

Ferramenta para auxílio na previsão de séries
temporais com intervalos de confiança usando
Máquinas de Vetor de Suporte

Trabalho de Conclusão de Curso

Engenharia da Computação

George Gomes Cabral
Orientador: Prof. Adriano Lorena Inácio de Oliveira

Recife, 18 de novembro de 2005



Ferramenta para auxílio na previsão de séries temporais com intervalos de confiança usando Máquinas de Vetor de Suporte

Trabalho de Conclusão de Curso

Engenharia da Computação

Este Projeto é apresentado como requisito parcial para obtenção do diploma de Bacharel em Engenharia da Computação pela Escola Politécnica de Pernambuco – Universidade de Pernambuco.

George Gomes Cabral
Orientador: Prof. Adriano Lorena Inácio de Oliveira

Recife, 18 de novembro de 2005



George Gomes Cabral

Ferramenta para auxílio na previsão de séries
temporais com intervalos de confiança usando
Máquinas de Vetor de Suporte

Resumo

Uma série temporal consiste de uma coleção de valores coletados seqüencialmente no tempo. A previsão de valores futuros de uma série temporal é um importante problema em várias áreas, incluindo economia, planejamento de produção, previsão de vendas e controle de estoque.

Este trabalho objetiva o desenvolvimento de uma ferramenta auxiliar para previsão de séries temporais usando máquinas de vetor de suporte (Support Vector Machines – SVMs). O escopo de funcionalidades da ferramenta proposta vai desde o pré-processamento das séries até o processamento inverso dos valores previstos. Para a previsão das séries é usada a biblioteca LibSVM, responsável pelo treinamento e teste das séries.

Nossa ferramenta possui um grande atrativo que consiste na facilidade no tratamento e amostragem gráfica dos dados. Séries temporais são, geralmente, disponibilizadas de forma inadequada para o uso dos métodos disponíveis de previsão, no caso SVMs. O tratamento dos dados envolve uma série de análises visando a obtenção de um processamento eficaz a ponto de tornar a série estacionária.

Três tipos de treinamentos são disponibilizados ao usuário:

- Treinamento com parâmetros personalizados;
- Treinamento com parâmetros otimizados e;
- Treinamento com melhores erros obtidos pela execução de um script na linguagem python.

A previsão de séries temporais apresenta um grande problema que consiste na impossibilidade da verificação antecipada da validade dos valores previstos pois, na prática esses valores não são conhecidos, portanto, é muito importante também prover um indicador de confiabilidade na previsão. Esse indicador de confiabilidade é dado pela implementação de intervalos de confiança robustos nos gráficos de teste.

Abstract

Time series consist of a collection of sequentially acquired values over time. The prediction of future values of a time series is an important problem in many areas, including economics, production planning, sales prediction and stock control.

This work aimed to develop a tool for time series forecasting using Support Vector Machines (SVMs). The proposed tool comprises all stages of time series forecasting, including pre-processing, normalization of the series and inverse processing of the predicted values. In our tool, time series prediction is carried out using the LibSVM library, which is used for training and testing.

Our tool makes it easy for users tasks such as pre-processing, training, testing and graphical data sampling. Generally, time series are arranged in a available prediction methods inadequate form, SVMs in the case. The data treatment involves consecutives analyses aiming at efficient processing capable to become the series stationary.

Three training types are available to the user:

- Training with personalized parameters;
- Training with optimized parameters; and
- Training with better EMQs gotten by the execution of one script in the python language.

Time series prediction presents a great problem that consists of the impossibility of the anticipated verification of the predicted values; its very important too to provide an indication of trustworthiness in the forecast. This reliability is given by the robust confidence interval implementation on the use of the graphics of test.

Sumário

Índice de Figuras	5
Índice de Tabelas	7
Agradecimentos	8
1 Introdução	9
2 Séries Temporais	12
2.1 Objetivos da análise de séries temporais	12
2.2 Tratamento dos dados	13
2.3 Características comuns a séries temporais	14
2.3.1 Sazonalidade	15
2.3.2 Tendência	16
2.3.3 Outras variações irregulares	17
3 Métodos para pré-processamento e previsão de séries temporais	19
3.1 Pré-processamento das séries	20
3.1.1 Auto-correlação da série	21
3.1.2 Eliminação da tendência	24
3.1.3 Eliminação da sazonalidade	25
3.1.4 Normalização	26
3.2 Previsão de séries temporais	26
3.2.1 Regressão Linear	27
3.2.2 Redes Neurais Artificiais	28
3.2.3 Redes MLP	29
3.2.4 Redes de Elman e Jordan	31
3.2.5 SVMs (Máquinas de vetor de suporte)	32
3.3 Previsões de séries temporais com intervalos de confiança robustos	36
4 A Ferramenta Desenvolvida	38

4.1 Características técnicas	39
4.2 Funcionalidades	40
4.2.1 Menu Arquivo	41
4.2.2 Menu Editar	42
4.2.3 Menu Gráficos	44
4.2.4 Menu Treinamento/Teste	46
5 Experimentos e Resultados Obtidos	50
5.1 Metodologia de treinamento para SVM e redes Neurais	50
5.2 Experimentos com a série CEMENT	51
5.2.1 Pré-Processamento da série	52
5.2.2 Resultados Obtidos para a série CEMENT	54
5.2.3 Gráficos das séries Previstas	56
5.3 Experimentos com a série BOVESPA	58
5.3.1 Pré-Processamento da série	59
5.3.2 Resultados Obtidos para a série Bovespa	61
5.3.3 Gráficos das séries previstas	63
6 Conclusões e trabalhos futuros	65
6.1 Conclusões	65
6.2 Trabalhos Futuros	67
Referências Bibliográficas	69

Índice de Figuras

Figura 1 - Série de índices mensais da bovespa no intervalo de julho de 1983 a abril de 2005	9
Figura 2 - Série estacionária.....	14
Figura 3- Série não estacionária quanto ao nível (mudança de nível)	15
Figura 4 - Série Sazonal	16
Figura 5 - Série com tendência.....	17
Figura 6 - Série com Outlier.....	18
Figura 7 - Padrões mistos	18
Figura 8 - resultado de previsão de uma série estacionária.....	20
Figura 9 - resultado de previsão de uma série não estacionária.	21
Figura 10 - Correlograma de uma série estacionária.....	22
Figura 11 - Correlograma de uma série com tendência	23
Figura 12 - Correlograma de uma série sazonal.....	23
Figura 13 - Correlograma de uma série com presença de sazonalidade e tendência	24
Figura 14 - série duplamente diferenciada	25
Figura 15 - Série com características sazonais.....	25
Figura 16 - Série sazonalmente diferenciada	26
Figura 17 - Função de regressão linear unidimensional.....	27
Figura 18 – Topologia de um perceptron simples.....	29
Figura 19 - Rede MLP típica com uma camada intermediária	30
Figura 20 - Topologia Elman	32
Figura 21 - Topologia Jordan	32
Figura 22- Mapeamento de espaço de entradas no espaço de características	34
Figura 23 - Separação otimizada entre classes.....	35
Figura 24 - EMQ X Generalização	36
Figura 25 – Previsão de séries temporais com intervalos de confiança robustos	37
Figura 26 – Screenshot da ferramenta.....	39
Figura 27 - Diagrama de classes do projeto	40
Figura 28 - Diagrama de casos de uso.....	41
Figura 29 - Painel de pré-processamento	43
Figura 30 - Correlograma gerado pela ferramenta.....	43
Figura 31 - Série original	45
Figura 32- Teste sem intervalos de confiança robustos	45
Figura 33 - Teste com intervalos de confiança robustos.....	46
Figura 34 - Painel de controle para realização de treino e teste das séries	48
Figura 35 - Série CEMENT original.....	52
Figura 36 - Correlograma da série CEMENT Original.....	53
Figura 37 - Correlograma final da série CEMENT.....	53
Figura 38 - Representação gráfica do melhor teste MLP - CEMENT	57

Figura 39 - Representação gráfica do melhor teste Elman - CEMENT	57
Figura 40 - Representação gráfica do melhor teste SVM - CEMENT.....	58
Figura 41 - Correlograma inicial - Bovespa.....	60
Figura 42 - Correlograma final – Bovespa.....	60
Figura 43 - Representação gráfica do melhor teste MLP - Bovespa.....	63
Figura 44 - Representação gráfica do melhor teste Elman - Bovespa	64
Figura 45 - Representação gráfica de melhor teste SVM utilizando intervalos de confiança robustos com uma confiança de 95% - Bovespa.....	64

Índice de Tabelas

Tabela 1 - Entradas e saídas desejadas para o treinamento com MLP e janela 4	31
Tabela 2 - Distribuição dos dados para a série CEMENT	54
Tabela 3 - Resultado dos treinamentos realizados para topologia MLP e para a série CEMENT	55
Tabela 4 - Resultado dos treinamentos realizados para topologia Elman e para a série CEMENT	55
Tabela 5 - Resultado dos treinamentos realizados utilizando SVMs para a série CEMENT	56
Tabela 6 - Distribuição dos dados para a série Bovespa	61
Tabela 7 Resultado dos treinamentos realizados para topologia MLP e para a série BOVESPA.....	62
Tabela 8 - Resultado dos treinamentos realizados para topologia Elman e para a série BOVESPA.....	62
Tabela 9 - Resultado dos treinamentos realizados utilizando SVMs para a série BOVESPA	63

Agradecimentos

Aos meus pais, José Gomes dos Santos e Juraci Gomes Cabral e irmãs Érika Gomes Cabral e Glenda Gomes Cabral pelo suporte incondicional dado em todos os momentos em que precisei.

A todos meus amigos e família que sempre estiveram presente ao meu lado me incentivando na conquista dessa vitória.

Ao meu orientador, Prof. Dr. Adriano Lorena Inácio de Oliveira, pela ajuda dada ao longo de todo o curso, seja no papel de professor, orientador ou conselheiro.

Aos professores Carlos Alexandre e Fernando Buarque pela excepcional atenção com os alunos e empenho na resolução de seus problemas, sejam eles acadêmicos ou não.

Ao colegiado do Departamento de Sistemas Computacionais, pelo ótimo papel desempenhado na estruturação e seguimento do curso ao longo desses poucos, porém notáveis, seis anos.

A todos os colegas que de alguma forma me ajudaram para a conclusão desse trabalho, em especial a Adélia Carolina que me ouviu atentamente durante a fase de escrita desse documento.

Aos componentes da minha banda de forró Ronaldo Cysneiros, Renato Augusto, Kelly Aguiar, Valmir Macário, Clóvis Santos e Éricles Medeiros por tornarem tão agradáveis esses cinco anos de convivência.

Capítulo 1

Introdução

Previsão de séries temporais[1] é uma área de importância crescente na solução de problemas como previsão de índice pluviométrico de uma região, previsão de valores em bolsas de valores, previsão de vendas de uma empresa para determinado período, etc. A partir de dados históricos é possível, de certa forma, se obter uma estimativa de dados futuros. No caso de uma série temporal contendo índices da Bolsa de Valores de São Paulo - BOVESPA, ver figura 1, por exemplo, se o valor de uma ação começa a subir, a ponto de atingir valores irreais, a evolução temporal da mesma pode indicar características como tendências ou variações periódicas. Para aqueles interessados em planejamento dentro do mercado de ações, ou para os que procuram conhecer ou melhorar sua carteira de investimentos, esses dados podem ser de crucial importância para criação de um cenário e no auxílio na tomada de decisões.

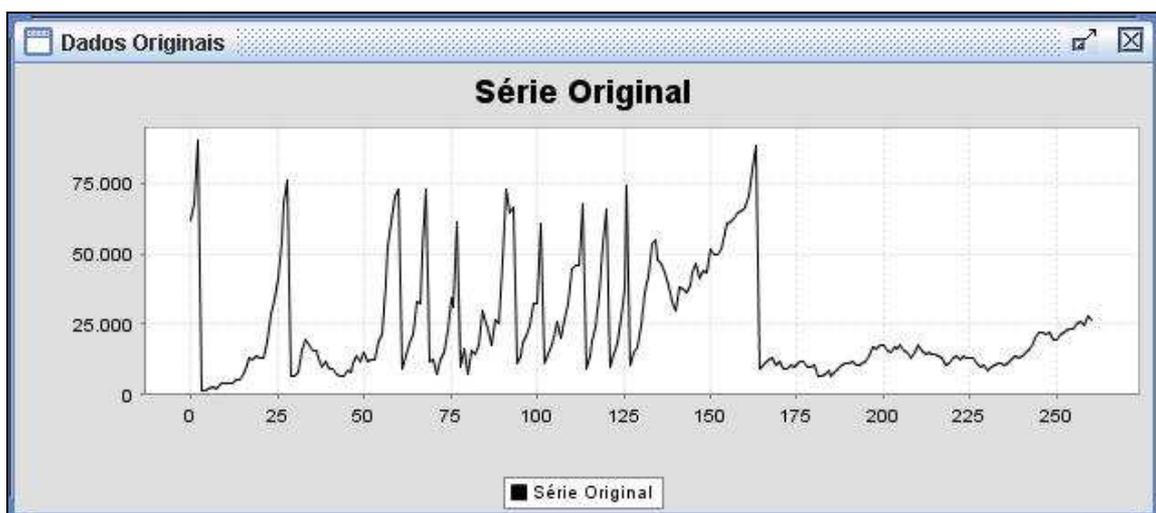


Figura 1 - Série de índices mensais da bovespa no intervalo de julho de 1983 a abril de 2005

A série temporal em seu estado original se apresenta, geralmente, com características como tendência e sazonalidade incorporadas. As ferramentas hoje disponíveis para previsão de séries temporais utilizam métodos para previsão de séries como redes neurais artificiais[2] e SVMs (Support Vector Machines – Máquinas de Vetor de Suporte)[4,5] que necessitam, para melhoria de desempenho, de um pré-processamento nos dados a serem utilizados.

O pré-processamento visa a atenuação, ou até mesmo eliminação, da tendência e sazonalidade assim como a normalização dos dados. Na eliminação da tendência e da sazonalidade é aplicado o processo de diferenciação que tem como objetivo a obtenção de uma série estacionária. Para a realização do pré-processamento é necessária uma análise, de preferência gráfica (pois transmite de forma clara, ao usuário, confiança de que o pré-processamento está sendo realizado de maneira eficiente, ao contrário da análise apenas dos dados numéricos que pode ser, de certa forma, mais complexa para um usuário desabituação), da série. De acordo com esta análise o usuário estará apto a identificar características indesejáveis na série e tratá-las posteriormente.

O teste com os valores previstos é uma tarefa trivial, tendo-se em mãos um simulador adequado, porém, os dados nesse estágio encontram-se totalmente diferentes dos dados originais, consequência dos pré-processamentos realizados sobre a série. Para a amostragem dos dados, se torna necessário um processamento inverso que é uma tarefa bastante complexa, pois requer o conhecimento de todo o histórico de pré-processamentos efetuados e lida com uma grande quantidade de dados. Os arquivos gerados, disponibilizados para o treinamento e teste, são compostos de vários padrões que consistem em valores consecutivos da série e a saída desejada, esses valores consecutivos (como, por exemplo, janeiro fevereiro março abril) consistem na janela de tempo responsável pela previsão do próximo elemento da série. Além da perda de dados na inclusão de janelas de tempo, existe a perda de dados também nas diferenciações que por sua vez também modificam os valores para treinamento, além da normalização.

Previsão de séries temporais apresenta um grande problema que consiste na impossibilidade da verificação antecipada da validade dos valores previstos; é muito importante também prover um indicador de confiabilidade na previsão. Esse indicador de confiabilidade é dado pela implementação de intervalos de confiança robustos [3] nos gráficos de teste.

Máquinas de vetor de suporte [4,5], SVMs, são uma família relativamente nova de métodos de aprendizagem que utilizam várias funções de kernel como funções de base radial,

funções polinomiais e sigmoid. É desconhecida a existência de uma ferramenta de uso simples e de boa interface para previsão de séries temporais através de SVMs.

A necessidade de uma ferramenta capaz de lidar com as dificuldades apresentadas anteriormente, motiva a execução desse projeto. Nosso projeto tem como objetivo o desenvolvimento de uma ferramenta de uso simples e de interface amigável que seja capaz de fornecer aos usuários, com um certo nível de familiarização com o assunto, meios de assegurar o correto andamento do processo de treinamento de máquinas de vetor de suporte para previsão de séries temporais desde o pré-processamento dos dados até aos testes. A ferramenta fará o uso da biblioteca para treinamento e teste com máquinas vetores de suporte LibSVM[6,7].

Este trabalho de conclusão de curso conta com um total de 6 capítulos. Neste capítulo o contexto do trabalho é apresentado e breves introduções dos próximos capítulos são disponibilizadas.

O capítulo 2 trata de séries temporais. Objetivos da análise de séries temporais são levantados assim como o tratamento dos dados também é visto e discutido. Nesse capítulo, o uso de séries temporais é exemplificado, assim como a descrição de características comuns a séries, como tendência e sazonalidade serão detalhadas.

O capítulo 3 introduz conceitos pertinentes à realização de previsões de séries temporais. O conceito de regressão linear é apresentado e uma breve explanação sobre métodos de previsão de séries temporais é dada. O pré-processamento das séries é descrito e a motivação para a realização do mesmo é apresentada.

No capítulo 4 a ferramenta proposta é descrita com todas suas funcionalidades. Características técnicas são apresentadas e o conceito de intervalos de confiança robustos é explicado. Todas as funcionalidades são explicadas e imagens da ferramenta são mostradas para uma melhor idéia do trabalho desenvolvido.

No capítulo 5 são realizados testes que visam a comparação dos resultados das previsões geradas pela ferramenta e duas topologias distintas de redes neurais, MLP[22] e Elman[17]. Os estágios, desde pré-processamento das séries aos testes, são relatados e imagens geradas são disponibilizadas para uma ilustração de como foram realizados os experimentos.

Finalmente, no capítulo 6 é apresentada a conclusão, bem como várias propostas para trabalhos futuros.

Capítulo 2

Séries Temporais

Em muitas áreas do conhecimento as observações de interesse são obtidas em instantes sucessivos no tempo, por exemplo a cada hora, durante 24 horas, ou então registradas por algum equipamento de forma contínua, como o traçado de um eletrocardiograma.

Chamamos de série temporal um conjunto de observações ordenadas no tempo. São exemplos de séries temporais[8]:

- (i) os valores diários do preço de ações de uma empresa na Bovespa;
- (ii) os valores de medições mensais da temperatura da cidade do Recife;
- (iii) as quantidades anuais de chuva na cidade de São Paulo;
- (iv) o registro do eletrocardiograma de uma pessoa;
- (v) o registro do movimento da crosta terrestre, obtido através de um sismógrafo.

Séries temporais podem ser discretas (i)-(iii) ou contínuas (iv)-(v). Mesmo que uma série seja obtida continuamente durante um intervalo de tempo de amplitude T , será necessário transformá-la em uma série discreta, através da amostragem em intervalos de tempo equispaçados.

2.1 Objetivos da análise de séries temporais

Quando se analisa uma série temporal $Z(t_1), \dots, Z(t_n)$, observada nos instantes t_1, \dots, t_n , os objetivos são os mais variados possíveis. Os objetivos seguintes são bastante relevantes [9]:

- a) Modelagem do fenômeno sob consideração;
- b) Obtenção de conclusões em termos estatísticos;

- c) Avaliação da adequação do modelo em termos de previsão;
- d) Sumário dos dados (verificação da existência de tendências, ciclos, construção de histogramas, etc.);
- e) Investigação do mecanismo gerador da série;
- f) Estabelecimento da causalidade (por exemplo, entre série de vendas e série de propagandas relativas ao produto);
- g) Classificação (a partir do ECG de um indivíduo pode se obter informações sobre a existência de algum problema cardíaco, por exemplo).
- h) Controle (uma empresa pode gerar medidas temporais de qualidade, por exemplo, e através da leitura dessas séries, através de gráficos, tomar decisões em relação ao controle de processos).

2.2 Tratamento dos dados

Dependendo dos objetivos da análise, vários problemas com as observações podem ocorrer e medidas devem ser tomadas para evitá-los ou, pelo menos, amenizá-los[9].

- *Planejamento*; sempre que possível, a obtenção das observações deve ser planejada previamente. O número de observações e o intervalo de amostragem devem ser determinados segundo os objetivos do investigador.
- *Estacionariedade*; a maioria das técnicas de previsão de séries temporais são aplicadas à séries estacionárias, por essa razão análise de séries temporais requerem uma conversão de séries não estacionárias em séries estacionárias.
- *Transformações*; uma das razões para transformar os dados originais é a presença de não estacionariedade. A transformação mais comum consiste em tomar diferenças consecutivas da série original, até obter-se uma série estacionária[8], diferenciação (equação 1).

- *Observações perdidas e irregulares;* observações podem ser perdidas, por exemplo, pela quebra do instrumento de medida durante um período de tempo, no caso do eletrocardiograma, por exemplo. Se o número de observações perdidas é pequeno e não concentrado em um intervalo, podemos usar interpolação sem grandes problemas. Caso contrário, técnicas de análise específica devem ser usadas.
- *Registros curtos;* às vezes é inevitável que se tenham poucas observações para análise. Nestes casos, todas conclusões devem ser tomadas com cautela e técnicas apropriadas devem ser utilizadas, quando elas existirem.

Equação 1, onde Z simboliza o vetor a ser diferenciado e t o posicionamento de cada amostra no tempo.

$$\Delta Z(t) = Z(t) - Z(t-1) \quad \text{(Equação 1)}$$

2.3 Características comuns a séries temporais

Uma série é dita estacionária se não existe mudança sistemática na média (sem tendência), se não existe mudança sistemática na variância e se variações periódicas não forem presentes. Exemplo de série estacionária pode ser visto na figura 2.

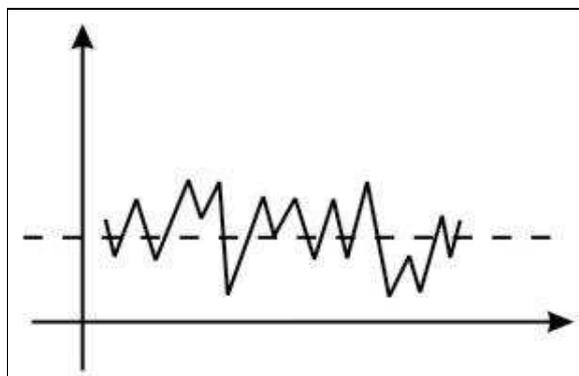


Figura 2 - Série estacionária

Uma das suposições mais freqüentes, feita a respeito de uma série temporal, é que ela seja estacionária, ou seja, que ela se desenvolve no tempo aleatoriamente ao redor de uma média constante, refletindo alguma forma de equilíbrio estável. Todavia, a maior parte das séries encontradas na prática apresentam alguma forma de não estacionariedade, como por exemplo na figura 3.

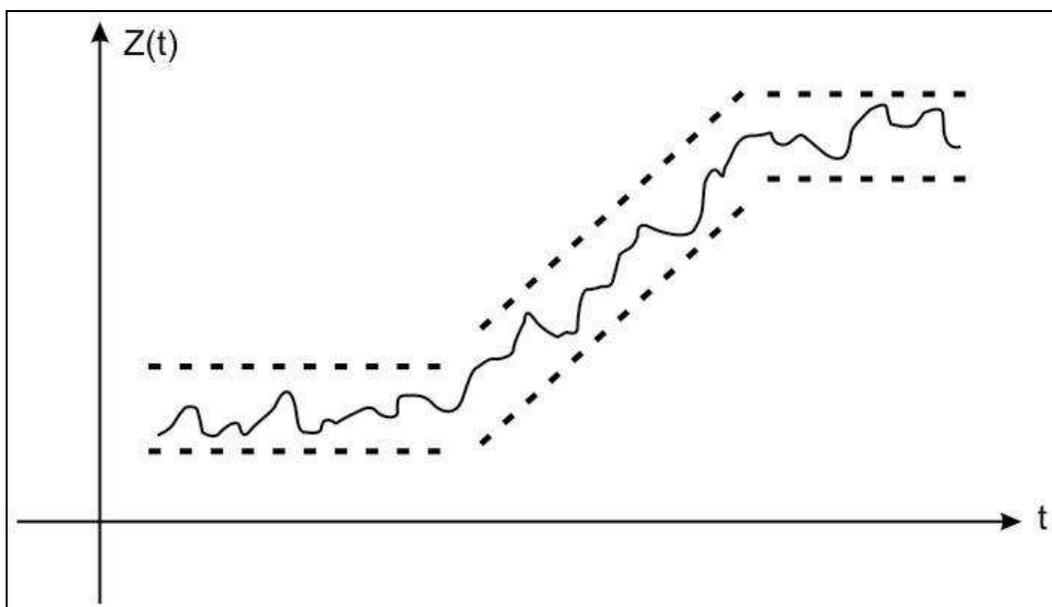


Figura 3- Série não estacionária quanto ao nível (mudança de nível)

2.3.1 Sazonalidade

Muitos exemplos de séries temporais como, séries de vendas e leituras de temperaturas, exibem variações que são periódicas. Por exemplo, a venda de protetor solar cresce consideravelmente no nordeste durante os meses de janeiro e fevereiro. Esse tipo de variação é de fácil entendimento e pode ser facilmente removida para a obtenção de uma série livre de sazonalidade. A figura 4 mostra um exemplo de série sazonal.

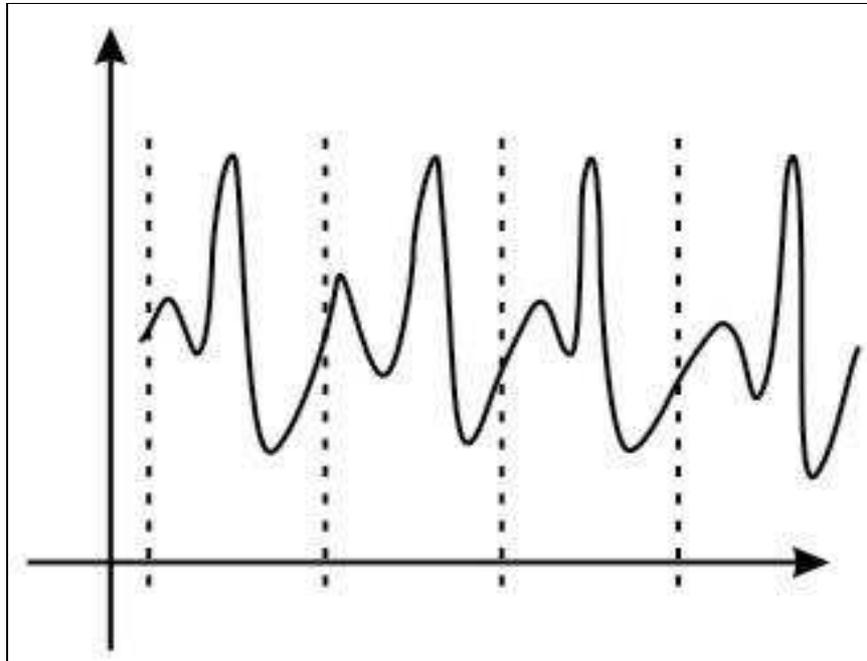


Figura 4 - Série Sazonal

2.3.2 Tendência

Tendência, no contexto de séries temporais, pode ser definida como uma mudança constante de nível. Uma dificuldade na definição de tendência consiste em encontrar um período ideal que prove existência de uma tendência e não outro efeito como sazonalidade[10]. Por exemplo, variações climáticas, às vezes, exibem variações cíclicas durante um longo período, como 50 anos. Se para essas variações for disponibilizada uma série que a represente por um período de apenas 20 anos, essa oscilação poderia parecer uma tendência, porém se várias centenas de anos forem disponíveis, a oscilação seria visível. A figura 5 mostra um exemplo de tendência.

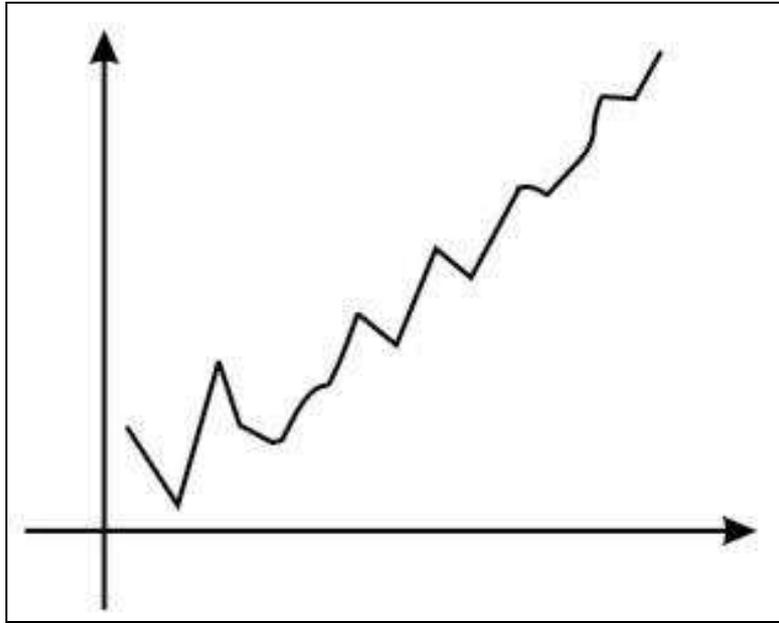


Figura 5 - Série com tendência

2.3.3 Outras variações irregulares

- Outliers – pontos da série situados significativamente fora da média (figura 6);
- Mudança de nível (figura 3);
- Padrões mistos – uma série pode conter várias características, como tendência e sazonalidade, simultaneamente (figura 7).

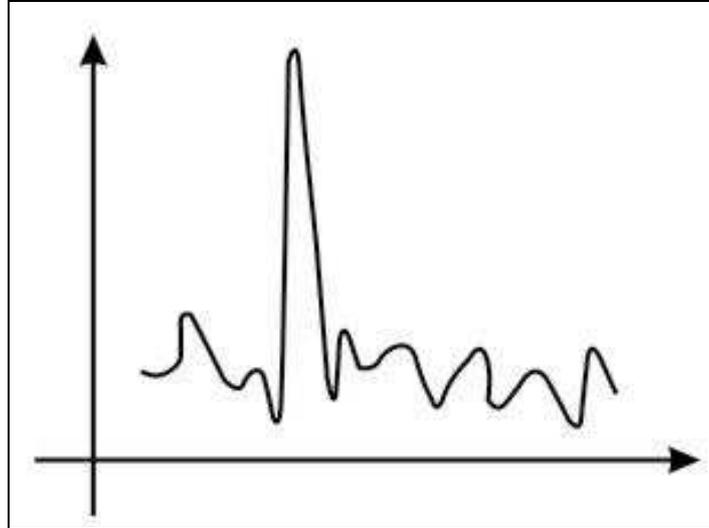


Figura 6 - Série com Outlier

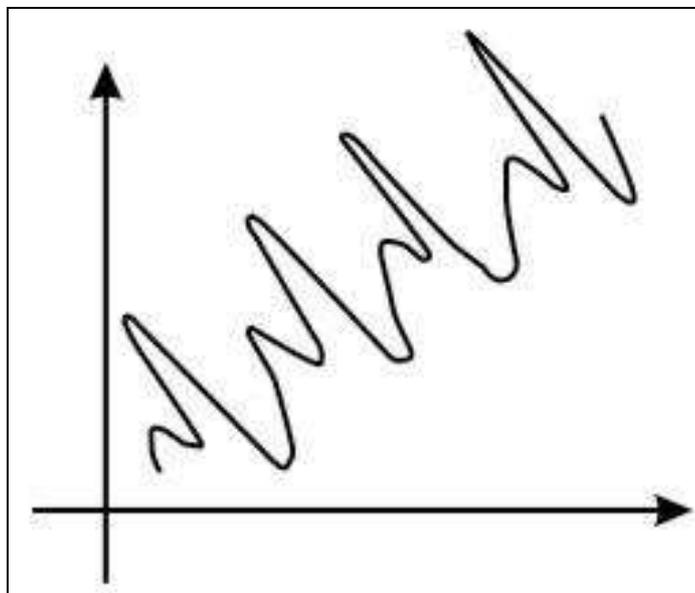


Figura 7 - Padrões mistos

Capítulo 3

Métodos para pré-processamento e previsão de séries temporais

Previsão de valores futuros de uma série temporal observada é um importante problema em várias áreas, incluindo economia, planejamento de produção, previsão de vendas, controle de estoque, etc.

Há uma grande variedade de diferentes métodos de previsão disponíveis e é essencial perceber que não existe um método universalmente aplicável. O analista deve ser apto à escolha de um método mais aplicável a um dado conjunto de condições.

Os procedimentos de previsão podem ser divididos grosseiramente em duas categorias:

- a) Automáticos: que são aplicados diretamente, com a utilização de programas de computador. Como exemplos de métodos automáticos pode se citar Redes Neurais [2] e SVMs [4,5];
- b) Não automáticos: que exigem a intervenção de pessoal especializado, para serem aplicados. Como exemplo de método não automático está o método Box-Jenkins [13] onde o usuário deve ser apto à escolha, com base na análise da série, do melhor modelo Box Jenkins para previsão de séries, que são 3:
 - Modelo auto-regressivo;
 - Modelo de médias moveis; e
 - Modelo auto-regressivo de médias móveis.

Métodos de previsão não automáticos fogem ao escopo do estudo. O leitor interessado poderá consultar [11], [12] e [13].

Mesmo os procedimentos incluídos entre os automáticos requerem do usuário um mínimo de conhecimento e prática para serem utilizados.

3.1 Pré-processamento das séries

Como introduzido anteriormente, séries temporais, geralmente, incorporam efeitos como tendência e sazonalidade. Métodos para previsão de séries não obtêm um bom desempenho diante dessas condições. Dessa maneira é necessário um pré-processamento das séries.

Previsões para séries não estacionárias podem obter resultados muito abaixo dos obtidos por previsões de séries estacionárias. As figuras 8 e 9 ilustram o desempenho de uma previsão realizada para uma série estacionária e para uma outra série não estacionária. Na figura 8, o resultado da previsão foi, significativamente, mais aproximado da série original que o resultado ilustrado na figura 9.

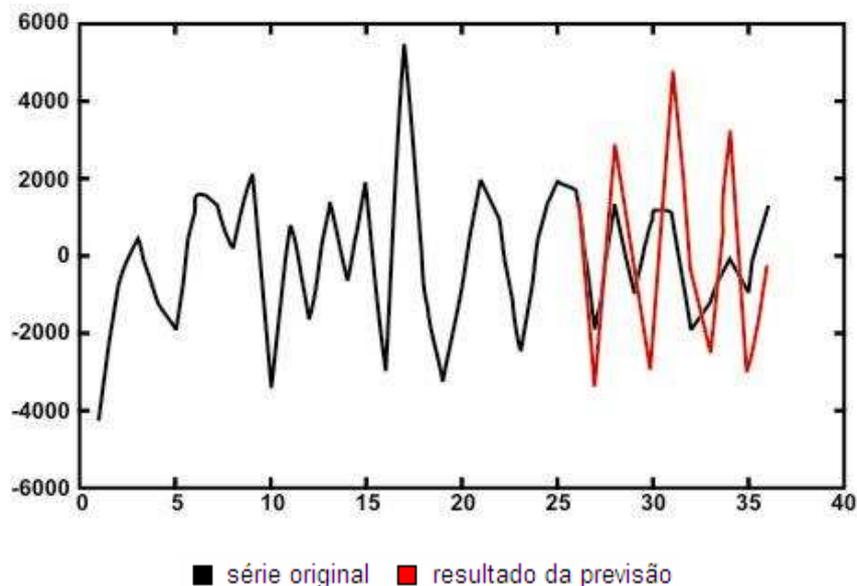


Figura 8 - resultado de previsão de uma série estacionária.

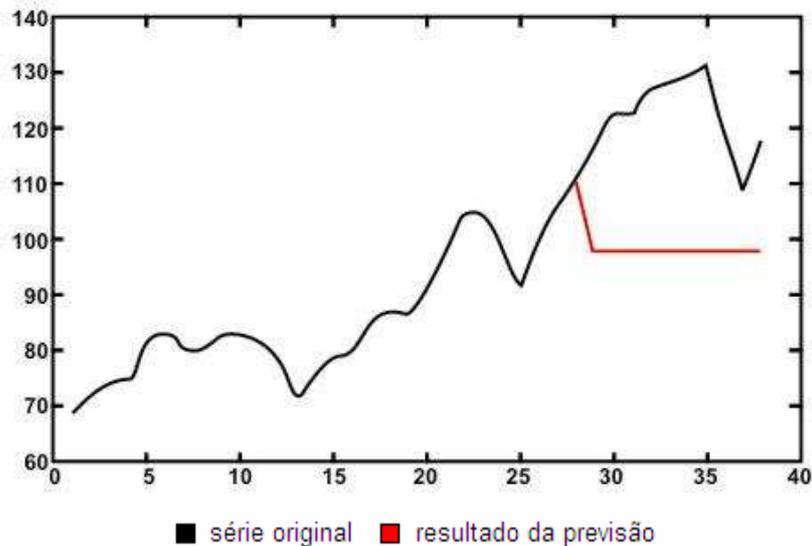


Figura 9 - resultado de previsão de uma série não estacionária.

O pré-processamento consiste de três etapas:

- Tornar a série estacionária;
- Mudança de escala da série; e
- Divisão dos conjuntos de dados.

3.1.1 Auto-correlação da série

Para identificação de efeitos como tendência e sazonalidade de uma forma mais técnica, é necessária a geração do correlograma que consiste no gráfico com os valores das auto-correlações[15] de ordem 1 a n-1, onde n é o tamanho da série. A equação 2 apresenta o operador de auto-correlação, tendo \bar{Z} como a média dos valores da série, k a janela do último processamento realizado, N o número de amostras da série e r_k o valor correlacionado de índice k.

$$r_k = \frac{\sum_{t=1}^{N-k} (z_t - \bar{Z}) * (z_{t+k} - \bar{Z})}{\sum_{t=1}^N (z_t - \bar{Z})^2} \quad \text{(Equação 2)}$$

Séries temporais, como dito anteriormente, são compostas por amostras discretas em relação ao tempo. O intervalo de tempo entre as amostras coletadas se denomina lag, por exemplo, numa série temporal com amostras mensais, o valor do lag é um mês.

O comportamento padrão do correlograma para séries estacionárias são poucos valores significativos além dos primeiros lags, valores alternando entre positivo e negativo e valores próximos a zero nos últimos lags. A figura 10 mostra um correlograma de uma série estacionária.

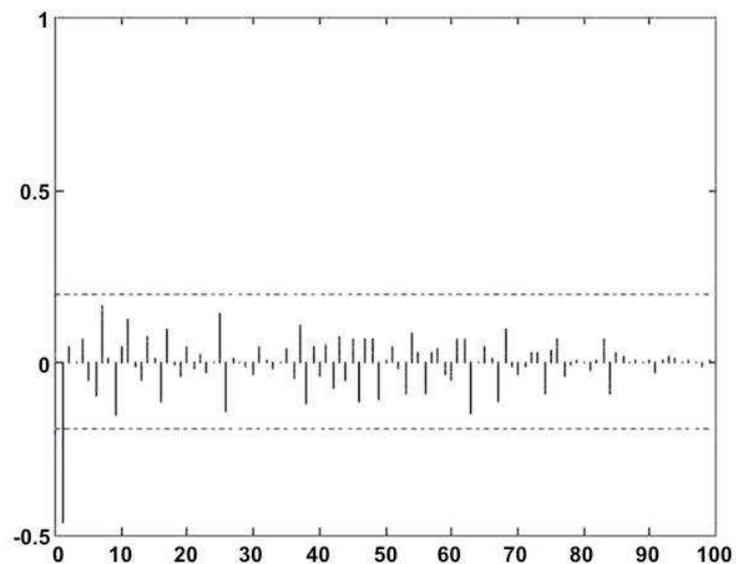


Figura 10 - Correlograma de uma série estacionária

Séries com tendência possuem um correlograma com muitos valores significativos, pouca alternância entre positivos e negativos e valores altos mesmo nos últimos lags. A figura 11 exemplifica um correlograma de uma série com presença de tendência.

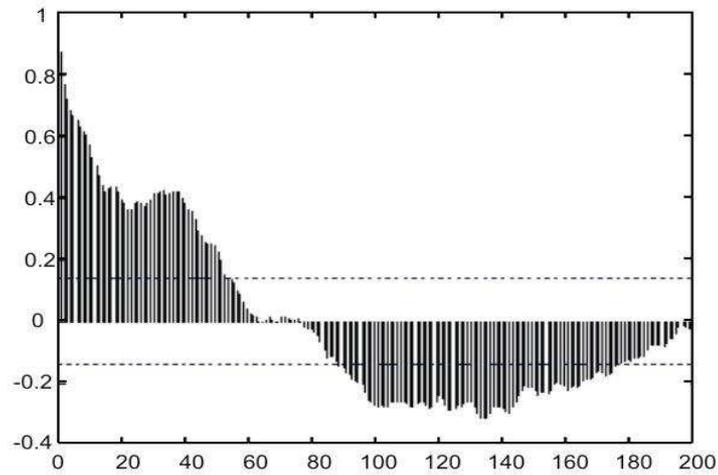


Figura 11 - Correlograma de uma série com tendência

Séries com sazonalidade apresentam um correlograma com presença de picos periódicos. A figura 12 mostra um exemplo de correlograma de uma série com sazonalidade.

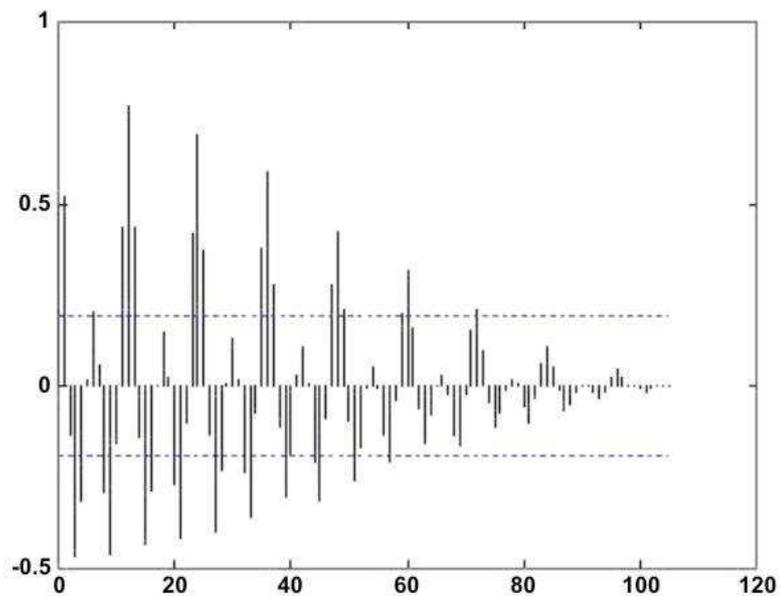


Figura 12 - Correlograma de uma série sazonal

Séries com sazonalidade e tendência apresentam correlogramas com muitos valores significativos, pouca alternância entre positivos e negativos, valores altos mesmo nos últimos lags e picos periódicos, como mostra a figura 13.

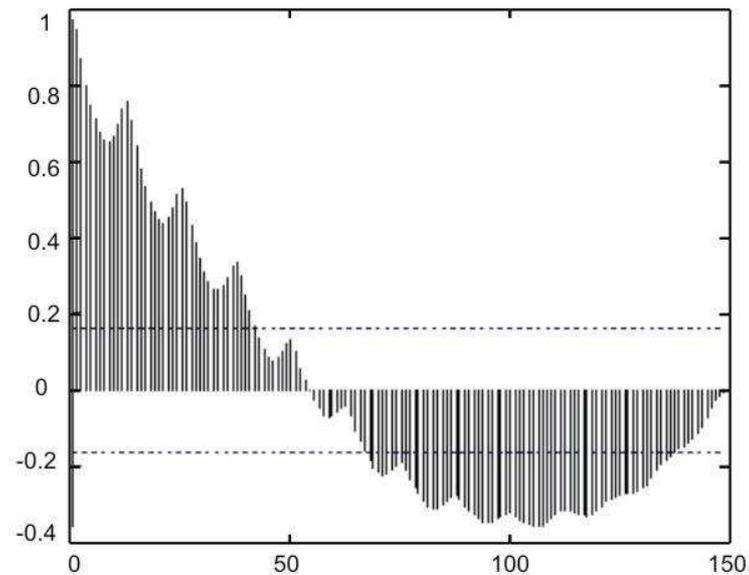


Figura 13 - Correlograma de uma série com presença de sazonalidade e tendência

3.1.2 Eliminação da tendência

Para eliminação da tendência deve se aplicar o operador de diferenças, $Z(t) = Z(t) - Z(t-1)$. Algumas vezes é necessária a aplicação do operador mais de uma vez para a garantia da eliminação da tendência, como mostra a figura 14.

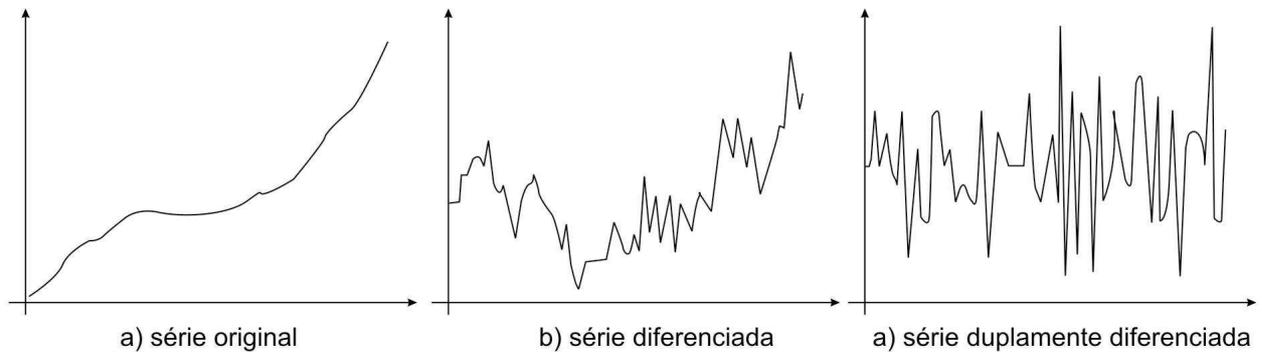


Figura 14 - série duplamente diferenciada

3.1.3 Eliminação da sazonalidade

De forma semelhante à eliminação da tendência, a eliminação da sazonalidade se dá com o uso do operador de diferenciação sazonal, porém nesse caso, o analista deve ter conhecimento do período da sazonalidade pois o operador deve incorporar esse período. Por exemplo, seja uma série sazonal com período de 12 meses, a fórmula de diferenciação seria $Z(t) = Z(t) - Z(t-s)$, onde s representa a janela de diferenciação que no caso seria 12. As figuras 15 e 16 mostram uma série sazonal e a mesma série sazonalmente diferenciada respectivamente.

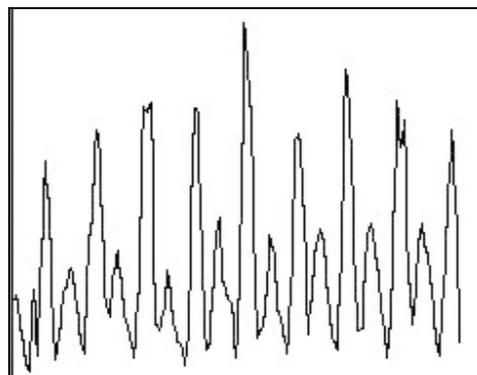


Figura 15 - Série com características sazonais

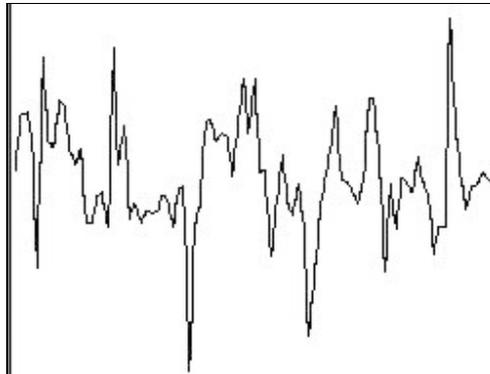


Figura 16 - Série sazonalmente diferenciada

3.1.4 Normalização

Para a obtenção de um bom desempenho os métodos de previsão, geralmente, trabalham com dados num intervalo definido, na maioria das vezes [0,1] ou [-1,1]. A equação 3 mostra o operador de normalização no intervalo [0,1] enquanto que a equação 4 mostra o operador de normalização no intervalo [-1,1].

$$Z'(t) = \frac{Z(t) - \min(Z)}{\max(Z) - \min(Z)} \quad \text{(Equação 3)}$$

$$Z'(t) = \frac{Z(t) - \max(Z) - \min(Z)}{\max(Z) - \min(Z)} \quad \text{(Equação 4)}$$

3.2 Previsão de séries temporais

Existem várias técnicas para a realização da previsão de séries temporais. Como citado anteriormente, dois dos mais populares e eficientes métodos para previsão de séries temporais são SVMs e redes Neurais Artificiais. A previsão de séries temporais deriva do conceito de regressão linear que tem como finalidade, dado um conjunto de pontos, encontrar a equação de uma reta que

melhor se adequa a esse conjunto de pontos. A seguir serão introduzidos de forma mais detalhada os conceitos anteriormente citados.

3.2.1 Regressão Linear

O problema da regressão linear consiste em achar uma função linear, $f(x) = \langle w \cdot x \rangle + b$, que melhor se adequa um conjunto de pontos (conjunto de treinamento)[5][14]. Geometricamente isso corresponde a um hiperplano que melhor se adapta aos pontos do conjunto. A Figura 17 mostra uma função de regressão linear unidimensional. A distância ξ é o erro obtido para o ponto em particular.

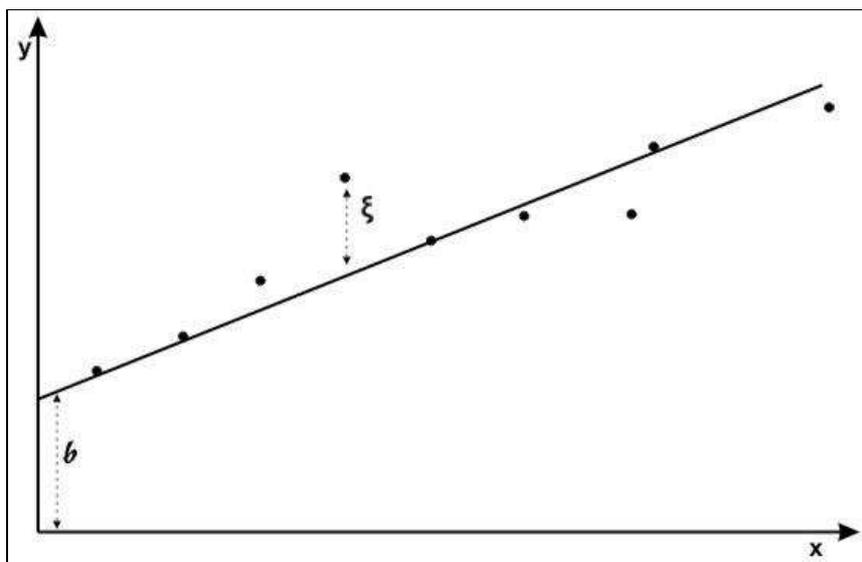


Figura 17 - Função de regressão linear unidimensional

O objetivo é achar uma função que minimize a soma dos erros quadráticos para cada ponto. A melhor função para esse fim acontece para w e b tais que:

$$L(w, b) = \sum_{i=1}^l (y_i - \langle w \cdot x_i \rangle - b)^2 \quad \text{(Equação 5)}$$

Um método automático e eficaz de verificação da qualidade do treinamento consiste no monitoramento dos erros médios quadráticos, EMQs, no momento do treinamento. O melhor desempenho de uma função de aprendizagem pode ser indicado por um baixo valor do EMQ, porém, o menor valor do EMQ não significa que o treinamento tenha obtido uma boa generalização, pois pode ocorrer overfitting, fenômeno em que o algoritmo de aprendizagem se adapta tão bem ao conjunto de treinamento que ruídos aleatórios são incluídos no modelo como características significativas. O EMQ é obtido através da Equação 6, onde o n é o número de amostras, X é o vetor contendo as saídas obtidas e Y é o vetor de saídas desejadas.

$$\frac{1}{n} \sum_{i=1}^n (x_i - y_i)^2 \quad \text{(Equação 6)}$$

3.2.2 Redes Neurais Artificiais

RNAs[2] são sistemas paralelos distribuídos compostos por unidades de processamento simples (nodos) que calculam determinadas funções matemáticas (normalmente não lineares). Tais unidades são dispostas em uma ou mais camadas e interligadas por um grande número de conexões, geralmente unidirecionais. Na maioria dos modelos estas conexões estão associadas a pesos, os quais armazenam o conhecimento representado no modelo e servem para ponderar a entrada recebida por cada neurônio da rede. O funcionamento destas redes é inspirado em uma estrutura física concebida pela natureza: o cérebro humano.

Redes neurais por si só não lidam satisfatoriamente com reconhecimento de padrões dinâmicos (temporais). Um artifício utilizado para processamento temporal envolve o uso de janelas de tempo, em que a entrada da rede utiliza trechos dos dados temporais como se eles formassem um padrão estático. Entretanto, essa não é a solução mais indicada para o processamento temporal. O ideal seria a utilização de uma rede cuja estrutura fosse adequada para esse tipo de processamento. Para que uma RNA seja considerada dinâmica é necessário que ela possua memória.

Existem basicamente duas maneiras de prover memória a uma RNA:

1. Introduzir atraso no tempo, como as técnicas TDNN, Time Delay Neural Network[16].
2. Utilizar redes recorrentes, tais como back-propagation Through Time[17], Real Time Recurrent Learning[18,19], Cascade Correlation

Recorrente [20], Redes de Elman[16] e Redes de Jordan[16].

3.2.3 Redes MLP

Redes MLP (Multi Layer Perceptron)[21] se originam do modelo Perceptron de redes neurais. O modelo Perceptron, concebido por McCulloch e Pitts [22], enfocou a modelagem de um neurônio biológico e sua capacidade computacional com a apresentação de vários exemplos de topologias de redes com capacidade de execução de funções booleanas. A topologia original, descrita por Rosenblatt, era composta por unidades de entrada (retina), por um nível intermediário formado pelas unidades de associação e por um nível de saída formado pelas unidades de resposta. A figura 18 mostra a topologia de um perceptron simples. Embora essa topologia inicial possua três níveis, ela é conhecida como perceptron de uma única camada, já que somente o nível de saída (unidades de resposta) possui propriedades adaptativas.

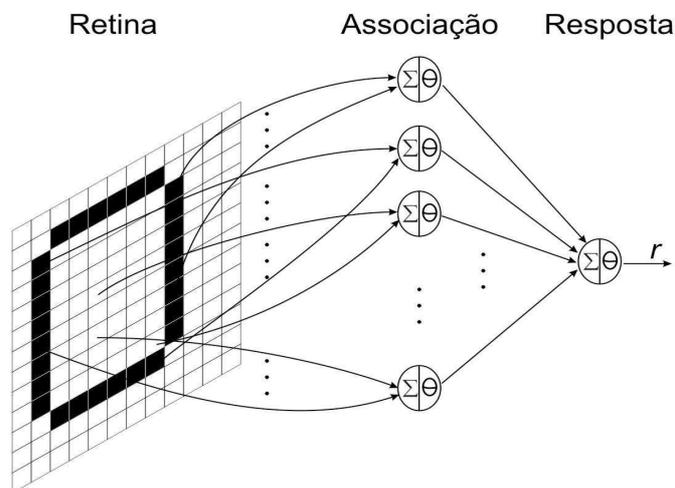


Figura 18 – Topologia de um perceptron simples

A solução de problemas não linearmente separáveis passa pelo uso de redes com uma ou mais camadas intermediárias ou escondidas (Redes Multi Layer Perceptron - MLPs). Segundo Cybenko [23], uma rede com uma camada intermediária pode implementar qualquer função

continua. A utilização de duas camadas intermediárias permite a aproximação de qualquer função [24]. A figura 19 apresenta uma rede MLP típica.

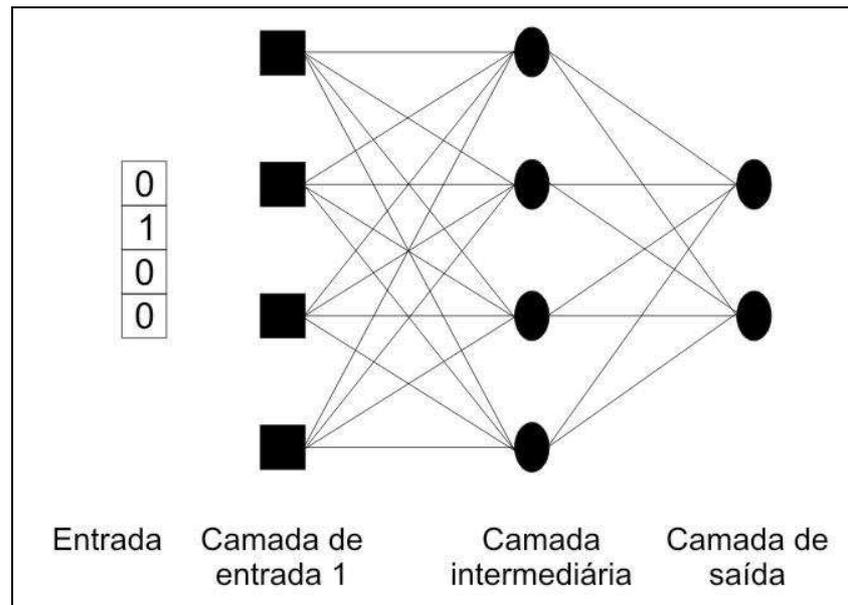


Figura 19 - Rede MLP típica com uma camada intermediária

Para uma rede com pelo menos duas camadas intermediárias, pode se dizer que o seguinte processamento ocorre em cada uma das camadas:

- **Primeira camada intermediária:** cada nodo traça retas no espaço de padrões de treinamento.
- **Segunda camada intermediária:** cada nodo combina as retas traçadas pelos neurônios da camada anterior conectados a ele, formando regiões convexas, onde o número de lados é definido pelo número de unidades a ele conectadas.
- **Camada de saída:** cada nodo forma regiões que são combinações das regiões convexas definidas pelos nodos a ele conectados da camada anterior. Os nodos definem, desta maneira, regiões com formatos abstratos.

No uso de MLPs[21] para previsão de séries temporais é necessário definir como entrada um intervalo de tempo (janela) que servirá como entrada para a rede e o valor imediatamente

após a seqüência servirá como saída. A tabela 1 mostra a configuração de entradas e saídas para um treinamento com janela de tempo 4.

Tabela 1 - Entradas e saídas desejadas para o treinamento com MLP e janela 4

Entradas	Saídas Desejadas
Z(1),Z(2),Z(3),Z(4)	Z(5)
Z(2),Z(3),Z(4),Z(5)	Z(6)
.....
Z(n-4),Z(n-3),Z(n-2),Z(n-1)	Z(n)

3.2.4 Redes de Elman e Jordan

Nas redes de Elman[16], além das unidades de entrada, intermediárias e de saída, há também unidades de contexto. As unidades de entrada e saída interagem com o meio externo, enquanto as unidades intermediárias e de contexto não o fazem. As unidades de contexto são usadas para memorizar as ativações anteriores das unidades intermediárias e podem ser consideradas como atraso no tempo de um passo. A figura 20 mostra um exemplo de topologia Elman.

As redes de Jordan[16] são similares às redes de Elman, porém as unidades de contexto são usadas para memorizar as ativações anteriores das unidades de saída. A topologia Jordan é ilustrada pela figura 21.

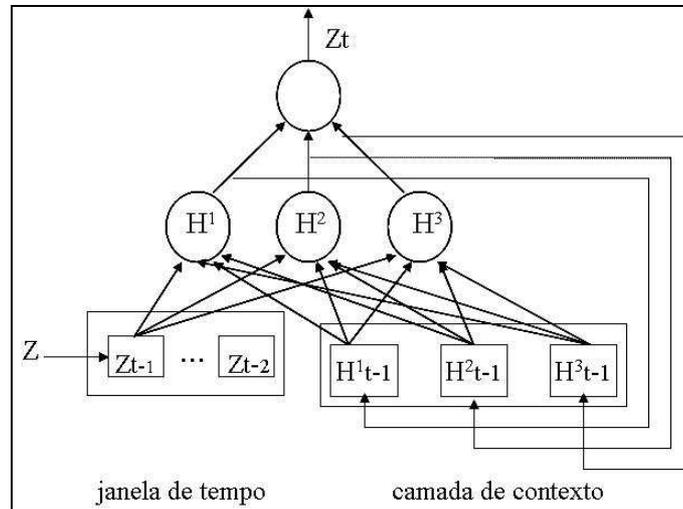


Figura 20 - Topologia Elman

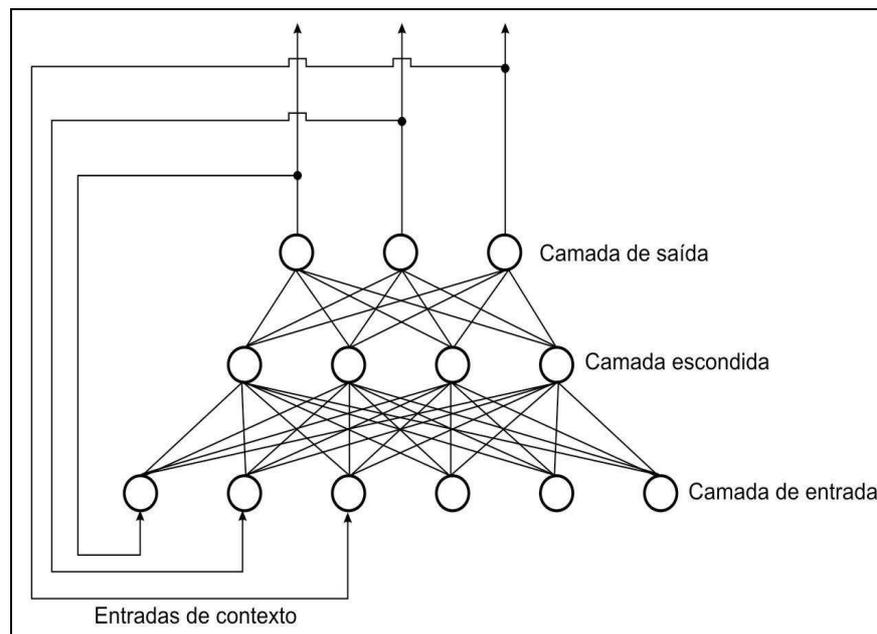


Figura 21 - Topologia Jordan

3.2.5 SVMs (Máquinas de vetor de suporte)

O algoritmo SV (*Support Vector*) é uma generalização não linear do algoritmo *Generalized Portrait* [26,27] desenvolvido na Rússia nos anos 60. No início dos anos 90 foram introduzidas, por Vapnik e equipe, máquinas de vetor de suporte que pertencem a uma nova geração de sistemas baseados em recentes avanços na teoria do aprendizado estatístico. Devido ao seu contexto industrial, pesquisas com Vetores de suporte ganharam um direcionamento a aplicações do mundo real [28]. SVMs [4,5] empregam o estado da arte na estatística em aplicações do mundo real como categorização de textos, reconhecimento de caracteres manuscritos, classificação de imagens, regressão, etc. Dentre as aplicações relativas às máquinas de vetor de suporte, regressão é a aplicação de interesse para o desenvolvimento deste trabalho. Em relação à previsão de séries temporais, ótimos resultados foram obtidos [29,30].

Em geral, aplicações do mundo real requerem um espaço de hipóteses, espaço representativo da função alvo $g: X \rightarrow Y$ (onde X corresponde ao vetor de instâncias para treinamento e Y corresponde ao vetor de saídas desejadas, alvos), mais expressivo do que funções lineares podem representar. Uma outra maneira de se visualizar esse problema é que freqüentemente o conceito alvo não pode ser expresso como uma simples combinação linear dos atributos dados, mas em geral requer que características mais abstratas dos dados sejam exploradas.

Funções de kernel oferecem uma solução alternativa projetando os dados em um espaço, de características, n -dimensional com a finalidade do aumento do poder computacional de máquinas de aprendizado linear. Os exemplos de treinamento nunca aparecem isolados, mas sempre na forma de produtos internos entre pares de exemplos. Substituindo o produto interno por uma função de kernel apropriada, pode-se, implicitamente criar um mapeamento de uma função não linear para um espaço de características n -dimensional sem que haja a necessidade do aumento do número de parâmetros ajustáveis.

SVMs fazem uso de várias funções de *kernel* como funções de base radial, funções polinomiais e sigmoid. De acordo com a finalidade da aplicação o usuário terá possibilidade de escolha do *kernel* que melhor atende à resolução de determinado problema. Muitas funções de kernel podem ser usadas, provavelmente um número infinito, porém, poucas funções de *kernel* têm obtido resultados satisfatórios para uma grande variedade de aplicações. A função de kernel mais recomendada é a RBF. São exemplos de funções de kernel:

- Linear: $K(x_i, x_j) = x_i^T x_j$

- Polinomial: $K(x_i, x_j) = (\gamma x_i^T x_j + r)^d, \gamma > 0$
- Sigmóide: $K(x_i, x_j) = \tanh(\gamma x_i^T x_j + r)$
- Função de base radial (RBF): $K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2), \gamma > 0$

A figura 22 mostra um exemplo de um mapeamento de um espaço de entradas bidimensional para um espaço de características bidimensional, onde os dados podem ser separados por uma função linear no espaço de características, mas não podem ser no espaço de entradas.

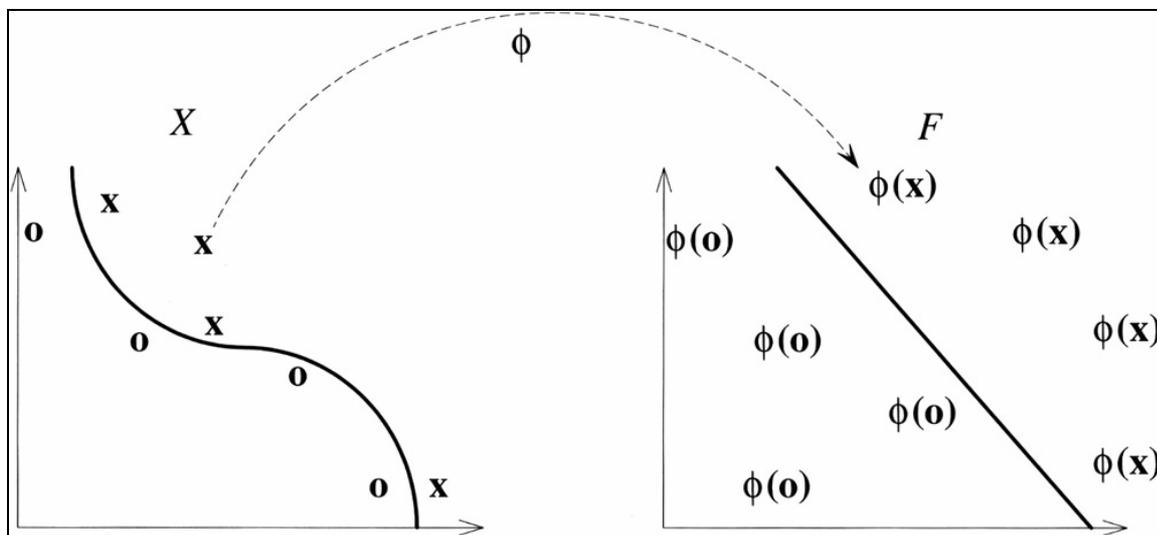


Figura 22- Mapeamento de espaço de entradas no espaço de características

Uma das características mais atrativas do método SVM consiste na maximização da margem separadora das diferentes classes. A figura 23 ilustra a separação otimizada das margens entre classes realizada pelo método SVM. No dois casos ilustrados, (1) e (2), a separação foi realizada com sucesso, porém no caso (1) ela não foi realizada de maneira ótima, a probabilidade de uma classificação equivocada no caso (1) é maior.

Teoricamente, o caso (1) corresponde à função obtida por um aprendizado realizado por RNAs, por exemplo, enquanto que o caso (2) ilustra a função obtida pelo método SVM.

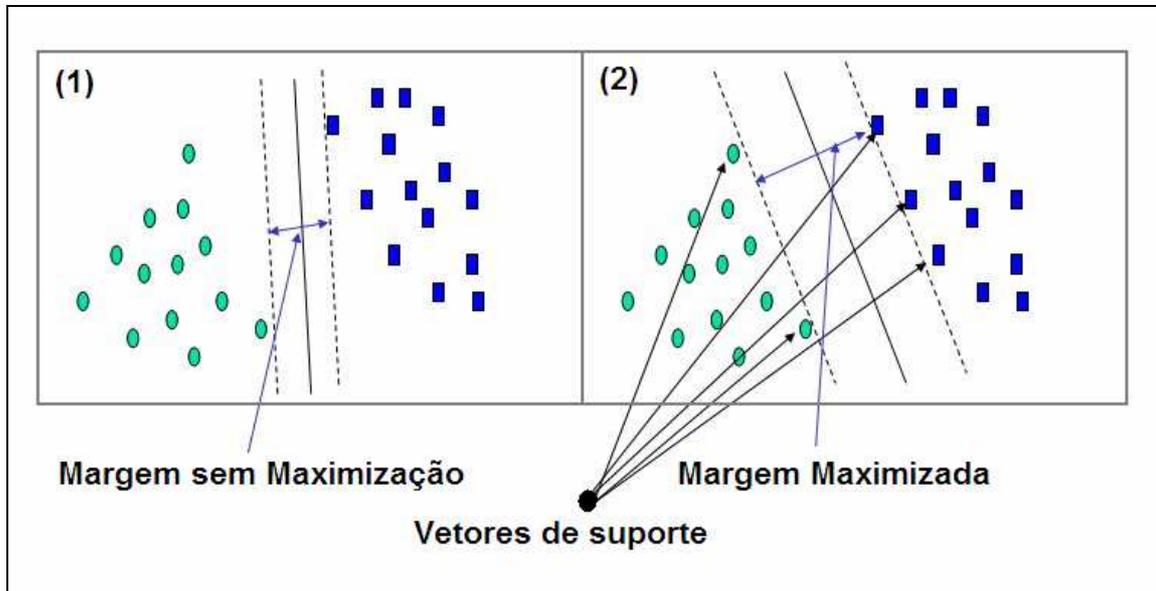


Figura 23 - Separação otimizada entre classes

Algumas vantagens de SVMs podem ser destacadas se comparadas a redes neurais[25]:

- SVMs não caem em mínimos locais, ou seja, garante-se a obtenção de um mínimo global, ver figura 24;
- SVMs conseguem uma separação otimizada entre planos, ver figura 23;
- SVMs conseguem uma boa representação do problema mesmo através de um pequeno conjunto de dados de treinamento.
- SVMs não necessitam de uma busca pelo melhor número de vetores de suporte, assim como redes neurais necessitam a indicação do número de neurônios na camada escondida, no caso das topologias de treinamento não construtivos;
- SVMs não precisam da divisão do conjunto de dados em treinamento, validação e teste. Apenas são necessários os conjuntos de treinamento e teste. Isso aumenta a eficiência do treinamento pois disponibiliza um maior número de amostras para o treinamento e teste.

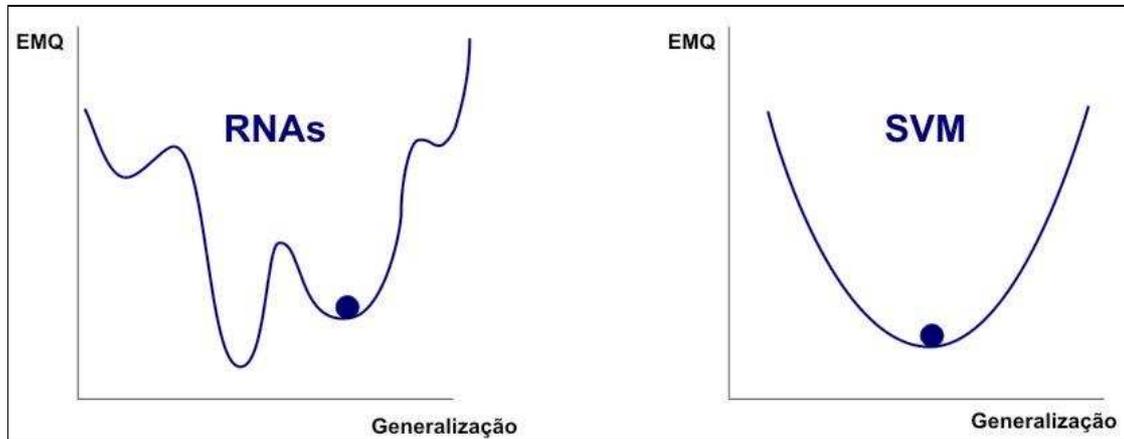


Figura 24 - EMQ X Generalização

3.3 Previsões de séries temporais com intervalos de confiança robustos

Em muitos trabalhos relativos à previsão de séries temporais apenas os valores previstos são providos. Na prática, é também bastante importante prover um indicativo de confiança da previsão [3]. Isso pode ser obtido pelo uso de intervalos de confiança.

Intervalos confiança robustos, são genéricos e podem ser computados a partir de previsões feitas por qualquer modelo, como SVM[4,5], Box-Jenkins[12] e redes neurais[2].

Intervalos de confiança robustos, ilustrados na figura 25, são computados a partir do conjunto de previsões dentro de um conjunto de dados conhecidos. Preferencialmente, esse conjunto de dados deve ser diferente do conjunto de treinamento.

Os erros de previsão são coletados tomando-se a diferença entre os valores previstos e os valores corretos, $x'(p+1) - x(p+1)$. O sinal do erro computado deve ser preservado.

A idéia por trás de intervalos de confiança robustos é que erros de um dado valor tendem a ocorrer com aproximadamente a mesma freqüência com que ocorreram anteriormente. Por exemplo, suponha que aproximadamente um quarto dos erros coletados do conjunto de treinamento excede 2.3. Então, é de se esperar que quando o modelo seja usado para a previsão, os erro excedam 2.3 em aproximadamente um quarto das amostras.

O seguinte exemplo ilustra o procedimento usado para computar intervalos robustos de confiança. Suponha que existem $n = 10$ erros coletados. Essa coleção é primeiramente ordenada para resultar no vetor $e = \{-0.4, -0.3, -0.1, 0.0, 0.0, 0.0, 0.1, 0.1, 0.2, 0.4\}$. Suponha que queiramos criar um intervalo de confiança com 60% de precisão baseado nessa coleção de erros. Nesse caso, $p = (1.0-0.6)/2 = 0.2$, que resulta $(n \times p) - 1 = 1$. Portanto, neste caso, uma amostra deve ser descartada de cada extremidade da coleção de erros. Como resultado, o intervalo de confiança com 60% de precisão para um valor futuro previsto $x'(p)$ seria $\{x'(p) - 0.3, x'(p) + 0.2\}$.

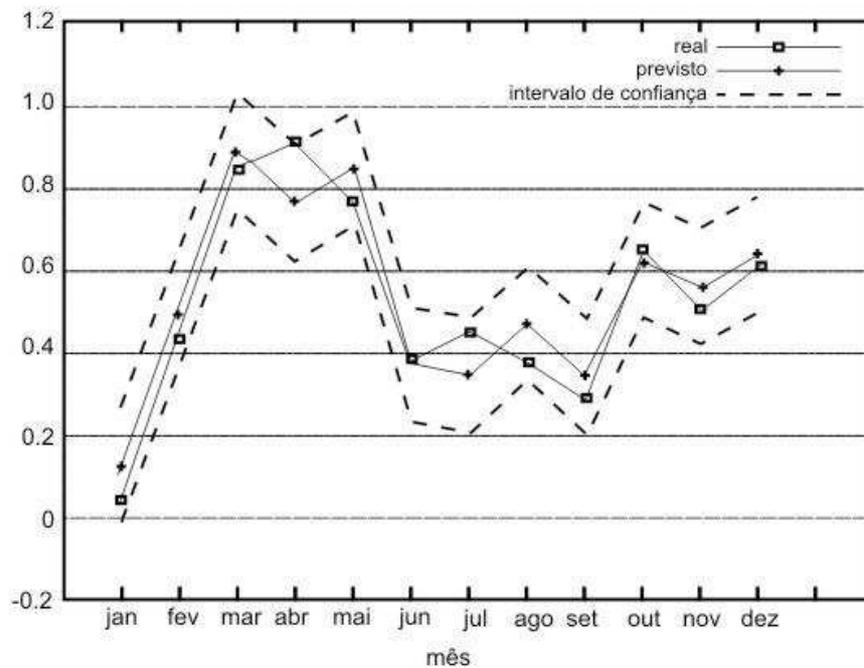


Figura 25 – Previsão de séries temporais com intervalos de confiança robustos

Capítulo 4

A Ferramenta Desenvolvida

A necessidade de uma ferramenta capaz de lidar com as dificuldades referentes ao processamento de dados temporais motiva a execução deste projeto. O projeto consiste de uma ferramenta de fácil uso e interface amigável com a capacidade de fornecer aos usuários, com um certo nível de familiarização com o assunto, meios de assegurar o correto andamento do processo de previsão de séries temporais. O escopo da ferramenta vai desde o pré-processamento até ao treinamento e testes das séries. Para a realização das diversas funcionalidades, gráficos e dados estatísticos são disponibilizados ao usuário de maneira a possibilitar a verificação da eficiência da tarefa realizada.

Com a utilização da ferramenta o usuário tem a capacidade de realizar treinamentos, utilizando a biblioteca LibSVM[6,7], de várias maneiras. No treinamento com os “N” melhores EMQs a ferramenta automaticamente gerará dados para a criação de intervalos de confiança robustos com tendência a uma distribuição de erros mais realista de acordo com o aumento do parâmetro “número de EMQs” escolhido. O teste dos valores previstos gera dados necessários para a criação dos gráficos de teste normal e de teste com intervalos de confiança robustos realizando também, de forma automática, um processamento inverso. O pré-processamento inverso é efetuado para que a série volte à sua escala original com remoção de todas as transformações realizadas sobre a série, no intuito de uma visualização de um gráfico com maior valor agregado.

Todos os resultados de pré-processamentos realizados são salvos em arquivos xml de maneira que se possa recuperá-los no futuro caso seja constatado um bom desempenho do modelo de treinamento gerado para aquele pré-processamento.

A figura 26 mostra a aparência da ferramenta durante o uso de suas várias funcionalidades.

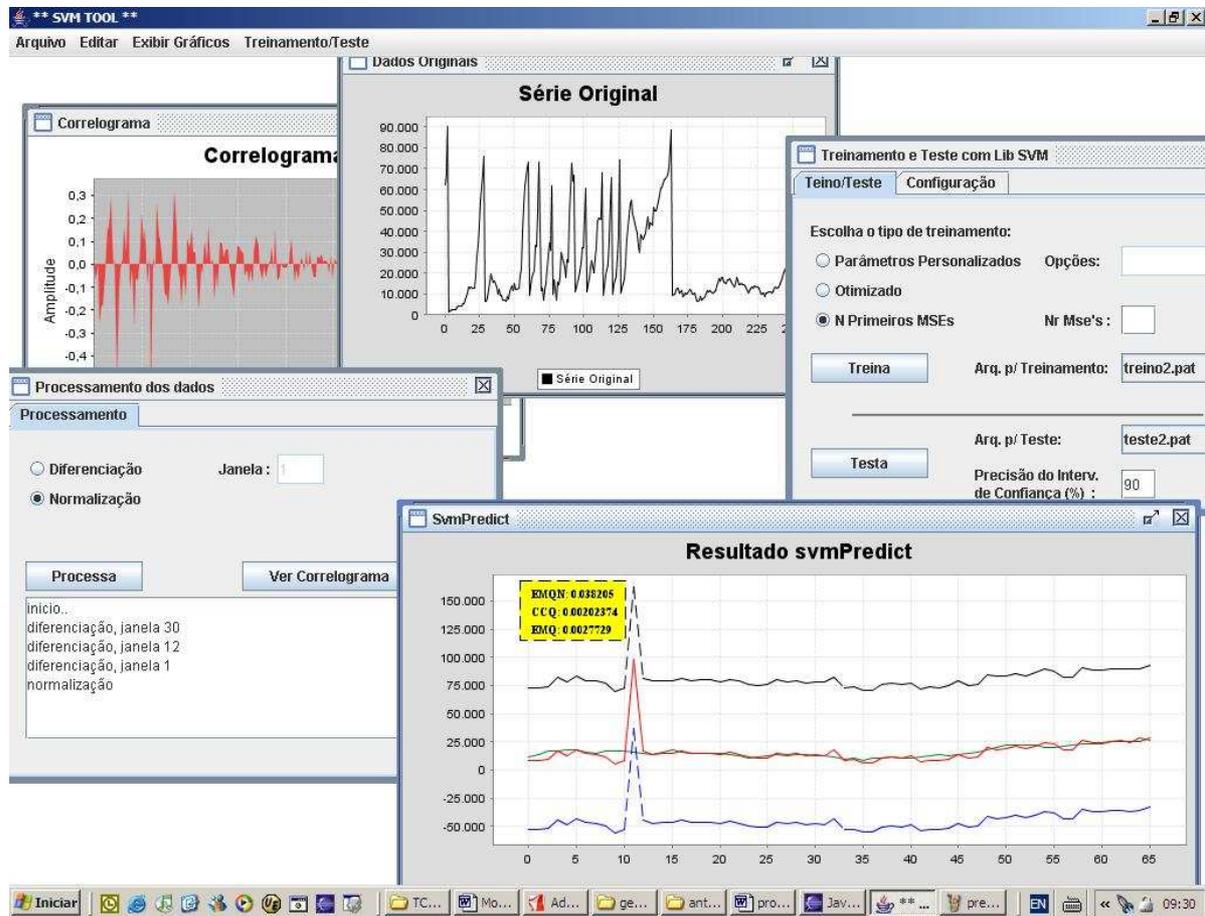


Figura 26 – Screenshot da ferramenta

4.1 Características técnicas

JAVA foi a linguagem escolhida para o desenvolvimento da ferramenta. Vários fatores foram determinantes para a escolha da linguagem. A vasta API disponibilizada para os usuários da linguagem é um grande atrativo, pacotes como javax.swing, java.io, e java.util englobam uma grande parte das necessidades para o desenvolvimento do projeto. Outro aspecto importante e determinante para a escolha da linguagem é a grande variedade de bibliotecas existentes.

Duas bibliotecas foram de grande importância no desenvolvimento do projeto:

- JFreeChart – responsável pela criação e customização dos gráficos gerados.

- JDom – responsável pela geração dos arquivos XML, contendo as informações necessárias ao treinamento e teste, criados após o processamento da série original.

O sistema está dividido em pacotes. O uso de pacotes é essencial por questões de engenharia de software. Com o uso de pacotes a leitura do código fica expressivamente mais simples. A figura 27 mostra o digrama de classes do sistema.

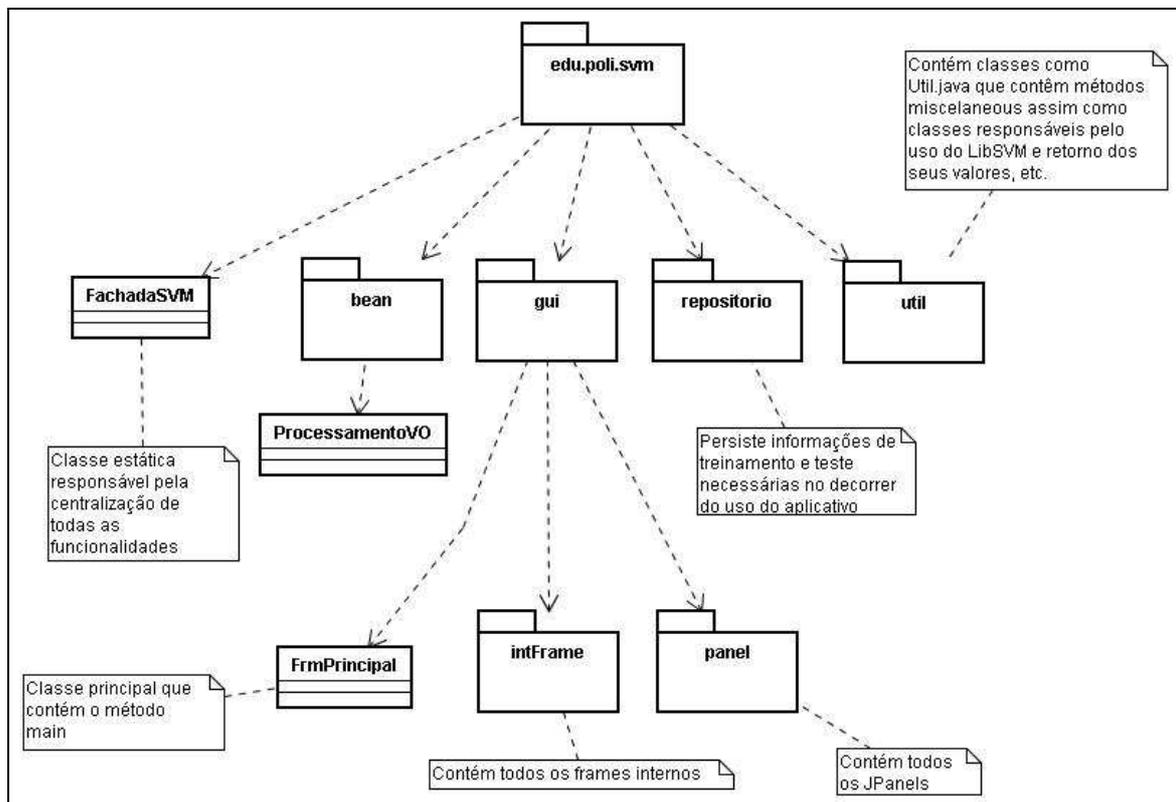


Figura 27 - Diagrama de classes do projeto

4.2 Funcionalidades

As funcionalidades da ferramenta podem ser acessadas através de 4 menus:

- Menu Arquivo;
- Menu Editar;
- Menu Gráficos; e
- Menu Treinamento/Teste.

O diagrama de casos de usos, mostrado figura 28, ilustra todas as funcionalidades da ferramenta e suas dependências entre si.

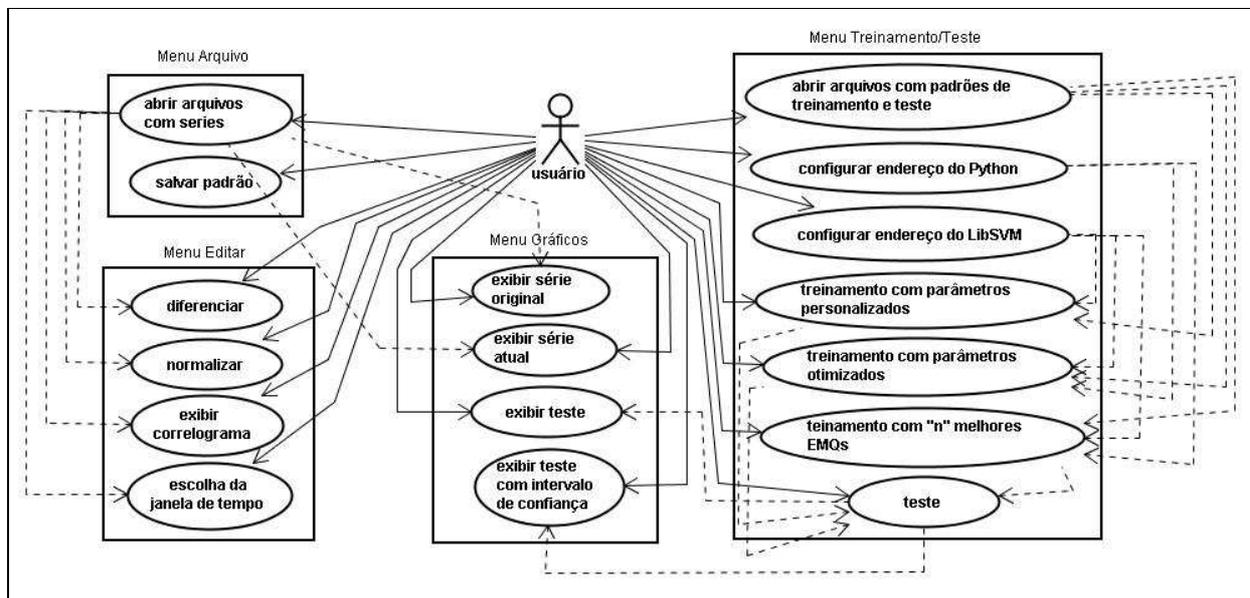


Figura 28 - Diagrama de casos de uso

4.2.1 Menu Arquivo

O Menu arquivo possui duas funcionalidades básicas:

- Abrir arquivos contendo séries temporais;
- Salvar arquivos no formato padrão para uso do LibSVM.

A funcionalidade de abrir arquivos, aceita arquivos de séries temporais contendo séries onde os valores das amostras estão separados por uma linha ou espaço em branco.

A funcionalidade salvar padrões depende do término do pré-processamento das séries assim como da escolha da janela de tempo para previsão da série. O padrão salvo segue o seguinte formato:

$$\begin{array}{cccccc} Z(n) & Z(n-j) & Z(n-(j+1)) & Z(n-(j+2)) & \dots & Z(n-(j+(j-1))) \\ Z(n+1) & Z((n+1)-j) & Z((n+1)-(j+1)) & Z((n+1)-(j+2)) & \dots & Z((n+1)-(j+(j-1))) \\ \dots & & & & & \end{array}$$

Onde j é o tamanho da janela. Para cada linha do arquivo, o primeiro elemento é o elemento a ser previsto, nesse caso $z(n)$ e $Z(n+1)$.

4.2.2 Menu Editar

O Menu editar possui duas opções:

- Painel de processamento;
- Escolha da janela.

O painel de pré-processamento engloba todas as funcionalidades necessárias à realização do pré-processamento das séries. Nele o usuário tem a possibilidade de diferenciar a série escolhendo a janela de diferenciação e ao mesmo tempo monitorar o resultado. O monitoramento do correlograma das séries, como discutido anteriormente, é necessário para a avaliação da eficiência do pré-processamento.

Uma vez removidos os efeitos indesejáveis da série o usuário poderá normalizá-la e salvar os padrões de treinamento e teste a serem usados pelo LibSVM. A figura 29 ilustra o layout do painel de pré-processamento dos dados para o treinamento e a figura 30 mostra um gráfico de correlograma obtido com nossa ferramenta.

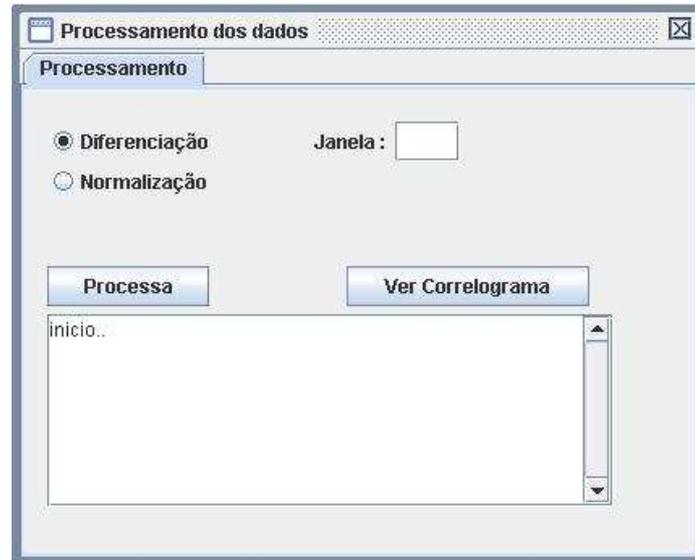


Figura 29 - Painel de pré-processamento

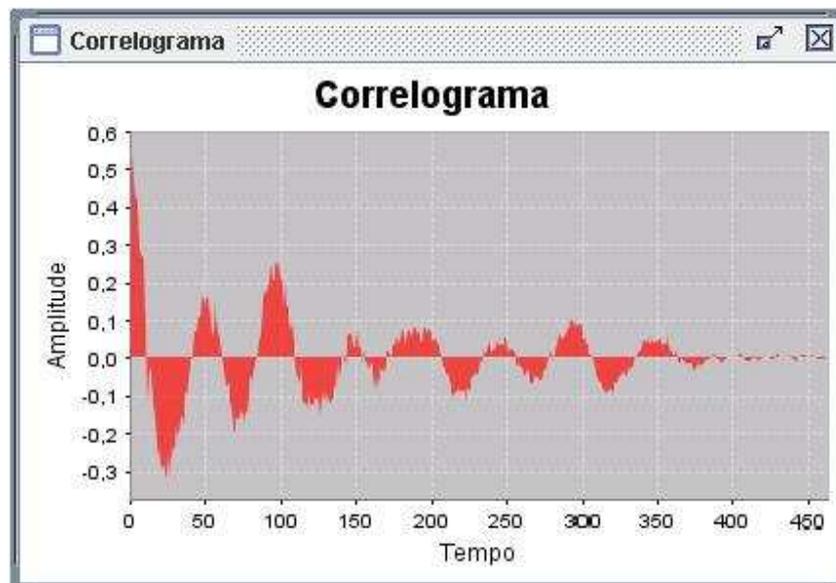


Figura 30 - Correlograma gerado pela ferramenta

4.2.3 Menu Gráficos

A partir do menu de gráficos o usuário terá a opção de visualização de 4 tipos de gráficos:

- Série original;
- Série atual;
- Teste;
- Teste com intervalos de confiança.

Os gráficos de testes podem apenas ser acessados após a conclusão do treinamento e teste, feitos no painel de controle de treinamento e teste, diferentemente dos gráficos da série original e série atual que podem ser acessados a qualquer momento após a série ser carregada.

Os gráficos de teste trazem também informações sobre a eficiência do teste. São disponibilizadas as medidas EMQ (Erro Médio Quadrático), CCQ (Coeficiente de Correlação Quadrática) e EMQN (Erro Médio Quadrático Normalizado).

As figuras 31, 32 e 33 mostram respectivamente os gráficos de uma série original, de teste e teste com intervalos de confiança robustos.

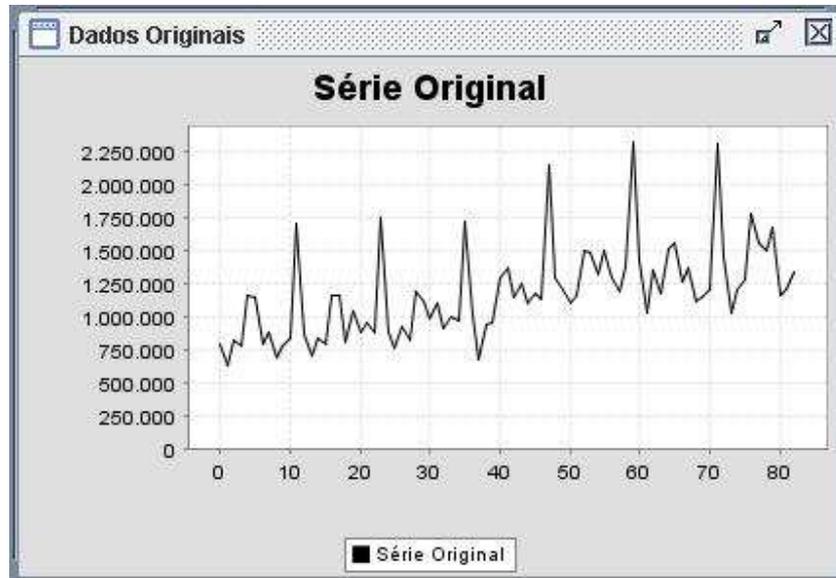


Figura 31 - Série original

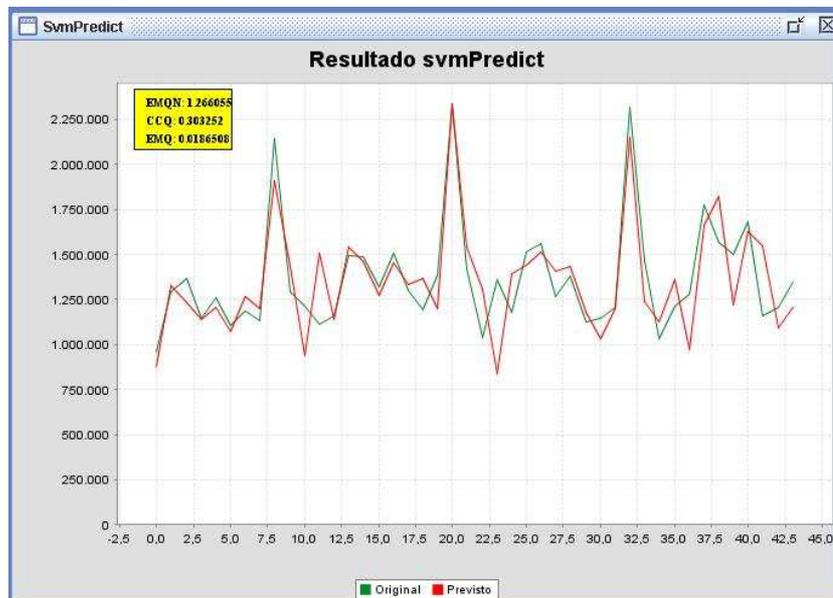


Figura 32- Teste sem intervalos de confiança robustos

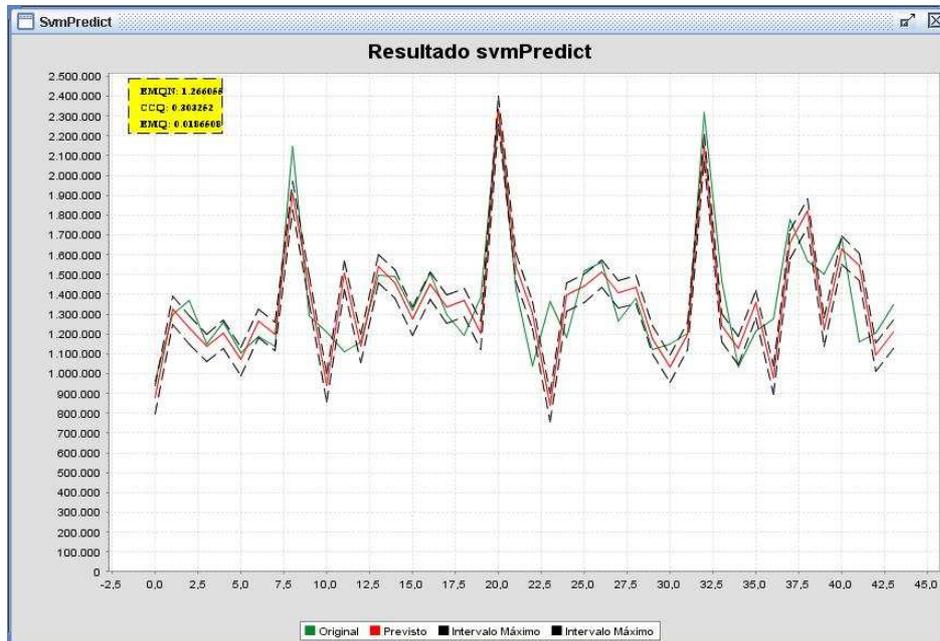


Figura 33 - Teste com intervalos de confiança robustos

4.2.4 Menu Treinamento/Teste

O menu “Treinamento/Teste” é responsável, basicamente, pela execução de treinamentos e teste dos padrões obtidos pelo pré-processamento das séries. O menu “Treinamento/Teste” possui as seguintes opções :

- Selecionar Arquivos Treinamento/Teste – essa opção disponibiliza ao usuário uma caixa de escolha de arquivos com extensão .pat (arquivos criados pela ferramenta) com fins de treinamento e teste.
- Painel de Controle – O painel de controle contém todas as funcionalidades necessárias para treinamento e teste.

O painel de controle de treinamento e teste possui duas abas, a aba de configuração e a de treino/teste. Na aba de configuração o usuário deverá adicionar os caminhos das pastas que contêm os aplicativos svmtrain.exe e svmpredict.exe para o uso da biblioteca LibSVM e python.exe para o uso do Python. Ao clicar no botão LibSVM e Python uma caixa de dialogo é aberta e nela

devem ser inseridos os caminhos dos executáveis acima citados. A aba configuração contém ainda uma área de texto onde são mostradas todas as informações saídas da execução do LibSVM.

A aba de treino/teste contém os elementos necessários ao treinamento e teste das séries com máquinas de vetor de suporte. Existem três tipos de treinamentos possíveis:

- Parâmetros Personalizados;
- Otimizado; e
- N primeiros EMQs.

4.2.4.1 Busca por valores ótimos dos parâmetros para treinamento

A precisão de um modelo SVM é extremamente dependente da seleção dos parâmetros do modelo. Na busca pelos valores ótimos de parâmetros é usado o script `gridregresion.py`, na linguagem Python, que executa uma busca em grid pelos parâmetros ótimos. Uma busca em grid tenta valores de cada parâmetro em um intervalo específico.

Buscas em grid são computacionalmente custosas pois o modelo deve ser avaliado para cada caso encontrado pela busca. Uma busca cresce de acordo com os intervalos escolhidos para cada parâmetro e com o número de parâmetros. Por exemplo, no nosso caso é usado o modelo ϵ -SVR que aceita três parâmetros (C , γ e ϵ), então uma busca em grid com intervalo de valores 10 para cada parâmetro requer avaliação de $10 \cdot 10 \cdot 10 = 1000$ modelos diferentes.

4.2.4.2 Tipos de treinamento

Para o treinamento, o usuário tem à sua disposição um Combo Box contendo todos os arquivos carregados no formato próprio para uso no LibSVM.

O treinamento com parâmetros personalizados requer do usuário uma certa familiaridade com a biblioteca LibSVM para a escolha dos parâmetros c , g e p de maneira a diminuir o valor do EMQ do treinamento. Com essa opção o usuário deverá inserir, da mesma forma da execução da biblioteca no prompt do dos, as opções desejadas. Por exemplo, “-s 3 -c 0.625 -g 32 -p

5”. Essa opção torna-se bastante necessária quando o usuário já conhece os valores ideais para cada parâmetro.

No treinamento com parâmetros otimizados, o script `gridregression.py` será chamado e os valores ótimos de cada parâmetro serão capturados e será executado um treinamento com esses valores.

O treinamento com os “N primeiros EMQs” também executa o script `gridregression.py`, e treina a série com os parâmetros otimizados encontrados pela execução do script. A principal idéia por trás desse tipo de treinamento é a realização de treinamentos com base nos melhores parâmetros encontrados e a partir desses treinamentos realizar testes para a obtenção de uma maior quantidade de conjuntos de erros de teste para a criação de intervalos robustos de confiança. A geração de um conjunto representativo de erros é essencial pois à medida que o número de amostras de erros aumenta pode se assumir que a função $S_n(e)$, *sample*, obtida através dos erros encontrados, esteja bem próxima à função $F(e)$, a distribuição real de erros.

A figura 34 ilustra o painel de controle responsável pela fase de treinamento e teste das séries.



Figura 34 - Painel de controle para realização de treino e teste das séries

4.2.4.3 Teste

Na realização do teste, o usuário deve escolher a série a ser testada com o modelo previamente treinado. Deve-se tomar cuidado para não ocorrer o teste de uma série com um modelo treinado para uma série distinta. No ato do teste o usuário pode especificar, através de um campo de texto, a precisão do intervalo de confiança. Se o campo responsável pelo intervalo de confiança for deixado em branco, a ferramenta assumirá o valor padrão 50%.

Após a realização do teste é obtida uma série ainda normalizada. Para uma melhor visualização dos resultados é necessária a execução do processamento inverso da série. Durante o pré-processamento são guardados dados sobre a operação de modo que seja possível a aplicação inversa da operação, processamento inverso.

Capítulo 5

Experimentos e Resultados Obtidos

Este capítulo tem como finalidade a realização de experimentos usando a ferramenta desenvolvida e também redes neurais artificiais para que haja a possibilidade de uma comparação entre os métodos. Para o uso de redes neurais duas topologias foram escolhidas:

- Redes MLP; e
- Redes Elman.

Como explicado anteriormente, a topologia de Elman difere da topologia MLP pela existência de recorrência parcial, além das entradas da janela de tempo dada pelo usuário ela recebe também como entrada as saídas dos neurônios na camada escondida que ao se conectarem à própria camada escondida se torna a camada de contexto.

5.1 Metodologia de treinamento para SVM e redes Neurais

Para o treinamento das séries com redes neurais foi usado o simulador de redes neurais SNNS, *Stuttgart Neural Networks Simulator*. O algoritmo de aprendizagem escolhido para o treinamento com as redes neurais, tanto Elman quanto MLP, foi o *Resilient Backpropagation*, RPROP[32]. Foram realizadas 9 execuções para cada topologia variando-se o tamanho da janela de tempo e o número de neurônios na camada escondida. A inicialização dos pesos foi aleatória variando-se de -1 a 1. A avaliação da melhor, ou pior rede, foi feita com base em seu EMQ, quanto menor seu EMQ maior a capacidade de previsão da rede.

Dois critérios de parada para o treinamento com redes neurais foram considerados durante o treinamento, GL5 do Proben1[33], que interrompe o treino se o EMQ de validação da época atual for 5% maior que o menor EMQ obtido até o momento (de acordo com a equação 7) , e a

parada por número máximo de 5000 iterações. O treinamento pára antecipadamente se o critério GL5 for alcançado duas vezes, este critério permite que o treinamento seja interrompido assim que for percebida a perda da capacidade de generalização da rede treinada, evitando, dessa forma, *overfitting*.

$$100 * \left(\frac{EMQ_VALID_{ATUAL} - 1}{EMQ_VALID_{MIN}} \right) > 5 \quad \text{(Equação 7)}$$

Para o treinamento com SVMs foi utilizada a ferramenta desenvolvida, objeto deste trabalho, com a opção de treinamento com parâmetros otimizados. Foram utilizadas janelas de tempo de 2, 4, e 6.

As séries usadas para o treinamento com redes neurais, devido à metodologia de medida de desempenho adotada, foram divididas em três conjuntos distintos: treinamento, validação e teste. Diferentemente de Redes Neurais Artificiais, SVMs não necessitam do uso do conjunto de validação o que implica numa maior disponibilidade de dados para o treinamento e teste. A divisão das séries para o treinamento com RNAs foi de 50% para o conjunto de treinamento, 25% para validação e 25% para teste, enquanto que para SVMs a divisão foi 75% para o conjunto de treinamento e 25% para o de teste.

5.2 Experimentos com a série CEMENT

A base CEMENT dispõe de dados mensais da produção australiana de cimento Portland em milhares de toneladas no período de janeiro de 1956 a agosto de 1995. Essa base está disponível em 1.

A base CEMENT apresenta claramente características sazonais e de tendência crescente. Visualmente é possível estimar seu período de sazonalidade prestando-se atenção nos picos e vales do gráfico da série original. Por ser uma série de natureza comercial e também pela periodicidade de um pouco mais que 4 vales a cada cinquenta amostras, é razoável inferir que essa

¹ <http://www-personal.buseco.monash.edu.au/~hyndman/TSDL/>

série sofre de sazonalidade com períodos de 12 amostras. A figura 35 ilustra a série CEMENT original.

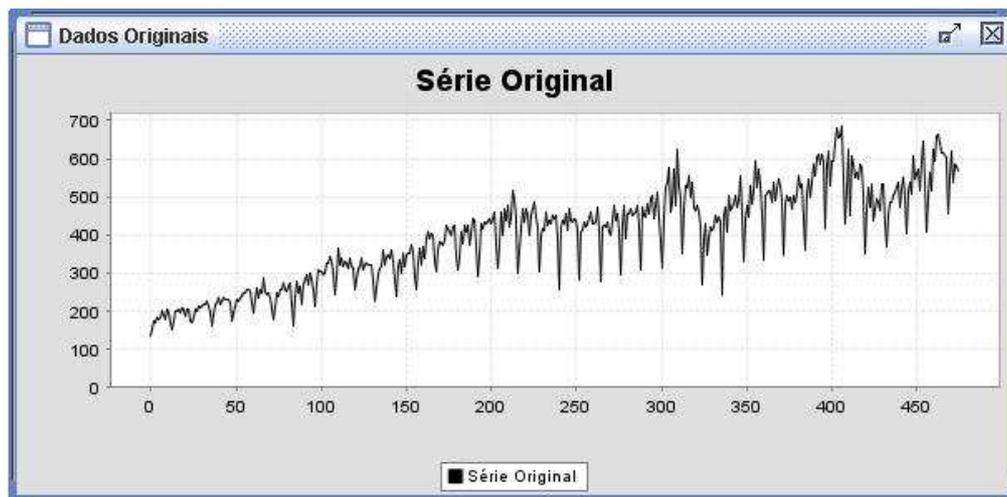


Figura 35 - Série CEMENT original

5.2.1 Pré-Processamento da série

De acordo com a análise da série original, figura 34, e de seus corelogramas a cada processamento, foram escolhidos três processamentos sobre os dados na sua preparação para o treinamento, foram eles:

- Diferenciação com janela de 12 – inicialmente foi identificada, através da análise do correlograma da série, uma sazonalidade com periodicidade 12;
- Diferenciação com janela de 1 – pela leitura do correlograma obtido da série após a diferenciação com janela 12 foi verificada apenas a presença de uma tendência simples; e
- Normalização – tendo a série estacionária, resta apenas a aplicação do operador de normalização.

A figura 36 ilustra o correlograma inicial da série enquanto que a figura 37 mostra o correlograma final, após o pré-processamento.

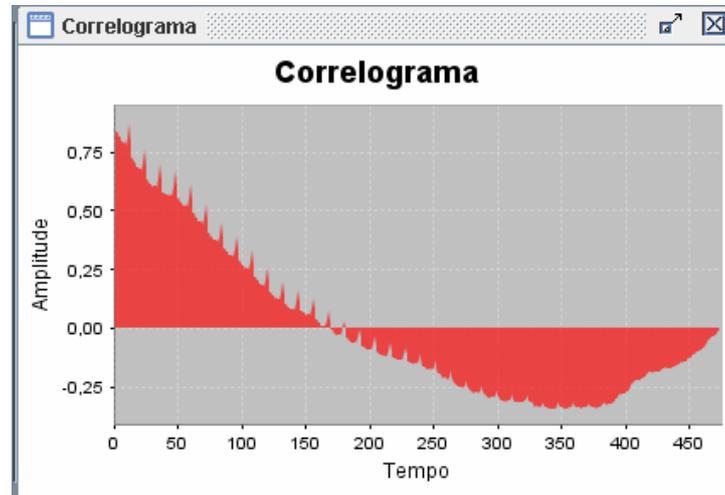


Figura 36 - Correlograma da série CEMENT Original

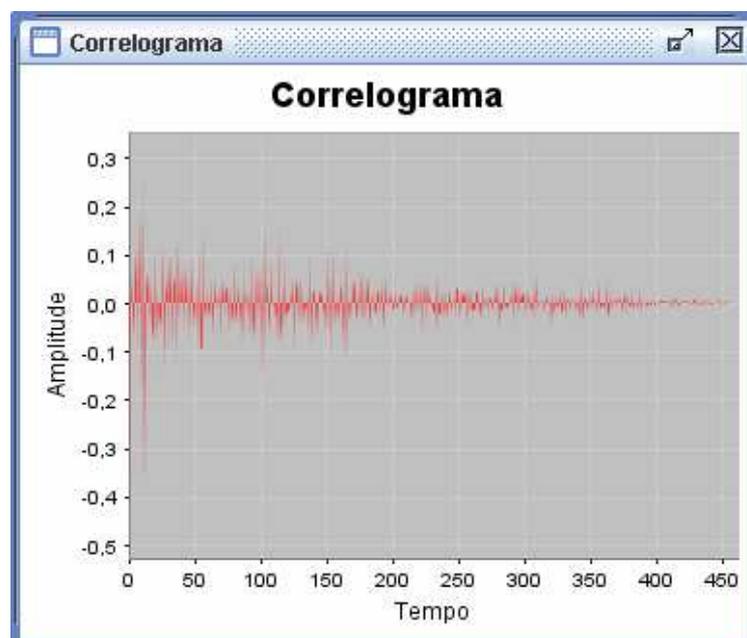


Figura 37 - Correlograma final da série CEMENT

Terminado o processamento da série CEMENT, tem início a divisão dos dados para treinamento. No caso do treinamento com as topologias Elman e MLP o conjunto de dados foi

dividido em 50% para treinamento, 25% para validação e 25% para testes. No método estatístico SVM não há a necessidade de um conjunto de validação. Nesse caso o conjunto de dados é dividido em 75 % para treinamento e 25 % para testes.

A tabela 2 mostra o número de amostras por geração de padrões de treinamento, teste e validação para a série CEMENT.

Tabela 2 - Distribuição dos dados para a série CEMENT

Janela	Padrões de treinamento	Padrões validação	Padrões Teste	Técnica
2	228	115	116	MLP e Elman
4	227	115	115	MLP e Elman
6	226	114	115	MLP e Elman
2	344	-	117	SVM
4	343	-	116	SVM
6	341	-	116	SVM

5.2.2 Resultados Obtidos para a série CEMENT

As tabelas 3, 4 e 5 mostram os resultados obtidos para o treinamento da série CEMENT com as topologias, tendo as linhas que representam o melhor resultado para cada técnica de previsão uma cor de plano de fundo em uma tonalidade de cinza mais escura, de redes neurais artificiais, MLP e Elman e com o método estatístico SVM, respectivamente. Através da observação das tabelas 3, 4 e 5 é possível verificar um desempenho levemente melhor na topologia Elman, vindo em seguida o método SVM e a topologia MLP, de redes neurais. O critério para avaliação de desempenho escolhido foi o menor EMQ para o resultados dos testes realizados com os conjuntos de testes para o treinamento correspondente. O EMQ, apesar de ser um bom indicativo, não é suficiente para a indicação do melhor ou pior desempenho pois o EMQ depende da distribuição do conjunto de testes e o conjunto de testes pode não ser o ideal para a avaliação do desempenho do treinamento.

Tabela 3 - Resultado dos treinamentos realizados para topologia MLP e para a série CEMENT

Janela de tempo	NCE(*)	EMQ Treinamento	EMQ Validação	EMQ Teste	Ciclos de Treinamento
** 2	2	0.00655062	0.0240534	0.0253515	5005
2	4	0.00653953	0.0244727	0.025954	5005
2	6	0.00654889	0.0240599	0.0253817	5005
4	2	0.00776836	0.0295684	0.0317212	55
4	4	0.00558661	0.0237829	0.026904	5005
4	6	0.00561878	0.0236843	0.0265039	5005
6	2	0.00551618	0.0234847	0.0278759	5005
6	4	0.00559381	0.0234745	0.0279213	5005
6	6	0.0055029	0.023576	0.027914	5005

(*) NCE = Neurônios na camada escondida

(**) Resultado do melhor treinamento realizado

Tabela 4 - Resultado dos treinamentos realizados para topologia Elman e para a série CEMENT

Janela de tempo	NCE(*)	EMQ Treinamento	EMQ Validação	EMQ Teste	Ciclos de Treinamento
2	2	0.00658719	0.0237821	0.0244857	5005
2	4	0.00658043	0.0238964	0.0251728	5005
2	6	0.00656893	0.023922	0.0251135	5005
4	2	0.00590109	0.0238571	0.0262595	270
4	4	0.00592834	0.0237854	0.0236379	345
4	6	0.00592349	0.0239965	0.0271455	500
** 6	2	0.00564129	0.0239945	0.0234337	380
6	4	0.00557286	0.0244595	0.0265061	5005
6	6	0.00557618	0.0248571	0.0267658	5005

(*) NCE = Neurônios na camada escondida

(**) Resultado do melhor treinamento realizado

Tabela 5 - Resultado dos treinamentos realizados utilizando SVMs para a série CEMENT

Janela de tempo	EMQ Treinamento	EMQ Teste
2	0.0120079	0.0262863
* 4	0.0114179	0.0253097
6	0.0112126	0.0258869

(*) Resultado do melhor treinamento realizado

5.2.3 Gráficos das séries Previstas

As figuras 38, 39 e 40 mostram graficamente os resultados das previsões dos conjuntos de testes da série CEMENT, sendo a figura 40 gerada usando-se as funcionalidades da ferramenta obtida. A linha verde representa a série original enquanto que a vermelha representa o resultado da previsão. Pela visualização dos gráficos nota-se uma maior similaridade das séries prevista e original nas previsões MLP e SVM que na previsão Elman.

Para o treinamento com SVMs efetuado pela ferramenta foi usada a opção dos melhores EMQs. Para essa série foram obtidos, na execução do script Python, sete conjuntos de parâmetros de treinamento. A quantidade de parâmetros obtidos usada foi estipulada em 6 pelo usuário. Para a criação do intervalo de confiança foi usada uma porcentagem de 95 % de confiança.

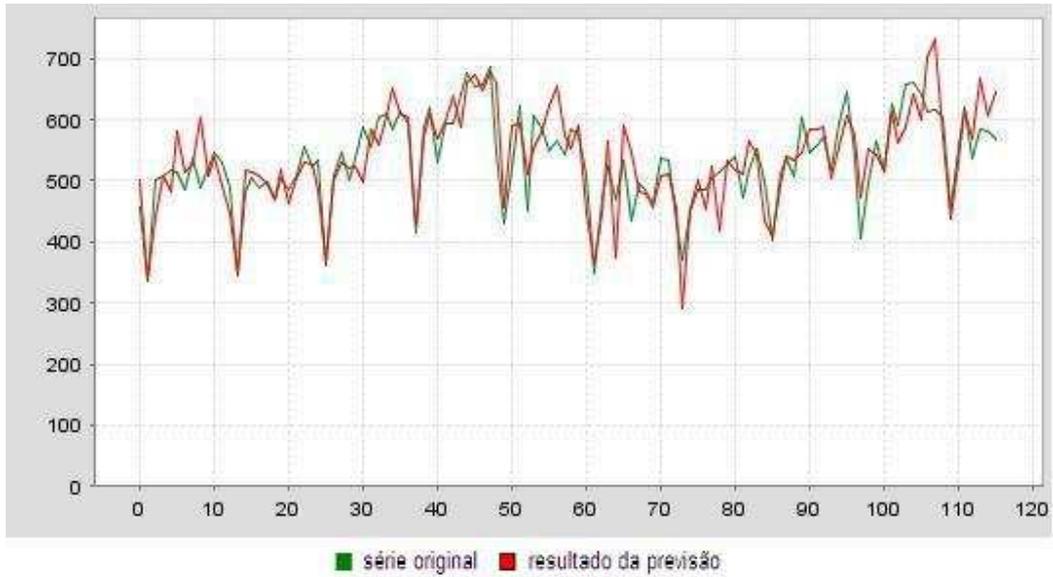


Figura 38 - Representação gráfica do melhor teste MLP - CEMENT



Figura 39 - Representação gráfica do melhor teste Elman - CEMENT

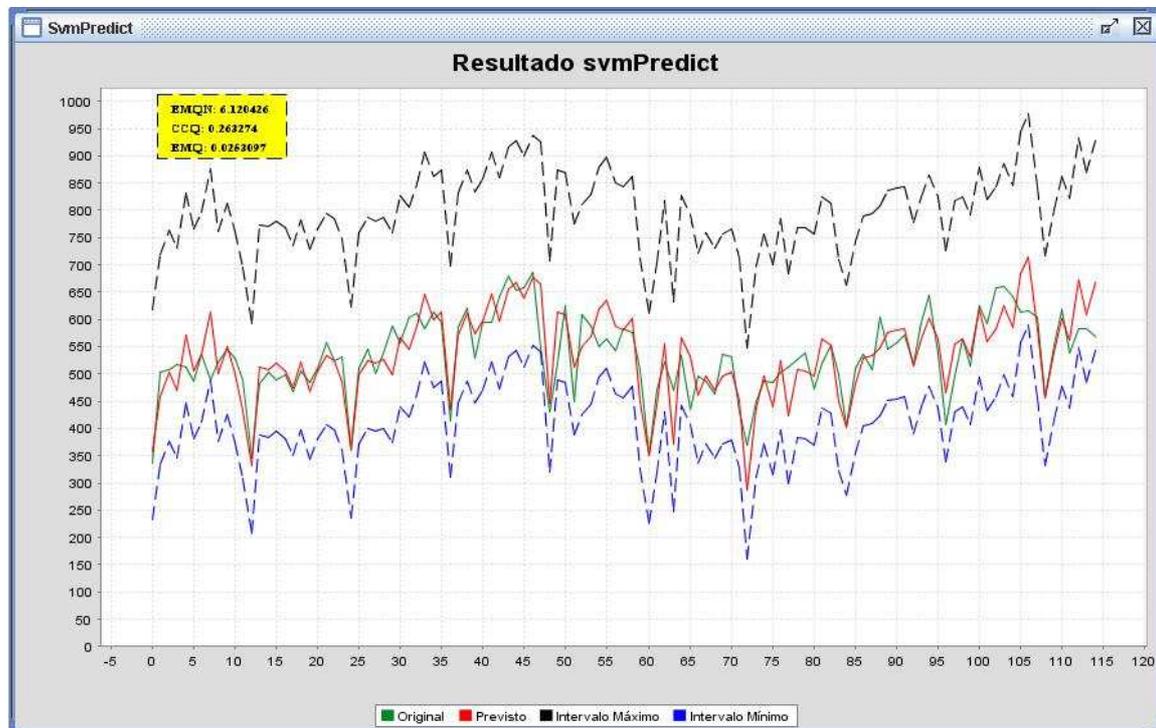


Figura 40 - Representação gráfica do melhor teste SVM - CEMENT

5.3 Experimentos com a série BOVESPA

No Brasil, dentre as Bolsas de Valores existentes a de maior importância no cenário econômico é a BOVESPA, Bolsa de Valores do Estado de São Paulo. Nela se concentra a negociação dos títulos das principais empresas do país. Seu índice, o IBV, é um dos indicadores mais significativos de todo o mercado brasileiro. O aumento na participação dos pequenos e médios investidores nas transações, respondendo hoje por aproximadamente 27,5% do total [31], indica o quanto pode ser vantajoso o investimento de recursos em título de ações nas Bolsas de Valores.

A série original pode ser obtida no website do Banco Central do Brasil ². Esta série conta com 262 amostras mensais coletadas no intervalo de julho de 1983 a abril de 2005. Devido à total irregularidade da série no período de julho de 1983 a janeiro de 1997 o desempenho do

² <http://www4.bcb.gov.br/pec/series/port/ls.asp?tp=1&g=113&PPer=>

treinamento pode ser prejudicado, porém, esse tipo de tratamento não vem ao caso no nosso contexto. Uma forma de driblar a irregularidade da série seria o uso apenas das amostras compreendidas no intervalo de tempo entre março de 1997 e abril de 2005. A figura 1 ilustra a série Bovespa original.

5.3.1 Pré-Processamento da série

De acordo com a análise da série original, figura 1, e de seus correlogramas a cada processamento, foram escolhidos quatro processamentos sobre os dados na sua preparação para o treinamento, foram eles:

- Diferenciação com janela de 30 – pela difícil análise de efeitos presentes na série, o valor 30 da janela de diferenciação foi obtido de maneira empírica;
- Diferenciação com janela de 12 – nesse ponto do pré-processamento, foram encontradas, pela análise do resultado da diferenciação anterior, características de sazonalidade com periodicidade 12;
- Diferenciação com janela de 1 – nesse ponto do pré-processamento foram identificadas apenas características de tendência simples na série; e
- Normalização - tendo a série estacionária, resta apenas a aplicação do operador de normalização.

A figura 41 mostra o correlograma inicial da série enquanto que a figura 42 ilustra o corelograma final, após o pré-processamento.

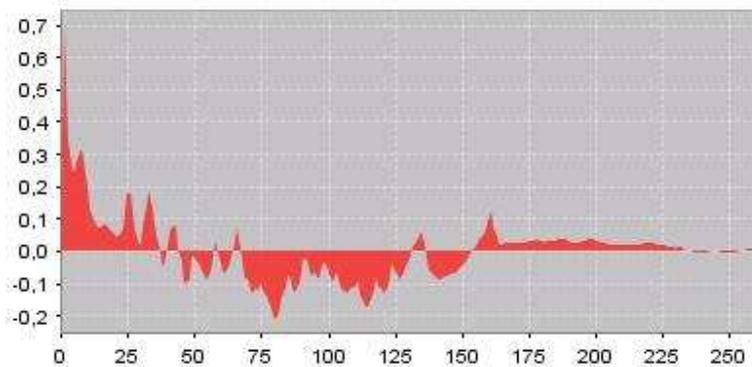


Figura 41 - Correlograma inicial - Bovespa

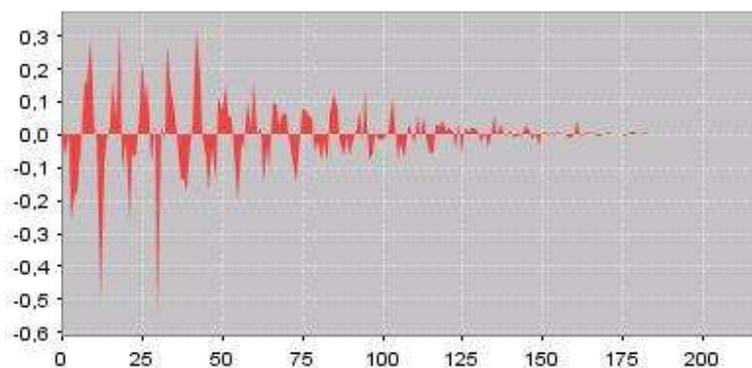


Figura 42 - Correlograma final - Bovespa

Verificada a eficiência do processamento da série Bovespa, tem início a divisão dos conjuntos de dados para treinamento, teste e validação. No caso de Elman e MLP o conjunto de dados foi dividido em 50% para treinamento, 25% para validação e 25% para testes. Na SVM não há a necessidade de um conjunto de validação, nesse caso o conjunto de dados é dividido em 75 % para treinamento e 25 % para testes.

Posteriormente, a partir da série no estado atual, foram criados seis novos arquivos compatíveis para treinamento de acordo com a tabela 6, onde cada linha é uma nova configuração da série original.

Tabela 6 - Distribuição dos dados para a série Bovespa

Janela	Padrões de treinamento	Padrões validação	Padrões Teste	Técnica
2	108	53	54	MLP e Elman
4	107	53	53	MLP e Elman
6	106	53	52	MLP e Elman
2	161	-	56	SVM
4	160	-	55	SVM
6	158	-	55	SVM

5.3.2 Resultados Obtidos para a série Bovespa

As tabelas 7, 8 e 9 mostram os resultados obtidos para o treinamento da série Bovespa com as topologias, de redes neurais artificiais, MLP e Elman e com o método estatístico SVM, respectivamente. Através da observação das tabelas 7, 8 e 9 é possível verificar um desempenho levemente melhor na topologia Elman, porém, foram testadas 9 configurações diferentes para a obtenção desses resultados assim como foi necessária a geração manual dos conjuntos de treinamento, validação e teste. O método SVM precisou apenas de três configurações distintas para a obtenção de um resultado similar.

Tabela 7 Resultado dos treinamentos realizados para topologia MLP e para a série BOVESPA

Janela de tempo	NCE (*)	EMQ Treino	EMQ Validação	EMQ Teste	Ciclos de Treinamento
2	2	0.039249	0.020154	0.0033528	25
2	4	0.039248	0.020155	0.0033528	20
2	6	0.039248	0.020155	0.0033527	20
4	2	0.036917	0.019971	0.0035426	250
4	4	0.034940	0.019728	0.0033891	320
4	6	0.034563	0.019694	0.0034571	340
6	2	0.031450	0.022463	0.0020187	100
** 6	4	0.031047	0.022368	0.0017095	70
6	6	0.029556	0.021756	0.0018092	40

(*) Neurônios na camada escondida

(**) Resultado do melhor treinamento realizado

Tabela 8 - Resultado dos treinamentos realizados para topologia Elman e para a série BOVESPA

Tam. Janela	NCE (*)	EMQ Treino	EMQ Validação	EMQ Teste	Ciclos
2	2	0.040199	0.019010	0.0034650	3950
2	4	0.040528	0.019000	0.0034202	4125
2	6	0.040407	0.018999	0.0034355	5565
4	2	0.036753	0.020285	0.0037197	3085
4	4	0.033007	0.020860	0.0037680	3170
4	6	0.033000	0.020837	0.0037414	4535
** 6	2	0.027084	0.021588	0.0013084	60
6	4	0.026662	0.022778	0.0017030	2695
6	6	0.025970	0.023672	0.0017945	4625

(*) NCE = Neurônios na camada escondida

(**) Resultado do melhor treinamento realizado

Tabela 9 - Resultado dos treinamentos realizados utilizando SVMs para a série BOVESPA

Tam. Janela	EMQ Treino	EMQ Teste
2	0.0329034	0.00319384
* 4	0.029065	0.0000382982
6	0.0248841	0.00145838

(*) Resultado do melhor treinamento realizado

5.3.3 Gráficos das séries previstas

As figuras 43, 44 e 45 mostram graficamente os resultados das previsões com os conjuntos de testes, sendo a figura 45 gerada usando-se as funcionalidades da ferramenta obtida. A linha verde representa a série original enquanto que a vermelha representa o resultado da previsão. Para os gráficos das previsões efetuadas pelas topologias MLP e Elman, de redes neurais, o processamento inverso foi realizado de forma manual enquanto que para o treinamento e teste com SVMs todas as operações foram feitas de forma automáticas.

O treinamento escolhido para a realização da experiência foi o treinamento com os “N melhores EMQs”. Todos os parâmetros encontrados foram usados o que gerou sete treinamentos e testes gerando um conjunto de erros de 7 vezes o número de amostras para o conjunto de testes. Para a criação do intervalo de confiança foi usada uma porcentagem de 95 % de confiança.



Figura 43 - Representação gráfica do melhor teste MLP - Bovespa



Figura 44 - Representação gráfica do melhor teste Elman - Bovespa

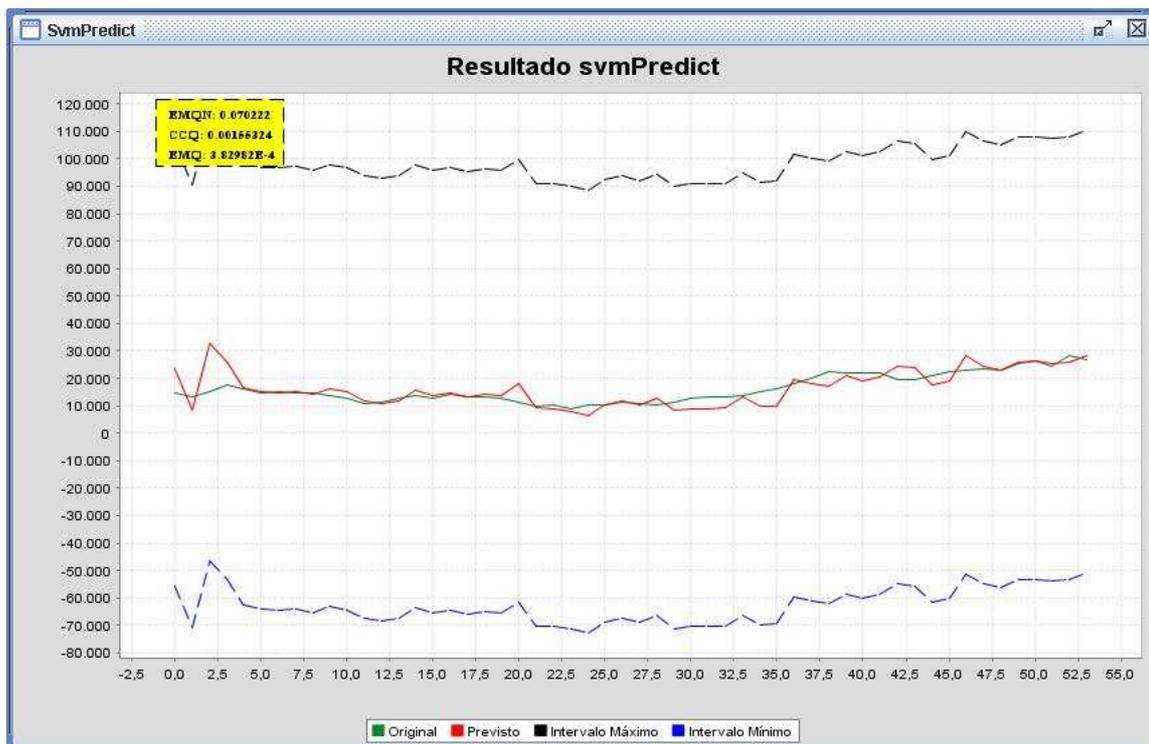


Figura 45 - Representação gráfica de melhor teste SVM utilizando intervalos de confiança robustos com uma confiança de 95% - Bovespa

Capítulo 6

Conclusões e trabalhos futuros

6.1 Conclusões

Ferramentas capazes de tratar dados para técnicas de análise de séries temporais de forma interativa e simples não são comumente encontradas na literatura ou na Internet, o que existe são ferramentas para previsão de séries temporais, aptas ao treinamento e teste de modelos para previsão, porém, essas ferramentas não são de fácil manuseio para um usuário sem o devido conhecimento de sua utilização.

A ferramenta gerada mostrou-se capaz de atuar de forma eficiente nas diversas etapas necessárias à realização de previsões de séries temporais. O módulo de pré-processamento das séries realiza este trabalho maçante, e muito susceptível a erros, de forma automática e clara poupando ao usuário tempo e preocupação na validação dos pré-processamentos realizados. A criação dos arquivos de treinamento e teste, como última etapa do pré-processamento das séries, consiste em um grande benefício pois séries com elevado número de amostras, ou mesmo um número mediano, exigem do usuário tempo, paciência e concentração na criação dos arquivos, assim como conhecimentos específicos do simulador a ser usado, no caso LibSVM.

O módulo de treinamento e testes usa de forma transparente a biblioteca LibSVM, disponibilizando ao usuário várias possibilidades de treinamento e teste. O treinamento otimizado se mostrou eficiente e de simples realização, enquanto que o treinamento com “N melhores EMQs” se mostrou bastante interessante para a criação mais eficiente de intervalos de confiança robustos.

A disponibilização de intervalos de confiança robustos consiste de uma inovação apresentada pela ferramenta pois faz com que não apenas os dados sejam disponibilizados na forma de gráficos, mas também indicativos de confiança no modelo gerado para previsão.

Foram realizados experimentos com duas séries temporais de grande importância no cenário econômico de seus países, a série de índices mensais da Bovespa e a série de vendas de cimento Portland na Austrália. Nos dois casos a ferramenta se comportou muito bem, ajudando na realização de uma tarefa, que um usuário com certa experiência levaria, provavelmente, mais que uma hora para a realização, em menos de 5 minutos.

Foi demonstrado que a ferramenta gerada cumpriu com todas as expectativas propostas mostrando ser de grande utilidade no que se refere à realização da tarefa de previsão de séries temporais e atividades correlatas.

6.2 Trabalhos Futuros

Como indicações para a continuação desse trabalho, seguem, abaixo, listadas cinco sugestões de possíveis trabalhos futuros:

1. Implementação de uma forma mais automática de pré-processamento de dados. A geração do correlograma seria responsável, também, pela indicação de quais efeitos a série sofre e qual janela, para diferenciação, usar simplesmente pela geração do correlograma da série em seu estado corrente. À medida que a ferramenta não conseguisse identificar efeitos sazonais na série ela passaria a buscar por tendência e sugerir a eliminação da mesma.
2. Incorporação de novas medidas de desempenho do modelo gerado para previsão. Atualmente a ferramenta conta com três diferentes medidas de desempenho (EMQ, NEMQ e CCQ, anteriormente explicadas), porém, a avaliação de um modelo é uma tarefa bastante crítica e quanto maior for o número de indicativos de desempenho da previsão gerada maior será a segurança, do usuário, da obtenção de um modelo satisfatório para aquele determinado problema.
3. de qual a próxima tarefa a ser realizada para tratamento dos dados sugerindo dados como janelas de diferenciação, assim como, também, o término do pré-processamento.
4. A integração com outros simuladores existentes, como o SNNS, agregaria grande valor à ferramenta, pois disponibilizaria ao usuário uma maneira de comparar os resultados obtidos pelos métodos utilizados. Com base nos resultados obtidos pelos diversos modelos utilizados, o usuário poderia optar pelo que melhor se adequou à realização do problema.

5. A disponibilização de métodos não só para previsão de séries mas também para classificação de padrões tornaria a ferramenta mais geral, porém, poderia torná-la uma ferramenta altamente didática no ensino de conceitos de inteligência artificial e mais especificamente redes neurais e métodos afins.

Bibliografia

- [1] Guoqiang Zhang, B. Eddy Patuwo, Michael Y. Hu. Forecasting with artificial neural networks: The state of art. *International Journal of Forecasting*, Elsevier, Páginas 35-62, 1998.
- [2] A. P. Braga, A. P. L. F. Carvalho, T. B. Ludermir. *Redes Neurais Artificiais: Teoria e Aplicações*, Rio de Janeiro: Livros Técnicos e Científicos, 2000.
- [3] A. L. I. Oliveira, F.B.L.Neto e S.R.L.Meira. Novelty Detection in Time Series by Neural Networks Forecasting with Robust Confidence Intervals. In 8 Simpósio Brasileiro de Redes Neurais, São Luiz, Maranhão, 2004.
- [4] V. David Sánchez A. Advanced support vector machines and kernel methods. *Neurocomputing*, Volume 55, Fascículos 1-2, Páginas 5-20, Setembro 2003.
- [5] N. Cristianini e J. Shawe-Taylor. *An Introduction to Support Vector Machines and Other Kernel-based Learning Methods*. Cambridge University Press ISBN:0521780195, 2000.
- [6] LIBSVM. LIBSVM - A library for support vector machines. Disponível em: <http://www.csie.ntu.tw>. Acesso em: 08 abril 2005.
- [7] LIBSVM. A practical guide to SVM classification. Disponível em: <http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf>. Acesso em: 08 abril 2005.

- [8] P. A. Morettin, C. M. Toloí. Métodos Quantitativos – Séries Temporais. Editora Atual, São Paulo. 1987.
- [9] Tukey, J.W. We Need Both Exploratory and Confirmatory. *The American Statistician*, 34:23-25, 1980.
- [10] C. Chatfield. *The Analysis of Time Series – An Introduction*. Chapman & Hall, quarta edição, 1989.
- [11] P. A. Moretin, C. M. Toloí. *Previsão de séries temporais*. Atual. São Paulo. 1985.
- [12] Box, G.E. e Jenkins, G.M. *Time series analysis, forecasting and control - Segunda Edição*. Prentice Hall San Francisco, 1976.
- [13] Brown, R. G. *Smoothing, Forecasting and prediction of discrete time series*. Englewood Cliffs. N. J.: Prentice Hall, 1962.
- [14] Linear Regression. Disponível em:
<http://cne.gmu.edu/modules/dau/stat/regression/linregsn/linregsn_frm.html>. Acesso em:
12 outubro 2005.
- [15] Zwillinger, D. (Ed.). *CRC Standard Mathematical Tables and Formulae*. Boca Raton, FL: CRC Press, p. 223, 1995.
- [16] A. Zell et al. *SNNS - Stuttgart Neural Network Simulator, User Manual, Version 4.2*. Disponível em:
<http://www-ra.informatik.uni-tuebingen.de/downloads/SNNS/SNNSv4.2.Manual.pdf> .
Acesso em: 05 de julho de 2005.

- [17] Rumelhart, D., Hinton, G., and Williams, R. Learning internal representations by error propagation. In *Parallel Distributed Processing*, capítulo 8. MIT Press, Cambridge, MA, 1986.
- [18] Williams, R. and Zipser, D. A learning algorithm for continually running fully recurrent neural networks. *Neural Computation*, 1(2):270-280, 1989.
- [19] Hertz, J., Krogh, A., and Palmer, R. G. *Introduction to the Theory of Neural Computation*. Addison-Wesley Publishing Company, Inc., Redwood City, CA, 1991.
- [20] Fahlman, S. E. The recurrent cascade-correlation architecture. In Lippmann, R. P., Moody, J. E., and Touretzky, D. S. *Advances in Neural Information Processing Systems 3*, paginas 190-196. Morgan Kaufmann, Denver, CO, 1991.
- [21] S. Haykin, *Neural Networks - A Comprehensive Foundation*. Prentice Hall, Segunda edição ed., 1998.
- [22] McCulloch, Warren & Pitts, Walter. A logical calculus of the ideas immanent in nervous activity, *Bulletin of Mathematical Biophysics*, volume 5, 1943.
- [23] G. Cybenko. Approximation by superpositions of a sigmoid function. *Mathematics of Control, Signals and Systems*, 2:303-314, 1989.
- [24] G. Cybenko. Continuous valued neural networks with two hidden layers are sufficient. Technical report, Department of Computer Science, Tufts University, 1988.

- [25] Suykens J. A Short Introduction to Support Vector Machines and Kernelbased Learning. ESANN 2003, Bruges, Abril 2003.
- [26] V. Vapnik and A. Lerner. Pattern recognition using generalized portrait method. Automation and Remote Control, 24: 774.780, 1963.
- [27] V. Vapnik and A. Chervonenkis. A note on one class of perceptrons. Automation and Remote Control, 1964.
- [28] A.J. Smola, B. Scholkopf . A tutorial on support vector regression, NeuroCOLT Technical Report NC-TR-98-030, Royal Holloway College, University of London, UK, 1998.
- [29] K.-R. Muller, A. Smola, G. Ratsch, B. Scholkopf, J. Kohlmorgen, and V. Vapnik. Predicting time series with support vector machines. In W. Gerstner, A. Germond, M. Hasler, e J.-D. Nicoud, editors, Artificial Neural Networks ICANN'97, Páginas 999 - 1004, Springer Lecture Notes in Computer Science, Vol. 1327. Berlin, 1997.
- [30] H. Drucker, C. J. C. Burges, L. Kaufman, A. Smola, and V. Vapnik. Support vector regression machines. In M. C. Mozer, M. I. Jordan, and T. Petsche, editores, Advances in Neural Information Processing Systems 9, páginas 155.161, 1997. MIT Press, 1997.
- [31] Sítio da BOVESPA, Estatísticas da BOVESPA – dados de 2004, Obtida via Internet. Último acesso em: 25 de maio de 2005, http://www.comunique-e.com.br/images/saladeimprensa/bovespa/planilha/planilha_bovespa_20050525.pdf.
- [32] M. Riedmiller e H Braun. A direct adaptative method for fast backpropagation learning: The RPROP algorithm. IEEE International Conference on Neural Networks 1993 (ICNN 93), 1993.

- [33] L. Prechelt. Proben1 – a set of neural networks benchmark problems and benchmarking rules. Technical Report 21/94, Universitt Karlsruhe, Alemanha, 1994.