' \cdot q)] = q' E[x \cdot x'] q = q' \cdot U \cdot q \quad (3.4)$$

A matriz U , m por m , é a matriz de correlação do vetor aleatório x , formalmente definida como o seguinte:

$$U = E[x.x'] \quad (3.5)$$

Pode-se observar que a matriz de correlação é simétrica. Isso significa que

$$U' = U \quad (3.6)$$

Dessa propriedade, segue que se a e b são dois vetores m por l quaisquer, então:

$$a'.Ub = b'.Ua \quad (3.7)$$

Pode-se perceber em (3.4) que a variância σ^2 da projeção P é uma função do vetor q . Pode-se escrever:

$$\psi(q) = \sigma^2 = q'.U.q \quad (3.8)$$

Esta última equação pode ser utilizada para medir a variância dos dados, em função da direção em que eles são projetados. A Extração de Componentes Principais realiza uma transformação dos dados que consiste exatamente em sua projeção nas direções que resultem em maior variância. Por isso, o objetivo é encontrar o vetor q que maximize (3.8), ou seja, a direção que maximize a variância de P . Esse vetor é um ponto para o qual (3.8) tem derivada igual a zero, e, portanto, satisfaz a seguinte equação, para uma pequena perturbação δq :

$$\psi(q + \delta q) = \psi(q) \quad (3.9)$$

A partir de (3.8), tem-se que

$$\psi(q + \delta q) = (q + \delta q)'.U.(q + \delta q) = q'.U.q + 2.(\delta q)'.U.q + (\delta q)'.U.\delta q \quad (3.10)$$

Em (3.10) foi feito uso de (3.7). Em seguida, ignorando o termo $(\delta q)'.U.\delta q$ e utilizando (3.8) tem-se

$$\psi(q + \delta q) = q'.U.q + 2.(\delta q)'.U.q = \psi(q) + 2.(\delta q)'.U.q \quad (3.11)$$

Portanto, a aplicação de (3.9) em (3.11) implica que:

$$(\delta q)' U q = 0 \quad (3.12)$$

Os possíveis valores δq são perturbações de q que resultem em vetores de norma igual a 1, pois esta é uma das premissas iniciais. Isto é,

$$\|q + \delta q\| = 1 \quad (3.13)$$

ou de maneira equivalente,

$$(q + \delta q)' (q + \delta q) = 1 \quad (3.14)$$

Então, em vista de (3.2), exige-se que:

$$(\delta q)' q = 0 \quad (3.15)$$

Isso significa que a perturbação δq deve ser ortogonal a q , isto é, só é permitida uma perturbação na direção de q .

Por convenção, os elementos do vetor unitário q são a-dimensionais no sentido físico. Portanto, para se combinar (3.12) e (3.15) é preciso introduzir um fator de escala, λ , com a mesma dimensão que os elementos da matriz de correlação U . Assim, pode-se escrever

$$(\delta q)' U q - \lambda (\delta q)' q = 0$$

ou, de maneira equivalente,

$$(\delta q)' (U q - \lambda q) = 0 \quad (3.16)$$

Para que a equação (3.16) seja satisfeita, é necessário e suficiente que se tenha

$$U q = \lambda q \quad (3.17)$$

Esta é a equação que governa os vetores unitários q para os quais a variância $\psi(q) = \sigma^2$ tem valores máximos locais.

A equação (3.17) caracteriza o problema de autovalores e autovetores. Portanto, os autovetores da matriz de correlação U definem as direções de maior variância para projeção dos dados. Pode ser mostrado também que a variância dos dados projetados sobre cada autovetor é medida pela magnitude do autovalor correspondente. A Extração de Componente Principais (ECP) realiza uma redução de dimensionalidade com os dados originais, \mathbf{x} , num espaço de dimensão n , para um espaço de dimensão d . Para isso, são tomados os d autovetores da matriz U que possuam os maiores autovalores e é realizada a seguinte transformação:

$$f_j = \mathbf{x}' \cdot \mathbf{q}_j, \quad j = 1, \dots, d \quad (3.18)$$

A idéia de se mapear os vetores do espaço original para um outro de menor dimensão em função das direções que apresentem maior variância é baseada no desejo de ter-se o menor erro possível ao se reconstruir os dados originais a partir do novo espaço. Nesse caso, para fins de classificação, a consideração é de que a direção de maior variância irá proporcionar a maior capacidade de discriminação. Isso não é sempre verdadeiro porque os atributos mais importantes são aqueles para os quais a diferença entre os pontos médios de cada classe seja grande com relação às variâncias das classes, e não aqueles para os quais as variâncias sejam grandes [Duda, 1973].

3.2 – Funções de Avaliação

A seguir serão revisadas algumas funções que podem ser utilizadas para comparação entre diferentes transformações candidatas.

DISCRIMINANTE DE FISHER

Dado um conjunto de treinamento com N amostras n dimensionais, $T = \{\mathbf{x}_i, y_i\}_{i=1}^N$, com $y_i \in \{C_1 = -1, C_2 = +1\}$, sendo N_1 pontos pertencentes à classe C_1 e N_2 pontos pertencentes à classe C_2 , a projeção de \mathbf{x}_i num espaço de uma dimensão é:

$$u = v' \cdot x_i \quad (3.19)$$

O vetor x_i será projetado na direção de v e os dados de treinamento, que poderiam estar separados no espaço de n dimensões, podem misturar-se na nova projeção. Então, deve-se encontrar um vetor, v , que produza a maior separação possível entre os dados. Isso pode ser feito calculando o vetor que maximiza a função Discriminante de Fisher:

$$FD(v) = \frac{v' S_B v}{v' S_w v} = \frac{|\tilde{m}_1 - \tilde{m}_2|}{\tilde{s}_1^2 + \tilde{s}_2^2} \quad (3.20)$$

A matriz de espalhamento intra-classe, S_w , é definida como

$$S_w = S_1 + S_2 \quad (3.21)$$

As matrizes S_1 e S_2 são definidas como

$$S_i = \sum_{x \in C_i} (x - m_i)(x - m_i)' \quad (3.22)$$

Os vetores médios (protótipos das classes) são dados por

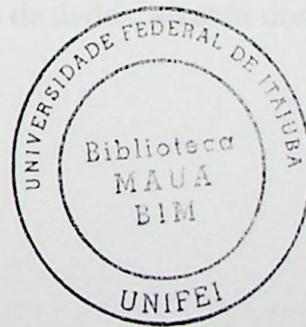
$$m_i = \frac{1}{N_i} \sum_{x \in C_i} x \quad (3.23)$$

Também,

$$\tilde{m}_i = v' \cdot m_i \quad (3.24)$$

e

$$\tilde{s}_i^2 = \frac{(u_i - \tilde{m}_i)^2}{N_i} \quad (3.25)$$



A matriz de espalhamento entre-classes, S_B , é definida como

$$S_B = (m_1 - m_2)(m_1 - m_2)' \quad (3.26)$$

A solução da maximização de FD pode ser escrita como

$$v = S_w^{-1}(m_1 - m_2) \tag{3.27}$$

Quando os dados pertencerem a N_c classes diferentes, em vez de duas, o Discriminante de Fisher pode ser generalizado, tal que a projeção aconteça do espaço de n dimensões para um espaço de no máximo $(N_c - 1)$ dimensões.

A função Discriminante de Fisher mede a distância entre os pontos médios das classes com relação às variâncias dos dados, considerando os atributos tomados como componentes do vetor x_i . Na Figura 3.1 um conjunto de dados é representado de duas maneiras diferentes, com dois subconjuntos de atributos, (x_1, x_2) e (x_1, x_3) . Cada subconjunto de atributos permite uma representação diferente dos dados, sendo a mais adequada aquela que produza o maior valor da função de Fisher.

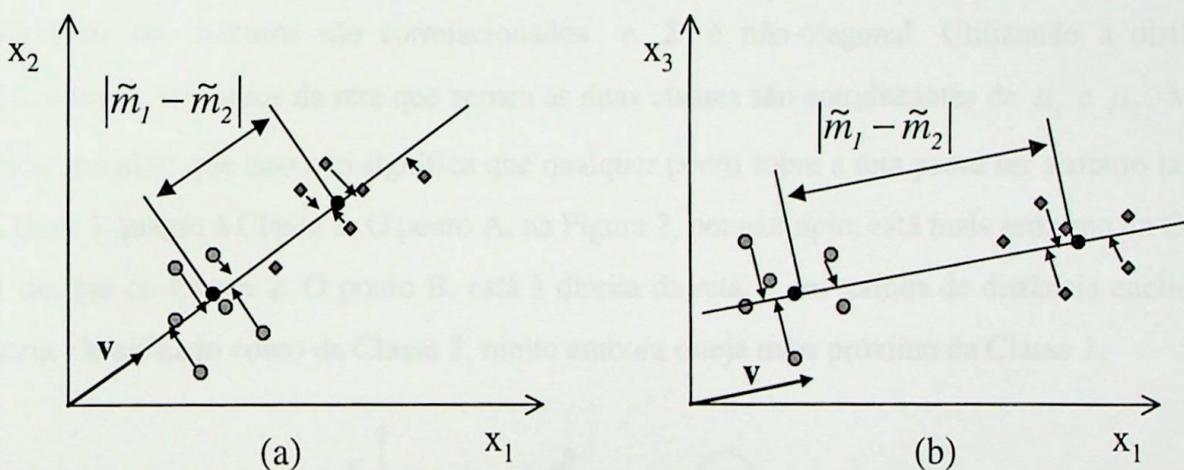


Figura 3.1 - Duas representações de um mesmo conjunto de dados, através dos atributos (a) (x_1, x_2) e (b) (x_1, x_3)

DISTÂNCIA EUCLIDIANA

A distância Euclidiana entre dois vetores é dada por

$$\delta_2(\mathbf{x}, \mathbf{z}) = \|\mathbf{x} - \mathbf{z}\| = \left(\sum_{j=1}^n (x_j - z_j)^2 \right)^{1/2} = \left((\mathbf{x} - \mathbf{z})' (\mathbf{x} - \mathbf{z}) \right)^{1/2} \quad (3.28)$$

Geometricamente, pode-se interpretá-la como a distância em linha reta entre dois pontos.

DISTÂNCIA DE MAHALANOBIS

Esta é uma distância ponderada que é dada por

$$\delta_M(\mathbf{x}, \mathbf{z}) = \left((\mathbf{x} - \mathbf{z})' \Sigma^{-1} (\mathbf{x} - \mathbf{z}) \right)^{1/2} \quad (3.29)$$

sendo Σ a matriz de covariância do vetor aleatório \mathbf{x} , correspondente aos dados de treinamento \mathbf{x}_i , ou sua estimativa a partir do conjunto de treinamento. O símbolo Σ representa uma matriz simétrica positiva definida. Para melhor entendimento da distância de Mahalanobis, considere a Figura 3.1, que apresenta um exemplo com duas classes, ambas com matriz de covariância Σ , em termos de dois atributos ou variáveis, x_1 e x_2 . No caso geral, as variáveis ou atributos são correlacionados e Σ é não-diagonal. Utilizando a distância Euclidiana, os pontos da reta que separa as duas classes são equidistantes de μ_1 e μ_2 . Mas é fácil perceber que isso não significa que qualquer ponto sobre a reta possa ser atributo tanto à Classe 1 quanto à Classe 2. O ponto A, na Figura 3, por exemplo, está mais próximo da Classe 1 do que da Classe 2. O ponto B, está à direita da reta, e em termos de distância euclidiana seria classificado como da Classe 2, muito embora esteja mais próximo da Classe 1.

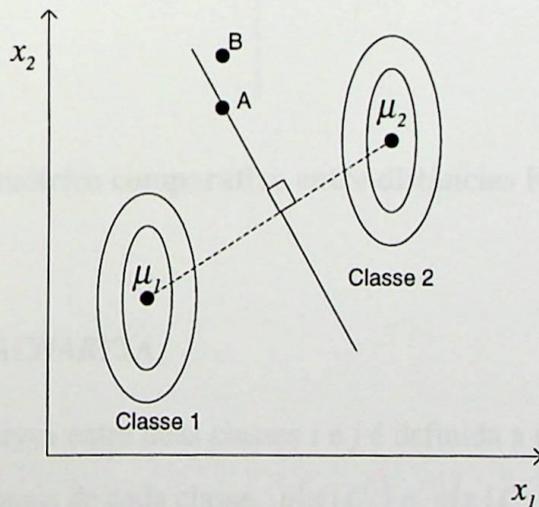


Figura 3.2 – Exemplo de duas distribuições com mesma matriz de covariância

Para que seja utilizada a distância Euclidiana, deve-se ter as variáveis não correlacionadas. A distância de Mahalanobis é uma medida ponderada que na verdade realiza uma transformação para tornar as variáveis não correlacionadas.

DISTÂNCIA DE CHEBYSHEV

$$\delta_{\infty}(\mathbf{x}, \mathbf{z}) = \max_{1 \leq j \leq n} |x_j - z_j| \quad (3.30)$$

Para um melhor entendimento sobre a distância de Chebyshev, na Figura 3.3 é apresentado um exemplo ilustrativo. Em duas dimensões, é apresentado o lugar geométrico dos pontos que têm a mesma distância até a origem, $\delta_2(\mathbf{x}, \mathbf{0}) = \delta_{\infty}(\mathbf{x}, \mathbf{0}) = 2$, com relação à distância Euclidiana e com relação à distância de Chebyshev, respectivamente.

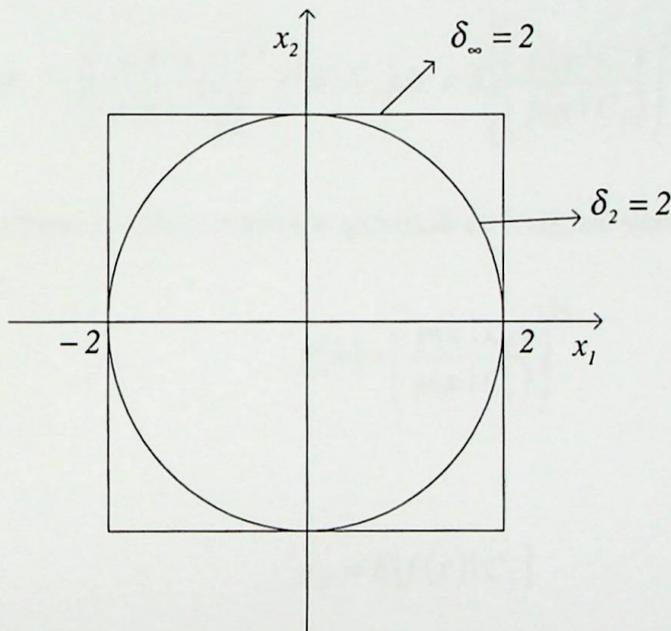


Figura 3.3 – Lugar geométrico comparativo entre distâncias Euclidiana e de Chebyshev

DISTÂNCIA DE BHATTACHARYYA

A Distância de Bhattacharyya entre duas classes i e j é definida a partir das funções densidade de probabilidade condicionais de cada classe, $p(\mathbf{x} | C_i)$ e $p(\mathbf{x} | C_j)$:

$$\delta_{ij,B} = -\ln \int_{\Omega} (p(\mathbf{x} | C_i) p(\mathbf{x} | C_j))^{1/2} d\mathbf{x} \quad (3.31)$$

sendo Ω o espaço de \mathbf{x} .

Essa medida de distância possui as seguintes propriedades:

- a) $\delta_{ij,B}$ é máxima se $p(\mathbf{x} | C_i) = 0$ quando $p(\mathbf{x} | C_j) \neq 0$ para qualquer \mathbf{x} , ou seja, quando as classes são disjuntas. Nesse caso, $\delta_{ij,B} = \infty$.
- b) $\delta_{ij,B} = 0$ se $p(\mathbf{x} | C_i) = p(\mathbf{x} | C_j)$.

Define-se o coeficiente de Bhattacharyya como

$$\rho_B = e^{-\delta_{ij,B}} = \int_{\Omega} [p(\mathbf{x} | C_i) p(\mathbf{x} | C_j)]^{1/2} dx \quad (3.32)$$

que pode ser escrito como

$$\rho_B = \int_{\Omega} \left[\frac{p(\mathbf{x} | C_i)}{p(\mathbf{x} | C_j)} \right]^{1/2} p(\mathbf{x} | C_j) dx = E \left[\left(\frac{p(\mathbf{x} | C_i)}{p(\mathbf{x} | C_j)} \right)^{1/2} | C_j \right] \quad (3.33)$$

Define-se a seguinte função, como raiz quadrada da razão de verossimilhança:

$$f(\mathbf{x}) = \left(\frac{p(\mathbf{x} | C_i)}{p(\mathbf{x} | C_j)} \right)^{1/2} \quad (3.34)$$

de onde

$$\rho_B = E[f(\mathbf{x}) | C_j] \quad (3.35)$$

Com isso tem-se

$$\delta_{ij,B} = -\ln(\rho_B) \quad (3.36)$$

É fácil notar que se as densidades $p(\mathbf{x} | C_i)$ e $p(\mathbf{x} | C_j)$ estão bem separadas entre si, então $f(\mathbf{x})$ assume valores pequenos na região em que $p(\mathbf{x} | C_j)$ é grande e vice-versa, fazendo com que $\rho_B = E[f(\mathbf{x}) | C_j]$ seja pequeno e, portanto, $\delta_{ij,B} = -\ln(\rho_B)$ seja grande. Se as densidades estão muito juntas então $\rho_B = E[f(\mathbf{x}) | C_j]$ é grande e $\delta_{ij,B}$ é pequena.

A dificuldade de cálculo dessa distância torna o seu uso muito difícil. Por isso, para obter uma forma mais prática, é preciso restringir-se à hipótese de que os dados tenham sido gerados a partir de uma distribuição normal. Nesse caso a Distância de Bhattacharyya torna-se

$$\delta_{ij,B} = \frac{1}{8} (\mathbf{m}_j - \mathbf{m}_i)' \left(\frac{\Sigma_i + \Sigma_j}{2} \right)^{-1} (\mathbf{m}_j - \mathbf{m}_i) + \frac{1}{2} \ln \frac{0.5 \det(\Sigma_i + \Sigma_j)}{(\det(\Sigma_i) \cdot \det(\Sigma_j))^{1/2}} \quad (3.37)$$

sendo Σ_i e Σ_j as matrizes de covariância dos dados de cada classe, e $\det(\bullet)$ o determinante de uma matriz.

DIVERGÊNCIA

A Divergência entre duas classes i e j é dada por

$$\delta_{ij,D} = \int_{\Omega} [p(\mathbf{x} | C_i) - p(\mathbf{x} | C_j)] \ln \frac{p(\mathbf{x} | C_i)}{p(\mathbf{x} | C_j)} d\mathbf{x} \quad (3.38)$$

De maneira semelhante à distância de Bhattacharyya, a Divergência possui as seguintes propriedades:

a) $\delta_{ij,D}$ é máxima se $p(\mathbf{x} | C_i) = 0$ quando $p(\mathbf{x} | C_j) \neq 0$ para qualquer \mathbf{x} , ou seja, quando as classes são disjuntas. Nesse caso, $\delta_{ij,D} = \infty$.

b) $\delta_{ij,D} = 0$ se $p(\mathbf{x} | C_i) = p(\mathbf{x} | C_j)$.

Assumindo que os dados tenham sido gerados a partir de uma distribuição normal, a Divergência torna-se

$$\delta_{ij,D} = \frac{1}{2} (\mathbf{m}_j - \mathbf{m}_i)' (\Sigma_i^{-1} + \Sigma_j^{-1}) (\mathbf{m}_j - \mathbf{m}_i) + \frac{1}{2} \text{tr} \{ \Sigma_i^{-1} \Sigma_j + \Sigma_j^{-1} \Sigma_i - 2I \} \quad (3.39)$$

sendo $\text{tr}\{A\}$ o traço da matriz A , $\text{tr}\{A\} = \sum_{i=1}^n a_{ii}$.

ENTROPIA E INFORMAÇÃO MÚTUA

Num problema de classificação, a entropia é uma função que pode medir a incerteza sobre a classe de saída. Num problema com N_c classes distintas, a entropia pode ser escrita como:

$$H(y) = -\sum_{i=1}^{N_c} P(C_i) \cdot \log_2(P(C_i)) \quad (3.40)$$

A entropia condicional, ou a incerteza sobre a classe de saída de dado o vetor de entrada, pode ser usada para medir a quantidade de informação contida em um dado vetor de entrada:

$$H(x) = H(y|x) = -\sum_{i=1}^{N_c} P(C_i|x) \log_2(P(C_i|x)) \quad (3.41)$$

Normalmente a incerteza sobre a classe de saída, medida por (3.41), é menor do que a incerteza medida por (3.27) porque naquele caso é acrescentada a informação sobre o vetor de entrada. Para mínima sobreposição de classes (máxima separabilidade), a entropia (quantidade de incerteza) deve ser a mínima possível, idealmente nula. Quando as variáveis de entrada são contínuas, os somatórios das expressões (3.27) e (3.28) tornam-se integrais. Para se selecionar atributos baseado em um conjunto de treinamento, deve-se utilizar a média de $H(x)$:

$$H = \int_{\Omega} H(x) p(x) dx = -\int_{\Omega} \sum_{i=1}^{N_c} P(C_i|x) \log_2(P(C_i|x)) p(x) dx \quad (3.42)$$

Ou seja, o melhor conjunto de atributos do vetor de entradas é aquele que minimiza (3.42), diminuindo a incerteza sobre a classe de saída.

A Informação Mútua (IM) é a quantidade pela qual a incerteza inicial, (3.40), é diminuída depois de acrescentada a informação do vetor de entrada:

$$I(y; x) = H(y) - H(y|x) \quad (3.42)$$

Portanto, a IM pode ser usada como critério de avaliação dos conjuntos de atributos que compõem o vetor x . O melhor conjunto é aquele que maximiza (3.42).

CLASSIFICADOR

Essa função de avaliação consiste em tomar o próprio classificador que será utilizado na tarefa de classificação para avaliar o conjunto de atributos no processo de EA. O erro de treinamento do classificador geralmente é tomado como critério de avaliação dos conjuntos de atributos considerados. Esse procedimento produz resultados de grande confiabilidade na EA, mas demanda um longo tempo de processamento e nem sempre pode ser aplicado na prática.

3.3 – Métodos de Busca

BUSCA SEQUENCIAL

Este é um método de busca muito simples e que apresenta tempo reduzido de processamento. Consiste em avaliar separadamente cada um dos atributos e tomar os primeiros d atributos com maior valor de avaliação. A consideração de que os atributos que têm boa avaliação serão os melhores para a tarefa de classificação nem sempre é verdadeira porque desconsidera os efeitos da combinação dos mesmos entre si.

BRANCH AND BOUND (B&B)

Este é um método de busca que exige que a função de avaliação utilizada para guiar a busca seja monotônica crescente. Essa propriedade garante que a função seja não decrescente à medida que novos atributos sejam adicionados ao subconjunto [Narendra, 1977]. Algumas funções que podem ser usadas são Distância de Mahalanobis, Discriminante de Fisher, Distância Bhattacharya e Divergência. O método *B&B* pode ser ilustrado com uma árvore cuja raiz representa todos os n atributos originais do vetor de entradas e cada ramo abaixo da mesma significa que um atributo foi removido, como no exemplo mostrado na Figura 3.4. Neste exemplo, 5 atributos são dados e a tarefa é escolher a melhor combinação de 2 atributos ($n = 5$ e $d = 2$). Se fosse aplicada a busca exaustiva, seria necessário calcular a função de

avaliação $\frac{5!}{3!2!}$ vezes. No entanto, aplicando a busca *B&B*, a primeira folha mais à direita (que contém os atributos x_1 e x_2 , já que x_3 , x_4 e x_5 foram removidos) é avaliada e o valor da função, B_0 , é armazenado. A seguir, o próximo ramo que ainda não foi avaliado será examinado, começando com o nó localizado logo abaixo da raiz (nó indicado por x_2 na Figura 3.4). Se a sua avaliação, B_1 , for menor do que B_0 então o ramo abaixo dele é descartado e o procedimento volta para a raiz. Nesse caso, nenhum valor maior que B_1 poderá ser encontrado para baixo do nó x_2 porque a monotonicidade garante que a função de avaliação é não crescente à medida que atributos são eliminados. Por outro lado, se B_1 for maior que B_0 então B_0 toma o valor de B_1 . Então o ramo pode ser explorado ainda mais porque uma combinação com número de atributos menor pode ser encontrada. O procedimento continua, sempre tendo o valor da função de avaliação como decisão sobre qual caminho tomar.

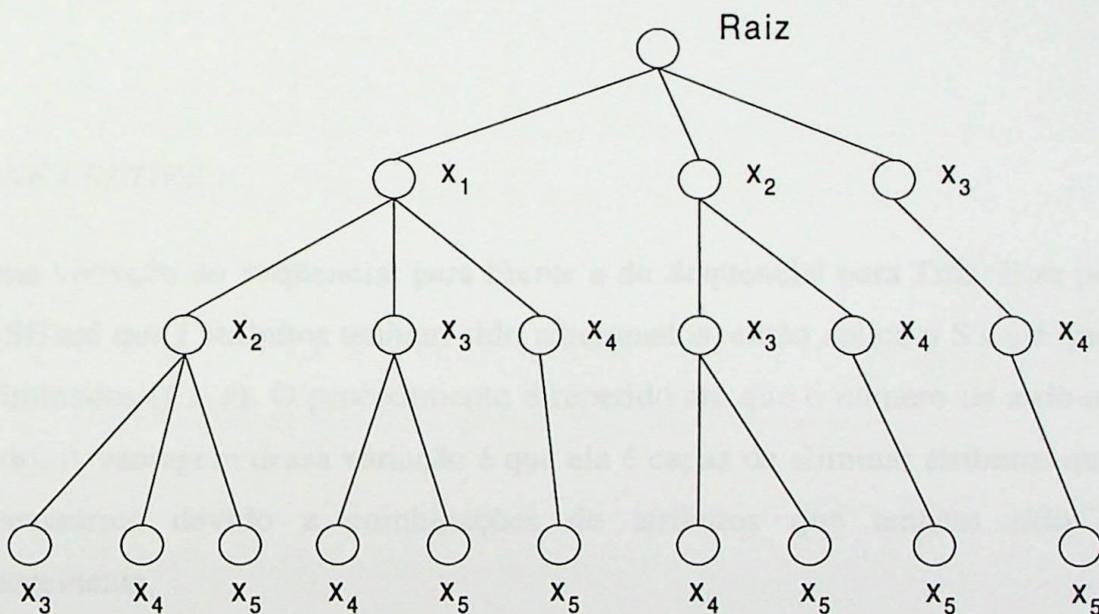


Figura 3.4 - Exemplo do método *Branch and Bound*

O processo de busca *B&B* sempre encontra a melhor solução com relação à função de avaliação considerada, mas no pior caso ela pode explorar mais nós do que na busca exaustiva. Nesse caso, ele iria explorar não apenas as folhas, mas também os ramos entre a raiz e as folhas.

SEQUENCIAL PARA FRENTE (SF) E SEQUENCIAL PARA TRÁS (ST)

Estes são métodos de busca sub-ótimos que requerem um tempo de processamento computacional bem pequeno. O SF primeiramente avalia os atributos individualmente e toma o melhor de todos. Em seguida todos os outros atributos remanescentes são testados em conjunto com o primeiro atributo escolhido e aquele que resultar em melhor valor da função de avaliação é o segundo atributo a ser escolhido. O conjunto de atributos é construído escolhendo-se entre os atributos remanescentes aquele que produz o melhor resultado quando combinado com os atributos escolhidos até o momento. Esse processo requer o cálculo da função de avaliação $n! - (n - d)!$ vezes.

Na busca ST, primeiro todos os atributos são combinados e avaliados em conjunto. O atributo a ser eliminado no passo seguinte será aquele que cuja eliminação causaria o maior aumento do valor da função de avaliação, quando os atributos remanescentes são avaliados em conjunto.

ADICIONE l -RETIRE r

Esta é uma variação do Seqüencial para Frente e do Seqüencial para Trás. Esse procedimento aplica o SF até que l atributos tenham sido adicionados, então aplica o ST até que r atributos sejam eliminados ($l > r$). O procedimento é repetido até que o número de atributos desejado seja obtido. A vantagem dessa variação é que ela é capaz de eliminar atributos que não sejam mais necessários devido a combinações de atributos que tenham sido adicionados subsequentelemente.

ALGORITMOS GENÉTICOS

Os Algoritmos Genéticos (AGs) representam um método de busca que reflete de uma maneira primitiva algo do processo natural evolutivo dos seres vivos. Os AGs proporcionam

mecanismos efetivos de busca em problemas de classificação e otimização.

Esta técnica evolucionária trabalha com uma população de pontos em vez de um único ponto. Cada um deles na verdade é um vetor que representa uma solução candidata para o problema de otimização em questão. Uma população é um conjunto de tais vetores. Cada vetor, também chamado indivíduo ou cromossomo, pode ser codificado com um número arbitrário de bits para representar o valor de interesse. Essa representação é específica de cada aplicação. Quando aplicado à Seleção de Atributos, por exemplo, cada bit de um cromossomo pode representar um atributo, e o comprimento dos cromossomos será igual ao número de atributos do vetor de entrada. Um bit com valor “1” indica que o atributo correspondente àquele bit está sendo usado naquela solução candidata. Um bit com valor “0” indica que o atributo correspondente àquele bit não está sendo usado naquela solução candidata. Como exemplo, considere um problema de Seleção de Atributos (SA) de um vetor originalmente de dimensão $n = 20$, como na Figura 3.5. O cromossomo representado indica que $d = 10$ e os atributos 3, 5, 6, 8, 11, 12, 14, 15, 16, 18 estão selecionados.

x_1	x_2	x_3	x_4	x_5	x_6	x_7	x_8	x_9	x_{10}	x_{11}	x_{12}	x_{13}	x_{14}	x_{15}	x_{16}	x_{17}	x_{18}	x_{19}	x_{20}
0	0	1	0	1	1	0	1	0	0	1	1	0	1	1	1	0	1	0	0

Figura 3.5 - Cromossomo representativo de uma solução candidata para Seleção de Atributos

O número total de bits define o tamanho do espaço de busca. Com um cromossomo de n bits há 2^n possibilidades de combinações a serem examinadas.

As operações executadas ao se implementar o AG genético “padrão” são as seguintes [El-Sharkawi, 2000], [Goldberg, 1989], [Holland, 1975]:

1. Inicializar a população;
2. Calcular a função de avaliação para cada indivíduo da população;
3. Selecionar indivíduos que servirão como reprodutores;
4. Executar Cruzamento e Mutação na população; e

5. Voltar ao Passo 2 caso a solução ainda não seja satisfatória.

A inicialização da população normalmente é feita de maneira aleatória, apesar de que às vezes pode ser útil iniciar com um ou mais indivíduos selecionados heurísticamente. Qualquer que seja o procedimento utilizado, a população deve representar uma grande variedade de indivíduos para enriquecer o processo de busca.

Um AG pode utilizar qualquer função de avaliação que seja apropriada para o problema em questão. No caso da SA, uma das funções mostradas na Seção 3.2 pode ser escolhida. A escolha da função de avaliação é importante para o bom desempenho do processo de busca e também para se atingir um bom desempenho computacional. Algumas funções são inviáveis para os AGs devido ao longo tempo de processamento que acarretam, por causa do elevado número de cálculos de avaliações, que dependem da dimensão original do vetor de entradas, do tamanho da população e do número de iterações do processo evolucionário. Como já mencionado, os operadores genéticos utilizados nos AGs são reprodução, cruzamento e mutação. Cada um deles será detalhado a seguir.

Seleção

O número esperado de vezes que cada indivíduo da população corrente será selecionado para servir como reprodutor é proporcional à avaliação daquele indivíduo com relação ao valor médio da avaliação da população inteira. O operador seleção normalmente é executado através da “roleta de seleção”. O conceito utilizado é o de uma roleta ou disco cuja superfície é dividida em setores que representam a probabilidade de ser selecionado ou avaliação (relativa) de cada indivíduo. Um ponto na borda da roleta é determinado como marco zero e cada arco em volta do círculo corresponde a uma área de setor proporcional ao valor da avaliação de cada indivíduo. Ao se girar a roleta, como se fosse uma roleta da sorte, e escolher um ponto da borda, o indivíduo correspondente ao setor sorteado é escolhido para a próxima população. Dessa maneira, indivíduos com melhor avaliação serão escolhidos com maior probabilidade. O operador seleção é repetido até que o número de indivíduos desejado seja escolhido. Um exemplo ilustrativo é mostrado na Tabela 3.1, em que é aplicado o operador seleção por oito vezes para gerar uma população de oito indivíduos a partir da população inicial.

Com a utilização da roleta de seleção, pode ser que o indivíduo que tenha o maior valor de avaliação não seja selecionado em uma determinada geração, ou ainda que ele seja modificado depois da execução dos operadores cruzamento e mutação. Muitas vezes é útil utilizar uma estratégia “elitista”, que garanta a sobrevivência do indivíduo com maior avaliação ou de um grupo de indivíduos com os maiores valores de avaliação. Isso é feito simplesmente copiando tais indivíduos para a geração seguinte depois que todos os operadores genéticos tenham sido executados.

TABELA 3.1

EXEMPLO ILUSTRATIVO DO OPERADOR SELEÇÃO

População Inicial	Função de Avaliação	Função de Avaliação Normalizada (Probabilidade de Seleção)	População Depois da Seleção
1 0 1 1 1 1 0 1	0.733	0.144	0 1 1 0 0 0 1 1
1 1 0 1 1 0 0 0	0.471	0.093	0 0 1 1 0 1 0 1
0 1 1 0 0 0 1 1	0.937	0.184	1 1 0 1 1 0 0 0
1 1 1 0 1 1 0 0	0.243	0.048	1 0 1 0 1 1 1 0
1 0 1 0 1 1 1 0	0.845	0.166	0 1 0 0 1 0 1 0
0 1 0 0 1 0 1 0	0.788	0.155	1 0 1 0 1 1 1 0
0 0 1 0 0 0 1 1	0.416	0.082	0 1 1 0 0 0 1 1
0 0 1 1 0 1 0 1	0.650	0.128	1 0 1 1 1 1 0 1

Cruzamento

Este operador é baseado na metáfora de combinação sexual em que os indivíduos são gerados pela combinação genética dos pais. Um dos atributos de um AG é a probabilidade de cruzamento, que determina para cada grupo de dois indivíduos se será aplicado o operador ou não.

O tipo de cruzamento mais comum é aquele executado em um ponto do cromossomo, que significa escolher uma posição do cromossomo aleatoriamente e trocar os bits que estejam à direita daquele ponto entre os cromossomos que estão sendo cruzados. A Figura 3.6 ilustra o operador cruzamento.

1 0 1 1 0 0 1 0	1 0 1 1 0 1 0 0
0 1 0 0 1 1 0 0	0 1 0 0 1 0 1 0
(a)	(b)

Figura 3.6 – Cruzamento em um ponto: (a) antes e (b) depois do cruzamento

Mutação

A mutação é a mudança aleatória de bits que pode ocorrer em cada geração. Essa operação é executada bit a bit na população inteira. Geralmente é feita com uma baixa probabilidade de ocorrência, por exemplo, 0,01%, 1% ou 10%.

A mutação é importante para adicionar variedade à população e levar a soluções que dificilmente seriam atingidas somente com os outros operadores. Mas o valor da probabilidade de mutação não pode ser muito alto para não transformar o algoritmo genético em uma busca aleatória.

3.4 – Outros Métodos de Extração de Atributos

Apesar de o assunto de Extração de Atributos ser estudado há muito tempo [Cherkassky, 1998], [Duda, 1973], ainda não há consenso quanto aos vários métodos existentes para as diferentes aplicações. Novas propostas e alternativas de EA têm surgido continuamente.

Algumas técnicas de EA para classificação via RNs existentes na literatura são:

- Poda de Nós (*Node Pruning*) [Bahbah, 1999],[Chung, 1997],[Jianchang, 1994],[Lee, 1994];
- Algoritmos Genéticos Rápidos [Brill, 1992],[Jain, 1997],[Raudys, 1992];
- Karhunen-Loève Modificado [Zayan, 1996].

As várias alternativas existentes da técnica de Poda de Nós para a SA utilizam o processo de treinamento da RN para determinar a importância dos atributos de entrada. Em [Bahbah, 1999] são utilizadas medidas de sensibilidade da saída com relação aos atributos de entrada num problema de Análise de Estabilidade Transitória de Sistemas de Potência. O método proposto é aplicado num sistema de nove barras e três geradores, e apresenta uma

análise qualitativa sobre os tipos de entradas mais apropriadas para a Análise de Estabilidade Transitória (AET). Em [Lee, 1994], uma RN de uma camada linear é treinada através do chamado *Pocket Algorithm* e seleciona d -melhores características baseado na máxima distância entre os pontos médios de cada classe. A técnica é aplicada para reconhecimento de assinaturas. Em [Chung, 1997], critérios de avaliação para poda de nós tais como Saliência de Nó, Sensibilidade de Nós, Medida de Contribuição, Magnitude de Pesos são comparados entre si, e aplicados ao reconhecimento de letras e números. Em [Jianchang, 1994], a poda de nós também é utilizada baseada na Saliência de Nós e aplicada ao reconhecimento de letras e números. Uma vantagem da técnica Poda de Nós é que ela otimiza simultaneamente o conjunto de características e o classificador. Porém, isso pode tornar-se uma desvantagem, se ela for aplicada a problemas que apresentem um conjunto de atributos iniciais muito grande, tornando-se impraticável devido ao longo tempo de treinamento da RN.

Em [Brill, 1992] é proposta uma técnica de seleção de atributos baseada em Algoritmos Genéticos com uma configuração rápida baseada em amostragem do conjunto de treinamento e avaliação parcial para diminuir o tempo de processamento. Esse tipo de método aproximado é criticado como inadequado devido à alta probabilidade de resultados incorretos [Jain, 1997],[Raudys, 1992].

Em [Zayan, 1996] a técnica de Karhunen-Loève Modificada é aplicada à AET de Sistemas de Potência. A EA é realizada para conjuntos de 30 atributos de entrada num caso e 32, em outro. Há uma redução do número de atributos para 26 e 29, respectivamente.

3.5 – Considerações Finais

Um método de SA ideal seria composto de uma busca exaustiva para procurar entre todas as possíveis combinações de atributos de entrada, tendo o próprio classificador como função de avaliação. Esse método proporcionaria uma confiabilidade imbatível na escolha dos atributos, se não fosse computacionalmente proibitivo. Tendo uma RN como classificador, por exemplo, seria muito difícil utilizá-la como a própria função de avaliação, devido ao elevado tempo de treinamento que geralmente é necessário. Mesmo utilizando um método de busca simples, como o Sequencial para Frente ou para Trás, em vez de uma busca exaustiva, seria difícil

executar tal SA.

Portanto, do ponto de vista prático, na SA, normalmente utiliza-se um método de busca computacionalmente tratável e uma função de avaliação de natureza diferente do próprio classificador para contornar o problema do tempo de processamento elevado.

Neste Capítulo foram expostos métodos de busca e funções de avaliação que podem ser usados para SA, além de técnicas de EA que podem ser usadas na tarefa de classificação via RNs. Essa exposição incluiu técnicas clássicas e algumas mais recentes contidas na literatura. Nos Capítulos seguintes serão desenvolvidas técnicas de EA baseadas em algumas das metodologias expostas neste Capítulo, de modo a generalizar os conceitos expostos. Essas técnicas serão desenvolvidas com o enfoque da aplicação de tais conceitos à AET, utilizando RNs do tipo *Perceptron* Multicamadas e Máquinas de Vetores Suporte. Serão analisados os efeitos da combinação de tais técnicas de EA com os modelos neurais utilizados, considerando as vantagens e desvantagens intrínsecas de cada modelo para a classificação de estabilidade de sistemas de potência de grande porte.

Antes disso, é preciso esclarecer e revisar os conceitos teóricos sobre as MVS, para que suas potencialidades possam ser mais bem entendidas e exploradas, bem como as implicações de sua utilização frente à Maldição da Dimensionalidade, capacidade de generalização e tempos de treinamento. No Capítulo 4 é apresentada uma descrição teórica sobre as MVS, contendo detalhes conceituais que são analisados em termos de vantagens e desvantagens, com vistas à aplicação em AET.

Máquinas de Vetores Suporte

As Máquinas de Vetores Suporte (MVS) ou Redes de Vetores Suporte (RVS) correspondem a um novo tipo de classificador não-linear que possui características teóricas muito interessantes. Elas utilizam transformações não-lineares do espaço de entrada original para um espaço intermediário de elevada dimensão, de onde realizam classificações (lineares) através de hiperplanos de separação. A motivação para tal abordagem vem do fato que hiperplanos classificadores nesse espaço expandido correspondem a superfícies de decisão não-lineares no espaço de entrada, permitindo uma análise formal relativamente simples de tal modelo não-linear.

Nesta tese as MVS são consideradas como um modelo de Redes Neurais (Redes de Vetores Suporte), seguindo a definição apresentada em [Haykin, 1999]:

Uma Rede Neural é um processador paralelo distribuído e massivo constituído de unidades processadoras simples, que têm uma propensão natural para armazenar conhecimento experimental e torná-lo disponível para uso. Ela assemelha-se ao cérebro em dois aspectos:

- O conhecimento proveniente do meio ambiente é adquirido pela rede através de um processo de aprendizado;
- Os pesos das conexões entre os neurônios, chamados de pesos sinápticos, são utilizados para armazenar o conhecimento necessário.

As MVS surgiram a partir de resultados provenientes da chamada Teoria do Aprendizado Estatístico (*Statistical Learning Theory*) [Vapnik, 1998]. Esta teoria generaliza

formalmente o princípio da minimização do risco empírico, isto é, o princípio de aprendizado de máquina que procura minimizar o erro de treinamento durante a estimação do modelo. É sabido que esse princípio por si só não consegue estimar o modelo de melhor capacidade de generalização e normalmente, na prática, técnicas heurísticas são usadas ao longo do treinamento para estimar a máquina com complexidade adequada ao problema em questão. A partir do princípio da minimização do risco empírico, a Teoria do Aprendizado Estatístico deriva formalmente o princípio do risco estrutural, em que a esperança matemática do erro verdadeiro (esperança matemática do erro de classificação calculada ao longo de toda a população que origina o conjunto de treinamento) é minimizada durante o treinamento e durante a estimação de máquinas de aprendizado em geral. O princípio do risco estrutural foi utilizado no desenvolvimento das Redes de Vetores Suporte num esforço de eliminar essas técnicas heurísticas através de um procedimento estruturado para a estimação do modelo não-linear.

Neste Capítulo as MVS são explicadas com o objetivo de proporcionar um bom entendimento de seus detalhes conceituais, e também são analisadas em termos de vantagens e desvantagens. O objetivo é a aplicação na AET de sistemas de potência.

Este capítulo inicia-se com uma revisão sobre a Teoria do Aprendizado Estatístico na Seção 4.1, contendo uma motivação teórica para o surgimento das MVS. Na Seção 4.2, as MVS do tipo linear são analisadas para o problema de classificação separável, em que é possível classificar perfeitamente os dados de treinamento através de um hiperplano de separação. Na Seção 4.3 essa idéia é generalizada para o caso não separável. E na Seção 4.4 é introduzida a utilização das Funções *Kernel* como elemento não linear das MVS. Algumas considerações complementares são reunidas na Seção 4.5. Apesar de as MVS poderem ser utilizadas em problemas de regressão, neste Capítulo será considerada somente sua aplicação na classificação em duas classes.

4.1 – Teoria do Aprendizado Estatístico (*Statistical Learning Theory*)

Nesta seção será considerado o problema da classificação em duas classes. Considere um conjunto de treinamento com N observações, exemplos, ou padrões de treinamento, contendo o seguinte [Burgess, 1998]:

- Vetores, ou padrões de entrada, $\mathbf{x}_i \in R^n, i = 1, \dots, N$;
- E a classe à qual pertencem $y_i \in \{-1, +1\}$.

Assumindo que esses dados tenham sido gerados a partir de uma função densidade de probabilidade, $p(\mathbf{x}, y)$ (embora sua forma seja desconhecida), e assumindo uma máquina de aprendizado qualquer cuja tarefa é aprender o mapeamento representado por $\mathbf{x}_i \mapsto y_i$, essa máquina pode ser representada por um conjunto de possíveis funções candidatas, $\mathbf{x} \mapsto f(\mathbf{x}, \beta)$, sendo β um vetor de parâmetros do qual depende o conjunto de funções. Uma vez determinado o valor de β , determina-se uma função específica e tem-se uma máquina treinada. Pode-se citar dois exemplos de conjuntos de funções:

O valor esperado do erro de teste para uma máquina treinada, $f(\mathbf{x}, \beta)$, é dado por

$$R(\beta) = \int \frac{1}{2} |y - f(\mathbf{x}, \beta)| dP(\mathbf{x}, y) = \int \frac{1}{2} |y - f(\mathbf{x}, \beta)| p(\mathbf{x}, y) dx \cdot dy \quad (4.1)$$

também chamado de “risco esperado” ou “risco verdadeiro”. Esse é o valor esperado da função de erro ou de custo, tomado ao longo de todos os possíveis valores de \mathbf{x} e y .

O “risco empírico” ou “erro de treinamento” de uma máquina treinada, $f(\mathbf{x}, \beta)$, é o valor médio da função de custo tomado ao longo do conjunto de treinamento:

$$R_{emp}(\beta) = \frac{1}{2N} \sum_{i=1}^N |y_i - f(\mathbf{x}_i, \beta)| \quad (4.2)$$

Um classificador MVS minimiza o erro de generalização ao otimizar a relação entre o erro de treinamento e a chamada dimensão Vapnik-Chervonenkis (dimensão VC). Este é um novo conceito de medida de complexidade que pode ser usado para diferentes tipos de funções.

Existe um limite teórico formal para a capacidade de generalização de uma MVS (medida pelo risco verdadeiro) que depende da quantidade de erros de treinamento, $R_{emp}(\beta)$, do tamanho do conjunto de treinamento, N , da dimensão VC associada ao classificador resultante, h , e de uma medida de confiança do próprio limite, η , que deve ser escolhida. Esse limite é dado pelo seguinte teorema [Vapnik, 1998]:

Escolhendo η tal que $0 \leq \eta \leq 1$, então com probabilidade $1 - \eta$, o seguinte limite superior é verdadeiro:

$$R(\beta) \leq R_{emp}(\beta) + \sqrt{\frac{h(\log(2Nh) + 1) - \log(\eta/4)}{N}} \quad (4.3)$$

A dimensão VC, h , é um inteiro não negativo que indica a capacidade da família de funções $f(\mathbf{x}, \beta)$. Portanto, a Teoria do Aprendizado Estatístico deriva a desigualdade (4.3) com o significado de que a capacidade de generalização de um classificador é medida por um limite superior do erro verdadeiro, dado pelo lado direito de (4.3), e esse limite superior é válido com probabilidade $1 - \eta$. À medida que h aumenta, a primeira parcela do limite superior (4.3) diminui e a segunda parcela aumenta, pois existe um compromisso entre os dois termos (erros de treinamento e complexidade). Nota-se que o cálculo do Risco R não exige o conhecimento prévio da função densidade de probabilidade dos dados.

Caso seja possível calcular h , pode-se utilizar um procedimento estruturado para escolher a máquina treinada que possua o maior poder de generalização possível. Dadas diferentes máquinas de aprendizado (diferentes famílias de funções, $f(\mathbf{x}, \beta)$), para um dado valor de η , pode-se obter a máquina com o menor valor do limite do risco verdadeiro, tomando-se uma máquina específica que minimize o lado direito de (4.3).

Definição da Dimensão VC

A dimensão VC é uma propriedade de um conjunto de funções, $f(x, \beta)$. Dado um conjunto de treinamento com N exemplos e a cada um podendo ser dada uma classificação, $y_i \in \{-1, +1\}$, então há 2^N maneiras diferentes de se classificar os dados de treinamento, de maneira semelhante à mostrada na Figura 4.1. Para cada uma das 2^N possíveis partições, se sempre for possível escolher uma função membro do conjunto $f(x, \beta)$ (determinando um valor específico de β) tal que ela classifique corretamente a partição, então diz-se que o conjunto de treinamento é “dicotomizado” pelo conjunto de funções. A dimensão VC é definida como o número máximo de exemplos de treinamento que podem ser “dicotomizados” pela família de funções em questão. Isso quer dizer que se a dimensão VC for igual a h , então há pelo menos um conjunto de h exemplos que pode ser quebrado pelo conjunto de funções. Mas isso não significa que necessariamente todos os conjuntos de h exemplos podem ser quebrados.

x_1	x_2	...	x_N
+1	-1	...	+1

Figura 4.1 - Conjunto de treinamento

Exemplo 4.3

Tome-se um exemplo com 3 padrões de treinamento, $x \in R^N$, e o conjunto de funções, $f(x, \beta)$, como um conjunto de retas orientadas classificadoras dos pontos em duas classes. Como mostrado na Figura 4.2, cada valor de β define a posição e o ângulo de orientação das retas, isto é, define uma função classificadora diferente. Como há 3 exemplos de treinamento, há 8 maneiras diferentes como esses pontos podem ser organizados nas duas classes existentes. Em todas as maneiras, é possível encontrar uma reta orientada que classifique corretamente os exemplos. Portanto, diz-se que o conjunto de treinamento é “dicotomizado” pelo conjunto de funções.

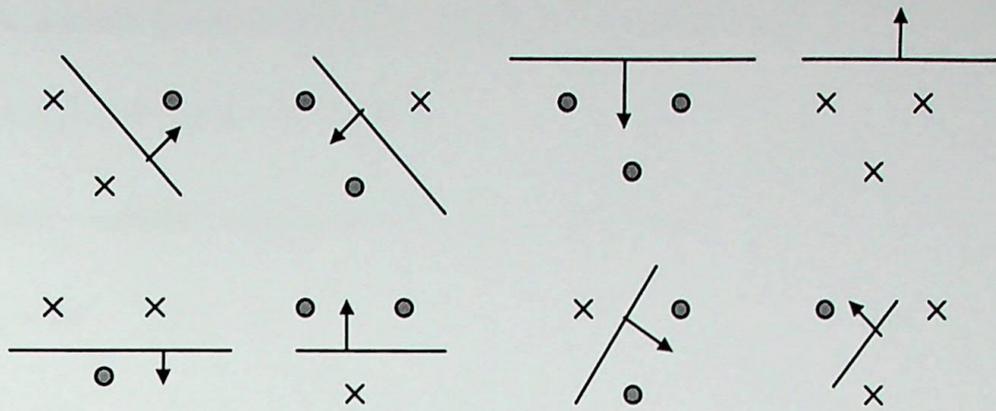


Figura 4.2 - Exemplo de retas orientadas classificadoras

Caso se considerasse 4 exemplos de treinamento, haveria 16 maneiras diferentes de organizá-los como elementos das duas classes. E neste caso, há pelo menos uma dessas combinações que não pode ser classificada corretamente por nenhuma reta orientada, como mostrado na Figura 4.3. Portanto o conjunto de 4 pontos não pode ser quebrado pelo conjunto de funções, e a dimensão VC do conjunto de retas orientadas é igual a 3.

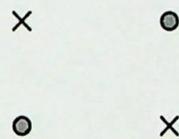


Figura 4.3 – 4 Exemplos de treinamento organizados em duas classes

Apesar da dimensão VC oferecer uma idéia da capacidade de um conjunto de funções, ela contradiz a noção intuitiva de que máquinas de aprendizado com um elevado número de parâmetros tenham um valor elevado de dimensão VC; ou de que máquinas de aprendizado com um baixo número de parâmetros tenham um valor baixo de dimensão VC.

Exemplo 4.4

Isso é mostrado considerando a função degrau [Burges, 1998]:

$$\theta(z), \quad z \in R: \begin{cases} \theta(z) = 1, & z > 0 \\ \theta(z) = -1, & z \leq 0 \end{cases}$$

e a seguinte família de funções, com apenas um parâmetro:

$$f(x, \beta) = \theta(\sin(\beta \cdot x)), \quad x, \beta \in \mathbb{R}$$

Dado o seguinte conjunto com N pontos de treinamento

$$x_i = 10^{-i}, \quad i = 1, \dots, N$$

então para qualquer classificação que se dê a eles

$$y_1, y_2, \dots, y_N, \quad y_i \in \{-1, +1\}$$

é possível classificar corretamente o conjunto de pontos com a função $f(x, \beta) = \theta(\sin(\beta \cdot x))$ e β dado por

$$\beta = \pi \left(1 + \sum_{i=1}^N \frac{(1 - y_i) 10^i}{2} \right).$$

Portanto esse conjunto de funções com um só parâmetro é capaz de “dicotomizar” o conjunto de treinamento dado. Isso é válido para qualquer N . Ou seja, o valor da dimensão VC de tal conjunto de funções é infinito.

O segundo termo do lado direito da desigualdade (4.3) é chamado de “confiança VC”. Dado um valor de η , é possível obter o valor desse termo em função da dimensão VC, h , e do tamanho do conjunto de treinamento, N . Tomando um valor fixo de $\eta = 0,05$ e $N = 10.000$, à medida que se aumenta o valor da dimensão VC, a Confiança VC varia com h/N como mostrado na Figura 4.4. Observa-se que o valor da Confiança VC é monotônica crescente com h . Por isso, uma dimensão VC que tenha valor baixo com relação ao tamanho do conjunto de treinamento, irá determinar também um baixo valor no segundo termo do lado direito de (4.3). E para um valor aproximadamente igual a 0,37, ela ultrapassa a unidade. A importância desse fato pode ser vista notando-se que o risco verdadeiro (4.1) toma valores somente entre 0 e 1:

$$R(\beta) = \int \frac{1}{2} |y - f(x, \beta)| dP(x, y) = \int \frac{1}{2} |y - f(x, \beta)| p(x, y) dx dy \quad (4.1)$$

Se o valor da Confiança VC for maior que um, o limite superior dado por (4.3) será muito conservativo, isto é, não será muito útil para avaliar a capacidade de generalização da máquina treinada.

$$R(\beta) \leq R_{emp}(\beta) + \sqrt{\frac{h(\log(2N/h) + 1) - \log(\eta/4)}{N}} \quad (4.3)$$

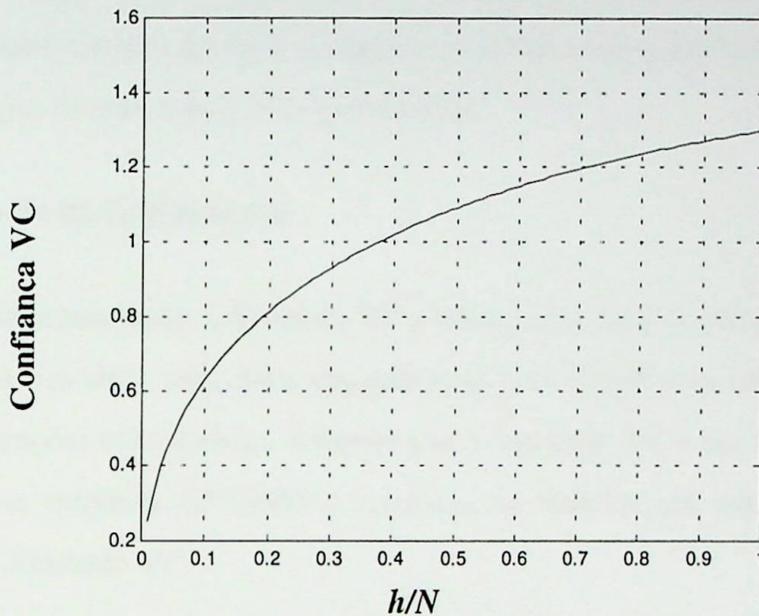


Figura 4.4 – Confiança VC x h/N

Além disso, note-se que o limite superior (4.3) é válido em probabilidade $(1 - \eta)$, não é uma garantia absoluta. Também, uma dada máquina treinada com limite superior menor que outra não necessariamente possui um valor menor do risco verdadeiro. Isso é ilustrado na Figura 4.5, onde são representados os riscos de duas máquinas treinadas, $R_1(\beta)$ e $R_2(\beta)$. $R_1(\beta)$ possui um limite superior menor, localizado pelo ponto fechado na reta, mas apesar disso seu valor verdadeiro, $R_1^*(\beta)$, é maior do que $R_2^*(\beta)$.

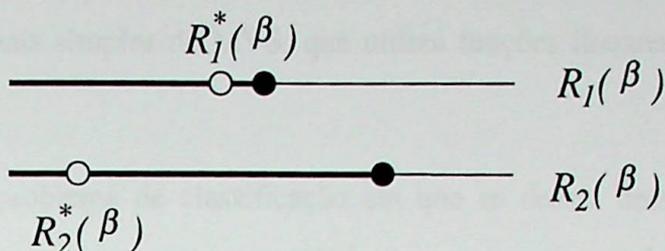


Figura 4.5 – Limites superiores e valores verdadeiros de $R(\beta)$

Portanto, o limite superior do risco verdadeiro é tomado como uma “indicação” da capacidade de generalização de uma máquina de aprendizado.

Minimização do Risco Estrutural

Pelo exposto anteriormente, a dimensão VC é uma propriedade de um conjunto de funções. O risco empírico e o risco verdadeiro dependem de uma função específica escolhida dentre o conjunto de funções consideradas. Sabendo que a dimensão VC é um valor discreto, pode-se introduzir uma estrutura dividindo o conjunto de funções em subconjuntos de valores crescentes da dimensão VC.

Pode-se então treinar várias máquinas, tendo como objetivo de cada treinamento, minimizar o risco empírico. Cada uma dessas máquinas pertencerá a um dos subconjuntos criados. O princípio denominado Minimização do Risco Estrutural consiste em encontrar a máquina treinada que possua o menor valor do limite superior do risco verdadeiro (4.3).

4.2 – Máquinas de Vetores Suporte Lineares – Caso Separável

Recentemente foram criados modelos de RNs, conhecidos como Máquinas de Vetores Suportes, com algoritmos de treinamento que procuram minimizar o limite superior do risco verdadeiro. Neste capítulo será tratado apenas o caso de classificação, embora as MVS também possam ser aplicadas a problemas de regressão.

O caso mais geral de tais máquinas é aquele com funções não lineares e treinadas com um conjunto de dados não separável, isto é, dados para os quais não é possível definir um

hiperplano que separe corretamente todos os exemplos de treinamento. Nesta seção será introduzido o caso mais simples de MVS, que utiliza funções lineares treinadas com dados separáveis.

Considere o problema de classificação em que se deseja determinar uma máquina capaz de aprender o mapeamento representado por um conjunto de treinamento, com o máximo poder de generalização possível, segundo o limite superior do risco verdadeiro dado na Seção 4.1. O conjunto de treinamento é dado por

$$\{\mathbf{x}_i, y_i\}_{i=1}^N, \quad \mathbf{x}_i \in R^n, \quad y_i \in \{-1, 1\}$$

As MVS lineares são uma família de funções baseadas em hiperplanos de separação usados para classificar os dados de treinamento. Pode-se representar tais funções por

$$f(\mathbf{x}) = \text{sign}(\mathbf{w}' \cdot \mathbf{x} + b) \quad (4.4)$$

sendo $\text{sign}(\bullet)$, a função sinal dada por

$$\text{sign}(z) = \begin{cases} +1, & z > 0 \\ -1, & z < 0 \end{cases}, \quad z \in R \quad (4.5)$$

Em (4.4), os pontos $\mathbf{x} \in R^n$ para os quais $\mathbf{w}' \cdot \mathbf{x} + b = 0$ definem o hiperplano de separação, como mostrado na Figura 4.6. Determinando o hiperplano que maximize a margem de separação, ρ , é intuitivamente esperado que o classificador tenha melhor capacidade de generalização.

Como os exemplos de treinamento são separáveis, é possível encontrar um hiperplano que classifique todos os pontos corretamente, tal que satisfaça as seguintes condições:

$$\begin{aligned} \mathbf{w}' \cdot \mathbf{x}_i + b &> 0 \text{ se } y_i = +1 \\ \mathbf{w}' \cdot \mathbf{x}_i + b &< 0 \text{ se } y_i = -1 \end{aligned}, \quad \forall i \quad (4.6)$$

Sempre é possível multiplicar \mathbf{w} e b por um escalar apropriado e transformar as condições (4.6) na seguinte forma:

$$\begin{aligned} w' \cdot x_i + b &\geq +1 \text{ se } y_i = +1 \\ w' \cdot x_i + b &\leq -1 \text{ se } y_i = -1 \end{aligned}, \quad \forall i \tag{4.7}$$

Essas condições podem ser combinadas num conjunto de desigualdades:

$$y_i(w' \cdot x_i + b) \geq 1 \Leftrightarrow y_i(w' \cdot x_i + b) - 1 \geq 0, \quad \forall i \tag{4.8}$$

Como mostrado na Figura 4.6, o vetor w é perpendicular ao hiperplano de separação. Os pontos para os quais

$$w' \cdot x_i + b = 1 \text{ e } w' \cdot x_i + b = -1 \tag{4.9}$$

(como x_1 e x_2 , na Figura 4.6), bem como a posição do hiperplano de separação, definem a margem de separação. Esses pontos são chamados de *vetores suporte*.

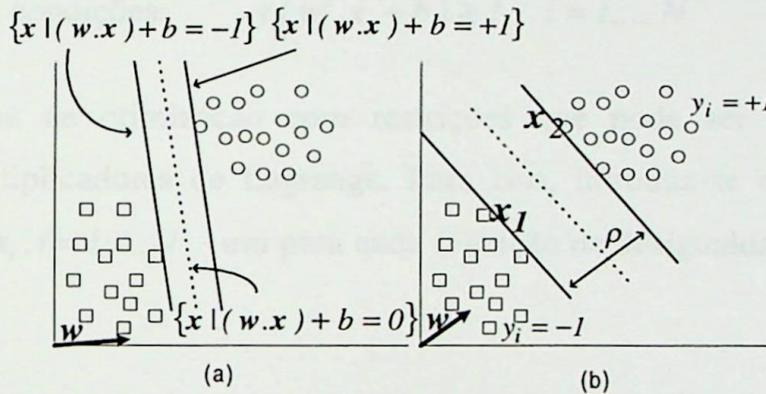


Figura 4.6 – Conjunto de treinamento separável e hiperplano de separação

Usando os vetores suporte x_1 e x_2 da Figura 4.6 e as equações (4.9), pode-se concluir que a margem de separação é dada por:

$$\rho = \frac{w' \cdot (x_2 - x_1)}{\|w\|} = \frac{2}{\|w\|} \tag{4.10}$$

Para dados linearmente separáveis, a dimensão VC do classificador MVS pode ser avaliada por [Vapnik, 1998]:

$$h < \min \left\{ n, \frac{4D^2}{\rho^2} \right\} + 1 = \min \left\{ n, D^2 \|w\|^2 \right\} + 1 \tag{4.11}$$

sendo n , a dimensão dos dados de treinamento, e D o mínimo raio de uma hipersfera que contenha os pontos de treinamento. Como é possível determinar um classificador com erro de treinamento zero, $R_{emp}(\beta) = 0$, o risco (4.1) pode ser diminuído diminuindo a complexidade da MVS, isto é, diminuindo h , que é equivalente a aumentar a margem de separação, ρ , ou a diminuir $\|\mathbf{w}\|$.

Portanto, para maximizar a margem de separação, deve-se determinar o hiperplano de separação definido pelo vetor \mathbf{w} que tenha a menor norma Euclideana possível, ao mesmo tempo que satisfaça as condições (4.8). Para encontrar o hiperplano de separação com máxima margem, pode-se formular o seguinte problema de otimização:

$$\begin{aligned} - \text{ Minimizar:} & \quad \tau(\mathbf{w}) = \frac{1}{2} \|\mathbf{w}\|^2; \\ - \text{ Sujeito às condições:} & \quad y_i(\mathbf{w}^t \cdot \mathbf{x}_i + b) \geq 1, \quad i = 1, \dots, N \end{aligned} \quad (4.12)$$

Este é um problema de otimização com restrições que pode ser resolvido através da formulação dos multiplicadores de Lagrange. Para isso, introduz-se os multiplicadores de Lagrange positivos $\alpha_i, i = 1, \dots, N$ - um para cada restrição de desigualdade - e o Lagrangiano torna-se

$$L(\mathbf{w}, b, \alpha) = \frac{1}{2} \|\mathbf{w}\|^2 - \sum_{i=1}^N \alpha_i y_i (\mathbf{w}^t \cdot \mathbf{x}_i + b) + \sum_{i=1}^N \alpha_i \Leftrightarrow L(\mathbf{w}, b, \alpha) = \frac{1}{2} \|\mathbf{w}\|^2 - \sum_{i=1}^N \alpha_i (y_i (\mathbf{w}^t \cdot \mathbf{x}_i + b) - 1) \quad (4.13)$$

O Lagrangiano (4.13) deve ser minimizado com relação a \mathbf{w} e b , e maximizado com relação aos multiplicadores de Lagrange, α_i . O ponto ótimo localiza-se num ponto de sela [Cristianini, 2000], no qual as seguintes condições são satisfeitas:

$$\frac{\partial L(\mathbf{w}, b, \alpha)}{\partial b} = 0 \Rightarrow \sum_{i=1}^N \alpha_i y_i = 0 \quad (4.14)$$

$$\frac{\partial L(\mathbf{w}, b, \alpha)}{\partial \mathbf{w}} = 0 \Rightarrow \mathbf{w} = \sum_{i=1}^N \alpha_i y_i \mathbf{x}_i \tag{4.15}$$

Ao substituir (4.14) e (4.15) em (4.12), pode-se formular o problema de otimização dual, equivalente ao problema (4.12):

- Maximizar:
$$W(\alpha) = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i,j=1}^N \alpha_i \alpha_j y_i y_j (\mathbf{x}_i \cdot \mathbf{x}_j)$$

- Sujeito às condições:
$$\alpha_i \geq 0, i = 1, \dots, N \text{ e } \sum_{i=1}^N \alpha_i y_i = 0 \tag{4.16}$$

Resolvendo o problema de otimização quadrática (4.16) e determinando os valores ótimos de $\alpha_i, i = 1, \dots, N$, o vetor \mathbf{w} é obtido a partir de (4.15). O escalar b pode ser calculado usando as igualdades (4.6), que são válidas para qualquer vetor suporte:

$$b = y_{VS} - \mathbf{w}^t \cdot \mathbf{x}_{VS} \tag{4.17}$$

Os vetores suporte são os pontos do conjunto de treinamento para os quais $\alpha_i > 0$, isto é, os pontos que determinam o hiperplano de máxima margem de separação.

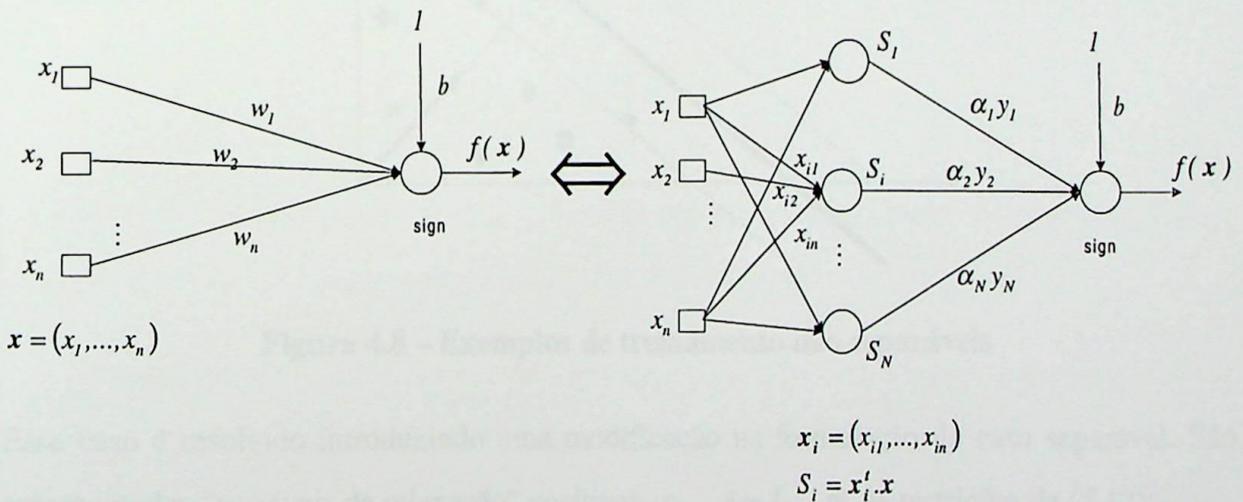


Figura 4.7 – Máquina de Vetores Suporte linear – caso separável

A função de classificação (4.4) pode ser escrita em função dos multiplicadores α_i como

$$f(x) = \text{sign}(w' \cdot x + b) = \text{sign}\left(\sum_{i=1}^N \alpha_i y_i (x_i' \cdot x) + b\right) \quad (4.18)$$

O classificador linear MVS, para o caso separável, é ilustrado na Figura 4.7.

4.3 – Máquinas de Vetores Suporte Lineares – Caso Não Separável

No caso não separável, os exemplos do conjunto de treinamento não podem ser todos classificados corretamente por um hiperplano de separação, como ilustrado na Figura 4.8. Se fosse aplicada a otimização da Seção 4.2 (caso separável), nunca seria encontrada uma solução porque seria impossível satisfazer as condições de restrição de (4.12).

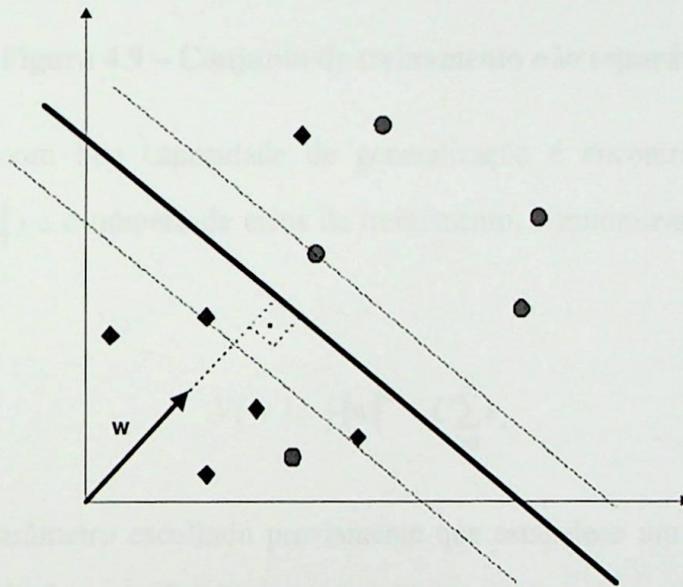


Figura 4.8 – Exemplos de treinamento não separáveis

Esse caso é resolvido introduzindo uma modificação na formulação do caso separável. São acrescentadas “variáveis de relaxação” positivas, ϵ_i , $i = 1, \dots, N$ às restrições de (4.12):

$$\begin{aligned} w' \cdot x_i + b &\geq 1 - \epsilon_i \text{ para } y_i = +1 \\ w' \cdot x_i + b &\leq -1 + \epsilon_i \text{ para } y_i = -1 \\ \epsilon_i &\geq 0, \quad i = 1, \dots, N \end{aligned} \quad (4.19)$$

Dessa maneira é permitido ao classificador cometer alguns erros de classificação. Esses erros acontecem com os pontos para os quais $\epsilon_i > 1$, como ilustrado na Figura 4.9.

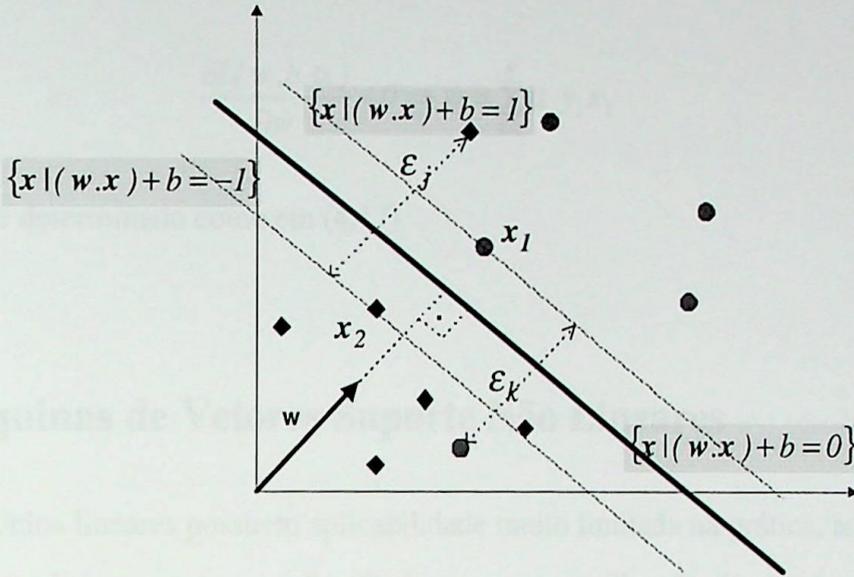


Figura 4.9 – Conjunto de treinamento não separável

Um classificador com boa capacidade de generalização é encontrado controlando-se a capacidade (via $\|w\|$) e o número de erros de treinamento, e minimizando a seguinte função objetivo:

$$V(w) = \frac{1}{2} \|w\|^2 + C \sum_{i=1}^N \epsilon_i \tag{4.20}$$

sendo $C > 0$ um parâmetro escolhido previamente que estabelece um compromisso entre a complexidade da máquina e o número de erros de treinamento.

Introduzindo multiplicadores de Lagrange e seguindo passos semelhantes aos da Seção 4.2, chega-se a um problema de otimização também bastante semelhante, que é:

- Maximizar:
$$W(\alpha) = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i,j=1}^N \alpha_i \alpha_j y_i y_j (x_i \cdot x_j)$$
- Sujeito às condições:
$$0 \leq \alpha_i \leq C, \quad i = 1, \dots, N \quad \text{e} \quad \sum_{i=1}^N \alpha_i y_i = 0 \tag{4.21}$$

A única diferença do problema de otimização (4.21) para o problema (4.16) do caso separável são as condições de restrição, com o novo limite superior, C . Novamente, a solução leva à determinação dos valores ótimos de α_i e do hiperplano de separação dado por

$$\frac{\partial L(\mathbf{w}, b, \alpha)}{\partial \mathbf{w}} = 0 \Rightarrow \mathbf{w} = \sum_{i=1}^N \alpha_i y_i \mathbf{x}_i \quad (4.15)$$

O valor de b é determinado como em (4.17).

4.4 – Máquinas de Vetores Suporte Não Lineares

Como os modelos lineares possuem aplicabilidade muito limitada na prática, as MVS lineares podem ser estendidas para sua versão não-linear com a utilização de um mapeamento não linear dos dados de entrada do espaço original, R^n , para um espaço de maior dimensão, R^m :

$$\begin{aligned} \Phi : R^n &\mapsto R^m \\ \mathbf{x} &\mapsto \Phi(\mathbf{x}) = (\phi_1(\mathbf{x}), \dots, \phi_m(\mathbf{x})) \end{aligned} \quad (4.22)$$

sendo $m > n$. Então, o classificador de máxima margem pode ser determinado no novo espaço expandido. Com esse procedimento, dados que a princípio sejam não separáveis podem tornar-se separáveis no novo espaço, $\Phi(\mathbf{x})$. O passo seguinte é estimar a MVS através da minimização de

$$\tau(\mathbf{w}) = \frac{1}{2} \mathbf{w}' \cdot \mathbf{w} = \frac{1}{2} \|\mathbf{w}\|^2 \quad (4.23)$$

sujeito às restrições de que todos os padrões de treinamento sejam corretamente classificados:

$$y_i (\mathbf{w}' \cdot \Phi(\mathbf{x}) + b) \geq 1, \quad i = 1, \dots, N \quad (4.24)$$

No entanto, dependendo do mapeamento não-linear (4.22), os dados de treinamento podem ser ainda não separáveis linearmente, mesmo no espaço expandido. Por isso, em vez de minimizar (4.23), uma nova função objetivo, similar à função (4.20), é utilizada no seguinte problema de otimização:

$$\begin{aligned}
 & \text{- Minimizar: } V(\mathbf{w}, \varepsilon) = \frac{1}{2} \mathbf{w}' \cdot \mathbf{w} + C \sum_{i=1}^N \varepsilon_i \\
 & \text{- Sujeito a: } \begin{aligned} & y_i (\mathbf{w}' \cdot \Phi(\mathbf{x}) + b) \geq 1 - \varepsilon_i, \quad i = 1, \dots, N \\ & \varepsilon_i \geq 0, \quad i = 1, \dots, N \end{aligned}
 \end{aligned} \tag{4.25}$$

na qual foram introduzidas N variáveis de relaxação ε_i para permitir que erros de treinamento existam, isto é, padrões de treinamento para os quais

$$y_i (\mathbf{w}' \cdot \Phi(\mathbf{x}) + b) \geq 1 - \varepsilon_i, \quad \varepsilon_i \geq 0 \tag{4.26}$$

Como anteriormente, em vez de resolver (4.25) diretamente, é mais fácil resolver o problema dual, em termos dos multiplicadores de Lagrange:

$$\begin{aligned}
 & \text{- Maximizar: } W(\alpha) = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i,j=1}^N \alpha_i \alpha_j y_i y_j (\Phi(\mathbf{x}_i)' \cdot \Phi(\mathbf{x}_j)) \\
 & \text{- Sujeito às condições: } 0 \leq \alpha_i \leq C, \quad i = 1, \dots, N \text{ e } \sum_{i=1}^N \alpha_i y_i = 0
 \end{aligned} \tag{4.27}$$

Depois de resolvido o problema de otimização, o classificador não linear terá a forma mostrada na Figura 4.10. É importante reafirmar que somente as parcelas do somatório em (4.28) correspondentes aos vetores suporte (pontos de treinamento próximos da região de separação entre as classes) são diferentes de zero. Todas as outras parcelas do somatório, correspondentes aos demais padrões de treinamento, são iguais a zero.

$$f(\mathbf{x}) = \text{sign}(\mathbf{w}' \cdot \Phi(\mathbf{x}) + b) = \text{sign}\left(\sum_{i=1}^N \alpha_i y_i (\Phi(\mathbf{x}_i)' \cdot \Phi(\mathbf{x})) + b\right) \tag{4.28}$$

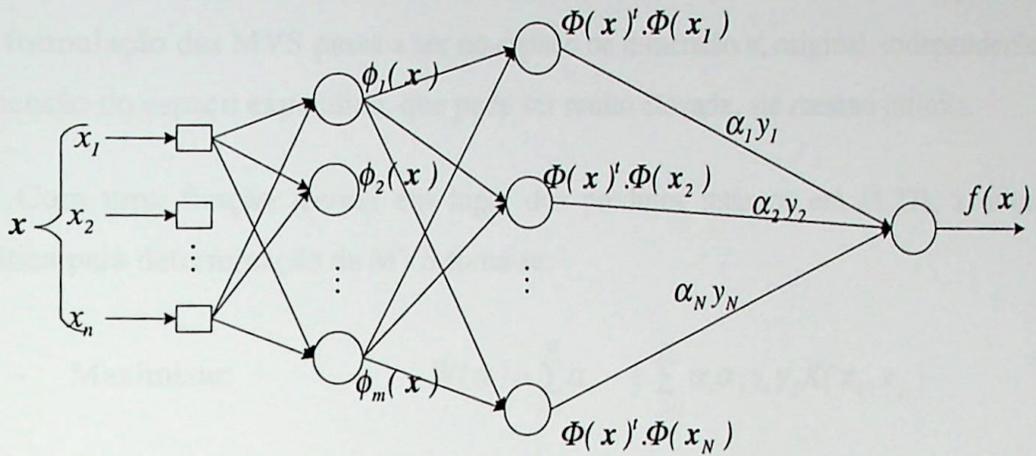


Figura 4.10 – Máquina de Vetores Suporte não linear

Verifica-se através de (4.27) e (4.28) que a modelagem e a estimativa das MVS se dão através de produtos internos, $\Phi(x_i)' \cdot \Phi(x_j)$, entre vetores do espaço expandido, $\Phi(x) \in R^m$. Na prática, esses produtos internos e o próprio mapeamento não-linear (4.22) são obtidos indiretamente através das chamadas funções *Kernel*. Essas funções são equivalentes aos produtos internos de vetores de dados no espaço expandido [Vapnik, 1998]:

$$K(a, b) = \Phi(a)' \cdot \Phi(b), \quad a, b \in R^n \tag{4.29}$$

Exemplo 4.5

No caso de uma transformação não linear como a seguir:

$$\begin{aligned} \Phi : R^2 &\mapsto R^3 \\ x &\mapsto \Phi(x) = [x_1^2, \sqrt{2}x_1x_2, x_2^2] \end{aligned}$$

pode-se calcular seus produtos internos indiretamente utilizando a seguinte função *Kernel*:

$$\Phi(x)' \cdot \Phi(y) = K(x, y) = (x' \cdot y)^2 = x_1^2 y_1^2 + 2x_1 y_1 x_2 y_2 + x_2^2 y_2^2$$

A grande vantagem de se usar a função *Kernel* em lugar dos produtos internos explícitos é que toda a formulação das MVS passa a ser no espaço de dimensão n , original, independentemente da dimensão do espaço expandido, que pode ser muito elevada, até mesmo infinita.

Com uma função *Kernel* em lugar dos produtos internos em (4.27), a otimização quadrática para determinação da MVS torna-se:

- Maximizar:
$$W(\alpha) = \sum_{i=1}^N \alpha_i - \frac{1}{2} \sum_{i,j=1}^N \alpha_i \alpha_j y_i y_j K(x_i, x_j)$$
- Sujeito às condições:
$$0 \leq \alpha_i \leq C, \quad i = 1, \dots, N \quad \text{e} \quad \sum_{i=1}^N \alpha_i y_i = 0 \quad (4.30)$$

E o classificador não linear transforma-se em (Figura 4.11):

$$f(x) = \text{sign}(w' \cdot \Phi(x) + b) = \text{sign}\left(\sum_{i=1}^N \alpha_i y_i K(x_i, x) + b\right) \quad (4.31)$$

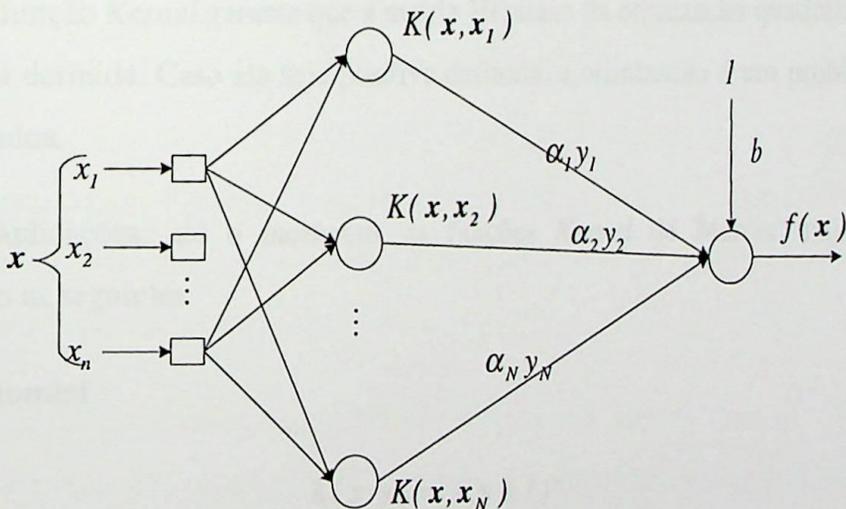


Figura 4.11 – MVS não linear com função *Kernel*

Para que uma função *Kernel* seja considerada equivalente a um produto interno entre vetores no espaço expandido, é preciso que ela satisfaça a chamada condição de Mercer [Cristianini, 2000]:

Existe uma expansão Φ e uma função K que satisfaçam

$$K(\mathbf{x}, \mathbf{y}) = \sum_i \phi_i(\mathbf{x}) \cdot \phi_i(\mathbf{y})$$

se e somente se,

$$\int K(\mathbf{x}, \mathbf{y}) g(\mathbf{x}) g(\mathbf{y}) d\mathbf{x} d\mathbf{y} \geq 0$$

para qualquer $g(\mathbf{x}) \in L_2(\mathbf{x})$.

Nota 6.1: $L_2(\mathbf{x})$ é o espaço Hilbertiano de funções $g(\mathbf{x})$ tais que $\int g(\mathbf{x})^2 d\mathbf{x} < \infty$.

A integral da Condição de Mercer corresponde, no caso de conjuntos de dados finitos $T = \{\mathbf{x}_i, y_i\}_{i=1}^N$, à exigência de que a matriz $\mathbf{K} = (K(\mathbf{x}_i, \mathbf{x}_j))_{i,j=1}^N$ seja positiva semi-definida.

Por isso, tais funções são chamadas *Kernel* de Mercer. O cumprimento da condição de Mercer por parte da função *Kernel* garante que a matriz Hessiana da otimização quadrática (4.30) seja positiva semi-definida. Caso ela seja positiva definida, a otimização é um problema convexo de solução única.

Em aplicações, até o momento, as funções *Kernel* de Mercer mais comumente utilizadas são as seguintes:

Kernel Polinomial

$$K(\mathbf{x}, \mathbf{y}) = (\mathbf{x} \cdot \mathbf{y} + 1)^p \quad (4.32)$$

Esta função *Kernel* corresponde a uma expansão não linear polinomial contendo componentes de até no máximo grau p , parâmetro que deve ser pré-especificado.



Kernel Funções de Base Radial Gaussiana

$$K(\mathbf{x}, \mathbf{y}) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{y}\|^2}{2\sigma^2}\right) \quad (4.33)$$

O parâmetro σ deve ser pré-especificado.

4.5 – Considerações Finais

Em resumo, uma expansão não-linear (4.22) pode ser definida indiretamente através de uma função *Kernel* (não há necessidade de especificá-la explicitamente), por exemplo (4.32) ou (4.33). Essa função pode ser qualquer uma que satisfaça a condição de Mercer. Os parâmetros p e σ das funções *Kernel* apresentadas na Seção 4.4 estão relacionados a parâmetros do mapeamento não-linear para o espaço expandido. Eles afetam a esparsidade e separabilidade dos dados no espaço expandido, e conseqüentemente, afetam a complexidade do classificador MVS resultante e a quantidade de erros de treinamento. O parâmetro C também afeta a complexidade do modelo. Atualmente, não há conhecimento de como ajustar os valores de C , de como escolher a melhor função *Kernel* (a expansão não-linear Φ) e de como ajustar os valores dos parâmetros das funções *Kernel*. Na prática, uma faixa de valores deve ser tentada para C e para os parâmetros da função *Kernel* utilizada, e então o desempenho do classificador MVS é avaliado para cada um desses valores.

A variável da otimização quadrática existente na identificação da MVS tem uma dimensionalidade elevada (dimensionalidade igual ao número de exemplares do conjunto de treinamento) e que demanda uma grande capacidade de memória de armazenamento. Apesar disso, algoritmos de treinamento relativamente rápidos já existem. Novos algoritmos de treinamento têm sido propostos a cada dia porque essa é uma área de pesquisas em andamento [Scholkopf, 1999].

Exemplo 4.6

Um exemplo do cálculo dos limites (4.3) e (4.11) é mostrado na Figura 4.12, onde os conjuntos de treinamento e teste foram obtidos de simulações de estabilidade transitória. Foi utilizada uma MVS com função *Kernel* do tipo Base Radial, com um valor alto de C para que o modelo pudesse atingir uma quantidade de erros de treinamento igual a zero. Para cada valor de σ^2 , variando em valores discretos entre 0 e 1, o conjunto de treinamento foi dividido numa partição aleatória de 10 subconjuntos com tamanhos iguais. Um subconjunto foi utilizado para teste e os outros para treinamento da MVS. O procedimento foi repetido para cada subconjunto de forma a obter 10 valores independentes de erros de classificação. A média e o desvio padrão desses erros de teste foram calculados e mostrados como pontos da curva inferior da Figura 4.12, chamada de “Held-Out Error”. A curva superior da Figura 4.12, chamada de “Risk Bound”, representa o limite superior do risco verdadeiro (4.3), calculado em função da estimativa da dimensão VC (4.11). Neste exemplo, pode-se perceber que apesar de o limite do risco verdadeiro possuir valores relativamente altos, ele prevê bem o comportamento do erro de teste da MVS, visto que as duas curvas são fortemente correlacionadas.

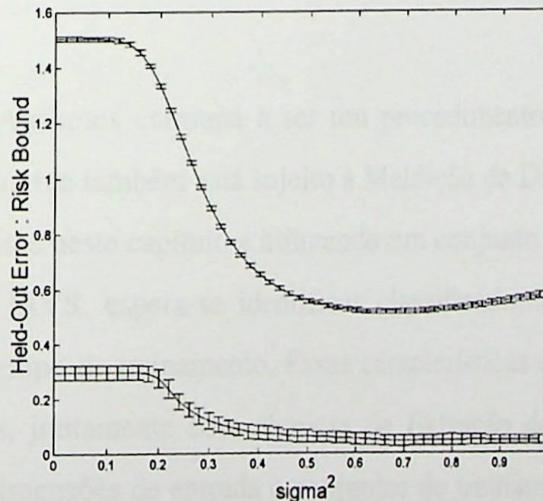


Figura 4.12 – Avaliação do Limite do Risco de MVS

Na prática, a desigualdade (4.11), como estimativa da dimensão VC, h , é válida somente quando os dados são linearmente separáveis no espaço expandido (isto é, quando não se têm

erros de treinamento). Para valores de C e dos parâmetros da função *Kernel* que resultam numa MVS com erros de treinamento, a estimativa do limite do risco verdadeiro (4.3) torna-se difícil porque ele depende da estimativa apresentada pela desigualdade (4.11). No entanto, a utilização de Redes de Vetores Suporte treinadas com os procedimentos mostrados neste capítulo tem-se mostrado satisfatória para aplicações em diversas áreas [Cortes, 1995], [Scholkopff, 1995], [Scholkopff, 1996], [Burgess, 1997], [Blanz, 1996], [Schmidt, 1996], [Principe, 1997], [Joachims, 1998a]. Na maioria desses casos, o desempenho das MVS tem sido igual ou significativamente melhor do que a de outros métodos.

As MVS são capazes de modelar relações não-lineares complexas com bom desempenho e baseiam-se em pontos de treinamento localizados na região de fronteira entre as classes, os chamados vetores suporte. Por isso elas parecem ser muito bem aproveitáveis para a aplicação de AET porque nessas regiões de fronteira a avaliação de estabilidade é crítica.

Uma sólida base teórica das MVS, devido a sua fundamentação na Teoria do Aprendizado Estatístico, permitiu o surgimento de técnicas de treinamento rápidas, mesmo com conjuntos de treinamentos grandes e dimensões de entrada elevadas [Osuna, 1997], [Joachims, 1998b].

A Extração de Atributos continua a ser um procedimento necessário para as MVS, pois esse modelo obviamente também está sujeito à Maldição da Dimensionalidade. Seguindo os procedimentos descritos neste capítulo e utilizando um conjunto de teste independente para avaliar os modelos de MVS, espera-se identificar classificadores com boa capacidade de generalização e baixo tempo de treinamento. Essas características das MVS serão exploradas nos Capítulos seguintes, juntamente com técnicas de Extração de Atributos, para atacar o problema de elevadas dimensões de entrada e conjuntos de treinamento grandes, inerentes ao problema de AET de sistemas de potência.

Metodologia, Aplicação e Resultados

Neste capítulo serão apresentadas as metodologias propostas para a aplicação de RNs à AET de sistemas de potência de grande porte, acoplados a técnicas de extração de atributos. Este capítulo inclui também os resultados obtidos de tal aplicação; os modelos neurais e as técnicas de Extração de Atributos, baseando-se nas idéias dos capítulos anteriores; os sistemas de teste e os arranjos experimentais utilizados para testar e demonstrar as idéias constantes neste trabalho.

5.1 – Modelos Neurais

5.1.1 – Perceptrons Multicamadas

Os modelos neurais utilizados em propostas de AET via RNs baseiam-se quase sempre em Perceptrons Multicamadas (MLPs) [Aresi, 1999], [Bahbah, 1999], [Mansour, 1997a], [Mansour, 1997b], [Omata, 1993], [Sobajic, 1989], [Kamwa, 2001], [Hwang, 1991], [Karapidakis, 2002]. Esse modelo permite que se ajuste a saída da RN para apresentar um resultado em termos de CCT, margem de estabilidade ou informação binária estável/instável. Em qualquer caso, o treinamento é realizado com o algoritmo Retropropagação de Erros. O modelo MLPs é tomado como representante do estado da arte, em termos de aplicação de RNs à AET.

Citadas como grandes desvantagens deste modelo estão a lentidão de treinamento, a carga de empirismo para a estimativa do modelo e a Maldição da Dimensionalidade a que está sujeito. Maiores detalhes sobre as RNs do tipo MLP e o algoritmo Retropropagação de Erros são apresentados no Anexo I.

5.1.2 – Redes de Vetores Suporte

As Redes de Vetores Suporte (RVS) ou Máquinas de Vetores Suporte (MVS) são capazes de mapear relações de entrada/saída não-lineares complexas com boa precisão. Elas parecem ser muito bem aplicáveis ao problema de AET porque são classificadores que se baseiam em pontos de treinamento próximos do limite de separação entre as classes, isto é, os vetores suporte. Considerando que as entradas normalmente são escolhidas como atributos que representam o ponto de operação do sistema, as RVS são classificadores que se baseiam em condições de operação próximas do limite de estabilidade, onde a avaliação de estabilidade é crítica.

O bom desenvolvimento teórico das MVS permitiu o surgimento de técnicas de treinamento rápidas mesmo com conjuntos de treinamento grandes e dimensionalidades de entrada elevadas, que são inerentes à AET via RNs. Portanto, espera-se que a necessidade da Extração de Atributos seja um fator de menor preocupação para as RVS, com relação aos tempos de treinamento. Mesmo assim a redução do número de atributos deve ser testada em termos do desempenho da classificação.

As funções *Kernel* RBF e Polinomial serão utilizadas nas aplicações que se seguem e serão comparadas entre si. A determinação dos parâmetros dessas funções deve ser realizada através de procedimentos heurísticos porque não há indicações teóricas de como isso deve ser feito. Os limites superiores de capacidade de generalização das máquinas de aprendizado que foram apresentados no Capítulo 4 podem ser calculados somente quando os dados de treinamento são totalmente separáveis. Essa premissa quase nunca é válida na prática.

Os parâmetros das funções *Kernel* influem na complexidade da RVS resultante porque indiretamente eles afetam a transformação não-linear do espaço de entrada para o espaço expandido (as funções *Kernel* equivalem a produtos internos de vetores no espaço expandido). A relação entre os parâmetros do *Kernel* e a complexidade do modelo é sensivelmente visível, à medida que se aumenta ou diminui a largura das RBF gaussianas, no caso do *Kernel* RBF, ou à medida que se varia o grau do polinômio, no caso do *Kernel* Polinomial. Dado um conjunto de treinamento, a estimativa da MVS é única a menos da escolha do tipo de *Kernel*,

seus parâmetros e a constante de otimização C . Portanto esse modelo de RNs permite a diminuição da carga de empirismo em sua estimativa e um melhor entendimento sobre sua adequação à aplicação. As RVS são treinadas através do programa SVM^{light} [Scholkopff, 1999].

5.2 – Técnicas Automáticas de Extração de Atributos

As propostas de aplicação de RNs para AET existentes baseiam-se na experiência e no julgamento humano para escolher atributos de entrada para as RNs. Essas entradas normalmente oferecem uma boa descrição do estado do sistema de potência. Geralmente, a análise de estabilidade realizada com esses atributos está do lado conservador.

No entanto, para a análise de estabilidade em tempo quase real de um sistema de grande porte, em esquema unificados como mostrado em [Jardim, 2000b] e [Jardim, 2000a], ou esquemas de identificação da borda de segurança como mostrado em [Kassabalidis, 2002], que consideram RNs como imprescindíveis a sua viabilidade, torna-se necessário o desenvolvimento de técnicas de Extração de Atributos automáticas. Conjuntos de atributos expandidos poderiam ser usados para garantir a representação ampla do sistema. O problema de tal abordagem seria o tempo computacional excessivamente longo para treinar a RN e os inconvenientes decorrentes da Maldição da Dimensionalidade. Se o conjunto de atributos for muito extenso, as RNs podem não ser mais uma opção viável.

Uma abordagem através da Extração de Atributos será realizada com um procedimento em dois passos. No primeiro, um grupo expandido de atributos é desenvolvido e atributos que representem o sistema são propostos. Especialistas humanos podem oferecer uma boa contribuição para isso. No segundo passo, o vetor de atributos é reduzido para um tamanho administrável que permita o treinamento da RN de maneira rápida e com bons resultados. Com esse procedimento, será possível levar em conta combinações numerosas e complexas de atributos de entrada que nunca foram tentadas por especialistas humanos até o momento, e também tornará possível atacar problemas de elevada dimensionalidade.

Nas subseções seguintes são apresentados métodos automáticos para Extração/Seleção de Atributos utilizados para AET via RNs. São apresentados três métodos utilizando Busca Sequencial (BS), Extração de Componentes Principais (ECP) e Algoritmos Genéticos (AGs). Eles são aplicados em avaliações de estabilidade via RNs e em testes com os sistemas de potência de 50 máquinas do IEEE e o subsistema brasileiro referente à bacia Paranaíba-Grande.

5.2.1 – Busca Sequencial

Como mostrado no capítulo 3, o método Busca Sequencial (BS) de SA é um dos métodos de busca mais simples que existem. Esse é um método sub-ótimo e os atributos assim selecionados não são garantidamente a melhor escolha para a RN. Os d -melhores atributos selecionados pela BS não são necessariamente os melhores d -atributos, isto é, os d atributos avaliados individualmente e em seguida utilizados em conjunto não levam em consideração a interação entre os mesmos.

Nesta técnica, os atributos são avaliados individualmente através da função Discriminante de *Fisher*, que calcula a distância entre os pontos médios das duas classes distintas, ponderada pelas variâncias dos dados de cada classe, como em (3.7). Cada atributo corresponde a uma direção ou componente de representação dos dados. O número de atributos final, d , deve ser escolhido previamente.

5.2.2 – Extração de Componentes Principais

A Extração de Componentes Principais ou Transformação de Karhunen-Loève é um dos métodos mais utilizados de EA na área de processamento de sinais. Ela consiste de uma transformação linear relativamente simples dos atributos de entrada, podendo reduzir o número de atributos através de um truncamento da dimensão dos vetores transformados resultantes. Surgida no âmbito da Teoria da Informação, a ECP tem o objetivo básico de reduzir a dimensão do vetor de atributos para realizar a transmissão de dados em canais de informação, transportando a maior quantidade possível de informações. Por isso, a transformação é baseada numa determinação de coordenadas que ofereçam as maiores variâncias possíveis dos dados, e que irão produzir o menor erro possível, no sentido dos mínimos quadrados, depois da reconstrução do sinal original.

Em termos de representação, a ECP pode ser considerada como um método ótimo de sintetização das informações. Essa abordagem tem sido transportada para a tarefa de classificação, com a premissa de que uma redução na dimensão da representação dos dados pode diminuir os custos computacionais de classificadores, ao mesmo tempo mantendo a performance de classificação, já que a perda de informação é minimizada depois da transformação. Essa premissa não é necessariamente verdadeira porque a ECP é um método não supervisionado que não utiliza nenhuma informação sobre a classificação dos dados de treinamento. Na ECP, o número de atributos final, d , deve ser determinado previamente.

5.2.3 – Algoritmos Genéticos

Quando a dimensão do vetor de atributos de entrada é muito alta, a SA torna-se muito pesada computacionalmente. Os AGs são um método de busca eficiente capaz de varrer grandes áreas do espaço de busca.

Com esse método de busca, é utilizada uma representação em que cada cromossomo equivale a um subconjunto de atributos candidato, com 'uns' representando quais atributos fazem parte daquele subconjunto. Os atributos são avaliados conjuntamente durante o processo evolutivo e espera-se encontrar uma solução global, como alternativa à busca exaustiva. A função de custo do AG é tomada como sendo o Discriminante de *Fisher*, que pode ser calculada de maneira relativamente simples e rápida.

Nesta implementação de AGs para a SA, é utilizado o AG padrão mostrado na Seção 3.3 do Capítulo 3, com operador Cruzamento de um ponto, e com algumas modificações explicadas abaixo.

O tempo de processamento é um elemento crucial durante a execução do AG. Caso a função de custo escolhida tome muito tempo para ser calculada, o elevado número de avaliações necessárias durante o processo torna-o impraticável, até mesmo para sua execução uma única vez. Uma solução alternativa para isso seria a utilização de processamento paralelo com máquinas de alta capacidade de processamento e várias populações concorrentes.

Como o Discriminante de *Fisher* é uma função monotônica crescente, a adição de novos atributos a um dado subconjunto de atributos sempre leva a um valor de avaliação maior. Portanto, a execução do AG sempre produziria soluções com o maior número de atributos

possível. O AG não é capaz de determinar a quantidade de atributos ótima para a utilização em RNs. Inicialmente foi implementado um artifício para contornar essa situação, utilizando uma função de custo como mostrado abaixo. Para um dado subconjunto de atributos, representado pelo vetor $\mathbf{g} \in R^k$, $k < n$, o valor da função de custo associada a ele é

$$FC(\mathbf{g}) = FD(\mathbf{g}, \mathbf{w}) + \gamma(k) \quad (5.2)$$

sendo

$FC \rightarrow$ função de custo;

$FD \rightarrow$ função Discriminante de Fisher;

$\gamma(k) \rightarrow$ fator que representa um prêmio para as soluções candidatas que tenham o número de atributos desejado ($\gamma = 0$ para $k \neq d$).

O fator $\gamma(k)$ serviria para incentivar a permanência na população de indivíduos com o número de atributos desejado. Mas com essa tentativa não se obtiveram resultados satisfatórios porque durante cada Cruzamento, mesmo que os cromossomos pais tenham o número de atributos desejado, $k = d$, e com a representação utilizada, a chance de que os cromossomos filhos resultem em $k \neq d$ é muito alta. Isso “corrompe” toda a população em poucas iterações.

Por isso é introduzida uma modificação no operador Cruzamento do AG padrão, da seguinte maneira: depois do cruzamento de cada par de cromossomos, os cromossomos resultantes somente são considerados para iterações seguintes se possuírem o número de atributos desejado, d , caso contrário eles são descartados e um novo cruzamento é realizado com o mesmo par de cromossomos pais. O cruzamento com os cromossomos pais é repetido até que cromossomos filhos com $k = d$ sejam gerados. Essa abordagem mostrou-se satisfatória, com aumento pouco considerável do tempo final de processamento. Portanto, nessa implementação, o número de atributos final, d , deve ser determinado previamente.

Uma vantagem da utilização de AGs para a SA é que eles proporcionam um método em que a avaliação dos atributos sempre é feita em conjunto. Dessa forma a busca é realizada levando em conta sempre os efeitos das combinações entre os atributos de entrada, o que não ocorre com a SA do tipo Busca Sequencial. Ao mesmo tempo, com uma representação como a

utilizada aqui, é possível que combinações de atributos importantes não estejam presentes na população inicial e por isso nunca sejam gerados durante as iterações. Por exemplo, caso seja importante a interação entre dois atributos específicos e a combinação dos dois resulte em bom desempenho de uma RN, e se a população inicial do AG não contiver nenhum cromossomo que apresente os dois atributos juntos, e se a localização dos mesmos for em pontos distantes do cromossomo, há uma grande chance de que nenhum cromossomo que contenha a combinação dos dois atributos seja gerado durante a execução do AG, devido à utilização do operador Cruzamento de um ponto.

Nesta implementação dos AGs são utilizadas populações com 200 indivíduos, 500 gerações, probabilidade de cruzamento, $P_c = 0,8$, e probabilidade de mutação, $P_m = 0,05$. Também é utilizada uma estratégia elitista em que os 10 melhores indivíduos de cada geração são automaticamente copiados para a geração seguinte. Cada vez que a SA é executada, o AG é executado 5 vezes e a melhor solução é utilizada. A obtenção da mesma solução em todas as execuções do AG indicaria que ela é global. Cabe notar, porém, que num espaço de busca tão grande como o existente nesta aplicação de SA, é difícil cobrir satisfatoriamente uma grande porção do mesmo. Em vista disso, uma alternativa seria executar o AG muitas vezes, cada uma com uma população inicial diferente. Mas com os recursos computacionais limitados, com essa prática o uso de AGs começaria a tornar-se demasiadamente custoso e inviável.

5.3 – Resultados

Para demonstrar as idéias propostas nesta tese foram utilizados dois sistemas de potência de teste: o sistema de 50 máquinas do IEEE [IEEE, 1992]; e o subsistema brasileiro referente à bacia dos rios Paranaíba e Grande (Sistema Priba) [GESIS, 2000].

5.3.1 – Sistema IEEE 50 Máquinas

Este sistema possui 50 geradores, 145 barras e 421 ramos. A criação dos conjuntos de treinamento e teste para as técnicas de EA e para o treinamento e teste das RNs foi realizada partindo de um caso base, e criando-se vários cenários com o seguinte procedimento:

1. Multiplicação dos valores base de carga de todos os barramentos por um fator aleatório entre 0,8 e 1,2. O mesmo fator multiplicativo foi usado para todas as barras de carga;
2. Multiplicação dos valores de carga obtidos no item 1 de cada barra por um valor aleatório entre 0,8 e 1,2. Em cada barra é utilizado um fator multiplicativo diferente;
3. Multiplicação dos valores base de geração de todos os barramentos pelo mesmo fator usado no item 1. O mesmo fator multiplicativo foi usado para todas as barras de geração;
4. Multiplicação dos valores de carga obtidos no item 3 de cada barra de geração por um valor aleatório entre 0,8 e 1,2. Em cada barra é utilizado um fator multiplicativo diferente.

Depois da criação de cada cenário com o procedimento acima, o mesmo é submetido à execução de um programa de fluxo de potência para checar sua viabilidade. Caso haja violações de tensões acima de 20% em alguma barra, o cenário é descartado. Caso o cenário seja viável, então é realizada a simulação no domínio do tempo da ocorrência de um distúrbio através da execução do programa ETMSP [EPRI, 1994] para a determinação do Tempo Crítico de Remoção da Falta (*Critical Clearing Time, CCT*). Em cada caso, o distúrbio considerado foi a ocorrência de uma falta trifásica num dado barramento com a retirada da linha de transmissão adjacente. Com a repetição desse procedimento, foi possível criar conjuntos de treinamento e teste com exemplos a serem utilizados para a EA e a identificação de RNs.

No sistema IEEE 50 Máquinas foram tomados como atributos de entrada, as potências ativa e reativa geradas em cada máquina e as potências ativa e reativa totais de carga, num total de 102 atributos de entrada. Como atributo de saída, foi utilizada uma variável binária para indicar que o ponto de operação representado pelos valores de entrada é estável ou instável, frente à ocorrência da falta. O atributo de saída possui valores 0 ou 1, obtidos a partir de um *threshold* fixado em 0,45 s, que indica o tempo de abertura e isolamento da falta. Os atributos de treinamento cujas simulações resultaram em valores de CCT maiores que 0,45 s foram tomados como estáveis e os restantes, instáveis.

Inicialmente as RNs foram treinadas utilizando todas as entradas consideradas originalmente e a saída binária. Em seguida, as técnicas de EA foram executadas utilizando o conjunto de treinamento para reduzir a dimensão do espaço de entrada original. Depois disso, RNs do tipo Perceptrons Multicamadas e Redes de Vetores Suporte foram treinadas, utilizando o conjunto de treinamento de dimensão reduzida para comparar os resultados com aqueles obtidos quando as RNs foram treinadas com todos os atributos.

Para este sistema foram utilizadas RNs treinadas com 102 atributos de entrada e uma saída binária para indicar condição estável ou instável. Foram utilizados 800 exemplares de treinamento e 400 de teste.

Como descrito anteriormente, dois modelos de RNs foram treinados nestes testes: MLPs e MVSs. Os MLPs foram treinados através da validação cruzada, utilizando o algoritmo Levenberg-Marquardt. Para cada conjunto de atributos de entrada foram testadas estruturas de MLPs com uma camada intermediária, 3, 5, 10, 15 e 20 neurônios de cada vez. A escolha da melhor estrutura de MLP foi baseada no menor valor médio do erro encontrado durante a validação cruzada, para as partições do conjunto não utilizadas no treinamento.

As MVSs foram treinadas utilizando o aplicativo SVM^{light} [Joachims, 1998b]. As MVSs foram treinadas utilizando funções *Kernel* do tipo Base Radial (RBF) Gaussiano e Polinomiais. Os parâmetros de tais *Kernels* foram determinados através de uma busca heurística, experimentando valores combinados dos mesmos juntamente com o parâmetro de otimização, C .

A Figura 5.1 apresenta curvas ROC (*Receiver Operating Characteristic*) para o desempenho de teste da MVS treinada com o conjunto completo de atributos e utilizando o *Kernel* RBF Gaussiano. O desempenho de teste refere-se a padrões de teste que não foram utilizados no treinamento, apresentados ao classificador. A taxa de falsos negativos no eixo x é a taxa de padrões de teste que foram incorretamente classificados como estáveis. A taxa de detecção no eixo y é a taxa de padrões estáveis que foram corretamente classificados. Várias MVSs foram treinadas com valores crescentes de σ^2 e os valores correspondentes das taxas de detecção e falsos negativos são mostrados na Figura 5.1 Esses são dois valores conflitantes que crescem juntos para um dado valor de C e à medida que σ^2 varia. Neste caso, valores de

$C = 0,1, 1, 10, 100, 1000$ e 10000 foram testados. A linha sólida da Figura 5.2 corresponde a $C = 1$. A linha pontilhada corresponde a MVSs treinadas com $C = 10, 100, 1000$ e 10000 , que mostram resultados idênticos. A curva para a MVS com $C = 0,01$ não é mostrada na Figura 5.1 devido à diferença de escalas. Ela apresenta valores muito maiores de taxas de falsos negativos, tão grandes quanto $0,08$ e valores menores de taxas de detecção que as curvas mostradas na Figura 5.1.



Figura 5.1 – Curva ROC da MVS RBF Gaussiano – IEEE 50 Máquinas

A Figura 5.2 apresenta a curva ROC da MVS treinada com 102 atributos de entrada e utilizando o *Kernel* polinomial. Os valores do gráfico são obtidos com o parâmetro p variando de 1 a 7 e $C = 1000$. Curvas semelhantes podem ser obtidas com outros valores de C .

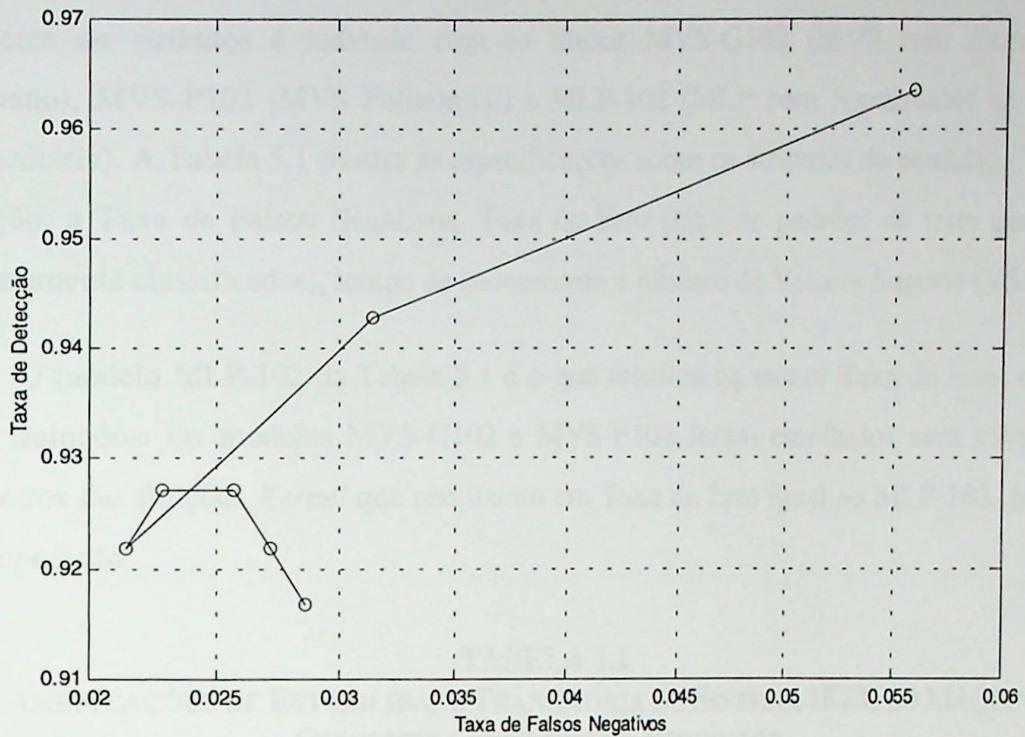


Figura 5.2 – Curva ROC da MVS Polinomial – IEEE 50 Máquinas

Os gráficos das Figuras 5.1 e 5.2 apresentam no eixo x a variável Taxa de Falsos Negativos, que é muito importante no caso de classificações de estabilidade de sistemas de potência. Ela representa a proporção de casos instáveis erroneamente descartados pelo classificador neural e deve ser minimizada ao máximo, idealmente deveria ser zero.

A busca de parâmetros das funções *Kernel* e da constante C constitui um ponto fraco das MVSs porque o resultado pode levar a um classificador não ótimo. A princípio não se sabe como se deve realizar essa busca. Porém, nas Figuras 5.1 e 5.2 pode-se perceber um relacionamento visível entre os parâmetros dos *Kernels* utilizados e a capacidade de generalização dos classificadores MVS, codificada através da Taxa de Falsos Negativos e da Taxa de Detecção.

Um classificador MLP também foi treinado com o conjunto completo de atributos de entrada, mas curvas ROC como as anteriores não podem ser traçadas para mostrar o desempenho no conjunto de teste. O número de fatores que afetam o classificador é grande e eles são interrelacionados.

Na Tabela 5.1, o desempenho dos classificadores neurais treinados com conjuntos completos de atributos é indicado com os títulos MVS-G102 (MVS com *Kernel* RBF Gaussiano), MVS-P102 (MVS Polinomial) e MLP-102 (MLP com 5 neurônios na camada intermediária). A Tabela 5.1 mostra as especificações sobre os atributos de entrada, a Taxa de Detecção, a Taxa de Falsos Negativos, Taxa de Erro (taxa de padrões de teste que foram incorretamente classificados), tempo de treinamento e número de Vetores Suporte (VSs).

O modelo MLP-102 da Tabela 5.1 é o que resultou na menor Taxa de Erro, entre os MLPs treinados. Os modelos MVS-G102 e MVS-P102 foram escolhidos com valores dos parâmetros das funções *Kernel* que resultaram em Taxa de Erro igual ao MLP-102, para fins de comparação.

TABELA 5.1
CLASSIFICAÇÕES DE ESTABILIDADE TRANSITÓRIA DO SISTEMA IEEE 50 MÁQUINAS
CONJUNTO COMPLETO DE ATRIBUTOS

Classificador	MVS-G102	MVS-P102	MLP-102
Atributos de Entrada	102	102	102
Taxa de Detecção	0.92	0.92	0.91
Taxa de Falsos Negativos	0.026	0.026	0.021
Taxa de Erro	0.047	0.047	0.047
Tempo de Treinamento	37 s	32 s	40min42s
No. de VSs	433	241	-

Em seguida, classificadores neurais MLP e MVS foram treinados em conjuntos reduzidos de atributos de entrada. Para o treinamento do MLP, novamente foi realizado o treinamento através da validação cruzada.

Depois de treinadas as MVSs com diferentes conjuntos de atributos de entrada, curvas ROC foram traçadas para diferentes valores de C . Então, para os valores de C que resultaram nas melhores curvas ROC, valores fixos de Taxas de Falsos Negativos foram fixadas: 0,02 e 0,03, que puderam ser obtidas com valores específicos correspondentes dos parâmetros da função *Kernel*. Esses modelos são mostrados na Tabela 5.2 como MVS-G0.02, MVS-G0.03 (*Kernel* RBF Gaussiano), e MVS-P0.02 e MVS-P0.03 (*Kernel* Polinomial). Para

o modelo MLP, MLP-1 é o modelo com a taxa de erro mais baixa e MLP-2 é o modelo com o menor tempo de treinamento.

A Tabela 5.2 mostra os modelos treinados com conjuntos reduzidos de atributos de entrada que resultaram nos melhores desempenhos de teste, em termos de Taxa de Detecção, Taxa de Falsos Negativos e Taxa de Erro. Por exemplo, para uma MVS com *Kernel* RBF Gaussiano e com parâmetro σ^2 ajustado para obter a Taxa de Falsos Negativos igual a 0,03 (MVS-G0.03) os 30 atributos de entrada selecionados pela técnica Busca Sequencial (BS) resultaram no melhor desempenho; para a MVS com *Kernel* polinomial e parâmetro p ajustado para obter a Taxa de Falsos Negativos igual a 0,03 (MVS-P0.03) os 30 atributos de entrada selecionados pela técnica Algoritmos Genéticos (AGs) resultaram no melhor desempenho.

TABELA 5.2
CLASSIFICAÇÕES DE ESTABILIDADE TRANSITÓRIA DO SISTEMA IEEE 50 MÁQUINAS –
CONJUNTOS REDUZIDOS DE ATRIBUTOS

Classificador	MVS-G0.02	MVS-G0.03	MVS-P0.02	MVS-P0.03	MLP-1	MLP-2
Atributos de Entrada	30-BS	30-BS	30-BS	30-AG	50-BS	10-AG
Taxa de Detecção	0.85	0.95	0.96	0.95	0.95	0.92
Taxa de Falsos Negativos	0.02	0.03	0.0215	0.0314	0.023	0.024
Taxa de Erro	0.06	0.043	0.033	0.046	0.036	0.046
Tempo de Treinamento	34 s	12 s	19 s	20 s	11min24s	1min42s
No. de VSs	1065	298	156	178	-	-

As Tabelas 5.1 e 5.2 mostram que os classificadores MVS são um modelo viável para aplicação na AET. A Tabela 5.2 mostra que é possível atingir uma grande redução na dimensionalidade do conjunto de atributos de entrada, com resultados melhores que os resultados do caso completo (caso que utiliza todos os atributos originais) e tempos de treinamento bem menores no caso do modelo MLP.

Pôde-se notar que o tempo de treinamento do MLP aumenta dramaticamente com o número de atributos de entrada e com o número de neurônios da camada intermediária. O tempo de treinamento da MVS depende do número de atributos de entrada, dos valores dos parâmetros da função *Kernel* e do número de VSs do classificador resultante.

Os resultados para o sistema IEEE 50 Máquinas mostram que o tempo de treinamento das MVSs é bem menor que o dos MLPs, tanto no caso completo quanto no caso reduzido. Com o modelo MVS foi possível atingir uma taxa de redução de aproximadamente 70% ($d = 30$), com desempenho medidos pela Taxa de Detecção, Taxa de Falsos Negativos e Taxa de Erro ligeiramente melhor do que o do caso completo. Com o modelo MLP pôde-se chegar a uma taxa de redução no número de atributos de entrada de aproximadamente 50% ($d = 50$) e 90% ($d = 10$), com desempenho de classificação um pouco piores, porém comparáveis aos das MVSs e tempo de treinamento maiores. Portanto, nesse caso não há grande evidência de que um tipo de classificador tenha sido melhor do que o outro e os dois mostram-se adequados.

Quanto às técnicas de EA, a ECP mostrou-se inadequada e por isso seus resultados não figuram na Tabela 5.2. Ambas técnicas BS e AG mostraram resultados satisfatórios e capacidade de grande redução do número de atributos. Como a BS é mais simples e demanda muito menor tempo de execução que o AG, ela mostra-se mais adequada.

O modelo MLP-2 da Tabela 5.2 foi utilizado recentemente numa proposta de um sistema de identificação da borda de segurança *on-line* baseado em um modelo de Otimização Enxame (*Particle Swarm Optimization*) [Kassabalidis, 2002]. Foram realizados testes com o sistema IEEE 50 Máquinas. Esse modelo de otimização multi-agente cria um conjunto de pontos de operação com o objetivo de identificar os que se encontram próximos ou sobre o limite de estabilidade. A partir do conjunto inicial, é executado um processo de otimização inspirado no movimento de seres vivos que viajam em grupos ou Enxames (como abelhas e certas espécies de pássaros). Um MLP é utilizado para produzir avaliações de estabilidade rápidas de cada ponto. Esse sistema também utiliza uma função objetivo e testa várias alternativas diferentes de penalização que indicam a distância entre os pontos gerados e o ponto mais próximo da borda de segurança, a distância dos pontos gerados uns dos outros, a uniformidade da distribuição dos pontos ao longo da borda de segurança, etc. Por isso uma grande quantidade de avaliações de distância é realizada ao longo do processo de otimização. Nesta aplicação, a redução da dimensionalidade do vetor de atributos que representa o ponto de operação é imprescindível não só para o treinamento e o bom desempenho do MLP, mas também para a viabilização do cálculo dessas distâncias de maneira *on-line*.

5.3.2 – Sistema Priba

Este sistema possui 200 barras de geração, 2484 barras de carga e 5730 ramos. Para a formação dos conjuntos de treinamento e teste, foram considerados vários casos base a partir de 14 indisponibilidades de linhas de transmissão (LTs) e transformadores (TFs). Essas indisponibilidades correspondem a elementos em manutenção. A partir dessas indisponibilidades, cenários de carga leve, pesada e média foram considerados, com fluxos de potência convergidos a partir de despachos de geração previamente estabelecidos. Também, para cada uma das 14 indisponibilidades, foi considerada a aplicação de defeitos do tipo curto-circuito trifásico seguido de isolamento do ramo. Esses defeitos foram aplicados em 9 LTs ou TFs. As indisponibilidades e os defeitos foram escolhidos através de conhecimento prévio sobre a vulnerabilidade do sistema. A análise da estabilidade transitória de cada caso foi realizada através do programa de simulações no domínio do tempo ANATEM (Cepel).

Dessa maneira foi obtido um conjunto inicial de dados, a partir do qual foram formados os conjuntos de 994 padrões de treinamento e 248 padrões de teste. Esses conjuntos representam um número limitado de 9 contingências que pode ser aumentado caso se julgue necessário. Na verdade, um dos maiores problemas existentes em supervisão e controle de sistemas elétricos ocorre quando contingências “incomuns” aparecem. Muitas vezes elas são as maiores causadoras de danos ao sistema. Caso se queira considerá-las no treinamento das RNs, basta que se realizem as simulações das mesmas e que se incluam os resultados no conjunto de treinamento.

Os modelos MLPs e MVS foram treinados como no caso do Sistema IEEE 50 Máquinas, incluindo a escolha da estrutura dos modelos.

Os atributos de entrada que representam as condições de operação antes da aplicação dos defeitos foram tomados inicialmente como sendo as potências ativa e reativa geradas em cada barra de geração, os fluxos em 26 ramos do sistema (dentre os quais estão as 14 indisponibilidades e os 9 defeitos), as cargas ativa e reativa totais, e representações binárias dos 9 defeitos, num total de 224 atributos de entrada. O atributo de saída foi tomado como sendo a informação binária estável/instável. Os fluxos nos 26 ramos são utilizados também para representar a topologia da rede, já que um fluxo igual a 0 indica que aquele ramo está aberto.

Como nos casos anteriores os conjuntos de treinamento e teste assim obtidos foram utilizados para treinar RNs com atributos completos e reduzidos.

Inicialmente, o conjunto de treinamento obtido a partir das simulações do Sistema Priba foi utilizado para treinar as RNs com o conjunto completo de atributos ($d = 224$). A Figura 5.3 apresenta a curva ROC (*Receiver Operating Characteristic*) da MVS com *Kernel* RBF Gaussiano. A Taxa de Detecção e a Taxa de Falsos Negativos são apresentadas no gráfico para cada valor do parâmetro σ^2 da função *Kernel* RBF. Uma série de valores do parâmetro C foi utilizado no treinamento das MVS, e a Figura 5.3 apresenta o gráfico resultante para $C = 1$.

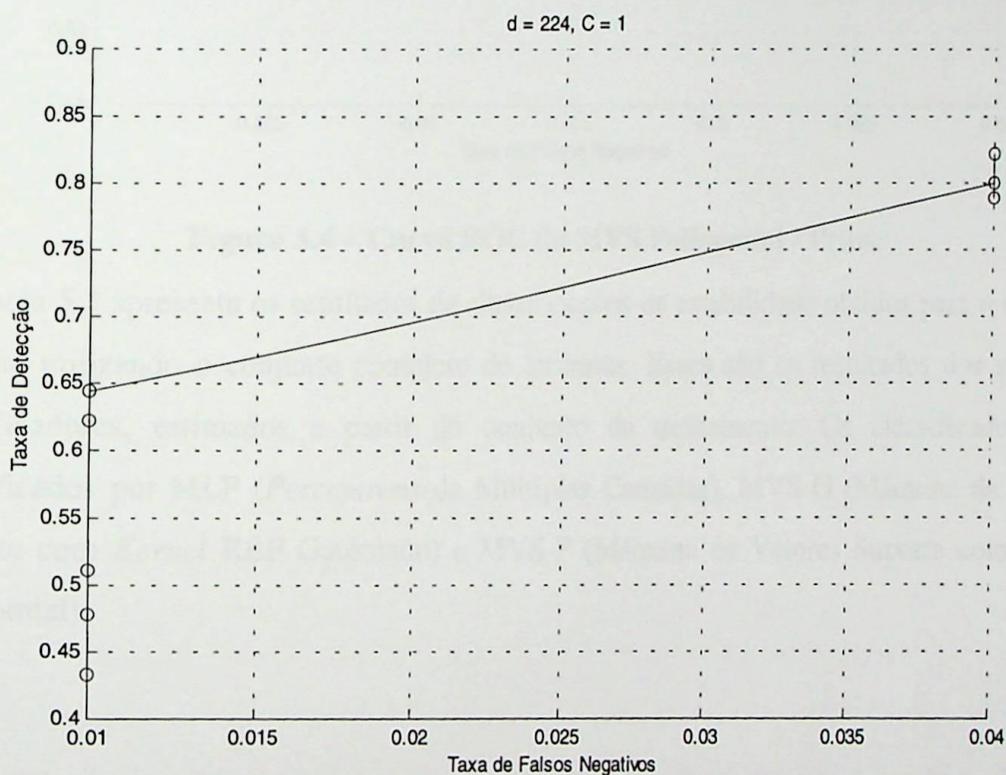


Figura 5.3 – Curva ROC da MVS RBF Gaussiano - Priba

A Figura 5.4 apresenta a curva ROC da MVS com *Kernel* Polinomial treinada usando o conjunto completo de atributos. O parâmetro p do *Kernel* Polinomial foi variado de 1 a 6. A Figura 5.4 apresenta pares de valores de Falsos Negativos e Taxas de Detecção para cada valor de p e para $C = 1$.

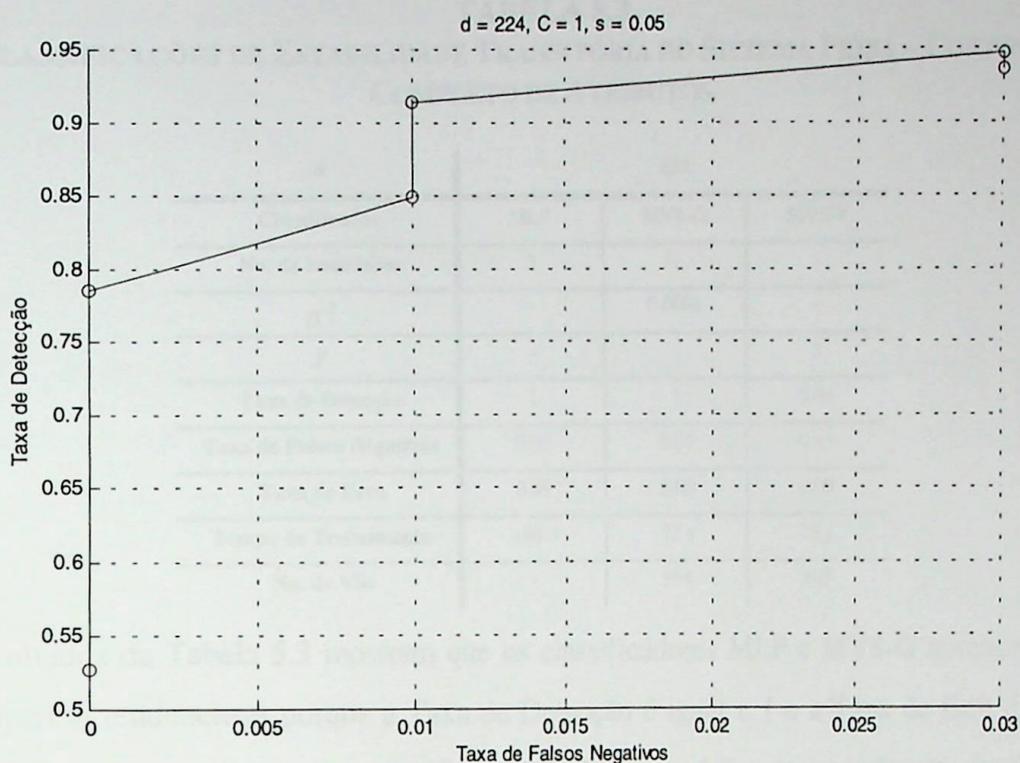


Figura 5.4 – Curva ROC da MVS Polinomial - Priba

A Tabela 5.3 apresenta os resultados de classificações de estabilidade obtidos para o conjunto de teste utilizando o conjunto completo de atributos. Esses são os resultados dos melhores classificadores, estimados a partir do conjunto de treinamento. Os classificadores são identificados por MLP (*Perceptrons* de Múltiplas Camadas), MVS-G (Máquina de Vetores Suporte com *Kernel* RBF Gaussiano) e MVS-P (Máquina de Vetores Suporte com *Kernel* Polinomial).

TABELA 5.3
CLASSIFICAÇÕES DE ESTABILIDADE TRANSITÓRIA DO SISTEMA PRIBA – CONJUNTO
COMPLETO DE ATRIBUTOS

<i>d</i>	224		
Classificador	MLP	MVS-G	MVS-P
No. de Neurônios	3	-	-
σ^2	-	0.0001	-
<i>p</i>	-	-	5
Taxa de Detecção	1	1	0.95
Taxa de Falsos Negativos	0.09	0.09	0.05
Taxa de Erro	0.09	0.09	0.09
Tempo de Treinamento	105 s	72 s	73 s
No. de VSs	-	994	249

Os resultados da Tabela 5.3 mostram que os classificadores MLP e MVS-G apresentam um desempenho tendencioso porque a Taxa de Detecção é igual a 1 e a Taxa de Erro é igual à Taxa de Falsos Negativos. Isso significa que todos os padrões de teste foram classificados como estáveis. Portanto esses dois classificadores são tendenciosos e inadequados.

Quando um classificador é tendencioso a avaliar os pontos de operação como estáveis, a Taxa de Detecção num conjunto de teste qualquer é alta (próxima de 1) mas o número de Falsos Negativos também é alto. Por outro lado, quando o classificador é tendencioso a avaliar os pontos de operação como instáveis, o número de Falsos Negativos é baixo, mas a Taxa de Detecção também é baixa.

Grande parte das simulações de estabilidade do subsistema Priba resultou em condição estável, por isso há um desbalanço entre o número de padrões estáveis e instáveis nos conjuntos de treinamento e teste, isto é, o número de padrões estáveis é maior do que o número de padrões instáveis. A MVS possui um mecanismo simples para evitar que isso leve a um classificador tendencioso. Através da constante *C* da função objetivo (4.20) da otimização quadrática.

$$V(\mathbf{w}) = \frac{1}{2} \|\mathbf{w}\|^2 + C \sum_{i=1}^N \varepsilon_i \quad (4.20)$$

em vez de usar um único valor da constante, dois valores diferentes, C_1 e C_2 , são utilizados para cada conjunto de padrões estáveis e instáveis, respectivamente. Uma nova função objetivo (5.3)

$$V(\mathbf{w}) = \frac{1}{2} \|\mathbf{w}\|^2 + C_1 \sum_{i=1}^{N_1} \varepsilon_i + C_2 \sum_{i=1}^{N_2} \varepsilon_i \quad (5.3)$$

possui C_1 como a constante utilizada para os padrões estáveis e C_2 , a constante utilizada para os padrões instáveis. N_1 e N_2 são os números de padrões estáveis e instáveis do conjunto de treinamento, respectivamente. Foram utilizados valores segundo a relação (5.4)

$$C_1 = \frac{N_1}{N_2} C_2 \quad (5.4)$$

para penalizar mais fortemente os padrões instáveis, e compensar a sua quantidade menor no conjunto de treinamento. Dessa maneira foi possível obter MVSs não tendenciosas. O mecanismo da função (5.4) foi utilizado em todos os treinamentos das MVSs.

A seguir, foram executadas as técnicas de Extração de Atributos indicadas no Capítulo 5. Devido ao desbalanço do conjunto de treinamento já comentado, todas os MLPs treinados resultaram em classificadores tendenciosos e inadequados. Por isso, nos resultados a seguir serão mostrados somente os classificadores MVS.

O Anexo III apresenta um conjunto de tabelas contendo todos os resultados de teste de tais classificadores, considerando cada técnica de EA, e reduções de dimensionalidade de 224 para $d = 150, 100, 50, 20$ e 10 . A Tabela 5.4 apresenta um resumo dos resultados do Apêndice III, contendo os melhores resultados em termos de Falsos Negativos.

TABELA 5.4**MELHORES RESULTADOS DO SISTEMA PRIBA EM TERMOS DE TAXA DE FALSOS NEGATIVOS**

<i>d</i>	150AG	100AG	50AG	20AG	10BS
Classificador	MVS-P	MVS-P	MVS-P	MVS-P	MVS-P
σ^2	-	-	-	-	-
<i>p</i>	4	5	1	3	5
Taxa de Detecção	0.8844	0.8578	0.6578	0.7111	0.52
Taxa de Falsos Negativos	0.032	0.048	0.032	0.024	0.02
Taxa de Erro	0.137	0.177	0.343	0.286	0.45
Tempo de Treinamento	70 s	65 s	105 s	53 s	19 s
No. de VSs	219	274	688	489	698

A Tabela 5.5 apresenta um resumo dos resultados do Apêndice III, contendo os melhores resultados em termos de Taxa de Erro.

TABELA 5.5**MELHORES RESULTADOS DO SISTEMA PRIBA EM TERMOS DE TAXA DE ERRO**

<i>d</i>	150BS	100BS	50BS	20AG	10AG
Classificador	MVS-P	MVS-P	MVS-P	MVS-G	MVS-P
σ^2	-	-	-	0.12	-
<i>p</i>	2	4	5	-	4
Taxa de Detecção	0.96	0.95	0.76	0.7511	0.6844
Taxa de Falsos Negativos	0.05	0.05	0.04	0.036	0.028
Taxa de Erro	0.08	0.10	0.26	0.262	0.315
Tempo de Treinamento	106 s	98 s	67 s	13 s	20 s
No. de VSs	180	225	342	503	667

A Tabela 5.6 apresenta um resumo dos resultados do Apêndice III, contendo os melhores resultados em termos de Tempo de Treinamento.

TABELA 5.6
MELHORES RESULTADOS DO SISTEMA PRIBA EM TERMOS DE TEMPO DE TREINAMENTO

<i>d</i>	150AG	100AG	50BS	20AG	10BS
Classificador	MVS-G	MVS-G	MVS-G	MVS-G	MVS-G
σ^2	0.6	2.04	0.6	0.12	0.0001
<i>p</i>	-	-	-	-	-
Taxa de Detecção	0.9956	0.9067	0.76	0.7511	0.66
Taxa de Falsos Negativos	0.089	0.052	0.06	0.036	0.06
Taxa de Erro	0.093	0.137	0.27	0.262	0.37
Tempo de Treinamento	46 s	31 s	18 s	13 s	13 s
No. de VSs	906	362	528	503	954

Os resultados provenientes da Extração de Componentes Principais não foram considerados nas Tabelas 5.4 a 5.6 porque eles são nitidamente tendenciosos. Portanto a técnica de ECP foi insatisfatória para a EA na a aplicação de AET.

Pode-se verificar na Tabela 5.3 que o treinamento dos classificadores com 224 atributos de entrada (caso completo) apresentou dificuldades, na medida em que dois deles (MLP e MVS-G) resultaram tendenciosos. Através da seleção de atributos e da redução da dimensionalidade foi possível compatibilizar melhor os dados de treinamento e os classificadores MVS-G de forma a obter desempenhos melhores.

As Tabelas 5.4 a 5.6 mostram que o número de VSs dos classificadores aumenta à medida que se diminui o número de atributos de entrada. Essa é uma indicação de que as superfícies de classificação tornam-se mais complicada e que mais VSs são necessários para realizar uma boa classificação. Nos casos reduzidos com $d = 150$ e $d = 100$ o número de VSs é menor do que no caso completo, e o desempenho de classificação é semelhante ou ligeiramente melhor. Portanto uma redução de aproximadamente 50% ($d = 100$) é benéfica para os classificadores nesse caso.

Quanto aos tempos de treinamento, as MVSs apresentam uma redução significativa somente nos casos em que o número de atributos é reduzido para $d = 20$ e $d = 10$. Na verdade, os tempos de treinamento desse modelo são relativamente pequenos, mesmo com um número elevado de atributos de entrada, variando de 13 s a 106 s. A redução de

dimensionalidade para esses classificadores é mais importante para a finalidade de melhoria no desempenho da classificação.

Nos testes com o sistema Priba não foi possível obter um classificador MLP satisfatório. Todos resultaram tendenciosos tanto no caso completo quanto nos casos reduzidos, devido ao desbalanço do conjunto de treinamento. Portanto, o modelo MVS mostra-se mais indicado para aplicações práticas porque na simulação de sistemas de potência reais como este é esperado um maior número de casos estáveis do que instáveis para constituir os conjuntos de treinamento e teste. Essa expectativa é justificada pelo fato de que o planejamento dos sistemas de potência visa sua operação de maneira mais estável possível, mesmo frente à ocorrência de distúrbios.

Não há evidências sobre qual das técnicas AG e BS conseguiu atingir melhor compatibilização entre os dados de treinamento e as MVSs. Nos resultados referentes a $d = 150$ e $d = 100$, os valores de Taxa de Erro e de Falsos Negativos decorrentes dos atributos escolhidos através da BS são menores, mas a diferença não é significativa. Preferencialmente deve-se optar pela técnica de BS porque ela demanda menor esforço computacional.

Conclusões e Trabalhos Futuros

Neste trabalho foi apresentada a aplicação de dois modelos de RNs, MLPs e MVSs para a AET considerando uma só contingência, ou várias contingências e variações de topologia. Foi mostrado que a MVS é um novo modelo que se presta a essa aplicação de maneira satisfatória. Possui boas qualidades como treinamento rápido e boa precisão em conjuntos de teste provenientes de sistemas elétricos e possui também um mecanismo simples para lidar com conjuntos de treinamento desbalanceados. A carga de empirismo inerente aos MLPs é substituída pela escolha da função *Kernel* e da busca entre os parâmetros do *Kernel* e do parâmetro *C*. Foi mostrado que existe uma relação explícita entre os parâmetros das funções *Kernel* e o desempenho das MVSs. Com relação ao tempo de treinamento, a vantagem das MVSs sobre os MLPs é evidenciada pelos resultados das Tabelas 6.1 e 6.2, tanto em conjuntos completos como reduzidos

No Sistema IEEE 50 Máquinas, ambos modelos de RNs apresentam resultados precisos e competitivos, com taxas de acerto acima de 95%. No sistema Priba, o MLP não conseguiu um treinamento satisfatório devido a sua dificuldade em lidar com conjuntos desbalanceados. Neste sistema, as MVSs atingem uma taxa de acerto acima de 90%.

Também foram apresentadas técnicas de EA capazes de atacar o problema de altas dimensionalidades presentes no problema de AET. Essas técnicas puderam reduzir a dimensionalidade de entrada de maneira automática, identificar os atributos de entrada mais importantes para as RNs, além de tornar o treinamento mais fácil.

Os resultados para o Sistema IEEE 50 Máquinas mostram que o tempo de treinamento das MVSs é bem menor que o dos MLPs, tanto no caso completo quanto no caso reduzido. Com o modelo MVS foi possível atingir uma taxa de redução de aproximadamente 70% ($d = 30$), com desempenho medidos pela Taxa de Detecção, Taxa de Falsos Negativos e Taxa de Erro ligeiramente melhor do que o do caso completo. Com o modelo MLP pôde-se chegar

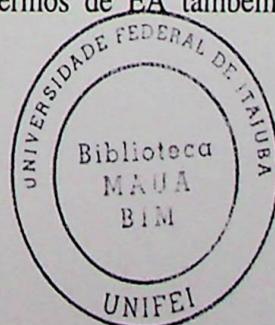
a uma taxa de redução no número de atributos de entrada de aproximadamente 50% ($d = 50$) e 90% ($d = 10$), com desempenho de classificação um pouco piores, porém comparáveis aos das MVSs e tempos de treinamento maiores. Portanto, nesse caso não há grande evidência de que um tipo de classificador tenha tido melhor desempenho do que o outro.

Quanto às técnicas de EA, a ECP mostrou-se inadequada e por isso seus resultados não figuram na Tabela 6.2. Ambas técnicas BS e AG mostraram resultados satisfatórios e capacidade de grande redução do número de atributos. Como a BS é mais simples e demanda muito menor tempo de execução que o AG, ela mostra-se mais adequada.

Nos testes com o Sistema Priba não foi possível obter um classificador MLP satisfatório. Todos resultaram tendenciosos tanto no caso completo quanto nos casos reduzidos, devido ao desbalanço do conjunto de treinamento. Portanto, o modelo MVS mostra-se mais indicado para aplicações práticas porque nas simulações de sistemas de potência reais como este é esperado um maior número de casos estáveis para constituir os conjuntos de treinamento e teste. Essa expectativa é justificada pelo fato de que o planejamento dos sistemas de potência leva à operação de maneira mais estável possível, mesmo frente à ocorrência de distúrbios.

Não há evidências sobre qual das técnicas AG e BS conseguiu atingir melhor compatibilização entre os dados de treinamento e as MVSs. Nos resultados referentes a $d = 150$ e $d = 100$ do Sistema Priba, os valores de Taxa de Erro e de Falsos Negativos decorrentes dos atributos escolhidos através da BS são menores, mas a diferença não é significativa. Preferencialmente deve-se optar pela técnica de BS porque ela demanda menor esforço computacional.

Para obter melhores taxas de redução para um sistema elétrico real como o Sistema Priba, em que pôde-se conseguir uma redução de aproximadamente 50% ($n = 224$, $d = 100$), pode-se usar uma RN treinada separadamente para cada contingência. Pôde-se observar que no sistema IEEE 50 Máquinas, foram obtidas maiores taxas de redução, visto que somente uma contingência foi considerada. Outra possibilidade é dividir o sistema em áreas elétricas, e utilizar um conjunto inicial de atributos menor, e diferente para cada área. Com a consequente redução do espaço de busca, melhores resultados em termos de EA também podem ser obtidos.



A referência [Mansour, 1997a] apresenta como parâmetro aceitável para a AET via RNs em sistemas de grande porte, um valor de Taxa de Falsos Negativos menor que 0,5%. O desempenho dos classificadores apresentados no Capítulo 6 com relação aos Falsos Negativos pode ser modificado utilizando uma “banda de tolerância”, isto é, numa classificação em que a RN deva produzir uma saída binária 0 ou 1, valores próximos de 0,5 são considerados indefinidos ou tomados conservativamente como instáveis. Esse procedimento equivale a adotar uma faixa de segurança que leva a um desempenho mais confiável, ao custo de uma maior taxa de erro.

A aplicação de RNs para AET descrita nesta tese presta-se à sua inclusão em esquemas de AET de sistemas de potência reais com diversos tipos de abordagem, incluindo sua utilização para o ranqueamento de contingências ou identificação da borda de segurança. Foi mostrado que existe uma flexibilidade para contemplar uma só ou variadas contingências e aspectos de topologia mais ou menos complexa. Uma proposta de sistema de identificação da borda de segurança apresentada em [Kassabalidis, 2002] mostra especificamente que a redução da dimensionalidade é imprescindível para sua viabilidade. Nessa proposta é demonstrado como a abordagem de EA para AET via RNs apresentados nesta tese podem ser aproveitados num esquema integrado para sistemas de potência de grande porte. A execução da EA é imprescindível para o mecanismo de identificação da borda de segurança, baseado num modelo de Otimização de Enxame.

As técnicas de EA para máquinas de aprendizado aqui apresentadas auxiliam o controle de sistemas elétricos, na medida em que elas podem identificar os atributos de controle mais importantes e facilitam o emprego das ações de controle. Essas técnicas são boas candidatas à utilização em ferramentas de inteligência artificial aplicadas em centros de controle para evitar estados de vulnerabilidade transitória de sistemas de potência. Elas produzem não só a redução de dimensionalidade, mas também identificam as variáveis mais importantes para evitar que o sistema mova-se na direção de situações de instabilidade.

Modelos e resultados apresentados nesta tese estão descritos de forma resumida nas referências [Kassabalidis, 2002], [Moulin, 2001a], [Moulin, 2001b] e [Moulin, 2002].

Alguns aspectos que não foram incluídos neste trabalho deverão ser estudados no futuro. Serão pesquisadas técnicas de EA que contemplem a natureza não linear das classificações de estabilidade, por exemplo através do cálculo da Informação Mútua entre atributos de entrada e saída, numa tentativa de melhorar os índices de desempenho dos classificadores neurais.

Serão estudadas maneiras de melhorar a AET via RNs com a divisão do sistema elétrico em sub-áreas, possivelmente através de conceitos de coerência utilizados em equivalentes dinâmicos, e função da dinâmica das máquinas e topologia da rede. Dessa maneira poderão ser utilizados atributos descritivos e classificadores dedicados a cada área analisada. Assim espera-se tornar mais leve a tarefa dos classificadores, permitindo a redução da quantidade de atributos do conjunto inicial e melhor desempenho de classificação. Também será estudada a alternativa de usar uma RN treinada para cada contingência.

Deverão ser consideradas também no futuro a inclusão de variáveis de natureza dinâmica no conjunto inicial de atributos, com valores adquiridos durante a simulação da ocorrência de distúrbios. Também deverão incluir a combinação destas com variáveis de natureza estática.

Os modelos neurais aqui apresentados juntamente com as técnicas de EA serão futuramente combinados num esquema de análise de estabilidade transitória *on-line*. A aplicação das MVSS para AET possivelmente deverá ser utilizada para um modelo de identificação da borda de segurança. A identificação dos VSs durante o treinamento podem auxiliar na indicação das condições de operação mais próximas do limite de estabilidade.

Referências

- [Aboytes,1996] F. Aboytes, R. Ramírez, “Transient Stability Assessment in Longitudinal Power Systems Using Artificial Neural Networks”. IEEE Transactions on Power Systems, Vol. 11, Nov. 1996, pp. 2003-2010.
- [Alkimoto, 1989] Y. Alkimoto, “Application of Expert Systems to Transient Stability Studies”. Proceedings of the Second Symposium on Expert Systems Applied to Power Systems, 1989, Seattle.
- [Alvarado, 1979] F. Alvarado, “Parallel Solution of Transient Problems by Trapezoidal Integration”. IEEE Transactions on Power Apparatus and Systems, Vol. 98, No. 3, May/Jun 1979, pp. 1080-1090.
- [Aresi, 1999] R. Aresi, B. Delfino, G. B. Denegri, S. Massucco, A. Morini, “A Combined ANN/Simulation Tool for Electric Power System Dynamic Security Assessment”. IEEE Power Engineering Society Summer Meeting, 1999, Vol. 2, pp. 1303-1309.
- [Athay, 1979] T. Athay, R. Podmore, S. Virmani, “A Practical Method for Direct Analysis of Transient Stability”. IEEE Transactions on Power Apparatus and Systems, Vol. PAS-98, No. 2, March/April 1979, pp. 573-584.
- [Bahbah, 1999] A. G. Bahbah, A. A. Girgis, “Input Feature Selection for Real-Time Transient Stability Assessment for Artificial Neural Network (ANN) Using ANN Sensitivity Analysis”. Proceedings of the 21st 1999 IEEE International Conference on Power Industry Computer Applications, PICA'99, 1999, pp. 295-300.
- [Blanz, 1996] V. Blanz, B. Scholkopf, H. Buthoff, C. Burges, V. Vapnik, T. Vetter, “Comparison of View-Based Object Recognition Algorithm Using Realistic 3D Models”. In C. Von der Malsburg, W. Von Seelen, J. C. Vorbruggen, B. Sendhoff, Editors, Artificial Neural

Networks – ICANN'96, pp. 251-256, Berlin, 1996. Springer Lecture Notes in Computer Science, vol. 1112.

[Brandwajn, 1997] B. Brandwajn, A. B. R. Kumar, A. Ipakchi, A. Bose, S. D. Kuo, "Severity Indices for Contingency Screening in Dynamic security Assessment". IEEE Transactions on Power Systems, Vol. 12, No. 3, August 1997, pp. 1136-1142.

[Brill, 1992] F. Z. Brill, D. E. Brown, W. N. Martin, "Fast Genetic Selection of Features for Neural Network Classifiers". IEEE Trans. On Neural Networks, Vol. 3, No. 2, March 1992.

[Burges, 1997] C. J. C. Burges, B. Scholkopf, "Improving the Accuracy and Speed of Support Vector Learning Machines". In M. Mozer, M. Jordan, T. Petsche, Editors, Advances in neural Information Processing Systems 9, pp. 375-381, Cambridge, MA, 1997. MIT Press.

[Burges, 1998] C. J. C. Burges, "A Tutorial on Support Vector Machines for Pattern Recognition". Data Mining and Knowledge Discovery, 2(2), 1998, pp. 1-47.

[Chadalavada, 1997] V. Chadalavada, V. Vittal, G. C. Ejebe, G. D. Irisarri, J. Tong, G. Pieper, M. McMullen, "An On-Line Contingency Filtering Scheme for Dynamic Security Assessment". IEEE Transactions on Power Systems, Vol. 12, No. 1, February 1997, pp. 153-161.

[Chang, 1990] C. S. Chang, T. S. Chung, K. L. Lo, "Application of Pattern Recognition Technique to Power System Security Analysis and Optimization". IEEE Transactions on Power Systems, Vol. 5, Issue 3, Aug. 1990, pp. 835-841.

[Chang, 1994] C. S. Chang, D. Srinivasan, A. C. Liew, "A Hybrid Model for Transient Stability Evaluation of Interconnected Longitudinal Power Systems Using Neural Network/Pattern Recognition Approach". IEEE Transactions on Power Systems, Vol. 9, No. 1, February 1994.

[Cherkassky, 1998] V. Cherkassky, F. Mulier, *Learning from Data: Concepts, Theory and Methods*. John Wiley & Sons, 1998.

- [Chiang, 1987] H. Chiang, F. Wu, P. Varaiya, "Foundations of Direct Methods for Power System Transient Stability Analysis". IEEE Transactions on Circuits and Systems, Vol. CAS-34, No. 2, February 1987, pp. 160-173.
- [Chiang, 1988] H. Chiang, F. Wu, P. Varaiya, "Foundation of the Potential Energy Boundary Surface Method for Power System Transient Stability Analysis". IEEE Transactions on Circuits and Systems, Vol. CAS-35, No. 6, June 1988, pp. 712-728.
- [Chiang, 1989] H. Chiang, "A Theory-Based Controlling UEP Method for Direct Analysis of Power System Transient Stability". Proceedings of the International Symposium on Circuits and Systems (IEEE ISCAS'89), Portland, Oregon, May 1989, pp. 1980-1983.
- [Chiang, 1994] H. Chiang, F. Wu, P. Varaiya, "A BCU Method for Direct Analysis of Power System Transient Stability". IEEE Transactions on Power Systems, Vol. 9, No. 3, Aug. 1994, pp. 1194-1208.
- [Chiang, 1999] H. D. Chiang, C. S. Wng, H. Li, "Development of BCU Classifiers for On-Line Dynamic Contingency Screening of Electric Power Systems". IEEE Transactions on Power Systems, Vol. 14, No. 2, May 1999, pp. 660-666.
- [Chung, 1997] K. Chung, J. Yoon, "Performance Comparison of Several Feature Selection Methods Based on Node Pruning in Handwritten Character Recognition". Proceedings of the Fourth International Conference on Document Analysis and Recognition, 1997, Vol. 1, pp. 11-15.
- [Cortes, 1995] C. Cortes, V. Vapnik, "Support Vector Networks". Machine Learning, 20, pp. 273-297, 1995.
- [Cristianini, 2000] N. Cristianini, J. Shawe-Taylor, *An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods*. Cambridge University Press, 2000.
- [Decker, 1992] I. C. Decker, D. M. Falcão, E. Kaszkurewicz, "Parallel Implementation of a Power System Dynamic Simulation Methodology Using the Conjugate Gradient Method". IEEE Transactions on Power Systems, Vol. 7, No. 1, Feb. 1992, pp. 458-465.

- [Demaree, 1994] K. Demaree, T. Athay, K. W. Cheung, Y. Mansour, E. Vaahedi, A. Y. Chang, B. R. Corns, B. W. Garrett, "An On-line Dynamic Security Analysis System Implementation". IEEE Transactions on Power Systems, Vol. 9, No. 4, Nov. 1994, pp. 1716-1722.
- [Dorrnsoro, 1997] J. R. Dorrnsoro, F. Ginel, C. Santa Cruz. IEEE Transactions on Neural Networks, Vol. 8, No. 4, July 1997.
- [Duda, 1973] R. O. Duda, P. E. Hart, *Pattern Classification and Scene Analysis*. John Wiley and Sons, 1973.
- [El-Abiad, 1966] A. H. El-Abiad, K. Nagappan, "Transient Stability Region of Multi Machine Power Systems". IEEE Transactions on Power Apparatus and Systems, Vol. PAS-85, No. 2, pp. 169-178, Feb. 1966.
- [El-Kady, 1986] M. A. El-Kady, "Dynamic Security Assessment Utilizing the Transient Energy Function Method. IEEE Transactions on Power Systems, Vol. 1, No. 3, August 1986, pp. 284-291.
- [El-Sharkawi, 1996a] M. A. El-Sharkawi, "Neural Network Application to Security Assessment". Neural Networks Applications in power systems, Editado por T. S. Dillon e D. Niebur, CRL Publishing Ltd, 1996.
- [El-Sharkawi, 1996b] M. A. El-Sharkawi, D. Niebur (Editores), *Application of Artificial Neural Networks to Power Systems*. IEEE Press, 96 TP 112-0, 1996.
- [El-Sharkawi, 2000] M. A. El-Sharkawi (Course Coordinator), R. Eberhart, R. J. Marks, C. Jensen, A. P. Alves da Silva, K. Tomsovic, "Tutorial on Evolutionary Techniques and Fuzzy Logic in Power Systems". Prepared for the IEEE-PES Summer Meeting in Seattle, July 2000.
- [EPRI, 1993] EPRI, "Analytical Methods for DSA Contingency Selection and Ranking, Users Manual for DIRECT Version 4.0", EPRI TR-105886, Palo Alto, California, USA, 1993.
- [EPRI, 1994] EPRI, ETMSP – Extended Transient Midterm Stability Program User's Manual, Vol. 1-6. Version 3.1. EPRI TR-102004-V2R1. Final Report, May 1994.

- [Fischl, 1989] R. Fischl, M. Kam, J.-C. Chow, S. Ricciardi, "Screening Power System Contingencies Using a Back-Propagation Trained Multiperceptron". IEEE International Symposium on Circuits and Systems, Vol. 1, Portland, OR, 1989, pp. 486-489.
- [Fouad, 1981] A. Fouad, S. Stanton, "Transient Stability of a Multi-Machine Power System Part I: Investigation of System Trajectories and Part II: Critical Transient Energy". IEEE Transactions on Power Apparatus and Systems, Vol. PAS-100, No. 7, July 1981, pp. 3408-3424.
- [Fouad, 1992] A. A. Fouad, V. Vital, *Power System Transient Stability Analysis Using the Transient Energy Function Method*. Prentice Hall, 1992.
- [Goldberg, 1989] D. E. Goldberg, *Genetic Algorithms in Search, Optimization and Machine Learning*, Adison-Wesley, 1989.
- [Hansen, 1997] J. V. Hansen, R. D. Nelson, "Neural Networks and Traditional Time Series Methods: A Synergistic Combination in State Economic Forecasts Neural Fraud Detection in Credit Card Operations". IEEE Transactions on Neural Networks, Vol. 8, No. 4, July 1997.
- [Haykin, 1999] S. Haykin, "Neural Networks – A Comprehensive Foundation". Second Edition, Prentice Hall, 1999.
- [Holland, 1975] J. H. Holland, *Adaptation in Natural and Artificial Systems*. University of Michigan Press, 1975.
- [Hwang, 1991] J.-N. Hwang, J. J. Choi, S. Oh, R. J. Marks II, "Query-Based Learning Applied to Partially Trained Multilayer Perceptrons". IEEE Transactions on Neural Networks, Vol. 2, No. 1, Jan. 1991, pp.131-136.
- [IEEE, 1992] IEEE Committee Report, "Transient Stability Test Systems for Direct Stability Methods". Transactions on Power Systems, Vol. 7, No. 1, February 1992.
- [GESIS, 2000] GESIS – Grupo de Engenharia de Sistemas, Escola Federal de Engenharia de Itajubá, "Estudos para Determinação de Limites Operativos em Redes Elétricas de 500 e 345 KV nas Regiões do Paranaíba e Grande". Relatório GESIS/FUPAI, Contrato ONS / FUPAI 107/2000.

- [Jain, 1997] A. Jain, D. Zongker, "Feature Selection: Evaluation, Application, and Small Sample Performance". IEEE Trans. On Pattern Analysis and Machine Intelligence, Vol. 19, No. 2, February 1997, pp. 153-158.
- [Jardim, 2000a] J. L. Jardim, C. A. da S. Neto, A. P. Alves da Silva, A. C. Zambroni de Souza, D. M. Falcão, C. L. T. Borges, G. N. Taranto, "A Unified Online Security Assessment System". Anais da Sessão Bienal de 2000 do CIGRÉ, pp. 36-40.
- [Jardim, 2000b] J. L. Jardim, "OnLine Dynamic Security Assessment: Implementation Problems and Potential use of Artificial Intelligence". IEEE PES Summer Meeting, 2000, Seattle, WA.
- [Jensen, 1999] C. Jensen, M. A. El-Sharkawi, R. J. Marks, "Power System Security Boundary Enhancement Using Evolutionary-Based Query Learning". Invited Paper, International Journal of Engineering Intelligent Systems, pp. 215-218, Dec. 1999.
- [Jianchang, 1994] M. Jianchang, K. Mohiuddin, A. K. Jain, "Parsimonious Network Design and Feature Selection Through Node Pruning". Proceedings of the 12th IAPR International Conference on Pattern Recognition, 1994, Vol. 2, pp.622-624.
- [Joachims, 1998a] T. Joachims, "Text Categorization with Support Vector Machines: Learning with Many Relevant Features". Proceedings of the 10th European Conference on Machine Learning (ECML), Springer-Verlag, 1998.
- [Joachims, 1998b] T. Joachims, "Making Large-Scale SVM Learning Practical". In *Advances in Kernel Methods and Other Kernel-Based Learning Methods – Support Vector Learning*, B. Scholkopf, J. C. C. Burges, A. J. Smola (editors). MIT Press, Cambridge, USA, 1998.
- [Kakimoto, 1984] N. Kakimoto, "Transient Stability Analysis of Large-Scale Power System by Lyapunov's Direct Method". IEEE Transactions on Power Apparatus and Systems, Vol. PAS-103, No. 1, Jan 1984, pp. 160-167.
- [Kamwa, 2001] I. Kamwa, R. Grondin, L. Loud, "Time-Varying Contingency Screening for Dynamic Security Assessment Using Intelligent-Systems Techniques". IEEE Transactions on Power Systems, Vol. 16, No. 3, August 2001, pp. 526-536.

- [Karapidakis, 2002] E. S. Karapidakis, N. D. Hatziargyriou, "Committee Machines for On-line Power System Dynamic Security Assessment". Anais da Conferência IEEE Intelligent System Application to Power, 2002, Budapest, Hungary, Paper No. 59.
- [Kassabalidis, 2002] I. N. Kassabalidis, M. A. El-Sharkawi, R. J. Marks II, L. S. Moulin, A. P. Alves da Silva, "Dynamic Security Border Identification Using Enhanced Particle Swarm Optimization". IEEE Transactions on Power Systems, em publicação.
- [Khoshgoftaar, 1997] T. M. Khoshgoftaar, E. B. Allen, S. J. Aud, "Application of Neural Networks to Software Quality Modeling of a Very Large Telecommunications System". IEEE Transactions on Neural Networks, Vol. 8, No. 4, July 1997.
- [Khotanzad, 1997] A. Khotanzad, R. Afkhami-Rohani, D. J. Maratukulam, "ANNSTLF – A Neural-Network-Based Electric Load Forecasting System". IEEE Transactions on Neural Networks, Vol. 8, No. 4, July 1997.
- [Kumar, 1991] B. R. Kumar, A. Pakchi, V. Brandwajn, M. A. El-Sharkawi, G. Gauley, "Neural Networks for Dynamic Security Assessment of Large-Scale Power Systems: Requirements Overview". First International Forum on Applications of Neural Networks to Power Systems, Seattle, July 23-26, 1991.
- [Kundur, 1993] P. Kundur, *Power System Stability and Control*. McGraw-Hill, 1993.
- [Lee, 1994] L. L. Lee, "On Two-Pattern Classification and Feature Selection Using Neural Networks". IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP-94, 1994, Volume 2, pp. II/617-II/620.
- [Mansour, 1997a] Y. Mansour, A Y. Chang, J Tamby, E Vaahedi, M. A. El-Sharkawi, "Large Scale Dynamic Security Screening and Ranking Using Neural Networks". IEEE Trans. On Power Systems, Vol. 12, No. 2, May 1997.
- [Mansour, 1997b] Y. Mansour, E. Vaahedi, M. A. El-Sharkawi, "Dynamic Security Contingency Screening and Ranking Using Neural Networks". IEEE Transactions on Neural Networks, Vol. 8, No. 4, July 1997, pp. 942-950.

- [Mansour, 2000] Y. Mansour, "Competition and System Stability – The Reward and the Penalty". Proceedings of the IEEE, Vol. 882, Feb. 2000, pp. 228-234.
- [Marceau, 1993] R. Marceau, R. Mailhot, F. Galiana, "A Generalized Shell for Dynamic Security Analysis in Operations Planning". IEEE Transactions on Power Systems, Vol. 8, No. 3, Aug. 1993, pp. 1098-1832.
- [McCalley, 1997] J. D. McCalley, Q. Zhao, S. Wang, G. Zhou, R. T. Treinen, A. D. Papalexopoulos, "Security Boundary Visualization for Systems Operation". IEEE Transactions on Power Systems, Vol. 12, No. 2, May 1997, pp. 940-947.
- [Minsky, 1969] M. Minsky, S. Papert, *Perceptrons*. MIT Press, 1969.
- [Miranda, 1994] V. Miranda, J. N. Fidalgo, J. A. Peças Lopes, L. B. Almeida, "Real Time Preventive Actions for Transient Stability Enhancement with a Hybrid Neural Network – Optimization Approach". IEEE PES Summer Meeting, July 1994.
- [Morison, 1999] K. Morison, H. Hamadanizadeh, L. Wang, "Dynamic Security Assessment Tools". IEEE Power Engineering Society Summer Meeting, 1999.
- [Moulin, 2001a] L. S. Moulin, M. A. El-Sharkawi, R. J. Marks II, A. P. Alves da Silva: "Automatic Feature Extraction for Neural Network Based Power Systems Dynamic Security Evaluation", anais da IEEE International Conference on Intelligent System Application to Power Systems, Budapest, Hungary, June 2001, pp. 41-46.
- [Moulin, 2001b] L. S. Moulin, A. P. Alves da Silva, M. A. El-Sharkawi, R. J. Marks II: "Neural Networks and Support Vector Machines Applied to Power Systems Transient Stability Analysis", International Journal of Engineering Intelligent Systems, Vol. 9, No. 4, December 2001.
- [Moulin, 2002] L. S. Moulin, A. P. Alves da Silva, M. A. El-Sharkawi, R. J. Marks II, "Support Vector and Multilayer Perceptron Neural Networks Applied to Power Systems Transient Stability Analysis with Input Dimensionality Reduction", IEEE Power Engineering Society Summer Meeting 2002, July, Chicago, Panel Session (a ser apresentado).

- [Narendra, 1977] P. M. Narendra, K. Fukunaga, "A Branch and Bound Algorithm for Feature Selection". *IEEE Transactions on Computers*, C-26(9):917-922, September, 1977.
- [Omata, 1993] K. Omata, K. Tanomura, "Transient Stability Evaluation Using an Artificial Neural Network". *Proceedings of the Second International Forum on Applications of Neural Networks to Power Systems, ANNPS'93*, 1993, pp. 130-135.
- [Osuna, 1997] E. Osuna, R. Freund, F. Girosi, "Training Support Vector Machines: An Application to Face Detection". *Proceedings of CVPR'97*, Puerto Rico, 1997.
- [Pai, 1989] M. A. Pai, *Energy Function Analysis for Power System Stability*. Kluwer Academic Publishers, 1989.
- [Pang, 1974] C. K. Pang, F. S. Prabhakara, A. H. El-Abiad, A. J. Koivo, "Security Evaluation in Power Systems Using Pattern Recognition". *IEEE Transactions on Power Apparatus and Systems*, Vol. PAS-93, May/June 1974, pp. 969-976.
- [Pao, 1992] Y.-H. Pao, D. J. Sobajic, "Combined Use of Unsupervised and Supervised Learning for Dynamic Security Assessment". *IEEE Transactions on Power Systems*, Vol. 7, No. 2, May 1992.
- [Principe, 1997] E. Osuna, R. Freund, F. Girosi, "An Improved Training Algorithm for Support Vector Machines". In *Proceedings of the 1997 IEEE Workshop on Neural Networks for Signal Processing*, J. Principe, L. Giles, N. Morgan, E. Wilson, Editors, pp. 276-28, Amelia Island, FL, 1997.
- [Raudys, 1992] S. Raudys, "Accuracy of Feature Selection and Extraction in Statistical and Neural Net Pattern Classification". *Proceedings of the 11th IAPR International Conference on Pattern Recognition*, 1992, Vol. 2, pp. 62-70.
- [Roadknight, 1997] C. M. Roadknight, G. R. Balls, D. Palmer-Brown, "Modeling Complex Environmental Data". *IEEE Transactions on Neural Networks*, Vol. 8, No. 4, July 1997.
- [Rosenblatt, 1958] F. Rosenblatt, "The Perceptron: A Probabilistic Model for Information Storage and Organization in the Brain". *Psychological Review*, Vol. 65, pp. 386-408, 1958.

- [Rovnyak, 1994] S. Rovnyak, S. Kretsinger, J. Thorp, D. Brown, "Decision Trees for Real-Time Transient Stability Prediction", IEEE Transactions on Power Systems, Vol. 9, No. 3, Aug. 1994, pp. 1417-1426.
- [Saito, 1975] O. Saito, K. Koizumu, M. Udo, M. Sato, H. Mukae, T. Tsuzi, "Security Monitoring Systems Including Fast Transient Stability Studies". IEEE Transactions on Power Apparatus and Systems, Vol. PAS-94, No. 5, Sep/Oct 1975, pp. 1789-1805.
- [Scala, 1996] M. La Scala, G. Lorusso, R. Sbrizzai, M. Trovato, "A Qualitative Approach to the Transient Stability Analysis". IEEE Transactions on Power Systems, Vol. 11, No. 4, November 1996, pp. 1996-2002.
- [Schimidt, 1996] M. Schmidt, "Identifying Speaker with Support Vector Networks". In Interface '96 Proceedings, Sydney, 1996.
- [Scholkopf, 1995] B. Scholkopf, C. Burges, V. Vapnik, "Extracting Support Data For a Given Task". In U. M. Fayyad and R. Uthurusamy, Editors. Proceedings of the First International Conference on Knowledge Discovery & Data Mining. AAAI Press, Menlo Park, CA, 1995.
- [Scholkopf, 1996] B. Scholkopf, C. Burges, V. Vapnik, "Incorporating Invariances in Support Vector Learning Machines". In C. Von Der Malsburg, W. Von Seelen, J. C. Vorbruggen, B. Sendhoff, Editors, Artificial Neural Networks – ICANN'96, pp. 47-52, Berlin, 1996. Springer Lecture Notes in Computer Science, Vol. 1112.
- [Scholkopff, 1999] B. Scholkopf, C. Burges, A. Smola (Editors), *Advances in Kernel Methods – Support Vector Learning*. MIT Press, Cambridge, MA, 1999.
- [Sobajic, 1989] D. J. Sobajic, Y.-H. Pao, "Artificial Neural-Net Based Dynamic Security Assessment for Electric Power Systems". IEEE Transactions on Power Systems, Vol. 4, No. 1, February 1989, pp. 220-228.
- [Sterngold, 2001] J. Sterngold, "California Endures a 2nd Straight Day of Power Blackouts". The New York Times, Jan. 19th 2001.

- [Tang, 1994] C. K. Tang, C. E. Graham, M. El-Kady, R. T. H. Alden, "Transient Stability Index from Conventional Time Domain Simulation". IEEE Transactions on Power Systems, Vol. 9, No. 3, Aug. 1994, pp. 1524-1530.
- [Taoka, 1992] H. Taoka, I. Iyoda, H. Noguchi, N. Sato, T. Nakzawa, "Real-Time Digital Simulator for Power System Analysis on Hypercube Computer". IEEE Transactions on Power Systems, Vol. 7, No. 1, February 1992, pp. 1-10.
- [Vapnik, 1998] V. N. Vapnik, *Statistical Learning Theory*. John Wiley & Sons, 1998.
- [Wehenkel, 1989] L. Wehenkel, T. Van Cutsem, M. Pavella, "An Artificial Intelligence Framework for On-Line Transient Stability Assessment of Power Systems". IEEE Transactions on Power Systems, Vol. 4, No. 2, May 1989.
- [Wehenkel, 1994] L. Wehenkel, M. Pavella, E. Euxibie, B. Heilbronn, "Decision Tree Based Transient Stability Method: A Case Study", IEEE Transactions on Power Systems, Vol. 9, No. 1, Feb. 1994, pp. 459-469.
- [Werbos, 1974] P. J. Werbos, *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences*. Ph. D. Thesis, Harvard University, Cambridge, M A, 1974.
- [Zayan, 1996] M. B. Zayan, M. A. El-Sharkawi, N. R. Prasad, "Comparative Study of Feature Extraction Techniques for Neural Network Classifier". Proceedings of the International Conference on Intelligent Systems Applications to Power Systems, ISAP'96, 1996, pp. 400-404.
- [Zhou, 1994] Q. Zhou, J. Davidson, A. A. Fouad, "Application of Artificial Neural Networks in Power System Security and Vulnerability Assessment". IEEE Transactions on Power Systems, Vol. 9, No. 1, February 1994.

Anexo I

Redes Neurais Artificiais

O desenvolvimento das RNs deu-se a partir do início da década de 60 com Rosenblatt, como um sistema de aprendizagem capaz de classificar conjuntos de padrões através da modificação de suas conexões internas [Rosenblatt, 1958]. Assim, criou-se o *perceptron* de única camada, que foi um dos marcos do período inicial de desenvolvimento da área. Esse sistema recebeu críticas por parte de Minsky e Pappert quanto à sua aplicabilidade em funções lógicas de maior complexidade, devido à sua natureza linear [Minsky, 1969]. Posteriormente os obstáculos foram removidos com a adição de camadas intermediárias e pelo uso de algoritmos de treinamento como o Retropropagação de Erros, desenvolvido por P. Werbos [Werbos, 1974]. Com isso, a partir do início dos anos 80 ressurgiu um ambiente propício ao desenvolvimento e um interesse pelas RNAs. Os avanços de *hardware* permitiram um grande aumento da capacidade de processamento, avanços teóricos foram conseguidos em algoritmos e técnicas computacionais, pesquisadores como Hopfield, Grossberg, Widrow, Anderson e Kohonen criaram diferentes arquiteturas de RNs e aplicações que demonstraram o potencial da nova tecnologia.

A partir de então deu-se o desenvolvimento das Redes Neurais treinadas através do aprendizado supervisionado, que possibilitou sua aplicação para a solução de problemas importantes de várias áreas, seja na indústria, no âmbito governamental ou na área financeira. Aplicações como nas áreas de Engenharia Ambiental [Roadknight, 1997], Engenharia de Software [Khoshgoftaar, 1997], Economia [Hansen, 1997], Finanças [Dorransoro, 1997], Previsão de Carga Elétrica [Khotanzad, 1997] e Análise de Estabilidade Dinâmica de Sistemas de Potência [Jensen, 1999] atestam o interesse crescente pelas RNs por parte de vários setores. Essa é uma lista de aplicações representativa, mas longe de ser completa. O número crescente de aplicações e o rápido progresso na pesquisa e desenvolvimento gerou a criação de organizações, conferências e periódicos para estudo e desenvolvimento do tema RNs.

As RNs já têm aplicações em numerosas áreas. Devido à sua capacidade de aprender relações não lineares e/ou complexas através de exemplos, uma grande quantidade de problemas podem ser atacados com essa abordagem. As RNs são adequadas para vários tipos de problemas em sistemas elétricos como análise de estabilidade, previsão de carga, detecção e avaliação de harmônicos, diagnóstico de faltas, controle adaptativo e processamento de alarmes.

As RNs são um método de sintetizar o mapeamento entre variáveis de entrada e de saída, ajustando um conjunto de valores denominados *pesos das interconexões*, num modelo conexionista e baseado em exemplos de treinamento. Seu desenvolvimento gerou uma gama de configurações que possuem algumas características subjacentes comuns. Todos os tipos de RNs tentam obter bom desempenho através de várias interconexões entre elementos simples. Elas são caracterizadas pelo modelo dos neurônios, pelas conexões entre eles e pelos métodos usados para treiná-las.

Uma RN pode ser definida como um arranjo de neurônios ou processadores elementares conectados. Os neurônios são ligados através de interconexões, análogas às sinapses biológicas. Por isso, diz-se que as RNs são treinadas ao invés de programadas no sentido convencional.

Há vários tipos de RNs, e estão entre eles as Perceptron Multicamadas (MLP), Kohonen e Hopfield. Com as Perceptron Multicamadas é utilizado o treinamento supervisionado, quando é apresentada a saída desejada para cada padrão de entrada. Até o momento, a grande maioria das aplicações prática tem utilizado as MLP como arquitetura predominante.

I.1 – Perceptrons Multicamadas

O elemento básico de uma RN é o neurônio, como mostrado na Figura I.1, sendo

$$v = \sum_{i=1}^n w_i x_i + b \quad (I.1)$$

e $\varphi(v)$, uma função não linear monotônica e continuamente diferenciável, chamada de função de ativação. Talvez a função de ativação mais comum de se encontrar seja a sigmoideal, que é dada por

$$\varphi(v) = \frac{1}{1 + e^{-v}} \quad (I.2)$$

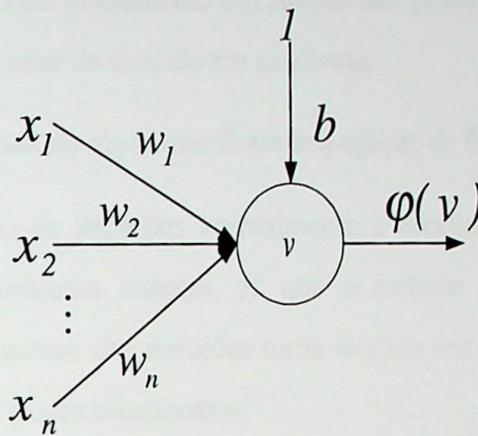


Figura I.1 Neurônio Matemático

Uma RN do tipo MLP constitui-se de um conjunto de neurônios organizados em camadas e conectados através dos pesos das conexões, como mostrado na Figura I.2. Um neurônio genérico, i , é conectado a outro neurônio genérico, j , através do peso w_{ji} . O número de camadas e de neurônios depende da aplicação em questão.

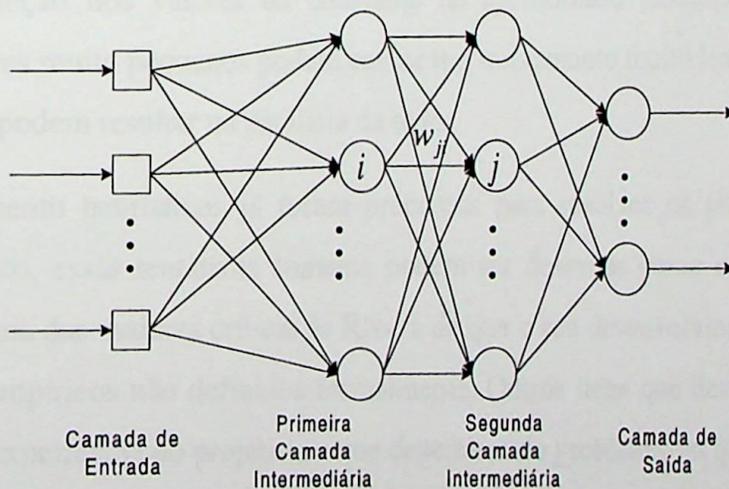


Figura I.2 Perceptron Multicamadas

Através da estrutura interconectada, os sinais de ativação dos neurônios de uma camada são transmitidos à camada posterior através dos pesos das conexões que atenuam ou amplificam o sinal.

A maneira mais utilizada de se treinar uma RN do tipo MLP é o algoritmo Retropropagação de Erros. Em sua forma fundamental, o Retropropagação de Erros calcula derivadas parciais dos erros de treinamento em função dos pesos das conexões, ajustando-os de forma iterativa num processo de descida em gradiente.

Alguns inconvenientes do algoritmo Retropropagação de Erros são:

1. Um número alto de iterações normalmente é necessário para treinar uma MLP, mesmo num problema simples. Já que o método é baseado em descida em gradiente, o progresso das iterações torna-se cada vez mais lento à medida que se aproxima de um ponto estacionário;
2. O treinamento pode ficar preso num mínimo local;
3. O treinamento pode chegar num platô virtual devido à paralisia da rede. Isso acontece num ponto em que os pesos são ajustados em valores muito altos, que causa saturação das funções de ativação nos neurônios e valor quase zero das derivadas;
4. A seleção dos valores da constante de aprendizado baseia-se na experiência. Valores muito pequenos podem causar um treinamento muito lento e valores muito altos podem resultar na paralisia da rede.

Vários procedimentos heurísticos já foram propostos para resolver os problemas descritos acima. No entanto, essas tentativas somente podem ser descritas como experimentais e na verdade essa é uma das maiores críticas às RNs, a de que o seu desenvolvimento baseia-se em conhecimentos empíricos não definidos formalmente. Outros itens que devem ser resolvidos recorrendo-se à experiência do projetista e que dependem do problema em questão são:

◆ Seleção da arquitetura da rede neural e do algoritmo de treinamento, tais como tamanho da RN, constante de aprendizado, número de padrões de treinamento e iterações;

- ◆ Determinação da consistência dos conjuntos de treinamento e teste, como faixa de variação dos dados e propriedades estatísticas dos mesmos;
- ◆ Definição de atributos que ofereçam boa correlação entre os padrões de treinamento e a resposta desejada da RN;
- ◆ Técnicas de Extração de Atributos para reduzir a dimensionalidade dos dados de entrada, e ao mesmo tempo preservar o desempenho da RN;
- ◆ Capacidade de generalização da RN dentro da faixa de operação do sistema. A RN deve “aprender” os padrões de entrada e não memorizá-los;
- ◆ Capacidade da RN em manter os pesos das conexões numa região viável sem causar saturação de um grande número de neurônios;

As RNs do tipo MLP podem ser utilizadas para classificação e regressão. Na classificação, a RN é treinada como função discriminante baseada num conjunto de exemplares de treinamento. O sucesso do classificador depende de sua capacidade de discriminar corretamente padrões que não tenham sido vistos previamente. As MLP possuem uma capacidade de generalização e de rejeição a ruídos que as torna bastante adequadas para a tarefa.

Nos problemas de regressão, o objetivo das MLP é aprender um mapeamento funcional entre um conjunto de variáveis de entrada e de saída. A capacidade de generalização das RN torna-as bastante adequadas para a tarefa de regressão/identificação.

I.2 – Algoritmo Retropropagação de Erros

A regra delta generalizada, mais comumente chamada de algoritmo *error back-propagation*, ou Retropropagação de Erro, realiza a descida em gradiente numa superfície de um espaço de pesos cuja abscissa em qualquer ponto do espaço é igual à medida de erro. Seja

$$E_p = \frac{1}{2} \sum_j (t_{pj} - o_{pj})^2 \quad (I.3)$$

a medida do erro num dado padrão de entrada/saída p , sendo t , correspondente à saída desejada, e o , correspondente à saída apresentada pela RN.

E seja

$$E = \sum_p E_p \quad (I.4)$$

a medida acumulada do erro ao longo de todos os padrões de treinamento.

Pode-se definir a soma ponderada das saídas da camada de neurônios anterior:

$$net_{pj} = \sum_i w_{ji} o_{pi} \quad (I.5)$$

A saída,

$$o_{pj} = f_j(net_{pj}) \quad (I.6)$$

utiliza uma função de ativação monotônica não-linear e continuamente diferenciável. Para obter a descida em gradiente, w_{ji} é ajustado da seguinte forma:

$$\Delta_p w_{ji} \propto -\frac{\partial E_p}{\partial w_{ji}} \quad (I.7)$$

Pode-se ver que essa derivada é formada pelo produto de duas partes: uma parte reflete a variação do erro com relação ao somatório de entrada no neurônio e outra parte reflete a variação do erro com relação à variação de um dos pesos do somatório de entrada no neurônio. Assim, pode-se escrever

$$\frac{\partial E_p}{\partial w_{ji}} = \frac{\partial E_p}{\partial net_{pj}} \frac{\partial net_{pj}}{\partial w_{ji}} \quad (I.8)$$

De (I.5), o segundo fator é dado por

$$\frac{\partial net_{pj}}{\partial w_{ji}} = \frac{\partial}{\partial w_{ji}} \sum_k w_{jk} o_{pk} = o_{pi} \quad (I.9)$$

Pode-se definir

$$\delta_{pj} = -\frac{\partial E_p}{\partial net_{pj}} \quad (I.10)$$

A equação (I.8) tem a forma equivalente:

$$-\frac{\partial E_p}{\partial w_{ji}} = \delta_{pj} o_{pi} \quad (I.11)$$

Isso quer dizer que para implementar a descida em gradiente em E , as atualizações dos pesos devem ser feitas da seguinte maneira:

$$\Delta_p w_{ji} = \eta \delta_{pj} o_{pi} \quad (I.12)$$

O resultado interessante é que há uma maneira recursiva simples de se calcular os δ 's e possível de ser implementada ao se propagar um sinal de erro para trás através da rede.

Para calcular (I.10), a regra da cadeia é aplicada para escrever essa derivada parcial como o produto de dois fatores: um fator que reflete a variação do erro como função da saída de uma unidade, e outro fator que reflete a variação da saída como função da entrada:

$$\delta_{pj} = -\frac{\partial E_p}{\partial net_{pj}} = -\frac{\partial E_p}{\partial o_{pj}} \frac{\partial o_{pj}}{\partial net_{pj}} \quad (I.13)$$

De (I.6) tem-se que

$$\frac{\partial o_{pj}}{\partial net_{pj}} = f'_j(net_{pj}) \quad (I.14)$$

que é simplesmente a derivada da função de ativação f_j para a j -ésima unidade, tomada no valor de entrada net_{pj} para aquela unidade.

Já calculado o segundo fator, para calcular o primeiro fator há dois casos:

a) Primeiro, considerando que a unidade u_j seja da camada de saída da rede. Neste caso, da definição de E_p , segue que

$$\frac{\partial E_p}{\partial o_{pj}} = -(t_{pj} - o_{pj}) \quad (I.15)$$

Substituindo os dois fatores em (I.13), tem-se que

$$\delta_{pj} = (t_{pj} - o_{pj}) f'_j(\text{net}_{pj}) \quad (I.16)$$

b) Agora, considerando que a unidade u_j não seja da camada de saída da rede.

Neste caso, a regra da cadeia é utilizada para escrever

$$\begin{aligned} \frac{\partial E_p}{\partial \text{net}_{pj}} \sum_k \frac{\partial E_p}{\partial \text{net}_{pk}} \frac{\partial \text{net}_{pk}}{\partial o_{pj}} &= \sum_k \frac{\partial E_p}{\partial \text{net}_{pk}} \frac{\partial}{\partial o_{pj}} \sum_i w_{ki} o_{pi} = \\ &= \sum_k \frac{\partial E_p}{\partial \text{net}_{pk}} w_{kj} = -\sum_k \delta_{pk} w_{kj} \end{aligned} \quad (I.17)$$

Substituindo os dois fatores em (I.13) tem-se

$$\delta_{pj} = f'_j(\text{net}_{pj}) \sum_k \delta_{pk} w_{kj} \quad (I.18)$$

As equações (I.16) e (I.18) fornecem um procedimento para se calcular os δ 's para todas as unidades da rede, os quais são utilizados para calcular as variações dos pesos de acordo com (I.12). Esse procedimento constitui a regra delta generalizada para as redes neurais tipo *feedforward*.

Assim, pode-se descrever o algoritmo de retropropagação de erro da seguinte forma:

PASSO 1) *Inicializar Pesos*: Ajusta todos os pesos para valores pequenos de forma aleatória.

PASSO 2) *Apresentar Entradas e Saídas Desejadas*: Apresenta um vetor com valores de entrada e especifica as saídas desejadas. Em cada tentativa, as entradas podem ser diferentes, se retiradas de um conjunto de treinamento.

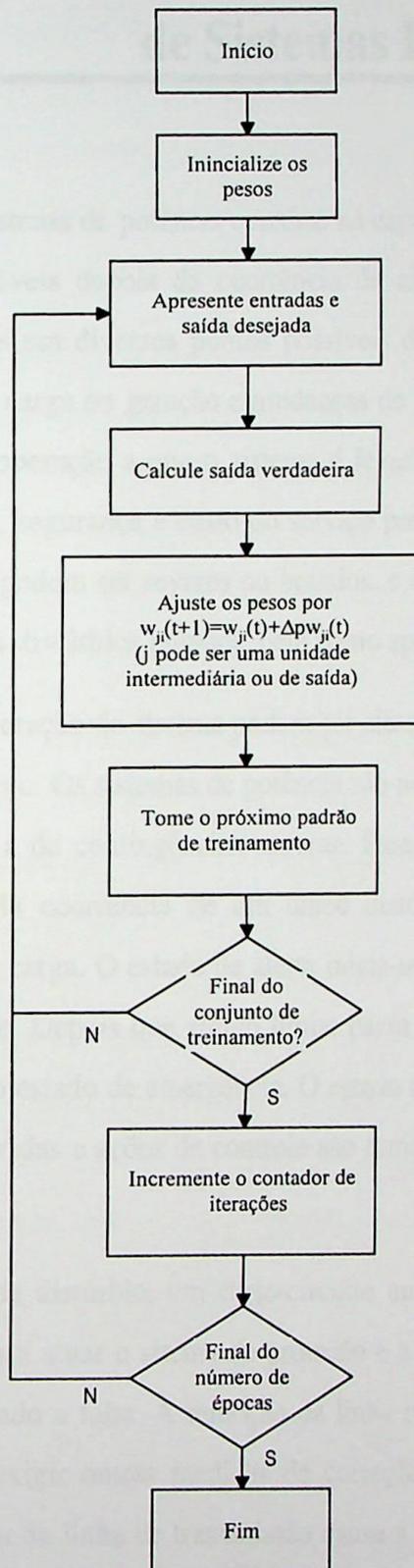
PASSO 3) *Calcular Saídas da Rede*. Utiliza as funções de ativação para calcular as saídas.

PASSO 4) *Ajustar Pesos*. Utiliza um algoritmo recursivo, começando da camada de saída até a primeira camada intermediária. Ajusta os pesos pela fórmula iterativa:

$$\boxed{w_{ji}(n+1) = w_{ji}(n) + \Delta w_{ji}(n) + \alpha \Delta w_{ji}(n-1)} \quad (\text{I.19})$$

onde $\Delta_p w_{ji} = \eta \delta_{pj} o_{pi}$, como mostrado em (I.12). η é a taxa de aprendizado e α é a taxa de momento. Se u_j é uma unidade da camada de saída, δ_{pj} é calculado por (I.16); caso contrário, é calculado por (I.18).

DIAGRAMA DE FLUXO DO ALGORITMO RETRO-PROPAGAÇÃO DE ERRO



Anexo II

Análise de Estabilidade Dinâmica de Sistemas Elétricos de Potência

A estabilidade de um sistema de potência consiste na capacidade de manutenção da operação dentro de limites aceitáveis depois da ocorrência de algum distúrbio. Tal distúrbio pode caracterizar-se por faltas em diversos pontos possíveis do sistema, falhas de equipamentos, bem como variações de carga ou geração e mudanças de topologia. A análise de estabilidade estuda as variações de operação a que o sistema é levado por dado distúrbio, em vista das exigências de qualidade, segurança e custo do serviço prestado aos consumidores de energia elétrica. Tais distúrbios podem ser severos ou brandos, e o sistema deve ser capaz de ajustar-se satisfatoriamente após distúrbios brandos, bem como após certos distúrbios severos.

Os estados de operação do sistema podem ser classificados como normal, de alerta, de emergência ou restaurativo. Os sistemas de potência são normalmente projetados com vistas a suportar o critério $n - 1$ de contingências severas. Esse critério determina a operação em estado normal depois da ocorrência de um único distúrbio, sem a violação dos limites elétricos e sem perda de carga. O estado de alerta inicia-se caso uma contingência resulte em violação de algum limite. Depois que algum limite passa a ser violado, mas as cargas ainda são supridas, passa-se ao estado de emergência. O estado restaurativo inicia-se quando cargas passam a não serem supridas e ações de controle são tomadas para levar o sistema ao estado normal.

Como exemplo de distúrbio, um curto-circuito numa linha de transmissão causando uma corrente de falta faria atuar o sistema de proteção e a abertura dos disjuntores retiraria a linha de operação, isolando a falta. A remoção da linha representaria um outro distúrbio no sistema e que poderia exigir outras medidas de correção, dependendo do novo estado de operação. Caso a retirada da linha de transmissão cause a retirada de outros componentes do sistema, isso pode levar a uma série de distúrbios em cascata e a um "black-out" total do sistema ou de grande parte dele. O desenvolvimento de técnicas de análise de estabilidade

dinâmica suficientemente rápidas e confiáveis reduzem os riscos de “black-out” e proporcionam um meio de quantificar os riscos inerentes às várias estratégias de operação.

Tradicionalmente a análise de estabilidade transitória estuda a capacidade das máquinas geradoras em manter sincronismo com o resto do sistema durante o período transitório imediatamente seguinte a um distúrbio. Tal aspecto da estabilidade do sistema é determinado pela natureza dinâmica dos ângulos dos rotores e pelas relações ângulo-potência das máquinas. A ocorrência de instabilidade também é possível no caso do colapso da tensão, mesmo com a manutenção do sincronismo dos geradores.

A análise da estabilidade dos ângulos dos rotores dos geradores estuda a capacidade dos mesmos em permanecerem em sincronizados ao longo da operação do sistema. Esse estudo é importante devido ao modo como se relacionam a potência gerada e as oscilações do rotor. A frequência da tensão gerada no estator depende da velocidade do rotor. Quando duas ou mais máquinas estão conectadas, as tensões dos estatores de todas as máquinas devem ter a mesma frequência. As velocidades dos rotores são sincronizadas com essa frequência.

As correntes dos circuitos do estator produzem um campo magnético girante que, em condições de regime permanente, gira à mesma velocidade que o rotor. Os campos magnéticos do estator e do rotor reagem entre si gerando um torque que resulta da tendência entre os dois se alinharem. Esse torque opõem-se à rotação do rotor, de forma que um torque mecânico deve ser aplicado pela turbina para manter o movimento. O efeito de aumentar o torque mecânico faz avançar a posição do rotor com relação ao campo magnético girante do estator. Inversamente, a redução do torque mecânico aplicado faz retardar a posição do rotor. Em regime permanente o campo magnético do rotor e o campo magnético girante do estator têm a mesma velocidade. Porém há uma separação angular, dependendo da potência gerada.

No caso de duas ou mais máquinas interconectadas, em operação de regime permanente há um equilíbrio entre o torque mecânico aplicado a cada máquina e a potência gerada, tal que a velocidade de rotação permanece constante. Quando ocorre algum distúrbio no sistema o equilíbrio é quebrado, resultando em aceleração ou atraso dos rotores. Se um gerador girar mais rápido que um outro, a posição angular do rotor avança e a diferença angular promove a transferência de parte da carga da máquina mais lenta para a máquina mais rápida, de acordo com a relação ângulo-potência. Isso tende a reduzir a diferença de

velocidades e levar o sistema de volta ao sincronismo. Além de um certo limite, o aumento da separação angular causa uma diminuição na transferência de potência; isso aumenta ainda mais a separação angular, levando à falta de sincronismo e instabilidade.

Vários fatores influenciam a análise de estabilidade transitória, incluindo:

- ◆ Tipo e duração do distúrbio;
- ◆ Topologia do sistema antes do distúrbio;
- ◆ Condições de carregamento e despacho antes do distúrbio;
- ◆ Características dinâmicas das máquinas;
- ◆ Ações de controles após o distúrbio.

Devido ao número ilimitado de condições de operação e tipos possíveis de distúrbios, as práticas de análise de estabilidade consideram um número limitado de casos críticos, que podem basear-se no critério $n - 1$ de contingências. Esses casos são escolhidos baseado em experiência profissional e em dados históricos de ocorrências.

A análise de estabilidade dinâmica ou de pequenas perturbações estuda a capacidade do sistema de manter o sincronismo após a ocorrência de distúrbios pequenos. Tais distúrbios acontecem a todo tempo no sistema devido às variações de carga e de geração em vários pontos.

A análise de estabilidade de tensão estuda a capacidade do sistema de manter níveis de tensão aceitáveis durante a operação de regime permanente e após a ocorrência de distúrbios. O sistema inicia um estado de instabilidade de tensão quando um distúrbio qualquer, como aumento de carga e mudança de ponto de operação, causa uma queda incontrollável da tensão.

Todos os tipos de análise de estabilidade são críticos e se não forem considerados apropriadamente podem acarretar falhas e danos ao sistema. Os vários aspectos da análise de estabilidade combinados servem de ferramentas para aplicação no dia a dia da operação do sistema. O objetivo é gerar procedimentos de operação que levem o sistema a regiões seguras. Como exemplo de aplicações, podem-se citar [Morison, 1999]:

- Análise do Planejamento da Operação – análise de estratégias de operação para condições futuras do sistema (pode ser com horizonte de um dia, uma semana ou um mês à frente). Isso é necessário para planejar mudanças de configuração (tais como em ocorrência de manutenções, por exemplo) e para desenvolver instruções e calendários de operação.

- Análise da Capacidade de Transmissão – cálculo da disponibilidade de transmissão para avaliar a possibilidade de transações no mesmo dia ou no dia seguinte. Sem esse cálculo, operações agendadas podem resultar em instabilidade.

- Análise de Segurança “On-line” - Avaliação do ponto de operação atual e da possibilidade de execução das transações e mudanças de operação agendadas. Mesmo com cálculos exatos de capacidade de transmissão, condições anômalas (como a retirada de um equipamento) não consideradas no cálculo podem levar a falhas do sistema.

A seguir será apresentado um sumário sobre a análise de estabilidade dinâmica ou de pequenas perturbações, a análise de estabilidade de tensão e a análise de estabilidade transitória, com enfoque concentrado nesta última.

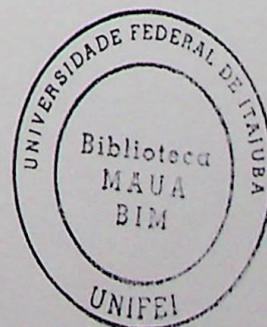
II.1 - Análise de Estabilidade de Pequenas Perturbações

Neste tipo de análise, consideram-se pequenas perturbações do sistema, tais como variações de carga e geração, e faltas de curta duração. Tais perturbações acontecem a todo momento no sistema, mas causam uma mudança no ponto de operação pequena. Isso permite uma análise através da linearização das equações dinâmicas do sistema de potência, que oferecem uma idéia do seu comportamento próximo do ponto de operação corrente.

O comportamento do sistema pode ser representado de forma genérica por um conjunto de n equações diferenciais de primeira ordem [Kundur, 1993]:

$$\dot{x}_i = f_i(x_1, x_2, \dots, x_n; u_1, u_2, \dots, u_r; t) \quad i = 1, 2, \dots, n \quad (\text{II.1})$$

sendo n a ordem do sistema e r o número de entradas. Essa equação pode ser escrita na forma compacta:



$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}) \quad (\text{II.2})$$

sendo

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \dots \\ x_n \end{bmatrix}, \quad \mathbf{u} = \begin{bmatrix} u_1 \\ u_2 \\ \dots \\ u_r \end{bmatrix}, \quad \mathbf{f} = \begin{bmatrix} f_1 \\ f_2 \\ \dots \\ f_n \end{bmatrix} \quad (\text{II.3})$$

Estando o sistema num ponto de operação $(\mathbf{x}_0, \mathbf{u}_0)$, a perturbação do mesmo por um distúrbio qualquer irá levar a um novo estado cujo comportamento deseja-se analisar:

$$\Delta \mathbf{x} = \mathbf{x} - \mathbf{x}_0 \quad (\text{II.4})$$

A variação do estado do sistema, $\Delta \mathbf{x}$, pode ser estudada linearizando a equação (II.2) em torno do ponto de operação e obtendo:

$$\Delta \dot{\mathbf{x}} = \mathbf{A} \Delta \mathbf{x} + \mathbf{B} \Delta \mathbf{u} \quad (\text{II.5})$$

sendo

$$\mathbf{A} = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \dots & \frac{\partial f_1}{\partial x_n} \\ \dots & \dots & \dots \\ \frac{\partial f_n}{\partial x_1} & \dots & \frac{\partial f_n}{\partial x_n} \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} \frac{\partial f_1}{\partial u_1} & \dots & \frac{\partial f_1}{\partial u_r} \\ \dots & \dots & \dots \\ \frac{\partial f_n}{\partial u_1} & \dots & \frac{\partial f_n}{\partial u_r} \end{bmatrix} \quad (\text{II.6})$$

A matrix \mathbf{A} é o Jacobiano do sistema avaliado no ponto de operação em que o distúrbio acontece. Através do cálculo dos auto-valores de \mathbf{A} pode-se inferir a natureza do movimento do sistema frente à perturbação:

$$\lambda_i = \sigma_i + j\omega_i \quad i = 1, \dots, n \quad (\text{II.7})$$

Se todos os auto-valores do sistema tiverem parte real negativa, isso indica a estabilidade do novo ponto de operação. O comportamento é não-oscilatório se a parte imaginária for igual a zero ou oscilatório decrescente, caso contrário. Caso algum auto-valor tenha parte real positiva, o sistema torna-se instável.

II.2 - Análise de Estabilidade de Tensão

A instabilidade de tensão normalmente acontece em sistemas altamente carregados. Ela é caracterizada pela operação num modo em que a redução da carga causa uma queda de tensão até o ponto de colapso. O conceito pode ser entendido analisando-se um sistema simples como o apresentado na Figura II.1 [Kundur, 1993].

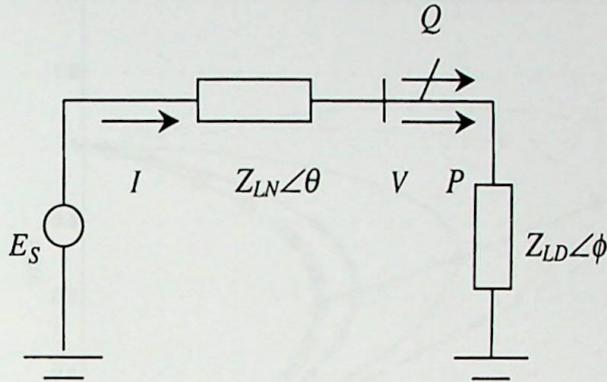


Figure II.1

Pode-se calcular a corrente do circuito por:

$$I = \frac{E_S}{Z_{LN} \angle \theta + Z_{LD} \angle \phi} = \frac{E_S}{\sqrt{F} Z_{LN}} \quad (\text{II.8})$$

sendo

$$F = 1 + \frac{Z_{LD}^2}{Z_{LN}^2} + 2 \frac{Z_{LD}}{Z_{LN}} \cos(\theta - \phi) \quad (\text{II.9})$$

A tensão na carga, V , e a potência entregue, P , podem ser calculadas por:

$$V = I \cdot Z_{LD} = \frac{Z_{LD}}{\sqrt{F} Z_{LN}} E_S \quad (\text{II.10})$$

$$P = V \cdot I \cdot \cos \phi = \frac{Z_{LD}}{F} \left(\frac{E_S}{Z_{LN}} \right)^2 \cos \phi \quad (\text{II.11})$$

Traçando num gráfico a relação entre P e V para vários fatores de potência da carga, obtém-se a curva P - V , como mostrado na Figura II.2. P_{max} é a potência máxima transferida para a carga quando o fator de potência (f.p.) é igual a 1. Esse valor é obtido se $Z_{LD} = Z_{LN}$. A Figura

II.2 mostra as curvas da esquerda para a direita com f.p. = 0,9 atrasado (o), f.p. = 0,95 atrasado (+), f.p. = 1 (-), f.p. = 0,95 adiantado (.-) e f.p. = 0,9 adiantado (--). A linha em triângulo indica o valor crítico de operação do sistema, a partir do qual há uma situação de instabilidade. Nesse caso, mesmo com a diminuição da carga haverá uma diminuição da tensão, o que levará ao colapso de tensão.

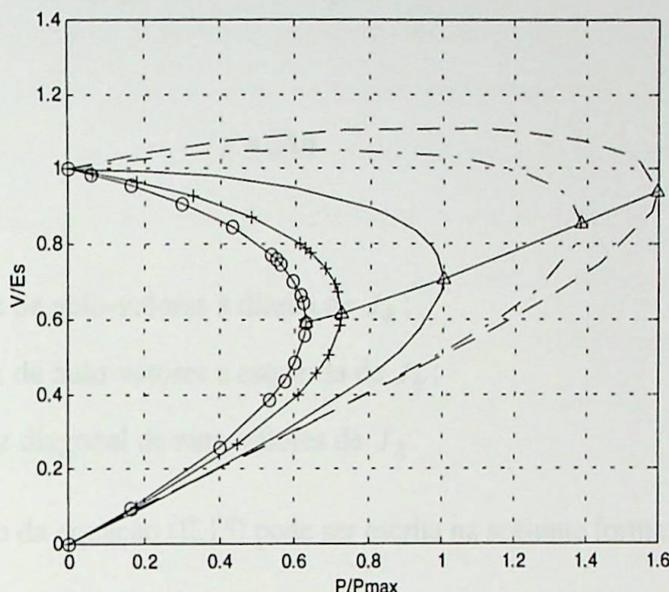


Figura II.2 – Curva P-V

Para analisar o comportamento de um sistema real interligado, considera-se a equação linearizada do fluxo de potência:

$$\begin{bmatrix} \Delta P \\ \Delta Q \end{bmatrix} = \begin{bmatrix} J_{P\theta} & J_{PV} \\ J_{Q\theta} & J_{QV} \end{bmatrix} \begin{bmatrix} \Delta\theta \\ \Delta V \end{bmatrix} \tag{II.12}$$

Os elementos do Jacobiano indicam a sensibilidade entre as variações de potências e as variações de tensões nos barramentos. Considerando $\Delta P = 0$, isso significa manter P constante em cada ponto de operação e considerar somente a relação entre Q e V , que pode ser obtida por:

$$\Delta Q = J_R \Delta V \tag{II.13}$$

sendo o Jacobiano reduzido

$$J_R = [J_{QV} - J_{Q\theta} J_{P\theta}^{-1} J_{PV}] \quad (\text{II.14})$$

Invertendo a equação (II.13), tem-se:

$$\Delta V = J_R^{-1} \Delta Q \quad (\text{II.15})$$

O Jacobiano reduzido, J_R , pode ser decomposto em função dos seus auto-valores e auto-vetores:

$$J_R = \xi \Lambda \eta \quad (\text{II.16})$$

sendo

$\xi \rightarrow$ matriz de auto-valores à direita de J_R ;

$\eta \rightarrow$ matriz de auto-vetores a esquerda de J_R ;

$\Lambda \rightarrow$ matriz diagonal de auto-valores de J_R .

Com isso, a relação da equação (II.15) pode ser escrita na seguinte forma:

$$\Delta V = \xi \Lambda^{-1} \eta \Delta Q = \sum_i \frac{\xi_i \eta_i}{\lambda_i} \Delta Q \quad (\text{II.17})$$

sendo

$\xi_i \rightarrow$ i-ésimo auto-vetor coluna à direita de J_R ;

$\eta_i \rightarrow$ i-ésimo auto-vetor linha à esquerda de J_R ;

$\lambda_i \rightarrow$ auto-valores de J_R .

A magnitude dos auto-valores de J_R permitem uma medida da proximidade do ponto de instabilidade à medida que se aproximam de zero. No entanto, essa é uma proximidade relativa porque na verdade não se sabe quão próximo de zero qualifica-se como uma situação de perigo.

Para determinar a margem de estabilidade, deve-se calcular a distância entre o ponto de operação corrente e o ponto crítico. Pode ser mostrado que o auto-vetor à esquerda referente ao auto-valor que aproxima-se de zero indica qual barramento está mais próximo do colapso de tensão. Com isso, ao se determinar a forma da curva P-V do barramento em

questão (Figura II.2), pode-se determinar a distância até o ponto crítico. Isso pode-se ser feito, por exemplo, através do método da continuação.

II.3 - Análise de Estabilidade Transitória

Na análise de estabilidade transitória interessa estudar o comportamento do sistema em decorrência de distúrbios severos como curto-circuitos e falhas de equipamentos de grande porte. Esses distúrbios causam uma grande variação no ponto de operação do sistema, por isso é preciso utilizar simulações computacionais razoavelmente confiáveis para se determinar o grau de estabilidade (ou instabilidade) do mesmo. Tipicamente, essa análise leva em conta as variações de sincronismo dos geradores do sistema até aproximadamente 10 s após a ocorrência do distúrbio. A análise de segurança transitória possui três etapas básicas que se iniciam em função de um determinado ponto de operação ou caso base:

O primeiro passo é a identificação de contingências que representem risco potencial para o sistema, o que pode levar em conta experiências passadas, a topologia corrente do sistema e o fluxo de potência do caso base. Idealmente todas as contingências possíveis deveriam ser estudadas para cada caso base, o que obviamente é impossível. A seleção de contingências críticas baseia-se em algum método aproximado e rápido para descobrir os casos que podem levar a violações do limite de estabilidade.

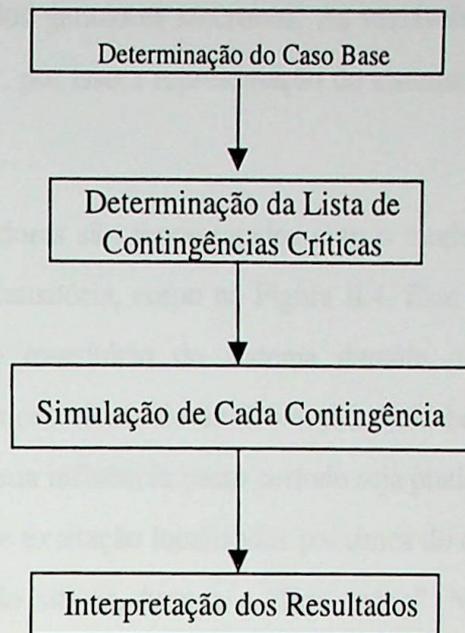


Figura II.3 – Processo de Análise de Estabilidade Transitória

Após a compilação da lista de contingências críticas, cada uma delas deve ser estudada e classificada em termos de estabilidade. Para isso uma quantidade de simulações é realizada, utilizando-se algum programa computacional, tal como a simulação no domínio do tempo, em que as equações diferenciais do sistema são resolvidas passo a passo. Tal simulação deve produzir uma classificação confiável de estável ou instável. Geralmente, calcula-se também um índice que indique o grau de estabilidade (instabilidade) de cada contingência. Com esse índice as contingências podem ser ordenadas de mais estável para menos estável.

O estágio final é a análise dos resultados por um especialista e a determinação de limites de operação, que podem ser tomados como válidos para outras condições de operação que não tenham sido estudadas explicitamente. Podem-se visualizar esses estágios como no diagrama da Figura II.3.

II.3.1 - Representação do Sistema de Potência

Para que sejam calculados os índices de estabilidade transitória face a distúrbio severos, é preciso modelar o sistema apropriadamente em função do objetivo da análise realizada. Os componentes chaves na análise de estabilidade transitória são as equações que descrevem o

comportamento dinâmico dos geradores síncronos. As variáveis de interesse são os ângulos dos rotores de cada gerador, por isso a representação do sistema concentra-se nos fatores que influenciam os ângulos.

Geralmente os geradores são representados com o modelo clássico, com uma f.e.m. interna atrás da reatância transitória, como na Figura II.4. Esse modelo é apropriado para o estudo do comportamento transitório do sistema durante o “first swing”, isto é, até aproximadamente 5 s após a ocorrência do distúrbio. A inércia da turbina e dos reguladores de velocidade fazem com que sua influência nesse período seja praticamente nula. No entanto, os geradores e seus sistemas de excitação localizados próximos do distúrbio exercem um grande efeito no comportamento do sistema durante o “first swing”. Nesse caso, os geradores são representados com um modelo detalhado de dois eixos, direto e em quadratura, como na Figura II.5, além do modelo do sistema de excitação.

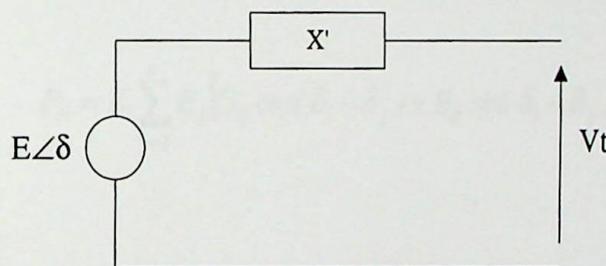


Figura II.4 – Modelo Clássico do Gerador Síncrono

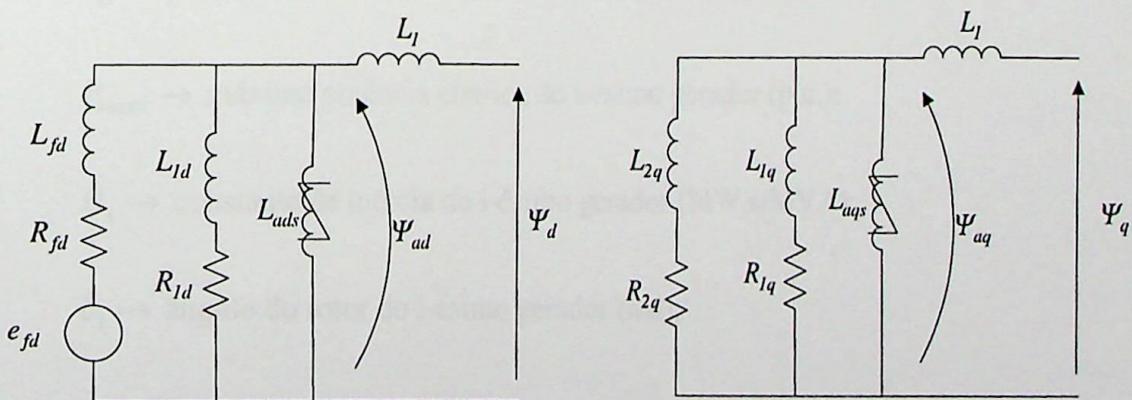


Figura II.5 – Modelo Detalhado do Gerador Síncrono

As cargas são representadas como impedâncias constantes e incorporadas à matriz de admitâncias do sistema.

II.3.2 - Análise de Estabilidade Transitória Através de Simulação no Domínio do Tempo

A simulação no domínio do tempo é executada para resolver um conjunto de equações passo a passo, dado um distúrbio específico. Essas equações representam o comportamento das máquinas síncronas (II.18) através de equações diferenciais não lineares, e o fluxo de potência na rede elétrica (II.19) através de equações algébricas [Kundur, 1993]:

$$\frac{2H_i}{\omega_0} \frac{\partial^2 \delta_i}{\partial t^2} = P_{mi} - P_{maxi} \sin \delta_i = P_{mi} - P_{ei} \quad (\text{II.18})$$

$$P_{ei} = E_i \sum_{j=1}^n E_j [G_{ij} \cos(\delta_i - \delta_j) + B_{ij} \sin(\delta_i - \delta_j)] \quad (\text{II.19})$$

sendo

P_{mi} → potência mecânica fornecida ao i-ésimo gerador (p.u.);

P_{ei} → potência elétrica de saída do i-ésimo gerador (p.u.);

P_{maxi} → máxima potência elétrica do i-ésimo gerador (p.u.);

H_i → constante de inércia do i-ésimo gerador (MW.s/MVA);

δ_i → ângulo do rotor do i-ésimo gerador (rad);

ω_0 → velocidade síncrona (rad/s);

E_i → tensão interna do i-ésimo gerador (p.u.);

$G_{ij}, B_{ij} \rightarrow$ componentes da matriz de admitâncias do sistema (p.u).

Portanto, P_{ei} é uma função não linear de δ e significa que a equação de movimento dos geradores não podem ser resolvidas diretamente. No entanto, o ângulo do rotor no instante t , $\delta(t)$, pode ser determinado integrando-se (II.18) duas vezes e assumindo que as condições iniciais sejam conhecidas.

Para ilustração, considere um sistema simples como o da Figura 3.6, ligado a um barramento infinito, com a ocorrência de uma falta tri-fásica como indicado.

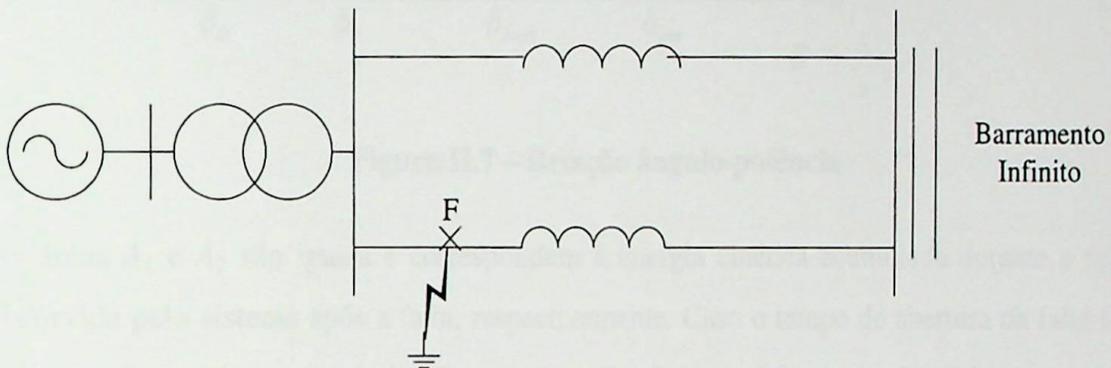


Figura II.6 – Configuração do Sistema

A relação ângulo-potência é mostrada na Figura II.7. A potência mecânica injetada, P_m , é considerada constante. Durante o regime permanente, o ponto de operação, a , corresponde ao o ângulo do rotor, δ_0 . Nesse caso a potência mecânica injetada é igual à potência elétrica fornecida pelo gerador. Na ocorrência da falta, a potência mecânica injetada torna-se maior que a potência elétrica e o ângulo do rotor avança até δ_c , quando a falta é isolada pelo sistema de proteção. Até esse momento, o sistema acumula energia cinética representada pela área AI . Neste momento, o sistema passa a um novo ponto de operação, b , quando a potência elétrica é maior que a potência mecânica. Por isso, o ângulo tende a continuar a avançar cada vez menos até atingir um valor máximo, δ_{max} . Em seguida, o ângulo do rotor volta a recuar até que o sistema atinja um novo ponto de operação pós-falta, c , com ângulo δ_f .

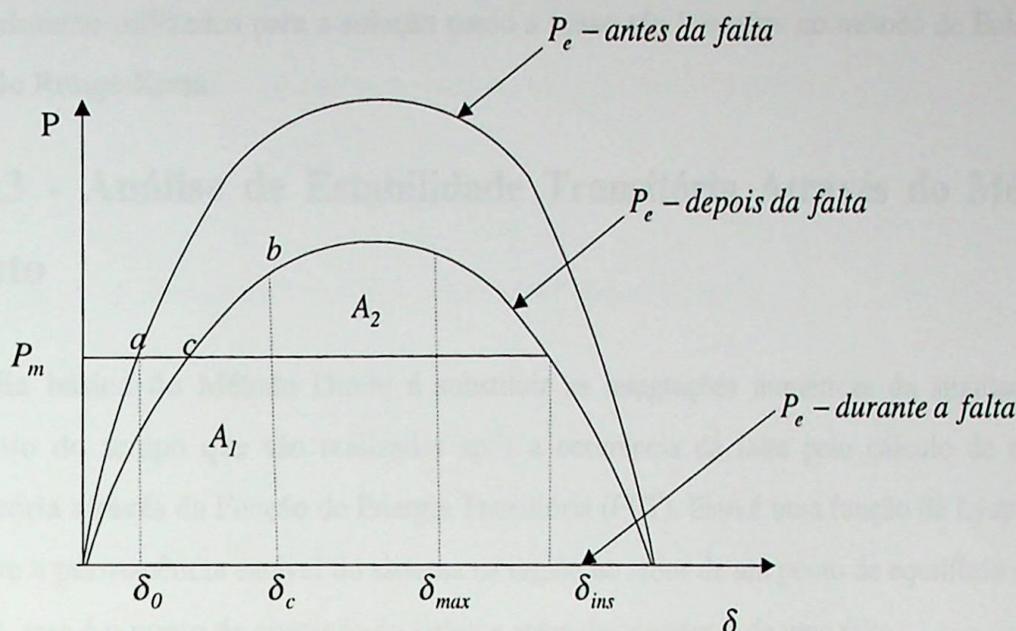


Figura II.7 – Relação ângulo-potência

As áreas A_1 e A_2 são iguais e correspondem à energia cinética acumulada durante a falta e absorvida pelo sistema após a falta, respectivamente. Caso o tempo de abertura da falta fosse muito maior, o sistema poderia não ser capaz de absorver toda a energia cinética acumulada durante a falta, pois o ponto limite até onde $P_e > P_m$ corresponde ao ângulo δ_{ins} . Além desse ângulo a potência mecânica injetada é maior que a potência elétrica gerada. Nesse caso, o gerador perderia sincronismo e a velocidade do rotor aumentaria descontroladamente.

Isso forma a base para o Critério das Áreas Iguais, que permite determinar o máximo aumento do ângulo admissível do δ e, conseqüentemente, a estabilidade do sistema. Portanto, há um limite conhecido por ângulo crítico estável antes do qual a falta deve ser isolada para que o sistema permaneça estável. A esse ângulo crítico corresponde o máximo período de tempo admissível de isolamento da falta, normalmente designado Tempo de Abertura Crítico (TAC). Se a falta for isolada após o TAC, o sistema provavelmente será levado à instabilidade.

Processos de integração numérica permitem calcular o comportamento transitório do sistema, resolvendo (II.18) através de algum algoritmo iterativo. O uso desse algoritmo, mais o Critério das Áreas Iguais permite calcular o Tempo de Abertura Crítico. Os algoritmos



normalmente utilizados para a solução passo a passo são baseados no método de Euler e no método Runge-Kutta.

II.3.3 - Análise de Estabilidade Transitória Através do Método Direto

A idéia básica do Método Direto é substituir as integrações numéricas da simulação no domínio do tempo que são realizadas após a ocorrência da falta pelo cálculo da energia transitória através da Função de Energia Transitória (FET). Essa é uma função de Lyapunov e garante a permanência estável do sistema na região ao redor de um ponto de equilíbrio estável (PEE), que é o ponto de operação do sistema antes da ocorrência de uma falta.

Este é um método que se encontra em desenvolvimento e várias pesquisas têm indicado boas possibilidades de aplicação ao problema de estabilidade transitória [Fouad, 1992]. Há programas comerciais disponíveis no mercado baseados no Método Direto [EPRI, 1993]. No entanto, ainda há uma certa restrição quanto a esse método devido à quantidade de aproximações necessária para tornarem práticos os cálculos da energia transitória.

A análise de estabilidade transitória através do Método Direto segue os seguintes procedimentos básicos:

- Cálculo da energia transitória no momento em que a falta é isolada, V_{ct} ;
- Cálculo da energia crítica correspondente à falta considerada, V_{cr} ;
- Cálculo da margem de energia transitória, $V_{cr} - V_{ct}$.

A energia crítica é definida como a máxima quantidade de energia transitória que o sistema consegue absorver e ainda manter a estabilidade. A energia transitória é a quantidade total de energia injetada no sistema pela falta. A margem de energia transitória define se o sistema continuará estável após a falta, caso seja for positiva, ou se o sistema tornar-se-á instável, caso seja negativa.

A função de energia normalmente é expressa em termos do centro de inércia do conjunto de geradores. Considerando a tensão interna de cada gerador:

$$\bar{E}_i = E_i \angle \delta_i, \quad i = 1, \dots, n$$

A posição do centro de inércia (CI) é definida por [Kundur, 1993]:

$$\delta_{CI} = \frac{1}{H_T} \sum_{i=1}^n H_i \delta_i \quad (\text{II.20})$$

sendo H_T a soma das constantes de inércia de todos os n geradores do sistema.

O movimento do CI é determinado por:

$$2H_T \frac{\partial \Delta \omega_{CI}}{\partial t} = P_{CI} = \sum_{i=1}^n (P'_{mi} - P_{ei}) \quad (\text{II.21})$$

$$\frac{\partial \delta_{CI}}{\partial t} = \Delta \omega_{CI} \omega_0 \quad (\text{II.22})$$

sendo

$$P'_{mi} = P_{mi} - E_i^2 G_{ii} \quad (\text{II.23})$$

$$P_{ei} = \sum_{\substack{j=1 \\ j \neq i}}^n [C_{ij} \text{sen}(\delta_i - \delta_j) + D_{ij}(\delta_i - \delta_j)] \quad (\text{II.24})$$

P_{mi} → potência mecânica fornecida à i -ésima máquina;

$$C_{ij} = E_i E_j B_{ij}$$

$$D_{ij} = E_i E_j G_{ij}$$

ω_0 → velocidade síncrona em (rad/s);

$\Delta \omega_{CI}$ → desvio de velocidade p.u. do CI com relação à velocidade síncrona.

O movimento dos geradores com relação ao CI pode ser expresso definindo-se:

$$\theta_i = \delta_i - \delta_{CI} \quad (rad) \quad (II.25)$$

$$\omega_i = \frac{\dot{\theta}}{\omega_0} = \left(\frac{\dot{\delta}_i}{\omega_0} - \Delta\omega_{CI} \right) \quad (p.u.) \quad (II.26)$$

As equações de movimento da *i*-ésima máquina em função da referência do CI são:

$$2H_i \frac{\partial \omega_i}{\partial t} = P'_{mi} - P_{ei} - \frac{H_i}{H^T} P_{CI} \quad (II.27)$$

$$\dot{\theta}_i = \omega_i \omega_0 \quad (II.28)$$

sendo ω_i a velocidade p.u. da *i*-ésima máquina com relação ao CI.

A função de energia que descreve a energia transitória total do sistema após a ocorrência de um distúrbio é definida como:

$$V = \frac{1}{2} \sum_{i=1}^n J_i \omega_i^2 - \sum_{i=1}^n P'_{mi} (\theta_i - \theta_i^s) - \sum_{i=1}^{n-1} \sum_{j=i+1}^n \left[C_{ij} (\cos \theta_{ij} - \cos \theta_{ij}^s) - \int_{\theta_i^s + \theta_j^s}^{\theta_i + \theta_j} D_{ij} \cos \theta_{ij} d(\theta_i + \theta_j) \right] \quad (II.29)$$

sendo

$\theta_i^s \rightarrow$ ângulo do barramento *i* referente ao ponto de equilíbrio estável após o distúrbio;

$J_i = 2H_i \omega_0 \rightarrow$ momento de inércia p.u. do *i*-ésimo gerador.

O cálculo da energia transitória no momento do isolamento da falta, V_{cl} , é realizado utilizando a função de energia (II.29). Para isso é necessário executar uma simulação no domínio do tempo desde o início do distúrbio até o momento de isolamento da falta para determinar os ângulos e velocidades dos geradores.

A parte mais difícil do Método Direto é o cálculo da energia transitória crítica, V_{cr} , que corresponde a um grande campo de estudo para a melhoria desse método.

A capacidade de absorção de energia por parte do sistema depende do “formato da superfície” da função de energia, e da trajetória que o ponto de operação realiza durante o período de ocorrência do distúrbio. Certas trajetórias levam o sistema a uma capacidade maior de absorção da energia transitória do que outras.

Definindo-se $x = \begin{bmatrix} \theta \\ \omega \end{bmatrix}$, diz-se que um PEE, x_e , localiza-se numa região da superfície

de energia a partir do qual a função de energia é sempre positiva, $\dot{V} > 0$. Nesse caso, há uma bacia de atração, $\partial A(x_e)$, em volta de x_e tal que a trajetória de qualquer ponto iniciando dentro dessa região irá convergir para x_e . Essa idéia é ilustrada de maneira simplificada na Figura II.8.

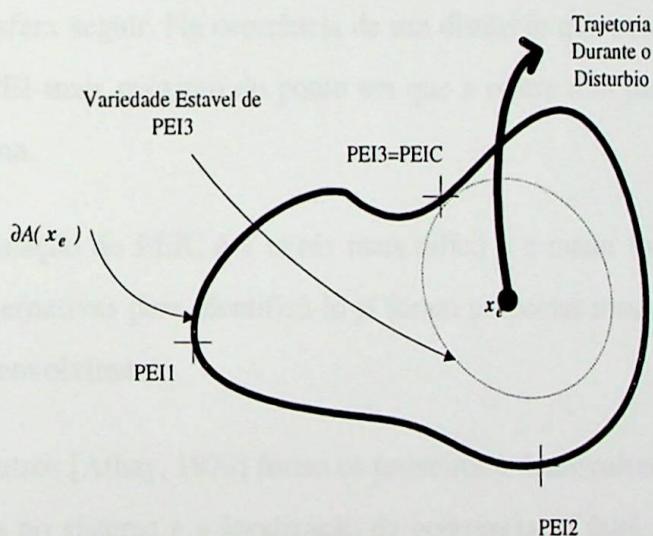


Figura II.8 – Conceito de Ponto de Equilíbrio Instável de Controle

Em volta do PEE há vários pontos de equilíbrio instável (PEI) que irão determinar o valor da energia crítica. Interessa localizar os PEI porque a partir dos mesmos é possível identificar $\partial A(x_e)$, que na verdade pode ser definida como a união das variedades estáveis dos PEI localizados em $\partial A(x_e)$ [Fouad, 1992]. A variedade estável de um PEI corresponde à região

em torno do mesmo e que converge para x_e . Portanto, dependendo da trajetória do sistema durante a ocorrência do distúrbio, há um PEI específico que determina a estabilidade do sistema. Este é chamado PEI de controle (PEIC) e a energia transitória correspondente a esse ponto é tomada como a energia crítica do sistema. Uma vez que o PEIC tenha sido identificado, a energia crítica é calculada a partir de (II.29) e utilizada para classificar a estabilidade do sistema.

Uma analogia pode ser feita considerando o PEE como uma pequena esfera colocada no fundo de uma bacia de formato convexo. Quando ocorre um distúrbio, a esfera é deslocada de sua posição inicial por uma certa quantidade de energia cinética, e rola pela parede da bacia percorrendo uma determinada trajetória. A tendência é que ela oscile em torno do fundo até que volte novamente ao ponto inicial de equilíbrio. Caso o distúrbio causado à esfera seja muito grande, ela poderá rolar para fora da bacia, caracterizando o movimento instável. Nesse caso os PEI estão localizados na borda da bacia. Se forem consideradas alturas variáveis ao longo da borda, diferentes valores de energia crítica poderão ser encontrados, dependendo da trajetória que a esfera seguir. Na ocorrência de um distúrbio que cause instabilidade, o PEIC corresponde ao PEI mais próximo do ponto em que a esfera rola para fora da bacia, dada a trajetória da mesma.

A determinação do PEIC é a tarefa mais difícil e a maior fonte de erros do Método Direto. Várias alternativas para identificá-lo já foram propostas mas este ainda é um tema de pesquisas em desenvolvimento.

Athay e outros [Athay, 1979] foram os primeiros a desenvolver um método prático que considera a perda no sistema e a localização da ocorrência da falta. Fouad e outros [Fouad, 1981] melhoraram o trabalho de Athay e apresentaram os resultados do chamado método dos modos de distúrbio (MDD=MOD) num paper de duas partes em 1981. Esse foi o primeiro paper a mostrar resultados num sistema real e mostrou a viabilidade do Método Direto para sistemas de tamanho real. O MDD ainda permanece como um dos mais aceitos métodos diretos. Em 1987, Chiang, [Chiang, 1987], [Chiang, 1988], publicou a primeira justificativa teórica para os métodos diretos. Mais tarde, o mesmo publicou uma série de artigos, [Chiang, 1989], [Chiang, 1994], sobre o que é conhecido como o Método do Ponto de Saída ou Borda



de Controle do PEI (BCPEI=BCU). O MDD e o BCPEI são bastante similares e diferem apenas no modo de encontrar o PEIC.

Na prática, há alguns problemas não resolvidos, como: a função de Lyapunov, cálculo do valor crítico, há necessidade da integração no domínio do tempo para obter os valores dos ângulos e das velocidades dos geradores no momento da ocorrência da falta, o critério é apenas uma garantia suficiente, pois para pontos de operação fora da região de estabilidade estimada, nada pode ser concluído.

Anexo III

Tabelas de Resultados do Sistema Priba

TABELA III.1
BUSCA SEQUENCIAL 1

<i>d</i>	150		100		50	
Classificador	MVS-G	MVS-P	MVS-G	MVS-P	MVS-G	MVS-P
σ^2	0.12	-	0.0001	-	0.6	-
<i>p</i>	-	2	-	4	-	5
Taxa de Detecção	0.99	0.96	0.98	0.95	0.76	0.76
Dispensas Falsas	0.08	0.05	0.09	0.05	0.06	0.04
Taxa de Erro	0.09	0.08	0.11	0.10	0.27	0.26
Tempo de Treinamento	77 s	106 s	39 s	98 s	18 s	67 s
No. de VSs	915	180	831	225	528	342

TABELA III.2
BUSCA SEQUENCIAL 2

<i>d</i>	20		10	
Classificador	MVS-G	MVS-P	MVS-G	MVS-P
σ^2	1.72	-	0.0001	-
<i>p</i>	-	6	-	5
Taxa de Detecção	0.5511	0.6044	0.66	0.52
Dispensas Falsas	0.04	0.048	0.06	0.02
Taxa de Erro	0.448	0.407	0.37	0.45
Tempo de Treinamento	17 s	37 s	13 s	19 s
No. de VSs	659	637	954	698

TABELA III.3
EXTRAÇÃO DE COMPONENTES PRINCIPAIS 1

<i>d</i>	150		100		50	
Classificador	MVS-G	MVS-P	MVS-G	MVS-P	MVS-G	MVS-P
σ^2	0.0001	-	0.0001	-	0.0001	-
<i>p</i>	-	6	-	5	-	5
Taxa de Detecção	1	0	1	0	1	0
Dispensas Falsas	0.093	0	0.093	0	0.093	0
Taxa de Erro	0.093	0.91	0.093	0.91	0.093	0.91
Tempo de Treinamento	78 s	316 s	41 s	291 s	28 s	94 s
No. de VSs	994	296	932	308	955	711

TABELA III.4
EXTRAÇÃO DE COMPONENTES PRINCIPAIS 2

<i>d</i>	20		10	
Classificador	MVS-G	MVS-P	MVS-G	MVS-P
σ^2	0.0001	-	0.0001	-
<i>p</i>	-	6	-	3
Taxa de Detecção	1	0	1	1
Dispensas Falsas	0.093	0	0.093	0.093
Taxa de Erro	0.093	0.907	0.093	0.093
Tempo de Treinamento	22 s	57 s	17 s	112 s
No. de VSs	993	566	925	629

TABELA III.5
ALGORITMOS GENÉTICOS 1

<i>d</i>	150		100		50	
	MVS-G	MVS-P	MVS-G	MVS-P	MVS-G	MVS-P
σ^2	0.6	-	2.04	-	0.0001	-
<i>p</i>	-	4	-	5	-	1
Taxa de Detecção	0.9956	0.8844	0.9067	0.8578	0.6578	0.6578
Dispensas Falsas	0.089	0.032	0.052	0.048	0.056	0.032
Taxa de Erro	0.093	0.137	0.137	0.177	0.367	0.343
Tempo de Treinamento	46 s	70 s	31 s	65 s	24 s	105 s
No. de VSs	906	219	362	274	959	688

TABELA III.6
ALGORITMOS GENÉTICOS 2

<i>d</i>	20		10	
	MVS-G	MVS-P	MVS-G	MVS-P
σ^2	0.12	-	0.0001	-
<i>p</i>	-	3	-	4
Taxa de Detecção	0.7511	0.7111	0.72	0.6844
Dispensas Falsas	0.036	0.024	0.069	0.028
Taxa de Erro	0.262	0.286	0.323	0.315
Tempo de Treinamento	13 s	53 s	16 s	20 s
No. de VSs	503	489	479	667