



Universidade Federal de Pernambuco
Centro de Informática

Pós-graduação em Ciência da Computação

**META-APRENDIZADO PARA SELEÇÃO
AUTOMÁTICA DE MODELOS DE SÉRIES
TEMPORAIS**

Renata Maria de Souza

DISSERTAÇÃO DE MESTRADO

Recife

01 de agosto de 2010

Universidade Federal de Pernambuco
Centro de Informática

Renata Maria de Souza

**META-APRENDIZADO PARA SELEÇÃO AUTOMÁTICA DE
MODELOS DE SÉRIES TEMPORAIS**

*Trabalho apresentado ao Programa de Pós-graduação em
Ciência da Computação do Centro de Informática da Uni-
versidade Federal de Pernambuco como requisito parcial
para obtenção do grau de Mestre em Ciência da Com-
putação.*

Orientador: *Prof. Dr. Ricardo Bastos Cavalcante
Prudêncio*

Recife
01 de agosto de 2010

Catálogo na fonte
Bibliotecária Jane Souto Maior, CRB4-571

Souza, Renata Maria de
Meta-aprendizado para seleção automática de
modelos de séries temporais / Renata Maria de Souza -
Recife: O Autor, 2010.
xi, 58 folhas : il., fig., tab.

Orientador: Ricardo Bastos Cavalcante Prudêncio.
Dissertação (mestrado) - Universidade Federal de
Pernambuco. CIn, Ciência da Computação, 2010.

Inclui bibliografia.

1. Inteligência artificial. 2. Aprendizado de máquina. 3. Meta
aprendizado. 4. Séries temporais. I. Prudêncio, Ricardo Bastos
Cavalcante (orientador). II. Título.

006.31

CDD (22. ed.)

MEI2011 - 113

4

Dissertação de Mestrado apresentada por Renata Maria de Souza à Pós-Graduação em Ciência da Computação do Centro de Informática da Universidade Federal de Pernambuco, sob o título "Meta-aprendizado para Seleção Automática de Modelos de Séries Temporais", orientada pelo Prof. Ricardo Bastos Cavalcante Prudêncio e aprovada pela Banca Examinadora formada pelos professores:

Prof. Cleber Zanchettin
Centro de Informática / UFPE

Prof. Messer Jorge Silva Valença
Departamento de Sistemas Computacionais / UFPE

Prof. Ricardo Bastos Cavalcante Prudêncio
Centro de Informática / UFPE

Visto e permitida a impressão;
Recife, 17 de agosto de 2010.

Prof. Nelson South Rosa
Coordenador da Pós-Graduação em Ciência da Computação do
Centro de Informática da Universidade Federal de Pernambuco.

Ao meu pai, por me ensinar com seu exemplo uma grande lição: recomeçar.

AGRADECIMENTOS

Em primeiro lugar, gostaria de agradecer a Deus, por ter me dado tanto amor e dons especiais: meus pais -Ayres e Jamile, minha irmã -Tatiana, meu namorado -Iuri, as meninas e os membros de simpatizantes do Movimento Regnum Christi, os meus orientadores espirituais - Padre Javier Sicilia, LC e Miriene Fernandes, por sempre mostrarem a vontade de Deus para minha vida de forma tão clara, simples e equilibrada. Sem o carinho, dedicação e paciência de cada um jamais seria capaz de realizar esse trabalho.

Em especial gostaria de agradecer também ao meu orientador Ricardo Prudêncio por me ensinar o que é ser Mestre, pela paciência, profissionalismo e por ser um exemplo como pessoa.

"A alegria nasce do amor"

—BENTO XVI (13 de dezembro de 2009)

RESUMO

Meta-Aprendizado tem crescido nos últimos anos devido ao desenvolvimento de assistentes para seleção de algoritmos, com o desafio de prever quando um algoritmo de aprendizagem é mais adequado do que outro a partir das características dos problemas sendo abordados. O meta-aprendizado surge originalmente para auxiliar a seleção de algoritmos em problemas de aprendizagem de máquina e mineração de dados, particularmente em classificação e regressão. Em anos recentes, meta-aprendizado tem sido extrapolado para seleção de algoritmos em outros domínios de aplicações, como sistemas de planejamento, otimização, bioinformática e previsão de séries temporais. Nesse trabalho, focamos particularmente, em meta-aprendizado no contexto de previsão de séries temporais que tem sido usado em diferentes contextos para diminuir riscos na tomada de decisão. Estudos foram realizados para seleção de modelos de previsão aplicados às séries anuais da *M3-competition*. Nesses estudos, diferentes algoritmos foram utilizados no meta-aprendizado como o algoritmo kNN, árvores de decisão e *support vector machines*. Os resultados mostraram que os algoritmos de aprendizado de fato são capazes de prever os melhores modelos de previsão a partir das características das séries temporais.

Palavras-chave: Aprendizado de Máquina, Meta-Aprendizado, Séries Temporais

ABSTRACT

Meta-Learning has grown in the last few years due to assists' development to algorithms selection with the challenge of predicting when a learning algorithm is more suitable than others from the problems characteristics that is the being addressed. The meta-learning appears originally to improve the algorithms selection in problems of machine learning and data mining, particularly in classification and regression. In recent years, meta-learning has been exploited for algorithms selection in other applications fields such as systems planning, optimization, bioinformatics, and time series forecasting. In this work, we focus particularly on meta-learning in the context of forecasting time series that has been used in different contexts to reduce risks in decision making. Studies were performed to select forecasting models applied to the M3-competition annual series. In these studies, different algorithms were used in the meta-learning algorithm such as kNN, decision trees and support vector machines. The results showed that the learning algorithms are indeed able to predict the best forecasting model from the characteristics of time series.

Keywords: Machine Learning, Meta-Learning, Time Series

SUMÁRIO

Capítulo 1—Introdução	1
1.1 Contexto da Dissertação	2
1.2 Objetivo da Dissertação	3
1.3 Organização da Dissertação	3
Capítulo 2—Meta-Aprendizado para Recomendação de Algoritmos	4
2.1 Meta-Aprendizado	4
2.2 Recomendação de Algoritmos com Meta-Aprendizado	6
2.3 Arquitetura	7
2.3.1 Modo de Aquisição de Conhecimento	7
2.3.2 Modo de Sugestão	8
2.4 Técnicas de Meta-Aprendizagem	10
2.4.1 Caracterização de Conjunto de Dados	10
2.4.2 Mapeamento de Conjunto de Dados para Modelos de Previsão	15
2.4.3 Combinação de meta-aprendizes	19
Capítulo 3—Modelo de Séries Temporais	24
3.1 Modelos Candidatos	25
3.1.1 <i>Random Walk</i>	25
3.1.2 Alisamento Exponencial de Holt	26
3.1.3 Auto-Regressivo de Box-Jenkins	27
3.2 Avaliação dos Modelos de Previsão de Séries Temporais	27
3.2.1 Medidas de Erro	28
3.2.2 Métodos de Amostragem	28
3.2.3 Bases de Experimentos	29
3.3 Trabalhos Relacionados	30

Capítulo 4—Análise Experimental	37
4.1 Construção dos Meta-Exemplos	37
4.2 Algoritmos Avaliados	38
4.3 Estudo de Caso I	40
4.3.1 Análise dos Resultados	40
4.4 Estudo de Caso II	44
4.4.1 Análise dos Resultados	44
Capítulo 5—Conclusão	48
5.1 Contribuições da Dissertação	49
5.2 Limitações e Trabalhos Futuros	50

LISTA DE FIGURAS

2.1	Sistema de Meta-Aprendizagem: Modo de Aquisição de Conhecimento . .	8
2.2	Sistema de Meta-Aprendizagem: Modo de Sugestão	9
2.3	Meta-Decision Tree	22
2.4	Design Básico de um CAMLET	23
4.1	Desempenho (precisão) dos Algoritmos Avaliados	42

LISTA DE TABELAS

3.1	Revisão de Literatura de Seleção de Modelo de Séries Temporais	36
4.1	<i>Ranking</i> dos algoritmos avaliados	41
4.2	Avaliação dos Modelos de Previsão pelas Medidas de Erro	42
4.3	Distribuição das Classes	43
4.4	Horizonte 1	44
4.5	Horizonte 2	45
4.6	Horizonte 3	45
4.7	Horizonte 4	46
4.8	Horizonte 5	46
4.9	Horizonte 6	47

INTRODUÇÃO

Meta-aprendizado pode ser definido como o estudo de métodos que exploram meta-conhecimento para obter soluções eficientes para adaptação de processos de aprendizagem de máquina [BGCSV08]. Meta-conhecimento tem sido usado para adaptar sistemas de aprendizado ao longo do tempo, à medida que os sistemas são sucessivamente aplicados em diferentes problemas. Meta-conhecimento será usado então para resolver novos problemas a partir da experiência adquirida na resolução de problemas passados.

Meta-aprendizado pode ser usado para auxiliar a definição de diferentes etapas da construção de sistemas de aprendizado como pré-processamento de dados, seleção de algoritmos de aprendizagem, combinação de algoritmos de aprendizagem, ajuste de parâmetros dos algoritmos, dentre outros. Estudos sobre meta-aprendizado têm crescido nos últimos anos devido principalmente ao desenvolvimento de assistentes para seleção de algoritmos, com o desafio de prever quando um algoritmo de aprendizagem é mais adequado do que outro a partir das características dos problemas sendo abordados.

No contexto acima, um *meta-exemplo* de treinamento é gerado a partir de um experimento comparativo de um conjunto de algoritmos candidatos em um dado problema de aprendizado. Cada meta-exemplo armazena: (1) as características de um dado problema (em especial, características dos dados como número de atributos, correlações dos dados, etc.); e (2) o melhor algoritmo avaliado para o problema (aquele de melhor desempenho experimental). A partir de um conjunto de tais meta-exemplos, um meta-algoritmo é usado então para relacionar características dos problemas e o melhor algoritmo. O meta-conhecimento adquirido a partir dos meta-exemplos é usado então para prever os melhores algoritmos para problemas futuros.

Meta-aprendizado foi proposto e avaliado originalmente para seleção de algoritmos de classificação e regressão, dentro da comunidade de aprendizagem supervisionada. Em anos recentes, meta-aprendizado tem sido extrapolado para seleção de algoritmos em outros domínios de aplicação, como sistemas de planejamento, otimização, bioinformática e também previsão de séries temporais. Esses domínios têm em comum a disponibilidade de algoritmos que competem para resolver problemas e a suposição de que nenhum al-

goritmo pode ser considerado o melhor independente das características dos problemas. Como veremos, nessa dissertação foi focada na aplicação de meta-aprendizado especificamente para o domínio de seleção de modelos de séries temporais.

1.1 CONTEXTO DA DISSERTAÇÃO

Previsões de séries temporais é uma área de pesquisa de grande interesse na atualidade. Vários modelos e algoritmos têm sido desenvolvidos e estudados para seleção ou combinação de modelos de séries temporais. No entanto, existem ainda muitas incertezas sobre a seleção do melhor modelo de previsão para um determinado problema.

Diversas abordagens podem ser utilizadas para selecionar o melhor modelo de previsão a ser utilizado, como exemplo, o uso de sistemas especialistas [TF92]. Nesses sistemas, o melhor algoritmo é selecionado a partir de regras definidas por um especialista no domínio de previsão, que avaliando as características das séries definem qual o melhor modelo de previsão a ser utilizado. Apesar das possíveis vantagens desses sistemas, eles dependem do conhecimento do especialista, que nem sempre está disponível. Além disso, dependendo da quantidade de séries temporais e da quantidade de modelos de previsão, torna-se inviável a utilização dessa técnica.

Nesse contexto, trabalhos anteriores utilizam meta-aprendizado como uma ferramenta para prever automaticamente os melhores modelos de previsão a partir das características das séries temporais [Pru04, SPL04].

Prudêncio, em [Pru04], utiliza meta-aprendizado para seleção de modelos para as séries anuais da *M3-competition* [MH00]. Essas séries pertencem ao domínio econômico e demográfico. Nesse trabalho, características das séries como tamanho, tendência da série e autocorrelações foram usadas para selecionar três modelos de previsão: modelo Autoregressivo, *Random Walk* e Alisamento Exponencial de Holt. Apesar dos resultados positivos dos experimentos, diferentes pontos ficaram em aberto como a avaliação de meta-aprendizado para seleção de modelos em diferentes horizontes de previsão, avaliação de diferentes algoritmos usados como meta-aprendiz, e a definição de diferentes medidas de erro para avaliação dos modelos de séries temporais.

Nesse trabalho, focamos em meta-aprendizado no contexto de previsão de séries temporais, realizando novos experimentos para investigar os pontos em aberto dos trabalhos anteriores.

1.2 OBJETIVO DA DISSERTAÇÃO

Essa dissertação tem como objetivo de um modo geral utilizar meta-aprendizado para seleção automática de modelos de séries temporais. Especificamente, ela também visa:

- realizar um levantamento bibliográfico sobre recomendação de algoritmos, bem como as técnicas de meta-aprendizagem utilizadas;
- apresentar modelos de previsão de séries temporais;
- descrever as principais métricas de avaliação utilizadas em trabalhos voltados para recomendação de algoritmos;
- realizar experimentos que possibilitem verificar a eficácia da utilização de meta-aprendizado para prever os melhores modelos de previsão a partir das características das séries temporais.

1.3 ORGANIZAÇÃO DA DISSERTAÇÃO

Esta dissertação está organizada da seguinte forma:

Capítulo 2 - Meta-Aprendizado para Recomendação de Algoritmos: esse capítulo apresentará uma arquitetura de meta-aprendizado para recomendação de algoritmos, incluindo a descrição dos seus dois modos: de aquisição de conhecimento e seleção de algoritmos ou combinação dos mesmos; além de abordar as principais técnicas de meta-aprendizagem: caracterização de conjuntos de dados, mapeamento de conjunto de dados para modelos de previsão e combinação de meta-aprendizes;

Capítulo 3 - Modelo de Séries Temporais: nesse capítulo, será apresentado os três modelos de previsão de séries temporais usados nos nossos experimentos e as métricas de avaliação utilizadas para realizar as previsões das séries;

Capítulo 4 - Análise Experimental: nesse capítulo, serão apresentados os experimentos realizados, incluindo a metodologia utilizada, descrição dos conjuntos de dados (séries temporais), algoritmos avaliados e análise do desempenho dos experimentos.

Capítulo 5 - Conclusão: nesse capítulo, será apresentado as conclusões da dissertação, os pontos mais relevantes da dissertação, suas contribuições e ainda suas limitações e possíveis trabalhos futuros.

CAPÍTULO 2

META-APRENDIZADO PARA RECOMENDAÇÃO DE ALGORITMOS

Um dos desafios da aprendizagem de máquina é prever quando um algoritmo de aprendizagem é mais adequado do que outro para adquirir conhecimento para um determinado problema. A seleção de algoritmos geralmente envolve procedimentos custosos de tentativa-e-erro e conhecimento especializado, pois não existe um único algoritmo que seja adequado para todos os problemas existentes [Aha92]. Meta-aprendizado, segundo [VGCB68, VGcBS04], surge como uma solução, capaz de prever automaticamente o desempenho de um algoritmo para uma dada tarefa, ajudando o usuário no processo de seleção de algoritmos.

Esse capítulo aborda como meta-aprendizado pode ser usado para a recomendação de algoritmos. Na Seção 2.1, uma visão geral sobre meta-aprendizado e alguns conceitos básicos é apresentada. Já na Seção 2.2, abordaremos como podemos recomendar algoritmos utilizando meta-aprendizado. A arquitetura de um sistema de meta-aprendizado é apresentada na Seção 2.3. E na Seção 2.4, é mostradas algumas técnicas de meta-aprendizado.

2.1 META-APRENDIZADO

Nessa seção apresentamos alguns conceitos básicos sobre meta-aprendizado. Inicialmente, definimos algoritmos-base como os algoritmos candidatos a resolver uma determinada tarefa de aprendizagem. Algoritmo-meta por sua vez é um modelo de aprendizado responsável pela seleção dos algoritmos-base.

Os algoritmos-base são treinados com conjunto de dados ou conjunto de instâncias, que são observações individuais de um dado fenômeno. Cada instância é formada por atributos que medem certas propriedades do fenômeno de interesse dados. Além disso, cada instância armazena um atributo alvo que consiste em um valor que estamos querendo prever. Nesse contexto, espaço de instâncias refere-se ao espaço de todas as combinações

possíveis de valores de atributos. A classificação denota o caso onde o atributo alvo é de natureza qualitativa, sendo que o modelo induzido no aprendizado é um classificador. Quando o atributo alvo é de natureza quantitativa temos um caso de regressão.

Uma tarefa de aprendizagem é a aplicação dos algoritmos-base em um dado conjunto de dados. Usamos um determinado subconjunto de dados, o qual chamamos de instâncias de treinamento para criar o modelo de indução, utilizamos outro subconjunto de dados (as instâncias de teste) para validar o modelo induzido. O desempenho de generalização de um modelo avalia comumente o número de acertos no conjunto de teste. O desempenho de um algoritmo de aprendizagem com um determinado conjunto de dados é fortemente afetado pela qualidade dos dados. A qualidade de um modelo pode ser afetada, por exemplo, por ruído nos dados, que pode ocasionar *overfitting* (quando um modelo induzido se comporta bem para as instâncias de treinamento, mas possuem alta taxa de erro de classificação). Outras características dos dados podem afetar o desempenho dos algoritmos de forma que se faz necessário então explorar métodos que adequadamente descrevam a qualidade dos dados de forma que se possa relacionar com o desempenho dos algoritmos-base.

Meta-aprendizado usa exemplos empíricos (i.e. experimentos realizados com os algoritmos-base em diferentes tarefas) para produzir um modelo de aprendizagem responsável por associar um algoritmo-base com as características das tarefas de aprendizagem. Os meta-aprendizes são algoritmos de aprendizagem indutiva supervisionada. No meta-aprendizado, uma tarefa de aprendizagem deve ser descrita adequadamente através de atributos (nesse caso, meta-atributos) para que possa realizar o relacionamento de uma experiência passada com um algoritmo de aprendizagem. Para isso, é necessário realizar uma caracterização de dados que supostamente ajuda a revelar informações pertinentes dos dados. O atributo alvo no caso de meta-aprendizado consiste no algoritmo-base que obteve o melhor desempenho para a tarefa de aprendizado em consideração.

Uma meta-aprendizagem bem sucedida pode oferecer uma série de benefícios, uma vez que a seleção de algoritmos para novas tarefas é feita de forma mais eficiente sem a necessidade de comparação direta dos algoritmos. O desempenho das técnicas de meta-aprendizado depende largamente de uma boa descrição das tarefas como dito acima.

O conceito de meta-aprendizado foi originalmente avaliado para selecionar algoritmos para tarefas de classificação. Nos últimos anos, meta-aprendizado tem extrapolado para seleção de algoritmo em outros domínios de aplicações, como sistemas de planejamento, otimização, bioinformática e previsão de séries temporais [Pru10].

2.2 RECOMENDAÇÃO DE ALGORITMOS COM META-APRENDIZADO

Existem muitos algoritmos que podem ser utilizados para uma determinada tarefa de aprendizagem como kNN, árvores de decisão, *support vector machines*, redes neurais, dentre outros. Assim, tentar todas as alternativas possíveis para escolher o melhor algoritmo não é viável. Soares e Brazdil [BGCSV08] afirmam que do ponto de vista do usuário, a recomendação de algoritmos pode ser indicada como uma atividade de poupar tempo, reduzindo o número de alternativas a serem testadas para um determinado problema. Isso deve ser feito com um mínimo de perda na qualidade dos resultados obtidos quando comparado com os melhores algoritmos.

Sem algum tipo de ajuda, o usuário, que na maioria das vezes não tem o conhecimento necessário, pode encontrar dificuldades para selecionar o melhor algoritmo. Uma solução para esse problema é a construção de um sistema de meta-aprendizado que sugira automaticamente algoritmos de acordo com o desempenho previsto.

No meta-aprendizado, a escolha do algoritmo é guiada pelo conhecimento gerado pelas características dos dados relacionadas ao desempenho do algoritmo, chamada de meta-conhecimento. Essa abordagem é utilizada para minimizar o número de alternativas disponíveis. A resolução do problema de seleção e recomendação de algoritmos é alcançada no meta-aprendizado através da utilização de meta-dados (ou meta-exemplos), que descrevem o desempenho dos algoritmos-base e as características das tarefas de aprendizado em questão. O desempenho dos algoritmos calculado usando os dados de um problema é utilizado para definir um ranking dos algoritmos-base. Esse ranking é referenciado como ranking alvo, ou é usado para definir o atributo alvo da tarefa de meta-aprendizagem. As medidas usadas para caracterizar os dados de um problema são chamados de meta-atributos ou meta-características.

Baseado nessa perspectiva, meta-aprendizado pode ser definido como uma abordagem para gerar meta-conhecimento que mapeia as características de um problema (meta-características) para o desempenho relativo do algoritmo [BGCSV08]. Segundo Vilalta [VGcBS04], de um ponto de vista prático, meta-aprendizagem otimiza o uso dos algoritmos em problemas de aprendizagem de máquina e mineração de dados, particularmente na área de classificação e regressão. Em primeiro lugar, o sucesso dos algoritmos é condicionado à seleção apropriada do modelo de predição (ou combinação de modelos) de acordo com o domínio da aplicação.

Além disso, o sucesso da aplicação de modelos num cenário do mundo real requer

continua adaptação as novas necessidades. Isso é viabilizado pelo meta-aprendizado uma vez que os algoritmos podem ser melhor selecionados à medida que são aplicados em diferentes problemas. É importante ressaltar que no meta-aprendizado, se faz uso de experiências anteriores para recomendar o uso de modelos para tarefas similares. Nesse caso, a solução para seleção de algoritmos é capaz de se adaptar continuamente as necessidades, ou seja, que ela possa de reaprender levando em consideração experiências passadas.

Mostraremos a seguir uma arquitetura de sistemas de meta-aprendizado e as técnicas necessárias para a sua construção de forma que seja capaz de aprender com experiências passadas, e então realizar a recomendação (seleção ou combinação) de algoritmos.

2.3 ARQUITETURA

Vilalta, Giraud-Carrier e Soares [VGcBS04] realizaram uma pesquisa com o objetivo de entender a interação entre o mecanismo de aprendizagem e o contexto concreto em que esse mecanismo é aplicado. Essa pesquisa é relatada no artigo *“Using Meta-Learning to Support Data Mining”* onde começam idealizando a arquitetura de um sistema de meta-aprendizado que utiliza uma série de técnicas.

A proposta divide o sistema de meta-aprendizagem em dois modos de operação: modo de aquisição de conhecimento (Subseção 2.3.1) e o modo de sugestão (Subseção 2.3.2) que será mostrado a seguir.

2.3.1 Modo de Aquisição de Conhecimento

O principal objetivo do modo de aquisição de conhecimento é aprender sobre o próprio processo de aprendizagem. A Figura 2.1 ilustra esse modo, que será descrita a seguir.

Conforme podemos visualizar na Figura 2.1-A, o modo recebe como entrada exemplos de conjuntos de dados (ou exemplos de tarefas). À chegada de cada conjunto de dados, o sistema de meta-aprendizagem invoca o componente responsável pela extração das características do conjunto de dados ou meta-características (Figura 2.1-B). O objetivo do componente é recolher informações que transcende o domínio particular da aplicação, buscando informação que possa ser generalizada para os outros conjuntos de exemplos do domínio.

Durante o modo de aquisição de conhecimento, um conjunto de técnicas de aprendizagem está disponível (Figura 2.1-C). As estatísticas inseridas na base de meta-conhecimento derivam da estratégia de aprendizagem (e.g., um classificador ou uma combinação de classificadores, Figura 2.1-D) e seus desempenhos (Figura 2.1-E), que podem ser usados como uma forma de caracterização das tarefas sob análise.

A informação derivada do gerador de meta-características e do módulo de avaliação de desempenho podem ser combinadas na base de meta-conhecimento (Figura 2.1-F). Essa base de conhecimento é o principal resultado da fase de aquisição de conhecimento. Ela reflete a experiência acumulada nas diferentes tarefas.

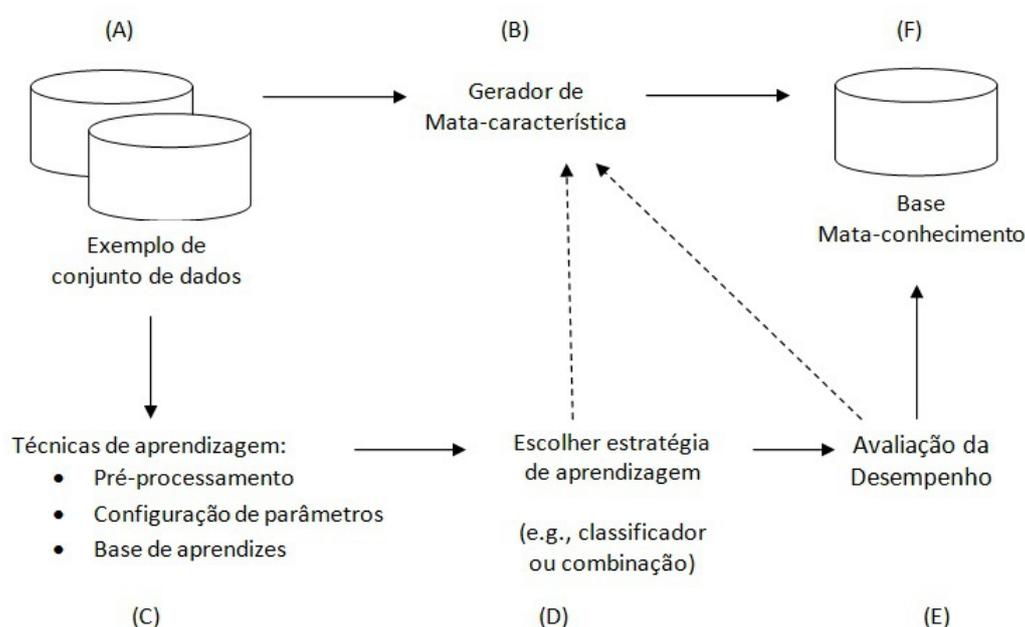


Figura 2.1. Sistema de Meta-Aprendizagem: Modo de Aquisição de Conhecimento

Meta-aprendizagem está intimamente ligada ao processo de aquisição e exploração de meta-conhecimento. Os avanços nesse campo dependem de uma questão específica; como podemos adquirir e explorar o conhecimento sobre sistemas de aprendizagem (e.g., meta-conhecimento) para entender e melhorar o seu desempenho.

2.3.2 Modo de Sugestão

No modo de sugestão (Figura 2.2), o meta-conhecimento adquirido no modo de exploração ou modo de aquisição do conhecimento é usada para configurar o sistema de

aprendizagem de forma a se explorar as características dos novos dados.

As meta-características retiradas dos conjuntos de dados (Figura 2.2-B) são combinadas com a base de meta-conhecimento (Figura 2.2-E) para produzir a recomendação relativa à melhor estratégia de aprendizado recomendada. A partir desse ponto, passamos da experimentação com aprendizes no nível base (usada previamente na fase de aquisição do conhecimento) para o uso de modelos de seleção ou combinação da base de aprendizes.

O classificador final (ou combinação de classificadores finais - Figura 2.2-D) é selecionado baseado não somente nas estimativas de generalização de desempenho do conjunto de dados atual, mas também na informação derivada do conhecimento adquirido em experiências passadas. O sistema muda de simplesmente experimentar estratégias de aprendizado diferentes (por escolha ou de forma randômica) para a habilidade de selecionar uma determinada estratégia dinamicamente.

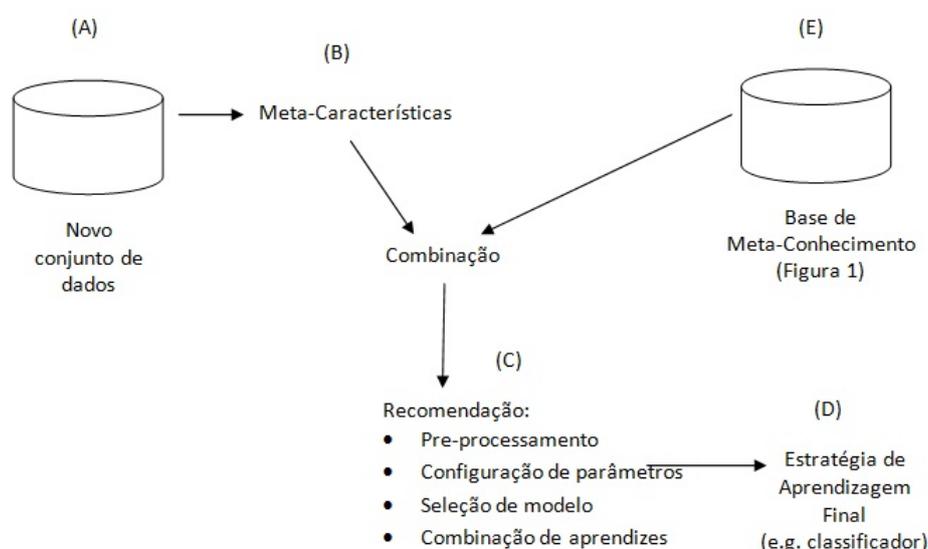


Figura 2.2. Sistema de Meta-Aprendizagem: Modo de Sugestão

A falta de experiência no começo da aplicação os algoritmos existentes forçaria a escolha randômica de modelos. Espera-se que ao longo do tempo o meta-aprendiz domine o processo de decisão de que estratégia de aprendizado melhor se adapta as características do problema. A efetividade do meta-aprendiz aumenta conforme o acumulo de meta-conhecimento.

A combinação das meta-características e a base de meta-conhecimento podem ter várias interpretações. A visão tradicional trata esse problema como um problema de aprendizagem supervisionado onde o meta-aprendiz é utilizado para gerar como saída

uma aproximação da função de mapeamento das meta-características em estratégias de aprendizagem sugeridas. Esse mapeamento tem como intuito simplesmente possibilitar a seleção de uma ou mais estratégias que parecem ser efetivas segundo as características do conjunto de dados em análise.

A seguir na Seção 2.4 vamos mostrar uma variedade de técnicas de meta-aprendizagem que podem ser utilizadas dentro dessa arquitetura.

2.4 TÉCNICAS DE META-APRENDIZAGEM

Apresentaremos nesta seção, as técnicas de meta-aprendizado, são elas: caracterização de conjunto de dados, que pode ser feita usando uma variedade de abordagens estatísticas, informações teóricas e baseada em modelo (Subseção 2.4.1); mapeamento de conjuntos de dados para modelos de previsão (Subseção 2.4.2); e, combinação de algoritmos-base, informações coletadas a partir do desempenho de um conjunto de algoritmos de aprendizagem no nível base para realizar combinação através de meta-aprendiz (Subseção 2.4.3).

2.4.1 Caracterização de Conjunto de Dados

Um componente crítico de qualquer sistema de meta-aprendizagem precisa extrair informações relevantes sobre a tarefa em análise, fornecendo algumas informações para diferenciar o desempenho de um determinado conjunto de estratégias de aprendizagem. Em outras palavras o desempenho das técnicas de meta-aprendizagem depende largamente da descrição da tarefa. Várias abordagens têm sido propostas, aqui apresentaremos as técnicas baseadas em caracterização de dados, são elas: caracterização baseada em informação teórica e estatísticas, caracterização baseada em modelos e *landmaking*.

Caracterização por Informação Teórica e Estatística

Segundo Engels e Theusinger [ET98], uma tarefa relacionada com seleção de algoritmo é a de processamento de dados. A seleção do algoritmo adequado para essa determinada tarefa depende do esforço na preparação dos dados mais do que qualquer outra coisa.

Muitos trabalhos sobre caracterização de conjunto de dados têm se concentrado na extração de parâmetros estatísticos e informações teóricas calculadas a partir do conjunto de treinamento [Aha92, ET98, GB95, HK01, MS94, Soh99].

No artigo “*Using a data metric for preprocessing advice for data mining applications*” [ET98], Engels e Theusinger afirmam que a tarefa de caracterização de conjunto de dados se concentra na extração simples de informação teórica e estatística dos parâmetros estimados do treinamento de um conjunto, incluindo número das instâncias, número de características, relação dos exemplos e as suas características, grau de correlação entre características e a classe, media da entropia das classes, obliquidade, presença de outliers, entre outros.

Nesse mesmo trabalho [ET98], ainda é apresentada uma ferramenta de caracterização de dados chamada DCT (*Data Characterization Tool*) a qual é responsável por computar a caracterização de dados. O DCT computa vários meta-dados de um determinado conjunto de dados. Subseqüentemente, extraídas as características mais relevantes dos dados. Essa característica pode ser separadas em três diferentes partes:

- medida simples ou características gerais dos dados;
- medida da análise discriminante e outras medidas, que podem ser computadas com atributos numéricos; e,
- medida de informação teórica, que pode ser calculada apenas com atributos simbólicos.

Além de fornecer medidas estatísticas, ela também fornece informação teórica para lidar com atributos discretos, como por exemplo:

- entropia do atributo - descreve a informação contida em certas variáveis discretas;
- entropia da classe - descreve informação contida nas variáveis dependentes;
- entropia de junção - relata à variável dependente das variáveis independentes, essa medida geralmente é usada para obter informação de relativa importância de atributos discretos no problema de classificação;
- heurísticas - calculadas com o intuito de estimar o grau de conformidade entre as medidas.

Lindner [LS99] desenvolveu uma ferramenta para selecionar algoritmos usando Raciocínio Baseado em Casos (RBC) chamado de AST (*Algorithm Selection Tool*). O problema da seleção de algoritmo é uma decisão baseada em restrições de aplicação (top-down), um

conjunto de dados dado com a sua meta-dados características (bottom-up) e no conhecimento sobre os algoritmos disponíveis. A arquitetura do AST é baseada em três fatores: restrições de aplicação, caracterização de dados e experiências existentes. A AST faz uso da DCT para caracterizar um conjunto de dados. As características eram calculadas para todos o conjunto de dados. Enquanto que as medidas estatísticas eram computadas para um subconjunto de atributos desses dados. No entanto, calcular essas medidas não é uma tarefa trivial.

Caracterização Baseada em Árvore de Decisão

Outra forma de obter a caracterização de um conjunto de dados é explorar as propriedades de hipótese de indução, como uma forma de representar o conjunto de dados em si. Podemos construir uma árvore de decisão a partir de um conjunto de dados e coletar as propriedades da árvore (por exemplo, forma, nós por recurso, a profundidade máxima da árvore, o desequilíbrio da árvore, etc.), como forma de caracterizar o conjunto de dados. Alguns trabalhos apresentar essa abordagem [BGCK00] e [PFBS02].

Em [BGCK00], podemos verificar que as principais abordagens de meta-aprendizado para a seleção de um modelo dependem de uma tarefa de caracterização de dados. Neste trabalho, também é afirmado que os principais tipos de caracterização são: a caracterização por informação estatística e teórica, landmarking e por árvore de decisão. Uma árvore de decisão é induzida a partir dos dados e uma série de medidas, tais como a profundidade, forma, e balanceamento, que são calculados a partir das mesmas. Supondo que as árvores de decisão induzidas a partir de conjuntos de dados, possui características que são fortemente dependentes de dados ou tarefa. Consideram que melhor do que extrair informação manualmente de um conjunto de dados ou de uma árvore induzida, é usar a árvore induzida para realizar a tarefa de caracterização. Seria possível realizar a caracterização de uma tarefa por vários modelos induzidos (por exemplo, uma árvore de decisão, uma rede neural, etc), mas os autores se restringimos a caracterização de uma árvore de decisão induzida usando os seguintes descritores: a razão entre o número de nós de árvore para o número de atributos, a razão entre o número de nós de árvore para o número de instâncias de treinamento, a homogeneidade da árvore, entre outros. Os experimentos realizados fizeram uso da informação computada da árvore de decisão induzida para caracterizar as tarefas de meta-aprendizagem, apresentam ainda os descritores e as representações das árvores de decisão induzidas, incluindo suas respectivas representações em Escher.

A grande motivação para árvore de decisão ter sido usada diretamente é que as pro-

priedades predefinidas usadas para caracterização baseada em árvore de decisão apenas tornam propriedades explícitas na estrutura de árvores. Além disso, essas propriedades devem ser calculadas independentemente da sua utilização para caracterização do problema. Ela provê uma representação de linguagem suficientemente expressiva. Uma tarefa pode ser caracterizada pelo número de hipóteses de aprendizes, os quais podem ser manipulado pelo meta-aprendiz baseado no framework de alta-ordem.

Em primeiro lugar, sugerem que mais do que realizar uma extração de informação manualmente dos conjuntos de dados ou de uma árvore induzida, é melhor usar a árvore induzida diretamente para realizar a tarefa de indução, e para isso, foi feito uso de uma estrutura de indução de aprendizagem de alta-ordem. Cada atributo da linguagem tem um tipo (nominal ou real) e os exemplos podem ser tuplas de constantes retiradas do domínio dos tipos correspondentes. Isso permite examinar possíveis relações entre elementos distintos que tem o mesmo tipo, embora limite a explosão do espaço de busca. A representação do conhecimento lógico de alta-ordem foi veiculada pela linguagem de programação Escher [Llo99].

Peng et al. [PFBS02] apresenta novas medidas para caracterizar conjuntos de meta-aprendizagem baseada na idéia de captar as características estruturais da forma e tamanho da árvore de decisão induzida a partir do conjunto de dados. Sua eficácia é ilustrada através da comparação com os resultados obtidos pelas características técnicas clássicas de dados, incluindo DCT que é a técnica mais usada em meta-aprendizagem e *landmarking* que é o método mais recentemente desenvolveu e produziu o melhor desempenho em relação ao DCT. A seguir abordaremos a técnica que utiliza *landmarking*.

Landmarking

Outra forma de caracterização é através da utilização de *landmarking* descritos em [BGC00] e [PGc00]. *Landmarking* tem como idéia explorar as informações obtidas a partir do desempenho de um conjunto de aprendizes simples, que apresentam diferenças significativas em seu mecanismo de aprendizagem. A precisão ou taxa de erro destes *landmarkers* é usado para caracterizar um conjunto de dados e identificar as áreas onde cada um dos formandos simples pode ser considerado como um *expert*.

Landmarking é uma abordagem para descrever tarefas em meta-aprendizado. Ela tenta determinar a localização de um determinado problema no espaço de todos os problemas de aprendizagem através da medição do desempenho de algoritmos de aprendizagem simples e eficientes.

Em [PGc00], Pfahringer mostra que o uso de *landmarkers* pode ajudar a distinguir entre diferentes áreas de aprendizagem que favorece diferentes aprendizes. Essa abordagem afirma que algumas meta-características usadas consomem muito tempo, e então, *landmarking* pode ser uma abordagem mais econômica para problemas de caracterização de aprendizagem.

Essa possibilidade de utilização de *landmarking* é feita através da exploração de informações obtidas em versões simplificadas dos dados, ou seja, em amostras. A precisão sobre os resultados destas amostras servem para caracterizar conjuntos de dados individuais e são referidos como subamostragem de *landmarkers*. Esta informação é posteriormente utilizada para selecionar um algoritmo de aprendizado adequado, como são mostrados em [FP01] e [GB95].

Pfahringer [PGc00] cita ainda que nenhum algoritmo é bom para todos os problemas possíveis, sendo assim apresenta uma série de experimentos com bases reais e artificiais para mostrar o sucesso dos *landmarkers* e a melhor desempenho de um conjunto de algoritmos de aprendizagem.

Landmarking tenta caracterizar diretamente um domínio pela relação de desempenho de alguns aprendizes (*landmarkers*) com o domínio de alguns outros algoritmos. Basicamente Pfahringer demonstra que “se *landmarker* A possui um melhor desempenho que o *landmarker* B para uma determinada tarefa, então o aprendiz X irá também apresentar um desempenho melhor que o aprendiz Y para a mesma tarefa”. *Landmarking* substitui o primeiro conjunto de aprendizes por um conjunto de outros mais sofisticados guiada pelos resultados dos aprendizes.

Essa abordagem é bem sucedida quando os *landmarkers* são capazes de medir as propriedades relevantes para o sucesso ou o fracasso de um algoritmo de aprendizagem para algum problema especificações. Assim, a escolha dos aprendizes vai influenciar a seleção de *landmarkers*. Meta-aprendizagem em geral auxilia o desenvolvimento e o aperfeiçoamento de algoritmos de aprendizagem. A determinação de certas deficiências de aprendizagem relativa aos algoritmos em relação aos seus concorrentes pode mostrar as indicações para melhoria deste algoritmo.

Pfahringer e Giraud-carrier [PGc00] propõem então o uso de uma versão mais simplificada do algoritmo. Enquanto [FP01] propõe uma versão simplificada dos dados. No entanto, [LB08] afirma que apesar de *Landmarking* possuir vantagens, a caracterização de dados ainda é útil e a combinação dos dois é seu objeto de estudo e seus experimentos mostram que esse método atinge uma desempenho melhor que os anteriores.

Nesse sentido, Leite e Brazdil [LB04] apresentam meta-aprendizado para pré-processamento de dados tendo como objetivo reduzir o número de amostras durante um processo de amostragem progressiva. O processo consiste em pára a seleção de amostragem quando a curva de aprendizado se estabiliza. A amostra correspondente é chamada de ponto de parada. O objetivo desse método consiste em prever o ponto de parada no conjunto de dados em estudo.

O método de Leite e Brazdil [LB04] compara os primeiros pontos na curva de aprendizagem construídas para um determinado algoritmo de aprendizagem com os dados sobre estudo. Os conjuntos de dados com as curvas mais semelhantes são selecionados e os pontos de paradas correspondentes são utilizados para estimar o ponto de parada para o conjunto de dados atual. Conclui ainda, dizendo que esta informação pode ser usada para diminuir as investigações nas amostras que, conseqüentemente, leva a uma economia de tempo.

2.4.2 Mapeamento de Conjunto de Dados para Modelos de Previsão

Meta-aprendizado também é utilizado, na prática, para a construção de um mecanismo de mapeamento dos espaços de entradas, compostos por conjunto de dados ou aplicações que gera como saída um espaço que é composto por modelos de previsão.

Critérios como exatidão, espaço de armazenamento, e tempo de execução podem ser usados para realizar avaliação de desempenho [GC98].

Meta-Regras Manuais

Brodley [Bro95] afirma que através de um especialista e utilizando conhecimento empírico, um conjunto de meta-regras que correspondam às características do domínio com técnicas de aprendizado podem ser geradas manualmente.

Vilalta et al. [VGcBS04] conclui então afirmando que construir regras manualmente tem a desvantagem de falhar quando é necessário identificar muitas regras importantes. Como resultado a maioria das pesquisas tem se concentrado em aprender essas meta-regras automaticamente. Como será apresentado a seguir.

Aprendizagem em Meta-Nível

A caracterização de um conjunto de dados é uma forma de meta-conhecimento que comumente é embutida em um meta-conjunto de dados construído após várias tarefas de

aprendizagem onde cada par de elementos é constituído pela caracterização do conjunto de dados (vetor de meta-características) e a classe *label* correspondente ao modelo com o melhor desempenho para aquele conjunto. Depois um algoritmo de aprendizagem pode ser aplicado a uma tarefa de aprendizagem bem definida para induzir uma hipótese de mapeamento de conjunto de dados para previsão de modelos [VGcBS04].

Berrer et al. [BPK00] apresenta uma variação da abordagem descrita acima. Essa abordagem se baseia em observar os elementos vizinhos da pesquisa no espaço de meta-características. Quando uma nova pesquisa é feita, o k elemento mais próximo dessa instância do conjunto de dados é identificado para selecionar o modelo com a melhor media de desempenho.

Merz propõe em [Mer96] uma abordagem diferente, que ao invés de mapear uma tarefa ou conjunto de dados para prever um modelo, ela tem como objetivo selecionar um modelo para cada exemplo de consulta. Essa abordagem é similar a abordagem do k vizinho mais próximo, onde se seleciona o modelo que exibe o melhor desempenho que seja mais próximo do exemplo de consulta.

Modelo de *Ranking*

Além de mapear um conjunto de dados a um único modelo de previsão, pode também produzir um ranking sobre um conjunto de modelos diferentes. Essa classificação é mais útil para o usuário, pois fornece informações adicionais para suportar a seleção de um determinado algoritmo.

Na prática, o *ranking* fornecer soluções alternativas para os usuários que desejem incorporar sua própria experiência ou a qualquer outro critério (por exemplo, as limitações financeiras) em seu processo decisório.

Várias abordagens têm sido propostas na literatura para atacar o problema do *ranking* de modelos preditivos [NS97, KT99, BS00, BPK00, BSC03, GB95].

Kalousis e Theoharis [KT99] apresentam a abordagem chamada *NOEMON ranking*. Esse sistema utiliza um conjunto de meta-aprendizes para fornecer um ranking (ou ordenação) dos algoritmos baseado nas informações de desempenho. Ele constrói um meta-aprendiz simples para cada par de algoritmos.

Segundo exemplifica Prudêncio e Ludermir [PL07], dado um novo problema de aprendizado, NOEMON coleta as respostas dos meta-aprendizes e credita pontos aos algoritmos candidatos dependendo das respostas fornecidas. Para um meta-aprendiz (X, Y) ,

por exemplo, se 'X' é a classe prevista, então um ponto é creditado para o algoritmo X; caso contrário é creditado para o algoritmo Y. A ordem final dos algoritmos candidatos é definida diretamente a partir do número total de pontos que cada algoritmo recebeu.

Já Bradzil e Soares [BS00] desenvolveram uma técnica chamada *Zoomed ranking*. Essa abordagem resolve o problema de seleção de algoritmos dividindo-a em duas fases distintas.

A primeira fase, chamada de *zooming*, é destinada a identificar um subconjunto de dados relevante levando em consideração a base de instâncias. Esta seleção é feita utilizando um algoritmo kNN com a função de distância baseada em informação estatística e teórica e outras medidas de caracterização para identificar o conjunto de dados mais similares.

Na segunda fase, um ranking dos algoritmos candidatos é construído baseado nas informações de desempenho (precisão e tempo total de execução) dos algoritmos candidatos sobre os dados selecionados na primeira fase.

Santos [SPL04] utilizou essa abordagem, *Zoomed ranking*, originalmente desenvolvida para gerar um ranking de algoritmos candidatos a solucionar um dado problema de classificação, para selecionar modelos de previsão de séries temporais com base em informação de desempenho, com o uso de uma abordagem de Meta-Aprendizado.

Strict Meta-learning

Seleção algoritmo pode ser visto como um problema de classificação convencional, onde cada meta-exemplo armazena um *label* da classe, indicando o melhor algoritmo para uma dada tarefa de aprendizagem, entre um conjunto de candidatos. Meta-aprendizagem é apenas um classificador que prediz o melhor algoritmo de acordo com as características da tarefa. Essa definição foi apresentada em [Aha92, KH03, LB05].

Aha propõe em [Aha92] a construção de variantes parametrizada de bases de dados e a análise do comportamento dos algoritmos sobre conjuntos de dados artificiais. Essa abordagem é utilizada para obter o conhecimento sobre o seu comportamento em circunstâncias diferentes, o que não seria possível com uma única experiência. Demonstra em seu estudo de caso utilizando o banco de dados de reconhecimento de letras.

Kalouisis e Hilario [KH03] propõem uma abordagem baseada nas noções de agrupamento que está relacionada às idéias de similaridade baseada aprendizagem relacional. Os resultados apresentados nesse trabalho mostram que a representação relacional junta-

mente com a medida de similaridade apropriada pode melhorar o desempenho. As idéias apresentadas são pertinentes não só para questões de representação em meta-aprendizado, mas para todos os domínios com requisitos similares.

Já Leite e Brazdil [LB05] apresentam um trabalho onde a preocupação se baseia no problema de prever o desempenho relativo de algoritmos de classificação. Focaliza em métodos que utilizam resultados em pequenas amostras e discute as deficiências dos métodos anteriores e uma nova variante é proposta para avaliá-los. Apresentam ainda experimentos realizados em algumas amostras, e a informação obtida foi usada para identificar a curva de aprendizagem mais próxima para que o processo de amostragem fosse realizado plenamente, que por sua vez, permite gerar uma previsão que diz respeito ao desempenho relativo de algoritmos. A avaliação experimental mostra que o método compete bem com as abordagens anteriores e fornece uma solução para o problema.

Meta-regressão

Essa abordagem de meta-aprendizado [BK01, KTK00] tenta prever diretamente a precisão ou alternativamente o erro de cada algoritmo candidato. Um meta-aprendiz pode então ser usado tanto para selecionar algoritmo com a maior precisão ou fornecer o *ranking* de algoritmo baseado na ordem de precisão prevista.

Bensusan e Kalousis [BK01] apresentam um trabalho que investiga o uso de meta-learning para estimar a precisão da previsão de um classificador, num cenário onde meta-aprendizado é visto como uma tarefa de regressão e analisam o seu potencial no âmbito de três estratégias de caracterização de conjunto de dados. Mostraram que é possível estimar o desempenho de classificação com um grau elevado de confiança e obter conhecimento sobre o classificador através de modelos de regressão gerados. E exploram os resultados dos modelos para prever o ranking dos indutores para mostrar que a melhor estratégia para a estimativa de desempenho não é necessariamente a melhor para a geração de ranking.

Um trabalho de Köpf et al. [KTK00] apresenta a abordagem de meta-análise, que é utilizada para ajudar o usuário com uma guia para seleção automática de modelos e transformação de dados. Dois campos de aplicação principais foram selecionados em Metal (Assistente para meta-aprendizado, projeto ESPRIT 26.357): classificação e regressão.

Na primeira fase do projeto, as características dos dados, medidas e testes foram avaliados para o uso automáticos de algoritmos de classificação. Para a regressão, também foram realizados testes com medidas estatísticas e informação teórica.

O conceito de meta-regressão é apresentado: aprendizagem que utiliza regressão sobre o meta-nível para meta-aprendizagem. Em outras palavras, abordaram o problema de ligar a caracterização de dados meta-aprendizagem de regressão. Várias medidas foram introduzidas para construir uma base das características dos dados, determinando quais variáveis eram úteis. Em comparação com a classificação, as taxas de erro calculadas para os testes de validação cruzada, essa abordagem apresentou um ganho de precisão.

2.4.3 Combinação de meta-aprendizes

Outra possibilidade do meta-aprendizado consiste em aprender com os algoritmos-base, através do uso explícito da informação obtida partir do desempenho de um conjunto de algoritmos de aprendizagem no nível base, essas informações são então incorporadas no processo de meta-aprendizado.

Stacked Generalization

Stacked Generalization é considerada uma forma de meta-aprendizado, pois a transformação do conjunto de treinamento transmite informações sobre as previsões dos meta-aprendizes, ou seja, adquire meta-conhecimento. Wolpert em [Wol92] afirma justamente que se pode incorporar previsões de meta-aprendizes.

O processo de *Stacked Generalization* funciona sob uma arquitetura em camada. Cada conjunto de meta-classificadores é treinado com um conjunto de dados, e as representações de características originais utilizadas para incluir a previsão dos classificadores. Sucessivas camadas recebem como entrada as previsões da camada imediatamente anterior e a saída é passada para a próxima camada. A classificação única no nível mais alto produz a previsão final.

A maioria das pesquisas sobre *Stacked Generalization* centra-se em uma arquitetura de duas camadas [Bre96, Ska97, Cha97]. Pesquisas nesta área investigam que aprendizes-base e meta-aprendizes produz o melhor resultado empíricos [DZ04, GB00], como representar previsões de classe [TW97], e também definição de meta-características [AP96, BL96].

Baseado na idéia original de Wolpert, Breiman [Bre96], por exemplo, propõe um método para a formação de combinações lineares de previsões diferentes para dar melhor precisão de previsão, utilizando os dados de validação cruzada de mínimos quadrados com restrições de não negatividade para determinar os coeficientes na combinação. A sua

eficácia é demonstrada no empilhamento de árvores de regressão de diferentes tamanhos e em uma simulação de empilhamento subconjunto linear e regressão mais alta.

Ting e Witten em [TW97] abordam duas questões que afirmam ser o êxito da implementação de *Stacked Generalization* para tarefas de classificação. Em primeiro lugar, aconselham que devem ser usada as probabilidades de classe em vez de uma única classe prevista como atributos de entrada para o aprendizado de alto nível. Em segundo lugar, técnica de regressão linear de respostas múltiplas dos mínimos quadrados deve ser utilizada como generalizador de alto nível.

Ao combinar três diferentes tipos de algoritmos de aprendizagem, a implementação de *Stacked Generalization* foi a que conseguiu a melhor precisão de previsão em comparação tanto com seleção de modelo baseado em validação cruzada e votação. Ao contrário de *Stacked Regression*, restrições não-negativas na regressão dos mínimos quadrados não é necessário para garantir a melhor precisão de previsão das tarefas de classificação. No entanto, são preferíveis essas restrições, porque aumentam a interpretabilidade do modelo de nível 1.

O trabalho de Ting e Witten em [TW97] concentrou-se na busca de condições de empilhamento que trabalha com generalização. A implicação do êxito na implementação de *Stacked Generalization* é que o método anterior para combinação de modelos que emprega voto da maioria (ponderada), média, ou de outros cálculos que não fazem uso da aprendizagem de nível 1, pode agora aplicar este aprendizado para melhorar a sua precisão de previsão.

Boosting

O objetivo dessa abordagem é gerar um conjunto de meta-aprendizes a partir de variantes do conjunto de dados de treinamento. Cada variante é gerado por amostragem com reposição do peso de distribuição. Esta distribuição é modificada a cada nova variante dando-se mais atenção aos exemplos classificados incorretamente pela hipótese mais recente.

Boosting é considerada uma forma de meta-aprendizagem, pois leva em consideração as previsões de cada hipótese sobre o conjunto de dados de treinamento original de modo a melhorar progressivamente a classificação dos exemplos em que a última hipótese falhou [VGcBS04].

Podemos encontrar na literatura alguns trabalhos sobre *boosting*, são eles [FS96,

FHT00, HTFF05].

Meta-decision tree

Meta-decision tree é outra abordagem que consiste em combinar vários modelos indutivos por meio da indução de árvores de decisão [TD03]. Uma árvore de decisão é construída, e cada nó interno é uma meta-característica; e, cada nó folha corresponde a um modelo de previsão. Por exemplo, uma *meta-decision tree* indica qual o modelo que lhe parece mais adequado na previsão do *label*.

Todorovski e Dzeroski [TD03] definem que a estrutura de uma *meta-decision tree* é idêntica à estrutura de uma árvore de decisão comum. O nó de decisão (interno) especifica um teste a ser realizado em um único valor de atributo e cada resultado do teste tem seu próprio ramo principal para a subárvore apropriada. Em um nó folha, um *meta-decision tree* prevê que classificador deve ser utilizado para a classificação de um exemplo, em vez de prever o valor de classe do exemplo diretamente como uma árvore de decisão comum.

O trabalho focou em combinação de múltiplos classificadores gerados pela utilização de diferentes algoritmos de aprendizagem em um único conjunto de dados. Na primeira fase, representada no lado esquerdo da Figura 2.3, um conjunto $C = (C_1, C_2, \dots, C_N)$, onde N é o número de classificadores de nível base, é gerado através da aplicação de algoritmos de aprendizagem A_1, A_2, \dots, A_N para um único conjunto de dados de treinamento L .

Cada classificador base em nível de C prevê uma distribuição de probabilidade sobre os valores da classe possível. Assim, a previsão da classificação de nível de base C , quando aplicada ao exemplo x é um vetor de distribuição de probabilidade, como pode ser visto na Equação 2.1:

$$P_C(x) = (p_C(c_1|x), p_C(c_2|x), \dots, p_C(c_k|x)), \quad (2.1)$$

onde (C_1, C_2, \dots, C_k) é um conjunto de valores de classe possível e $p_C(C_i|x)$ denota a probabilidade de que x pertence a exemplo C_i da classe estimada e prevista pelo classificador C . A classe C_j com maior probabilidade da classe $p_C(C_j|x)$ é prevista pelo classificador C .

A classificação de um novo exemplo x usando os classificadores de meta-nível em C é exibido no lado direito da Figura 2.3.

Em primeiro lugar, as N previsões $(P_{C_1}(x), P_{C_2}(x), \dots, P_{C_N}(x))$ dos classificadores de

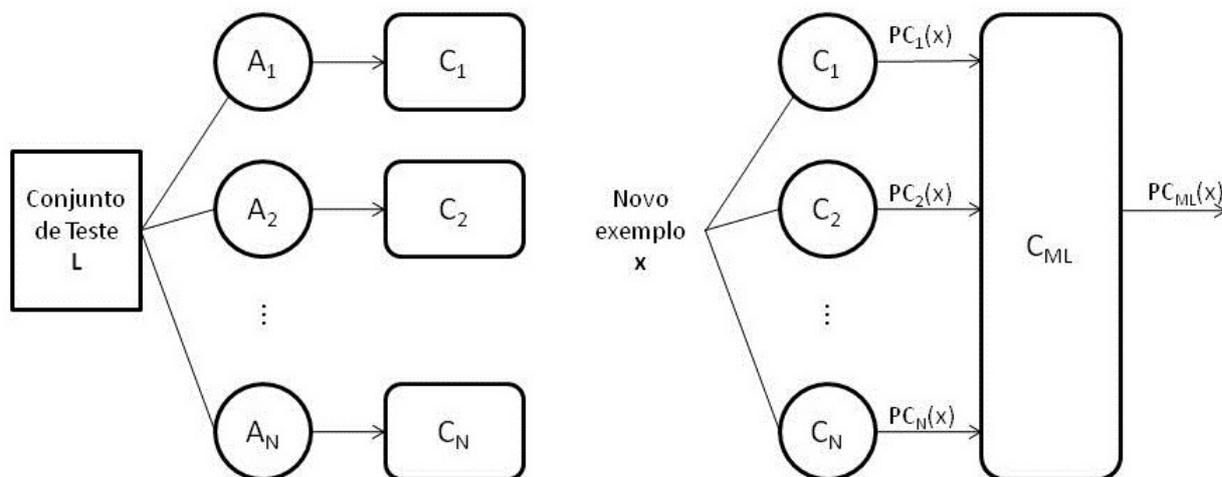


Figura 2.3. Meta-Decision Tree

nível base de C em x são gerados. As previsões obtidas são combinadas usando o método C_{ML} . Diferentes métodos de combinação podem ser usados para em diferentes estruturas de combinação.

Composição de aplicações indutivas

O sistema CAMLET ou Composição de Aplicações Indutivas constrói modelos usando componentes com diferentes tendências (viés) [AY04]. Um sistema CAMLET é baseado num *template* que abstrai o processo de aprendizagem indutiva.

Por exemplo, para um dado conjunto de dados o *template* é instanciado usando componentes que estão organizados em diferentes repositórios. O modelo final é obtido através de uma pesquisa interativo dos melhores componentes conectados a este modelo [VGcBS04].

A Figura 2.4 mostra o design básico de um CAMLET [SNY99]. A atividade básica para construção de sistemas de conhecimento usa o método de resolução de problemas (PSM - *Problema Solving Methods*).

O primeiro passo é que a atividade de construção construa uma especificação inicial para a aplicação indutiva. Depois o CAMLET seleciona a estrutura de nível superior para um sistema de aprendizado indutivo. Então, o CAMLET recupera o processo da folha a raiz, verificando a interligação dos relacionamentos do pré e pós-processamento dos processos selecionados na folha. Constrói então uma especificação inicial para a aplicação indutiva, descrita pelo processo da folha.

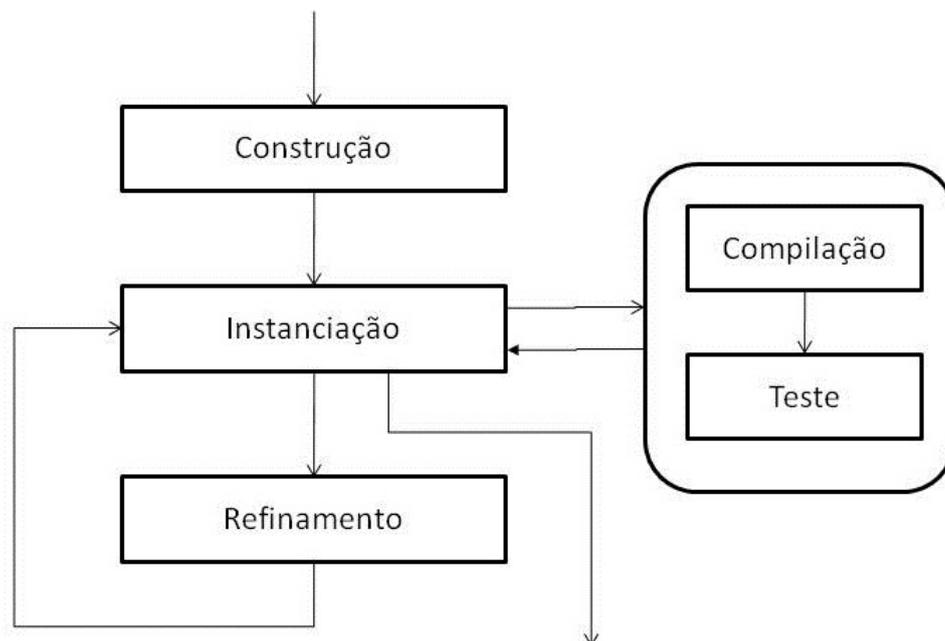


Figura 2.4. Design Básico de um CAMLET

A atividade de instanciação preenche a entrada e a saída dos relacionamentos dos processos da folha para a especificação inicial, usando os tipos de dados obtidos do conjunto de dados. Os valores dos outros papéis, como referência, pré-processamento e pós-processamento, não têm sido instanciados, mas vem diretamente do esquema de processo. Assim, uma especificação instanciada é sugerida. Adicionalmente, o processo da folha tem sido preenchido na lista de papéis dos objetos identificados pelos tipos de dados.

A atividade de compilação transforma a especificação instanciada em código executável usando uma determinada biblioteca. Quando o processo é conectado a outro processo em detalhes de implementação, a especificação para entrada e saída de tipo de dados deve ser unificada. Para fazer isso, a atividade tem como um mecanismo de conversão de dados que converte a árvore de decisão em um classificador.

A atividade de teste verifica se o código para a especificação instanciada executa bem através da checagem de requisitos (precisão) do usuário. A estimativa irá para a atividade de refinamento. A atividade de refinamento é uma tarefa de busca para achar um sistema (ou estrutura de controle) que satisfaça o objetivo da precisão.

CAPÍTULO 3

MODELO DE SÉRIES TEMPORAIS

Segundo [BJ70], série temporal é um conjunto de observações de um fenômeno ordenadas no tempo. A previsão de séries temporais tem como função fornecer subsídios que ajudem a diminuir os riscos na tomada de decisão no ambiente onde as séries estão inseridas e antecipar quadros futuros que servirão para o planejamento atual de estratégias [Arm01].

A previsão de séries temporais é o processo de identificação das características, padrões e propriedade importantes da série utilizando para descrever em termos gerais o fenômeno gerador [Mit97]. Como exemplos de séries temporais podemos citar: registro do consumo mensal de uma casa, vendas diárias de um determinado produto ao longo do ano, registro diários das bolsas de valores ao longo do mês, entre outros.

Aqui utilizaremos meta-aprendizado para avaliar o desempenho dos algoritmos para previsão de séries temporais. A motivação para usar algoritmos de aprendizagem de máquina é aperfeiçoar a construção de modelos a partir da experiência [VD02] obtida na resolução de diferentes problemas de previsão.

No entanto, selecionar o melhor algoritmo de aprendizado de acordo com as características dos conjuntos de treinamento e os modelos de previsão de forma adequada para uma dada série temporal pode ser uma tarefa difícil dependendo dos modelos candidatos e das características das séries.

Os métodos de previsão podem ser:

- subjetivos - baseado no julgamento subjetivo, na intuição, no conhecimento de um ou mais especialistas do domínio de previsão;
- univalorados - a série é prevista usando valores passados da própria série, relacionados através de modelos matemáticos (também chamados de modelos de extrapolação ou modelos de séries temporais); e,
- multivalorados - a série é prevista depende de uma ou mais variáveis adicionadas das chamadas variáveis de previsão ou explanatórias relacionadas através de modelos

matemáticos.

A vantagem de se utilizar algoritmos de aprendizagem é que eles podem realizar a aquisição automática do conhecimento. No entanto, são limitados em relação às funcionalidades fornecidas ao usuário da previsão. Alguns trabalhos que usam algoritmos de aprendizagem indicam apenas um único modelo candidato de acordo com um critério de precisão pré-definido. Prudêncio e Ludermir [PL04] abordam o problema de previsão de séries temporais como um problema de aprendizado e adapta idéias e conceitos de meta-aprendizado a problemas de seleção e combinação de modelos.

Utilizando meta-aprendizado para seleção de séries temporais faremos uso das técnicas de seleção de algoritmo para realizar a recomendação automática de modelos de previsão de séries temporais.

3.1 MODELOS CANDIDATOS

Nesta seção, nos deteremos em apresentar modelos usados para previsão das séries anuais da *M3-Competition*. São eles: Auto Regressivo de Box-Jekings (AR) [BJ70], *Random Walk* (RW) [Har93] e Alisamento Exponencial de Holt (HL) [HMMS60].

3.1.1 *Random Walk*

O Modelo *Random Walk* (RW) é um modelo simples para realizar previsão, mas eficiente para prever séries sem sazonalidade ou tendência [Pru04]. Segundo [Har93], a previsão de uma dada série Z no tempo t (Z_t) é igual ao valor observado no anterior (Z_{t-1}). O modelo RW pode ser escrito da seguinte forma:

$$Z_t = Z_{t-1} + e_t \quad (3.1)$$

Onde e_t é um termo aleatório com média 0 (zero) e variação constante. Sendo então a previsão a um passo da série Z no tempo t (\tilde{Z}_t) representada pela seguinte equação:

$$\tilde{Z}_{t+1} = Z_t \quad (3.2)$$

$$\tilde{Z}_{t+h} = Z_{t-1} \quad (3.3)$$

A previsão de uma determinada série temporal Z_1, Z_2, \dots, Z_t pode ser definida por Z_{t+h} , onde h (horizonte de previsão) é o número de passos a serem previstos a partir do tempo t .

3.1.2 Alisamento Exponencial de Holt

O modelo de alisamento exponencial de Holt (HL) [HMMS60] assume que uma série temporal possui uma tendência linear variando suavemente com o tempo. O modelo pode ser representado pela seguinte equação:

$$Z_t = b_t + a_t * h + e_t, h = 1, 2, 3, \dots \quad (3.4)$$

Na equação acima, a_t é o valor da inclinação da reta, b_t é o valor do intercepto e h é o horizonte de previsão. A cada nova série esses valores são ajustados de forma similar ao alisamento exponencial simples, através das fórmulas:

$$\tilde{a}_t = \alpha * Z_t + (1 - \alpha) * (\tilde{a}_{t-1} + \tilde{b}_{t-1}) \quad (3.5)$$

$$\tilde{b}_t = \beta * (\tilde{a} - \tilde{a}_{t-1}) + (1 - \beta) * \tilde{b}_{t-1} \quad (3.6)$$

Onde α e β são as constantes de alisamento de Holt, e seus valores definidos dentro do intervalo [0-1]. A previsão a um passo da série é dada pela Equação 3.7:

$$\tilde{Z}_t = \tilde{b}_t + \tilde{a}_t \quad (3.7)$$

Nesse modelo, essa previsão de h passos é dada pela Equação 3.8:

$$\tilde{Z}_{t+h} = \tilde{b}_t + \tilde{a}_t * h \quad (3.8)$$

3.1.3 Auto-Regressivo de Box-Jenkins

O modelo Auto-regressivo de Box e Jenkins (AR) apesar de simples obtém resultados satisfatórios em uma diversidade de problemas reais [Mas95]. O modelo AR prevê uma dada série através de uma combinação linear dos valores passados da série. O modelo AR pode ser escrito pela equação:

$$\tilde{Z}_t = \phi_1 * Z_{t-1}, \dots, \phi_p * Z_{t-p} + e_t \quad (3.9)$$

Os valores dos parâmetros ϕ_i descreve como Z_t se relaciona com o valor para Z_{t-i} onde $i = 1, 2, \dots, p$. Seus valores conforme a série que se deseja modelar, visando diminuir o erro da previsão. O parâmetro p representa o número de valores passados considerados e é denominado de janela de tempo ou *lag* do modelo. A variável e_t representa a parte aleatória do modelo. O experimento é realizado variando esses parâmetros de forma sistemática no intervalo entre zero a um, escolhendo o que obteve melhor erro de previsão no conjunto de treinamento.

3.2 AVALIAÇÃO DOS MODELOS DE PREVISÃO DE SÉRIES TEMPORAIS

A definição do atributo classe armazenado em cada meta-exemplo é feita através de um torneio onde os modelos candidatos são estimados e avaliados com a série em questão. Um torneio entre modelos pode ser caracterizado pela:

- medida de avaliação para definir o melhor modelo. Nesse aspecto, diversos fatores podem ser levados em consideração como a precisão do modelo, a sua complexidade, facilidade de interpretação, manutenção e uso, dentre outros;
- a amostra de dados da série sob a qual a medida de avaliação é calculada.

Aqui usaremos o erro de precisão (erro de previsão) para avaliar os modelos. Diferentes medidas de previsão podem ser usadas. Aqui, as medidas de avaliação que usaremos serão: a Média dos Erros Quadrados (*Mean of Squared Error* - MSE); Média dos Erros Absolutos (*Mean Absolute Error* - MAE); e a medida *Percentage Better* (PB).

3.2.1 Medidas de Erro

Média dos Erros Quadrados

A Média dos Erros Quadrados é obtida através da equação abaixo.

$$MSE = \frac{1}{H} \sum_{t=T+1}^{T+H} (Z_t - \tilde{Z}_t)^2 \quad (3.10)$$

Onde H é o número de ponto previstos da série, Z_t e \tilde{Z}_t são respectivamente o valor real e a previsão da série no tempo t .

A medida MSE é uma das medidas mais utilizadas no contexto de previsão. No entanto apresenta algumas desvantagens, como exemplo, grandes erros se destacam devido ao cálculo da média, erros dos *outliers* são supervalorizados, dentre outras.

Média dos Erros Absolutos

A Média dos Erros Absolutos é apresentada como uma alternativa para superar a desvantagens apresentada do MSE. A medida MAE é obtida através da seguinte equação.

$$MAE = \frac{1}{H} \sum_{t=T+1}^{T+H} |Z_t - \tilde{Z}_t| \quad (3.11)$$

Onde H é o número de ponto previstos da série, Z_t e \tilde{Z}_t são respectivamente o valor real e a previsão da série no tempo t .

Percentage Better

A medida *Percentage Better* mede a percentagem de vezes em que o erro de previsão do modelo analisado é menor que o erro de um dado modelo de referência. A principal vantagem dessa medida é que ela é imune a presença de *outliers* nos erros de previsão.

3.2.2 Métodos de Amostragem

Uma vez definido a medida de avaliação, o outro aspecto importante seria definir o procedimento de amostragem da série para avaliação. Thashman [Tas00] apresenta duas

formas de avaliação: *in-sample* e *out-of-sample*.

A avaliação *in-sample* usa dados da própria série temporal para treinamento dos modelos. Ela pode então superestimar o desempenho do modelo de previsão, em especial no caso de *overfitting*. Outra desvantagem que pode ser apontada é uma deficiência em relação ao método selecionado, que por melhor que se adequem a amostra de treinamento podem não prever bem os dados futuros da série.

A avaliação *out-of-sample* usa dados fora da amostra de treinamento. Sendo então uma alternativa a avaliação *in-sample*. Nessa avaliação, um determinado período da série é usado para estimar os modelos enquanto outro período é reservado para realizar testes dos modelos treinados. Essa avaliação também pode ser chamada de *ex-ante* ou *holdout*.

3.2.3 Bases de Experimentos

Como o domínio da aplicação é previsão de séries temporais, nossa base de experimentos será formada por séries temporais. Um das competições mais importantes é a *M1-Competition* [MAC⁺82] e suas versões *M2-Competition* [MCH⁺93] e *M3-Competition* [MH00].

A *M1-Competition* é uma competição formada 1001 séries temporais subdivididas em várias categorias (micro, macro, indústria, etc). A primeira competição utilizou 15 modelos para o conjunto de séries disponíveis com mais 9 variações desses métodos. As conclusões mais importante da *M1-Competition* foram: métodos estatísticos sofisticados ou complexos não são necessariamente melhores para fornecer previsões mais precisas do que os mais simples; a classificação relativa do desempenho dos diversos métodos varia de acordo com a medida de precisão a ser utilizado; a precisão de vários métodos combinados supera, em média, a precisão individual dos métodos; e, a precisão dos diferentes métodos depende da duração do horizonte de previsão.

A *M2-Competition* foi mais uma tentativa de fornecer um fórum adicional para o estudo da precisão de vários métodos de previsão e compreender melhor os fatores que afetam a precisão da previsão. A *M2-Competition* incluiu 6 séries macroeconômica e foi projetado e executado com base em tempo real. As empresas envolvidas forneceram especialistas e dados reais sobre o passado e o presente, que também se comprometeram a responder às suas perguntas sobre os dados, os fatores que afetaram o seu negócio e as variáveis que eles estavam pensando, enquanto a previsão da série que foram dadas

aos participantes. Os resultados da *M2-Competition* foram praticamente idênticos aos da *M1-Competition*.

O objetivo principal da *M3-Competition* foi tanto replicar e quanto de estender as competições da *M1-Competition* e da *M2-Competition*. A extensão envolve a inclusão de mais métodos, investigadores (em especial nas áreas de redes neurais e sistemas especialistas) e mais séries. A replicação foi concebida para determinar se as principais conclusões da *M1-Competition* iria se aplicar ou não a *M3-Competition*. A *M3-Competition* é uma competição que mantém uma base de 3003 séries que incluem vários tipos de séries temporais (micro-economia, indústria, macro-economia, entre outras) com diferentes intervalos de observação (anual, trimestral, etc.).

Recentemente a rede EUNITE organizou uma competição mundialmente conhecida na previsão de carga de energia elétrica. Dada a temperatura e carga de energia elétrica entre os meses de Janeiro de 1997 e 1998, era solicitado a previsão da carga máxima diária de do mês de Janeiro de 1999. A principal técnica de aprendizagem de máquina utilizada foi *support vector machine* [CCL01]. EUNITE utilizou, ao total, 56 registros de 21 países. Os dados foi fornecido pela companhia eslovaca *Eastern Electricity Corporation* ¹ que tem grande interesse em previsão de carga elétrica [RP04].

Ainda podemos citar o repositório *UCR Time Series Classification/Clustering* ² criado como a intenção de ser um serviço público para a comunidade de mineração de dados e aprendizagem de máquina de forma a incentivar a investigação reproduzível para classificação de séries temporais e *clustering*.

Além dessas bases de experimentos, temos as *NN3-Competition* e *NN5-Competition*, que são um novo conjunto de dados de múltiplas frequências. A *NN3-Competition* são séries temporais mensais obtidas entre os anos de 2006 e 2007 e *NN5-Competition* são séries temporais diárias obtidas em 2008 ³.

3.3 TRABALHOS RELACIONADOS

Pesquisas de diversas disciplinas têm investigado o desempenho de algoritmos. Muitos estudos empíricos voltados para descobrir que algoritmo tem o melhor desempenho para um domínio particular ou um subconjunto de classes. Nesse contexto, como foi mostrado

¹<http://www.vse.sk/>

²http://www.cs.ucr.edu/~eamonn/time_series_data/

³<http://www.neural-forecasting-competition.com/>

no capítulo anterior, precisamos entender mais sobre as características do problema a fim de selecionar o algoritmo mais adequado em consideração das propriedades estruturais do problema ou da instância.

Segundo pesquisas de Smith-Miles [SM08], a comunidade de aprendizagem de máquina tenta responder perguntas como: “quais são as características ou instâncias particulares do problema que se correlacionam com o desempenho de algoritmo?”, “podemos definir a relação entre essas características e desempenho do algoritmo?”, entre outras.

O termo meta-aprendizado, uma aprendizagem sobre a aprendizagem, foi usado pela primeira vez, neste contexto, por [Aha92], no projeto europeu StatLog, cujo objetivo era relacionar o desempenho de algoritmos com as características ou medidas de classificação dos conjuntos de dados.

Como conseqüência do progresso dos estudos, houve um avanço no uso de conceitos de meta-aprendizado para resolver o problema de seleção algoritmo em uma variedade de domínios de aplicação. Os pré-requisitos para resolver o problema de seleção de algoritmos usam um (meta) algoritmo incluem:

- a. a disponibilidade de grandes conjuntos de instâncias de problemas complexos;
- b. a existência de um grande número de diferentes algoritmos;
- c. as métricas de desempenho para avaliar o desempenho do algoritmo; e,
- d. a existência de recursos adequados para caracterizar as propriedades das instâncias.

Combinando as características (d) com as métricas de desempenho (c) através de um grande número de instâncias (a) resolvido por diferentes algoritmos (b) cria um conjunto de meta-dados ou meta-conhecimento sobre o desempenho do algoritmo.

Então, métodos de aprendizagem de máquina podem ser usados para desenvolver modelos de seleção automática de algoritmos, modelos de algoritmo de classificação, as combinações de algoritmos, algoritmos auto-adaptativos, entre outros, a depender da necessidade, como por exemplo: aprendizagem sobre regressão, previsão, classificação, satisfação de restrição, algoritmo de otimização de desempenho.

O uso de características das séries temporais para selecionar o modelo de previsão apropriado é algo que vem sendo estudado desde a década de 1990. Essa Seção, apresentará os trabalhos relacionados a meta-aprendizado para o problema de seleção de modelos

de previsão. A Tabela 3.1 mostra um resumo dos trabalhos relacionados sobre abordagens para previsão de séries temporais.

Collopy e Armstrong [CA92] usaram as características das séries temporais que através de um sistema baseado em regras que gerou 99 regras para ponderação de quatro modelos diferentes: Alisamento Exponencial de Holt e Brown, *Random Walk*, Regressão Linear; as 18 características foram obtidas através de julgamentos de inspeção visual das séries temporais e utilizando o conhecimento do domínio. As séries temporais utilizadas foram 126 séries da competição *M1-Competition*.

Vokurka et al. [VFP96] extraiu 5 características automaticamente, utilizando 3 modelos individuais e 1 combinação baseada em regras criada automaticamente, mas que requeria revisão manual das saídas. Vokurka também utilizou 126 séries da competição *M1-Competition*.

Arinze et al. [AKA97] demonstrou a aproximação com resultados empíricos, utilizando: 67 séries temporais; 3 métodos de previsão: Filtragem Adaptativa (AD), Alisamento exponencial de Holt (HT), e Alisamento Exponencial de Winter (WT), que também foram combinados em 3 métodos híbridos: Média Móvel + HT, AD + WT, e WT + HT; foi utilizado média do erro padrão; 6 características, granularidade de dados (trimestral ou anual), *turning point*, o coeficiente de autocorrelação, a tendência, coeficiente de regressão, coeficiente de determinação do modelo de regressão, e médio do quadrado do erro do modelo de regressão. O método de meta-aprendizado utilizado foi um sistema baseado em conhecimento.

Shah [Sha97] desenvolveu uma regra de seleção individual através de análise discriminante e comparou seu desempenho com a seleção agregado para as séries trimestrais dos dados da *M1-Competition*. Medidas de precisão de previsões foram utilizadas para avaliação e intervalos de confiança foram construídos utilizando *bootstrapping*. Os resultados indicaram que a regra de seleção individual com base em escores de discriminantes é mais preciso, significativamente, do que qualquer outro método de seleção de agregação. A análise de discriminante foi utilizada para selecionar entre os 3 métodos de previsão e utilizou 26 características. Shah utilizou 203 séries temporais da *M1-Competition*.

Venkatachalam e Sohl [VS99] seguiram os estudos de Arinze et al. [AKA97], propondo uma abordagem em duas fases de rede neural. Na primeira fase, uma série temporal foi mapeada, com base nas características, para um dos 3 grupos de algoritmos: algoritmos flexíveis adaptáveis a uma variedade de tendências, ou seja, Alisamento Exponencial Linear de Holt (*2-Parameter*), Alisamento Exponencial de Winter *3-Parameter*, e Tri-

plo Alisamento Esponencial de Brown; algoritmos que respondem a tendências lineares: Duplo Alisamento Esponencial de Brown, Regressão Linear, e *Adaptative Response*; e, algoritmos simples (*naïve deseasonalized*, Alisamento Esponencial Simples, *Simple Moving Average*).

Assim, há 9 algoritmos considerados, agrupados em três grupos. Uma vez que a primeira rede neural determinou que o grupo fosse mais adequado para uma dada série, uma segunda rede neural é então aplicada para determinar qual dos três algoritmos dentro do grupo é susceptível de produzir o menor erro de previsão.

Foram utilizadas 180 séries temporais da *M3-Competition*, 9 algoritmos divididos em três grupos, conforme descrito anteriormente; erro médio percentual absoluto (MAPE) e um conjunto de 6 características para medir o tamanho das séries temporais, período de tempo, o tipo de dados (por exemplo, macro, micro, demográfica, etc), a tendência básica, a tendência recente, e variabilidade da série medida pelo modelo de regressão R2. Os resultados demonstraram uma precisão de previsão comparado com o algoritmo mais preciso de 50%.

Adya et al. [AACK00, ACJAK01] mais tarde modificou o sistema de Collopy e Armstrong [CA92] e reduziram a necessidade de entradas de pessoas, não abandonando por completo a intervenção de especialistas. O sistema baseado em regras utilizado foi originalmente desenvolvido, testado e validado apenas em dados anuais. Para utilizar a *M3-Competition*, três grandes modificações foram feitas no sistema. Uma delas foi à redução no número de regras na base de 99 para 64 regras, que resultou em perda de precisão. Foram utilizada 26 características das séries temporais e os seguintes modelos de previsão: *Exponential Smoothing (Holt and Brown)*, *Random Walk*, Regressão Linear. As melhores previsões do sistema foram para os dados anuais do que para mensal e trimestral. Esta conclusão é particularmente verdade para as previsões relativas aos horizontes longos.

Prudêncio e Ludermir [PL04] apresentaram dois estudos de caso. O primeiro estudo de caso utilizou 99 séries estacionárias a partir da série temporal, 2 algoritmos de previsão (Alisamento Exponencial Simples e Rede Neural), 14 características séries temporais, as propriedades estatísticas de autocorrelação, o coeficiente de variação para verificar a estabilidade, a assimetria e curtose, e os testes de *turning points*, como medida de aleatoriedade. Os algoritmos utilizados foram J4.8 (uma implementação Java do algoritmo de árvore de decisão C4.5 Weka) para determinar as regras para decidir qual dos dois algoritmos é esperado para produzir o menor erro.

O segundo estudo de caso de Prudêncio e Ludermir [PL04] investigou a seleção de modelos para prever as 645 séries anuais de *M3-Competition* relacionadas ao domínio econômico e demográfico. Três modelos candidatos foram utilizados para previsão de séries, *Random Walk*, *Holt Smoothing*, Auto-regressivo. A abordagem NOEMON de Kalousis e Theoharis [KT99] foi utilizada como meta-aprendiz para gerar um ranking de modelos para cada série temporal fornecida como entrada, de acordo com as características da série e uma função de erro. Esse ranking foi gerado de acordo com a medida MAE, no qual o modelo de melhor posição obtinha o menor erro MAE e assim por diante. Os dados foram divididos em três conjuntos: treinamento, usado para ajustar o peso das redes; validação, usado para selecionar a melhor configuração da rede; e, teste, usado para avaliar as redes selecionadas.

Simultaneamente Wang [WAN05], propõe um método para agrupamento de séries temporais com base em suas características estruturais. Ao contrário de outras alternativas, este método não agrupa valores usando uma métrica de distância, mas tomam como base características globais extraídas da série histórica. As características são obtidas a partir de cada série individual e pode ser alimentado em algoritmos de agrupamento arbitrário, incluindo um algoritmo de rede neural não supervisionada, de auto-organização do mapa, ou algoritmo de agrupamento hierárquico. As medidas globais que descrevem a série temporal são obtidas através da aplicação de operações estatísticas que melhor capturam as características subjacentes: tendência, sazonalidade, periodicidade, correlação serial, assimetria, curtose, o caos, não-linearidade e auto-similaridade. Isso, reduz a dimensionalidade da série temporal e é menos sensível a ruídos. Além disso, fornecem um mecanismo de busca para encontrar a melhor seleção do conjunto de características que devem ser utilizadas como agrupamento de entrada. Essa técnica foi testada usando conjuntos de séries temporais de referência para o agrupamento e um outro conjunto de séries temporais com características conhecidas. Os resultados empíricos mostram que a abordagem foi capaz de produzir agrupamentos significativos.

Wang [WSMH09] utilizou 315 séries temporais do repositório *UCR Time Series Data Mining Archive*, e vários conjuntos de dados sintéticos com características definidas em matéria de tendência, sazonalidade, ruído, etc.; 4 métodos de previsão, *Random Walk*, *Smoothing*, *ARIMA*, Rede Neural; 13 características (cerca de com base em dados brutos e em tempo algum série ajustada para tendência e sazonalidade), caracterizando tendência, sazonalidade, correlação serial, não-linearidade, assimetria, curtose, auto-similaridade, caos e periodicidade. Dois algoritmos de aprendizagem foram adotados. Em primeiro lugar, as regras foram aprendidas com *C4.5*. Em segundo lugar, as séries

temporais foram agrupados com base em 13 características, e regras foram inferidos com base nos membros do cluster e analisou o desempenho dos algoritmos diferentes dentro de cada cluster. Precisão de mais de 80% foram obtidos para prever qual algoritmo deve ser selecionada.

O trabalho de Lemke e Grabys [LG10] teve como objetivo investigar a aplicabilidade de meta-aprendizado em duas abordagens, a primeira para adquirir conhecimento sobre que modelo funcionaria melhor para um problema específico; e a segunda, para melhorar a desempenho da previsão.

Investigaram uma aproximação automática, uma vez que uma análise exaustiva de séries temporais por seres humanos muitas vezes não é viável em aplicações práticas desse processo um grande número de séries de tempo em tempo limitado. Algumas séries temporais características apresentadas neste trabalho são semelhantes aos utilizados na literatura. Um experimento inicial foi realizado com árvores de decisão que tentam encontrar uma ligação entre as características das séries temporais e o desempenho dos modelos. Quatro técnicas de meta-aprendizado foram comparados avaliando potenciais melhorias no desempenho.

Dois conjuntos de dados tanto constituído de 111 séries temporais têm sido utilizados neste estudo, foram obtidos a partir das competições NN3 e NN5 previsão da rede neural. Na base NN3, os dados incluem séries cronológicas mensais empírica negócios com 52-126 observações, enquanto que a base NN5 as séries temporais são diárias de retiradas da máquina de dinheiro com 735 observações cada.

Os algoritmos utilizados nos experimentos forma: árvore de decisão, rede neural e SVM. Foram utilizados ainda os seguintes modelos de previsão: *Moving Average*, Alisamento Exponencial Simples, Alisamento Exponencial de Taylor, Regressão Polinomial, ARIMA; e, as combinações *Simple Average*, *Simple Average with Trimming*, *Variance-based Model*, *Outperformance Method*, *Variance-based Pooling*.

Tabela 3.1. Revisão de Literatura de Seleção de Modelo de Séries Temporais

Referência	Características	Meta-aprendizado	Séries	Modelos Utilizados
Collopy and Armstrong 1992	18	Baseado em Regras	126 (M1)	Alisamento Exponencial de Holt e Brown, <i>Random Walk</i> , Regressão Linear
Vokurka et al. 1996	5	Baseado em Regras (parcialmente autotômático)	126 (M1)	Alisamento Exponencial de Garden e Simplex, Estrutural e Combinação de árvores
Arinze et al. 1997	6	Baseado em Regras	67	Alisamento Exponencial de Holt e Winter, <i>Adaptive Filtering</i> , Árvores Híbridas de Previsão
Shah 1997	26	Análise das Discriminantes	203 (M1)	Alisamento Exponencial de Holt-Winter e Simplex, Estrutural
Venkat. e Sohl 1999	6	Redes Neurais	180 (M3)	Alisamento Exponencial de Holt, Alisamento Exponencial de Winter, Alisamento Exponencial de Brown (Triple), Alisamento Exponencial de Brown (Double), Regressão Linear, <i>Adaptive Response</i> , <i>Naive deseasonalized</i> , Alisamento Exponencial Simplex, <i>Simplex Moving Average</i>
Adya et al. 2000	26	Baseado em Regras (julgamento)	122 (M1)	Exp. Smoothing (Holt and Brown), Random Walk, Regressão Linear
Prudencio and Lude-mir 2004	14	Árvore de Decisão (C4.5)	99 (M3)	Alisamento Exponencial Simplex, Rede Neural
Prudencio and Lude-mir 2004	5	NOEMON	645 (M3)	<i>Random Walk</i> , Alisamento Exponencial de Holt, Auto-regressivo de Box-Jenkins
Wang et al. 2009	13	Árvore de Decisão	315 (UCR)	<i>Random Walk</i> , Alisamento Exponencial Simplex, ARIMA, Rede Neural
Lemke e Grabys 2010	18 /56	Árvore de Decisão / Rede Neural / SVM	111 (NN3)/ 111 (NN5)	<i>Moving Average</i> / Alisamento Exponencial Simplex / Alisamento Exponencial de Taylor / Regressão Polinomial / ARIMA / <i>Simple Average</i> / <i>Simple Average with Trimming</i> / <i>Variance-based Model</i> / / <i>Outperformance Method</i> / <i>Variance-based Pooling</i>

ANÁLISE EXPERIMENTAL

Na literatura encontramos trabalhos que já utilizaram meta-aprendizado para seleção de modelos de previsão. Em [SPL04], meta-aprendizado é utilizado para selecionar modelos baseada em informação de desempenho.

Segundo [Pru04] a aplicação técnica de meta-aprendizado depende da construção de um conjunto de exemplos de treinamento que representam series temporais associadas a modelos candidatos. Cada série é descrita por um conjunto de atributos que caracterizadas de forma automática a partir dos dados da série. O uso de características do domínio dependeria de uma análise feita por um analista em cada série temporal usada como exemplo poderia ser muito demorada.

Nesse capítulo, apresentaremos a avaliação dos resultados obtidos nos experimentos que foram realizados para demonstrar a eficácia do meta-aprendizado para seleção automática de modelos de séries temporais.

Na próxima seção, veremos detalhes sobre a construção dos meta-exemplos e as características usadas para descrição das séries anuais. Detalhes de cada modelo serão vistos na subseção 4.2. O processo de avaliação dos modelos para definição do atributo classe associado aos meta-exemplos é descrito na seção 4.3.

4.1 CONSTRUÇÃO DOS META-EXEMPLOS

Os experimentos foram realizados utilizando séries temporais da *M3-Competition* [Tas00] para geração de meta-exemplos. O nosso trabalho utiliza um subconjunto de séries anuais que contém 645 séries, com tamanho mínimo de 14 observações e o tamanho médio de 19 observações.

A partir das 645 séries foram gerados 645 meta-exemplos. Cada meta-exemplo armazena 5 meta-características de uma série e o valor de um atributo-alvo (classe). Esse atributo indica que modelo de previsão obteve o melhor desempenho experimental para

a série, dentre três modelos candidatos comumente usados para prever as séries anuais da *M3-Competition*: Autoregressivo de Box-Jekings (AR) [BJ70], *Random Walk* (RW) [Har93] e Alisamento Exponencial de Holt (HL) [HMMS60] conforme apresentado na Seção 3.1.

Um conjunto de 5 meta-atributos foram usados para descrever as series anuais, entre eles:

- * Tamanho da série temporal - número de observações da série.
- * Tendência básica - indicado pela estatística t da inclinação do modelo de regressão linear. Quanto mais alto o valor da variável, mais forte é a tendência global da série.
- * *Turning points* - essa medida tenta capturar o grau de oscilação da série.
- * Autocorrelação - mede o grau de correlação entre o valor da série no tempo t e o valor no tempo $t - 1$.
- * Classe - variável categoria que representa o domínio da série.

4.2 ALGORITMOS AVALIADOS

Os algoritmos utilizadas neste trabalho são implementações do WEKA (*Waikato Environment for Knowledge Analysis*) implementado na linguagem Java. Cada implementação representa uma família diferente de algoritmos. Os experimentos foram realizados no ambiente *Experimenter* do WEKA versão 3.6. Inicialmente, foi realizada a avaliação de desempenho de cinco algoritmos descritos a seguir:

- a. ZeroR: Esse classificador fornece predições definido a classe majoritária dos exemplos de treinamento como a classe predita para os exemplos de teste. Pode ser chamado também de classificador *default*. Embora o classificador ZeroR seja trivial, ele serve como base de comparação de desempenho para algoritmos mais complexos. De fato, se um algoritmo mais complexo obtiver uma precisão mais baixa comparada ao classificador ZeroR para um dado conjunto de dados, isso indica que o algoritmo não é adequado para o problema ou que o conjunto não contém informação útil para aprendizado.

- b. IBk: O algoritmo IBk é uma derivação do algoritmo k-NN (k-Nearest Neighbors) [AKA91]. O k-NN é um dos algoritmos mais simples dentre os algoritmos baseados em instâncias. O conjunto de treinamento é formado por vetores n-dimensionais e cada elemento deste conjunto representa um ponto no espaço n-dimensional. A vizinhança de uma instância é definida em termos de uma função de distância (e.g., distância Euclidiana) ou de uma função de similaridade. A predição da instância pode ser feita considerando a classe majoritária dentre a vizinhança da instância (com e sem ponderação das instâncias mais similares). No nosso experimento, utilizamos a opção *cross-validate* igual a true para definir tamanho da vizinhança (parâmetro k). Utilizamos ainda a opção de ponderar a contribuição de cada um dos k vizinhos de acordo com a distância, dando maior peso para os vizinhos mais próximos.
- c. Segundo [Bre01], *Random Forest* é um dos algoritmos mais versáteis, pode trabalhar com um grande número de dados e seu funcionamento é baseado em árvores de decisão. O algoritmo *Random Forest* gera N árvores de classificação. Cada árvore possui uma determinada classificação, indicando em que classe deve votar. Quando um novo objeto é fornecido para a classificação, uma votação é realizada para definir a classe mais popular, e esse novo objeto é rotulado com essa classe. Cada árvore é construída usando-se uma amostra de *bootstrap* (método genérico para estimar variabilidade em estatística), diferente dos dados originais. Um terço dos casos é omitido da amostra de bootstrap e não são usados na construção da árvore. Cada caso omitido na construção da árvore é colocado abaixo dela, e faz-se a classificação. Deste modo, um teste de classificação é obtido para cada caso sobre um terço das árvores. Ao término da execução, tem-se a classe mais votada.
- d. J48: O classificador J48 tem como base o algoritmo C4.5 [Rod09], classificador tradicional e pertencente à família TDIDT (Top-down Induction of Decision Tree). O classificador J48 gera uma árvore de decisão a partir da abordagem recursiva de particionamento da base de dados. O algoritmo descarta atributos que geram regras com quantidade de objetos inferior ao parâmetro definido.
- e. SMO: O algoritmo SMO é uma versão do WEKA para *Support Vector Machines* (SVMs). O algoritmo SVM se baseia na idéia de encontrar um hiperplano ótimo que separe dois conjuntos de pontos linearmente separáveis no espaço. Ao se encontrar um hiperplano que separe esses dois conjuntos, a classificação de um novo ponto torna-se trivial, pois basta verificar em que região (à esquerda ou à direita do hiperplano) se encontra o novo ponto. No entanto, podem existir infinitos hiperplanos

que separam dois conjuntos de pontos linearmente separáveis no espaço. É possível demonstrar que o hiperplano cuja margem para os pontos mais próximos (vetores suporte) é a maior, é o hiperplano que minimiza o risco de se classificar erroneamente um novo ponto. O desafio do algoritmo de SVM é, portanto, o de encontrar o hiperplano ótimo que tenha a maior margem para os pontos mais próximos a ele. Isso é feito através de técnicas de otimização com restrições. Para encontrar hiperplanos em situações onde os conjuntos de pontos não são inteiramente separáveis (i.e. aproximadamente separáveis de forma linear), as SVMs permitem que pequenos erros sejam tolerados, atribuindo, no entanto, uma penalidade (parâmetro C) para as classificações incorretas. A generalização não-linear de SVM é baseada em funções de kernel, que mapeiam vetores de características em um espaço de alta dimensão, onde os pontos mapeados sejam linearmente separáveis. Nesse, novo espaço de alta dimensão, cria-se o hiperplano ótimo separando as classes. No nosso experimento, utilizamos o kernel RBF (com parâmetro γ igual ao valor default do WEKA). O valor do parâmetro C também foi definido como o valor default do WEKA ($C = 1$).

4.3 ESTUDO DE CASO I

Esse estudo tem como objetivo avaliar o desempenho de diferentes algoritmos para selecionar modelos de previsão (conforme as bases de experimentos descritas na seção anterior).

No primeiro estudo de caso, foi voltado para analisar o desempenho dos modelos para prever séries temporais. Esse estudo de caso utilizou 3 modelos candidatos: Auto Regressivo de Box-Jekings (AR), Random Walk (RW) e Alisamento Exponencial de Holt (HL); 5 algoritmos para aquisição de conhecimento automático: ZeroR, IBk, *Random Forest*, SMO e J48. Os algoritmos avaliados tiveram a seguintes configurações:

- ZeroR: foi utilizado como base;
- IBk: foi configurado com o cross-validation igual à true;
- *Random Forest*: foi utilizado o número de árvore igual a 100;
- J48: foi selecionada a configura default; e,

- SVM: foi selecionado o algoritmo SMO com o kernel RBFKernel e com as configurações default.

Cada um dos algoritmos acima foi avaliado em cada uma das 6 bases de experimentos. A metodologia de experimentos foi a *k-fold cross-validation* com $k = 10$ e 10 repetições. O *k-fold cross-validation* consiste em dividir os exemplos em grupos de k elementos (no nosso caso, $k = 10$), e depois treinar o sistema com todos os grupos, à exceção do *grupo i* que é deixado fora durante o treino para testar o classificador. Este processo é repetido k vezes, variando i de 1 até k . Como esse experimento, depende da ordem das instâncias, ele foi repetido 10 vezes com ordem aleatória para os exemplos.

4.3.1 Análise dos Resultados

Podemos observar, na Tabela 4.1, o desempenho dos algoritmos avaliados, considerando os diferentes horizontes de previsão das séries da *M3-Competition*.

O algoritmo *Random Forest* obteve o melhor resultado, apresentando um melhor desempenho em relação ao algoritmo base ZeroR em todos os horizontes de previsão das séries temporais. E também teve ganho estatístico para os seis horizontes.

O segundo melhor algoritmo identificado foi o IBk, superando o desempenho do algoritmo ZeroR para todo os horizontes, mostrando um bom desempenho para a maioria dos casos, perdendo apenas para o J48 quando o horizonte de previsão foram 3 e 5. IBk teve ganho estatístico para quase todos os horizontes, menos para o horizonte 5.

Já o J48 apresentou um bom desempenho apenas para os horizontes 3 e 5, em relação ao IBk. Apresentou um ganho estatístico em relação a quase todos os horizontes, menos o horizonte 4.

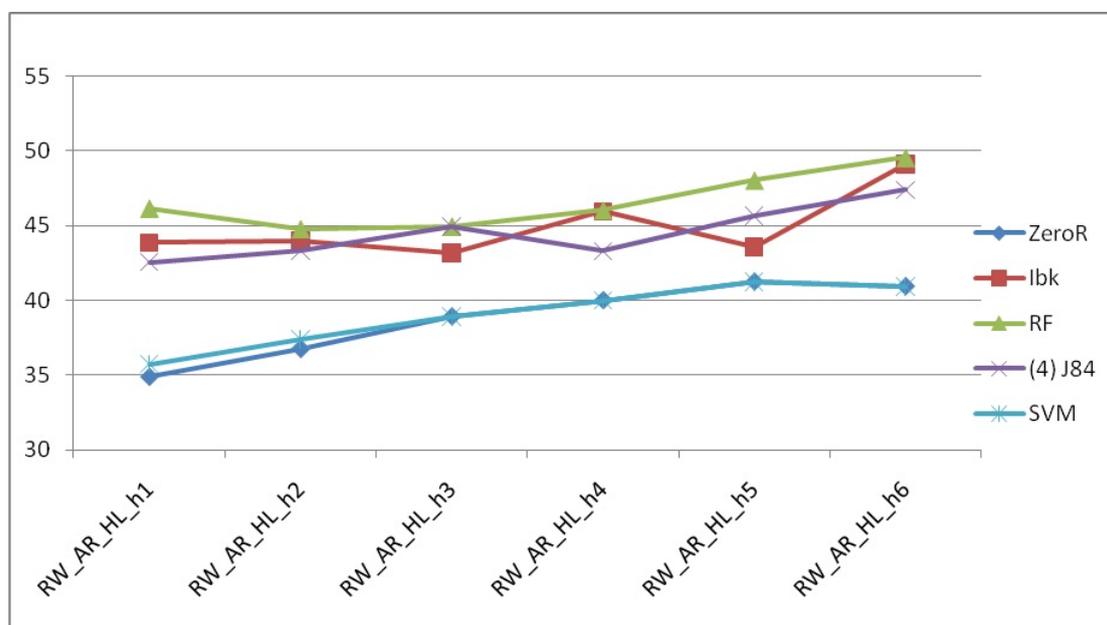
E, por fim, o SVM, que apresentou desempenho equivalente ao ZeroR. É interessante observar que nesse experimento inicial com SVMs, foram utilizados os valores *default* para o parâmetro γ do kernel RBF e o parâmetro C , que não se mostraram adequados para as bases em questão. E não houve nenhum ganho estatístico.

A Figura 4.1 nos possibilita observar que o algoritmo *Random Forest* teve o melhor desempenho independente do horizonte de previsão. Os algoritmos IBk e J48 apresentaram um desempenho menor, mas próximo do *Random forest*. No entanto, para os horizontes 3 e 5 o IBk obteve um desempenho menor que o J48. E o SVM, demonstrou

Tabela 4.1. *Ranking* dos algoritmos avaliados

Dataset	ZeroR	IBk	Random Forest	J48	SVM
RW AR HL h1	34.88	43.86 v	46.15 v	42.56 v	35.75
RW AR HL h2	36.74	44.01 v	44.77 v	43.35 v	37.42
RW AR HL h3	38.92	43.18 v	44.95 v	44.94 v	38.92
RW AR HL h4	40.00	45.95 v	46.02 v	43.29	40.00
RW AR HL h5	41.24	43.57	48.03 v	45.67 v	41.24
RW AR HL h6	40.93	49.11 v	49.55 v	47.38 v	40.93

um desempenho próximo ao algoritmo tomado como base, ZeroR.

**Figura 4.1.** Desempenho (precisão) dos Algoritmos Avaliados

A Tabela 4.2 apresenta a avaliação das medidas de erro: MSE, MAE e PB para cada modelo de previsão. Podemos observar que considerando a métrica MSE, o algoritmo *Random Forest* possui um desempenho melhor que os demais algoritmos. O IBk ficou em segundo lugar, seguido pelo algoritmo J48 com uma diferença mínima. O algoritmo de pior desempenho foi o SVM que apresentou precisão igual ao do classificador base ZeroR. Esses resultados eram esperados uma vez que o mesmo ranking dos algoritmos foi observado para os horizontes de previsão individualmente.

Para a medida de avaliação MAE, o algoritmo *Random Forest* também apresentou

Tabela 4.2. Avaliação dos Modelos de Previsão pelas Medidas de Erro

Dataset	ZeroR	IBk	Random Forest	J48	SVM
RW AR HL MSE	43.72	46.52	49.47 v	46.50	43.72
RW AR HL MAE	42.02	45.70	48.37 v	46.80 v	42.02
RW AR HL PB	45.43	49.56 v	52.01 v	49.40 v	45.43

o melhor desempenho em relação aos demais algoritmos. No entanto, mostra que o algoritmo J48 tem um bom desempenho ficando em segundo lugar. O IBk ficou em terceiro lugar. E o algoritmo de pior desempenho continuou sendo o SVM com precisão igual ao do classificador base ZeroR.

Finalmente, a medida de avaliação PB mostra que os algoritmos *Random Forest*, IBk e J48 com um bom desempenho, se destacando ainda o *Random Forest* com o melhor desempenho. E o SVM permanece com a precisão igual ao desempenho do algoritmo ZeroR.

Além disso a Tabela 4.2 nos permite observar que entre os resultados obtidos nas bases com as medidas de erro MSE, MAE e PB, é mais fácil selecionar modelos utilizando a medida PB.

Tabela 4.3. Distribuição das Classes

Modelos de Previsão	Holt	AR	RW
Horizonte 1	225	197	223
Horizonte 2	237	209	109
Horizonte 3	251	196	198
Horizonte 4	258	194	193
Horizonte 5	173	206	266
Horizonte 6	264	188	193

Observando-se detalhadamente a Tabela 4.3, verificamos que para horizontes mais recentes a previsão dos modelos de séries temporais HL, AR e RW se mostram equivalentes em relação à distribuição das classes.

Para o Horizonte 1 podemos verificar que não houve uma diferença significativa na distribuição das classes.

Ao longo do tempo (passo 2, 3 e 4), pode-se observar que a distribuição das classes

passa a não ficar mais balanceada, o modelo HL possui mais classes que os demais modelos AR e RW, que se mostram equivalentes.

Para previsões mais distantes (passo 5 e 6), o modelo HL permanece com uma grande quantidade de classes, no entanto os modelos AR e RW não se mostram mais equivalentes. No passo 5, o modelo AR possui mais classes; e, no passo 6, o modelo RW possui mais classes.

O trabalho utilizou a avaliação *out-of-sample*, dividindo o grupo o conjunto de dados em dados de treinamento e dados de teste. A base foi dividida em subconjuntos de 10 meta-exemplos. Bases de meta-exemplos foram geradas considerando cada horizonte individualmente. As medidas MSE, MAE e PB foram calculadas levando em consideração todos os horizontes.

Cada série temporal foi dividida em dados de treinamento e dados de teste (uma avaliação *out-of-sample*). Os dados de teste correspondem a 6 horizontes de previsão, ou seja, 6 últimos anos. Os dados de treinamento correspondem a parte inicial restante da série. Com os dados de teste foi possível avaliar os modelos de previsão em 6 pontos separadamente, o que resulta em 6 bases de meta-exemplos (uma base associada a cada horizonte).

Foi possível ainda avaliar os modelos levando em consideração os 6 pontos conjuntamente. Para isso, usamos as medidas MSE, MAE e PB para avaliar os modelos nos 6 pontos e conseqüentemente mais 3 bases de meta-exemplos foram gerados.

4.4 ESTUDO DE CASO II

Devido ao fraco desempenho apresentado pelo algoritmo *support vector machine* com a configuração padrão no primeiro estudo de caso, realizamos um segundo experimento variando os parâmetros C e γ . Os parâmetros variaram de 2^{-3} , 2^1 , 2^1 , 2^3 .

A Subseção 4.4.1 mostrará a análise das variações dos parâmetros para cada horizonte. As tabelas que serão apresentadas mostram a variação do parâmetro C nas linhas e a variação do parâmetro γ nas colunas, evidenciando em verde os melhores desempenhos do algoritmo e em vermelho o piores desempenho.

4.4.1 Análise dos Resultados

Conforme exibido na Tabela 4.4, tabela a qual exibe o desempenho do algoritmo *support vector machine* para o horizonte 1, o melhor desempenho apresentado foi quando os parâmetros $C = 8$ e $\gamma = 8$, seguido de $C = 2$ e $\gamma = 8$, e $C = 8$ e $\gamma = 2$. E os piores desempenhos para $C = 0.125$ e $\gamma = 0.125$, seguido de $C = 0.125$ e $\gamma = 0.5$, e $C = 0.5$ e $\gamma = 0.125$.

Pode-se observar ainda, que para o horizonte 1, houve um ganho estatístico de 50% do algoritmo *support vector machine* em relação ao algoritmo base ZeroR das configurações testadas (esse ganho é indicado pela letra v ao lado direito dos valores na tabela).

Tabela 4.4. Horizonte 1

C / Gama	0.125	0.5	2	8
0.125	37.23	37.31	37.63	40.80 v
0.5	37.59	38.15	39.23 v	42.47 v
2	39.42 v	39.65 v	40.13 v	44.71 v
8	39.91 v	40.64 v	42.52 v	45.89 v

A Tabela 4.5 exibe o desempenho para o horizonte 2, o melhor desempenho registrado foram para os parâmetros $C = 8$ e $\gamma = 8$, seguido de $C = 2$ e $\gamma = 8$, e $C = 8$ e $\gamma = 2$, coincidindo com a configuração para o horizonte 1. Já os piores desempenhos foram registrados para as configuração $C = 0.125$ e $\gamma = 0.125$, seguido de $C = 8$ e $\gamma = 0.5$, e $C = 2$ e $\gamma = 2$. Os dois últimos piores desempenho não coincidem com a configuração de pior desempenho para o horizonte 1.

Além disso, o horizonte 2, houve ganho estatístico para quase 100% do algoritmo *support vector machine* em relação ao algoritmo base ZeroR, com a única exceção da configuração $c = 0.125$ e $\gamma = 0.125$.

Tabela 4.5. Horizonte 2

C / Gama	0.125	0.5	2	8
0.125	38.98	44.72 v	45.19 v	45.67 v
0.5	45.05 v	44.94 v	44.65 v	46.03 v
2	44.62 v	44.69 v	43.99 v	46.48 v
8	44.79 v	43.61 v	44.76 v	44.87 v

A Tabela 4.6 exibe o desempenho para o horizonte 3, que diferente dos desempenhos

exibidos para o horizonte 1 e 2 não tem a configuração $C = 8$ e $\gamma = 8$ como melhor desempenho, e sim, a configuração $C = 8$ e $\gamma = 2$, seguidas das configuração $C = 2$ e $\gamma = 8$ e $C = 2$ e $\gamma = 2$. Os piores desempenhos foram registrados para as configuração $C = 0.125$ e $\gamma = 0.125$, seguido de $C = 0.5$ e $\gamma = 0.125$, e $C = 0.125$ e $\gamma = 8$.

O horizonte 3 teve ganho estatístico para de quase 70% do algoritmo *support vector machine* em relação ao algoritmo base ZeroR.

Tabela 4.6. Horizonte 3

C / Gama	0.125	0.5	2	8
0.125	38.81	43.14	43.24 v	39.27
0.5	42.60	43.51 v	43.92 v	47.89 v
2	42.97	43.82 v	46.97 v	48.85 v
8	43.44 v	44.59 v	49.44 v	46.57 v

A Tabela 4.7 exhibe o desempenho para o horizonte 4, com o melhor desempenho para as configurações $C = 8$ e $\gamma = 8$, $C = 2$ e $\gamma = 8$ e $C = 8$ e $\gamma = 2$. Os piores desempenhos foram registrados para as configuração $C = 0.125$ e $\gamma = 0.125$, seguido de $C = 8$ e $\gamma = 0.125$, e $C = 8$ e $\gamma = 0.5$.

Houve ganho estatístico, para o horizonte 4, de quase 90% do algoritmo *support vector machine* em relação ao algoritmo base ZeroR. Para esse horizonte o algoritmo *support vector machine* se mostrou muito sensível ao parâmetro γ , pois para $C = 8$, a variação do parâmetro γ obteve os piores e melhores desempenhos.

Tabela 4.7. Horizonte 4

C / Gama	0.125	0.5	2	8
0.125	39.85	43.24 v	43.16 v	43.55 v
0.5	42.78 v	43.33 v	42.95 v	43.11 v
2	43.60 v	42.78	43.69 v	45.66 v
8	42.73	42.76	44.00 v	46.60 v

A Tabela 4.8 exhibe o desempenho para o horizonte 5, com o melhor desempenho para as configurações $C = 8$ e $\gamma = 8$, $C = 2$ e $\gamma = 8$ e $C = 0.5$ e $\gamma = 8$. E os piores desempenhos foram registrados para as configuração $C = 0.125$ e $\gamma = 8$, seguido de $C = 2$ e $\gamma = 0.5$, e $C = 0.125$ e $\gamma = 0.125$.

Para o horizonte 5, observamos que ocorreu o oposto ao horizonte 4. No horizonte 5,

o algoritmo *support vector machine* se mostrou muito sensível ao parâmetro C , pois para $\gamma = 8$, a variação do parâmetro X obteve o pior e os melhores desempenhos.

Além disso, podemos verificar que para horizontes mais distantes, o desempenho do *support vector machine* foi similar ao desempenho do algoritmo base, com apenas um ganho, quando a configuração dos parâmetros foram igual a 8.

Tabela 4.8. Horizonte 5

C / Gama	0.125	0.5	2	8
0.125	41.24	42.59	42.47	41.12
0.5	43.06	43.10	41.26	44.08
2	42.73	41.13	42.75	44.79
8	41.80	41.68	44.02	46.26 v

A Tabela 4.9 exibe o desempenho para o horizonte 6, com o melhor desempenho para as configurações $C = 8$ e $\gamma = 8$, $C = 2$ e $\gamma = 8$ e $C = 0.5$ e $\gamma = 8$, similar ao desempenho apresentado pelo algoritmo para o horizonte 5 na Tabela 4.7. Os piores desempenhos foram registrados para as configuração $C = 0.125$ e $\gamma = 2$, seguido de $C = 0.125$ e $\gamma = 0.5$, e $C = 0.125$ e $\gamma = 0.125$. Para esse horizonte podemos verificar que o desempenho também é similar ao algoritmo base.

Tabela 4.9. Horizonte 6

C / Gama	0.125	0.5	2	8
0.125	40.93	40.97	40.91	41.95
0.5	41.91	42.04	42.26	42.54
2	42.28	41.51	41.67	46.59 v
8	41.92	41.70	43.41	49.50 v

De um modo geral, o algoritmo *support vector machine* se mostrou muito mais eficaz para a previsão com horizontes mais próximos, tendo ganho estatísticos bastante elevado em relação ao algoritmo base, ZeroR. Principalmente para os horizontes 3 e 4, que podem ser observados nas Tabela 4.6 e Tabela 4.7. E mostrando um desempenho muito próximo ao ZeroR quando os horizontes são mais distantes, como podemos observar nas Tabela 4.8 e Tabela 4.9.

Concluimos que a variação dos parâmetros do *support vector machine* trouxeram um ganho no desempenho. Dentre todas as configurações a que mostrou melhor desempenho em quase todos os horizontes foi a configuração $C = 8$ e $\gamma = 8$.

CONCLUSÃO

A tarefa de seleção automática de algoritmos está na intersecção de várias disciplinas: a teoria da complexidade computacional, teoria da informação algorítmica, inteligência artificial, aprendizado de máquina, para citar alguns. Técnicas de meta-aprendizado aplicado aos modelos de previsão de séries temporais tem sido objeto de estudo da comunidade de Inteligência Artificial. Nesse trabalho, focamos particularmente, em meta-aprendizado no contexto de previsão de séries temporais que tem sido usada em diferentes contextos para diminuir riscos na tomada de decisão.

No Capítulo 2 foram apresentados conceitos básicos sobre meta-aprendizado e uma introdução sobre sua utilização para a recomendação de algoritmos. Seguida por uma arquitetura de meta-aprendizado para recomendação de algoritmos, incluindo a descrição dos seus dois modos: de aquisição de conhecimento e seleção de algoritmos ou combinação dos mesmos. O modo de aquisição de conhecimento é responsável realizar a caracterização da série temporal e guardar informação relativa aos desempenhos desses modelos para sugerir os melhores modelos de previsão.

Inicialmente, quando a base de conhecimento ainda não possui informação, os modelos escolhidos de forma aleatória. Uma vez que a base de meta-conhecimento foi alimentada com a descrição dos desempenhos, o modo de sugestão pode utilizar o conhecimento armazenado para sugerir automaticamente o melhor modelo de previsão. Além disso, foram abordadas as principais técnicas de meta-aprendizagem, são elas:

- caracterização de conjunto de dados: caracterização por informação teórica e estatística, caracterização baseada em árvore de decisão e *landmarking*;
- mapeamento de conjunto de dados para modelos de previsão: meta-regras manuais, aprendizagem em meta-nível; modelo de *ranking*, *strict meta-learning* e meta-egressão;
- combinação de meta-aprendizes: *stacked generalization*, *boosting*, *meta-decision tree* e composição de aplicações indutivas.

O Capítulo 3 apresentou uma breve introdução a séries temporais, seguida das descrições dos modelos de previsão de séries temporais mais comuns, são eles: Alisamento Exponencial de Holt, Autoregressivo de Box-Jenkins e *Random Walk*. Foram apresentadas, ainda, abordagens de avaliação de previsão das séries temporais, como medidas de erros, métodos de amostragem e bases de experimentos. Uma última seção apresentou ainda, os trabalhos relacionados a essa área de estudo.

O Capítulo 4 abordou os algoritmos utilizados para realizar os experimentos, e para casa estudo de caso foram apresentados a metodologia utilizada, descrição dos conjuntos de dados (séries temporais), algoritmos avaliados e suas respectivas configurações e análise do desempenho dos experimentos. Os seguintes experimentos foram desenvolvidos para verificar a viabilidade da proposta:

- Estudo de Caso I: diferentes algoritmos foram utilizados no meta-aprendizado como o algoritmo kNN, árvores de decisão e *support vector machines* para realizar a seleção de modelos de previsão aplicados às séries anuais da *M3-competition* e método de amostragem *out-of-sample*. O desempenho dos modelos de previsão foi avaliado através das medidas de erro: MSE, MAE e PB.
- Estudo de Caso II: como o desempenho do algoritmo *support vector machines* não foi satisfatório no primeiro estudo de caso. Assim, realizamos um segundo experimento utilizando a mesma técnica empregada no Estudo de Caso I, variamos os parâmetros C e γ do *support vector machines*.

Os resultados desses estudos de caso mostraram que os algoritmos de aprendizado de fato são capazes de predizer os melhores modelos de previsão a partir das características das séries temporais.

5.1 CONTRIBUIÇÕES DA DISSERTAÇÃO

Nessa dissertação, apresentamos um estudo sobre meta-aprendizado para seleção automática de modelos de previsão de séries temporais. Previsão de séries temporais é uma área de crescente interesse, tanto por parte dos pesquisadores, quanto por parte das companhias e indústrias, por auxiliar a tomada de decisão. Abordagens de meta-aprendizado foram utilizadas para auxiliar a recomendação e/ou combinação de modelos para previsão de séries temporais. Considerando a previsão de séries temporais como um problema de

aprendizado uma arquitetura e técnicas foram adaptadas para realizar a recomendação automática de modelos. Podemos então apresentar como as principais contribuições:

- i. Revisão bibliográfica de técnicas de meta-aprendizado: o capítulo 2 apresenta diferentes abordagens de técnicas de meta-aprendizado;
- ii. Revisão bibliográfica dos trabalhos relacionada à aplicação de meta-aprendizado para previsão de séries temporais: o capítulo 3 aborda os trabalhos relacionados, apresentando as técnicas de meta-aprendizado utilizadas, a quantidade de meta-atributos utilizados para realizar a caracterização das séries temporais, as séries temporais utilizadas, e os modelos de previsão avaliados;
- iii. Contribuição na área de meta-aprendizado para seleção de modelos de previsão de séries temporais: foram realizados experimentos utilizando algoritmos de aprendizado de máquina ainda não avaliados em trabalhos anteriores. Além disso, foram realizados experimentos com diferentes bases de meta-dados, variando a forma de etiquetagem dos meta-exemplos considerando horizonte de previsão e medidas de erro de previsão.

5.2 LIMITAÇÕES E TRABALHOS FUTUROS

Uma das limitações da nossa proposta é utilização de um número reduzido de características das séries temporais usadas. Fazendo-se necessária também uma maior investigação sobre que características seriam mais relevantes.

Outro ponto a ser investigado pode ser a análise de parâmetros dos algoritmos, como visto no segundo estudo de caso, a análise do *support vector machines* verificou que o desempenho melhorou em relação ao desempenho obtido no primeiro estudo de caso, que inclusive para os horizontes com 2, 3 e 4 passos superaram o desempenho do melhor algoritmo que foi o *Random Forest*.

REFERÊNCIAS BIBLIOGRÁFICAS

- [AACK00] Monica Adya, J. Scott Armstrong, Fred Collopy, and Miles Kennedy. An application of rule-based forecasting to a situation lacking domain knowledge. *International Journal of Forecasting*, 16(4):477 – 484, 2000.
- [ACJAK01] Monica Adya, Fred Collopy, Scott J. Armstrong, and Miles Kennedy. Automatic identification of time series features for rule-based forecasting. *International Journal of Forecasting*, 17(2):143 – 157, 2001.
- [Aha92] David W. Aha. Generalizing from case studies : A case study. *Machine Learning*, 1(Section 2), 1992.
- [AKA91] D. W. Aha, D. Kibler, and M. K. Albert. Instance-based learning algorithms. *Machine learning*, 6(1):37–66, 1991.
- [AKA97] Bay Arinze, Seung-Lae Kim, and Murugan Anandarajan. Combining and selecting forecasting models using rule based-induction. *Comput. Oper. Res.*, 24(5):423–433, 1997.
- [AP96] Kamal M. Ali and Michael J. Pazzani. Error reduction through learning multiple descriptions. *Machine Learning*, 24(3):173–202, sep 1996.
- [Arm01] J. S. Armstrong. Extrapolation for time-series and cross-sectional data. *Principles of forecasting: a handbook for researchers and practitioners*, page 217, 2001.
- [AY04] Hidenao Abe and Takahira Yamaguchi. Constructive meta-learning with machine learning method repositories. In *IEA/AIE'2004: Proceedings of the 17th international conference on Innovations in applied artificial intelligence*, pages 502–511. Springer Springer Verlag Inc, 2004.
- [BGC00] Hilan Bensusan and Christophe Giraud-Carrier. *Discovering Task Neighbourhoods through Landmark Learning Performances*, volume Volume 1910/2000 of *Lecture Notes in Computer Science*, pages 136–137. Springer Berlin / Heidelberg, January 2000. 10.1007/3-540-45372-5.

- [BGCK00] H. Bensusan, C. Giraud-Carrier, and C. Kennedy. A higher-order approach to meta-learning. *ECML'2000 Workshop on Meta-Learning: Building Automatic Advice Strategies for Model Selection and Method Combination*, pages 109–117, 2000.
- [BGCSV08] Pavell Brazdil, Christophe Giraud-Carrier, Carlos Soares, and Ricardo Vialta. *Metalearning Applications to Data Mining*. Springer, June 2008.
- [BJ70] G. E. Box and G. M. Jenkins. *Time Series Analysis: Forecasting and Control*. Holden-Day, San Francisco, CA, 1970.
- [BK01] Hilan Bensusan and Alexandros Kalousis. Estimating the predictive accuracy of a classifier. In *EMCL '01: Proceedings of the 12th European Conference on Machine Learning*, pages 25–36, London, UK, 2001. Springer-Verlag.
- [BL96] Carla Brodley and Terran Lane. Creating and exploiting coverage and diversity. In *In Work. Notes AAAI-96 Workshop Integrating Multiple Learned Models*, pages 8–14. In Work. Notes AAAI-96 Workshop Integrating Multiple Learned Models, 1996.
- [BPK00] Helmut Berrer, Iain Paterson, and Jorg Keller. Evaluation of machine-learning algorithm ranking advisors. In *Proceedings of the PKDD-2000 Workshop on Data-Mining, Decision Support, Meta-Learning and ILP: Forum for Practical Problem Presentation and Prospective Solutions*. Citeseer, 2000.
- [Bre96] Leo Breiman. Stacked regressions. *Mach. Learn.*, 24(1):49–64, 1996.
- [Bre01] Leo Breiman. Random forests. *Machine learning*, pages 1–33, 2001.
- [Bro95] Carla E. Brodley. Recursive automatic bias selection for classifier construction. *Machine Learning*, 20(1-2):63–94, 1995.
- [BS00] Pavel B. Brazdil and Carlos Soares. Zoomed ranking: Selection of classification algorithms based on relevant performance information. *Principles of Data Mining and Knowledge Discovery*, pages 6–8, 2000.
- [BSC03] Pavel B. Brazdil, Carlos Soares, and Joaquim Pinto da Costa. Ranking learning algorithms: Using ibl and meta-learning on accuracy and time results. *Mach. Learn.*, 50(3):251–277, 2003.

- [CA92] Fred Collopy and J. Scott Armstrong. Rule-based forecasting: development and validation of an expert systems approach to combining time series extrapolations. *Manage. Sci.*, 38(10):1394–1414, 1992.
- [CCL01] Ming-wei Chang, Bo-Juen Chen, and Chih-Jen Lin. Eunite network competition: Electricity load forecasting. Technical report, National Taiwan University, 2001.
- [Cha97] Philip K. Chan. On the accuracy of meta-learning for scalable data mining. *Computer*, 28:5–28, 1997.
- [DZ04] Saso Dzeroski and Bernard Zenko. Is combining classifiers with stacking better than selecting the best one? *Machine Learning*, 54(3):255–273, mar 2004.
- [ET98] Robert Engels and Christiane Theusinger. Using a data metric for preprocessing advice for data mining applications. In *In Proceedings of the European Conference on Artificial Intelligence ECAI-98*, pages 430–434. John Wiley e Sons, 1998.
- [FHT00] J. Friedman, T. Hastie, and R. Tibshirani. Additive logistic regression: a statistical view of boosting (with discussion and a rejoinder by the authors). *The annals of statistics*, 2000.
- [FP01] J. Furnkranz and Johann Petrak. An evaluation of landmarking variants. *Working Notes of the ECML/PKDD 2001 Workshop on*, 2001.
- [FS96] Yoav Freund and Robert E. Schapire. Experiments with a new boosting algorithm. *MACHINE LEARNING-INTERNATIONAL*, 1996.
- [GB95] João Gama and Pavel Brazdil. *A Characterization of Classification Algorithms*, pages 189–200. Springer Berlin / Heidelberg, 1995.
- [GB00] João Gama and Pavel Brazdil. Cascade generalization. *Machine Learning*, pages 315–343, 2000.
- [GC98] Christophe Giraud-Carrier. Beyond predictive accuracy: What? Technical report, University of Bristol, Bristol, UK, UK, 1998.
- [Har93] A. C. Harvey. *Time Series Models*. MIT Press, Cambridge, MA, 1993.

- [HK01] Melanie Hilario and Alexandros Kalousis. Fusion of meta-knowledge and meta-data for case-based model selection. In *PKDD '01: Proceedings of the 5th European Conference on Principles of Data Mining and Knowledge Discovery*, pages 180–191, London, UK, 2001. Springer-Verlag.
- [HMMS60] C. C. Holt, F. Modigliani, J. F. Muth, and H. A. Simon. *Planning Production Inventories and Work Force*. Prentice-Hall, Englewood Cliffs, NJ, 1960.
- [HTFF05] Trevor Hastie, Robert Tibshirani, Jerome Friedman, and James Franklin. The elements of statistical learning: data mining, inference and prediction. *The Mathematical*, 27(2):83–85, 2005.
- [KH03] Alexandros Kalousis and Melanie Hilario. Representational issues in meta-learning. *Machine Learning*, 2003.
- [KT99] Alexandros Kalousis and Theoharis Theoharis. Noemon: Design, implementation and performance results of an intelligent assistant for classifier selection. *Intell. Data Anal.*, 3(5):319–337, 1999.
- [KTK00] Christian Köpf, Charles C. Taylor, and Jörg Keller. Meta-analysis: From data characterisation for meta-learning to meta-regression. *Mining, Decision Support, Meta-Learning and*, pages 209–216, 2000.
- [LB04] Rui Leite and Pavel Brazdil. Improving progressive sampling via meta-learning on learning curves. In *ECML*, pages 250–261, 2004.
- [LB05] Rui Leite and Pavel Brazdil. Predicting relative performance of classifiers from samples, 2005.
- [LB08] R. Leite and P. Brazdil. Selecting classifiers using meta-learning with sampling landmarks and data characterization. In *Second Planning to Learn Workshop (PLANLEARN) at the ICML/COLT/UAI 2008*, page 35. Cite-seer, 2008.
- [LG10] Christiane Lemke and Bogdan Gabrys. Meta-learning for time series forecasting and forecast combination. *Neurocomputing*, 73(10-12):2006 – 2016, 2010. Subspace Learning / Selected papers from the European Symposium on Time Series Prediction.
- [Llo99] J. W. Lloyd. Programming in an integrated functional and logic language 1 introduction. *Lloydia (Cincinnati)*, pages 1–51, 1999.

- [LS99] Guido Lindner and R. Studer. Ast: Support for algorithm selection with a cbr approach. *Principles of Data Mining and Knowledge Discovery*, pages 418–423, 1999.
- [MAC⁺82] S. Makridakis, A. Andersen, R. Carbone, R. Fildes, M. Hibon, R. Lewandowski, J. Newton, E. Parzen, and R. Winkler. The accuracy of extrapolation (time series) methods: Results of a forecasting competition. *Journal of Forecasting*, 1(2):111–153, 1982.
- [Mas95] Timothy Masters. *Neural, Novel and Hybrid Algorithms for Time Series Prediction*. John Wiley & Sons, Inc., New York, NY, USA, 1995.
- [MCH⁺93] Spyros Makridakis, Chris Chatfield, Michèle Hibon, Michael Lawrence, Terence Mills, Keith Ord, and LeRoy F. Simmons. The m2-competition: A real-time judgmentally based forecasting study. *International Journal of Forecasting*, 9(1):5 – 22, 1993.
- [Mer96] C. J. Merz. Dynamical selection of learning algorithms. *Lecture Notes in Statistics - New York Spring Verlag*, pages 281–290, 1996.
- [MH00] Spyros Makridakis and Michèle Hibon. The m3-competition: results, conclusions and implications. *International Journal of Forecasting*, 16(4):451 – 476, 2000.
- [Mit97] Tom Mitchell. *Machine Learning*. McGraw-Hill Education (ISE Editions), October 1997.
- [MS94] Donald Michie and David J. Spiegelhalter. *Machine learning, neural and statistical classification*, 1994.
- [NS97] Gholamreza Nakhaeizadeh and Alexander Schnabl. Development of multi-criteria metrics for evaluation of data mining algorithms. *Third International Conference on Knowledge Discovery and Data Mining*, 1997.
- [PFBS02] Yonghong Peng, Peter A. Flach, Pavel Brazdil, and Carlos Soares. Decision tree-based data characterization for meta-learning. In *ECML/PKDD'02 Workshop on Integration and Collaboration Aspects of the Data Mining, Decision Support and Meta-Learning*, pages 111–122, 2002.
- [PGc00] Bernhard Pfahringer and Christophe Giraud-carrier. Meta-learning by landmarking various learning algorithms. In *Proceedings of the Seventeenth International Conference on Machine Learning*, 2000.

- [PL04] Ricardo Bastos Cavalcante Prudêncio and Teresa Bernarda Ludermir. Meta-learning approaches to selecting time series models. *Neurocomputing*, 61:121–137, 2004.
- [PL07] Ricardo B. C. Prudêncio and Teresa Bernarda Ludermir. Aprendizagem ativa para seleção de exemplos em meta-aprendizado. In *6th Brazilian Meeting on Artificial Intelligence (Encontro Nacional de Inteligencia Artificial-ENIA)*, pages 3–6. Citeseer, 2007.
- [Pru04] Ricardo B. C. Prudêncio. *Meta-aprendizado para seleção e combinação de séries temporais*. PhD thesis, Universidade Federal de Pernambuco, 2004.
- [Pru10] Ricardo B. C. Prudêncio. Meta-learning research. www.cin.ufpe.br/rbcp/, May 2010. Accessed in 05/08/2010.
- [Rod09] Joseane Pereira Rodrigues. *Sistemas Inteligentes Híbridos para classificação de Texto*. PhD thesis, University Federal of Pernambuco, 2009.
- [RP04] I. Rojas and H. Palomares. Soft-computing techniques for time series forecasting. In *Proc. of the European Symposium on Artificial Neural Networks*, pages 93–102. Citeseer, April 2004.
- [Sha97] Chandra Shah. Model selection in univariate time series forecasting using discriminant analysis. *International Journal of Forecasting*, 13(4):489 – 500, 1997.
- [Ska97] D. B. Skalak. *Prototype selection for composite nearest neighbor classifiers*. PhD thesis, University of Massachusetts Amherst, 1997.
- [SM08] Kate A. Smith-Miles. Cross-disciplinary perspectives on meta-learning for algorithm selection. *ACM Computing Surveys*, 41(1):1–25, dezembro 2008.
- [SNY99] Akihiro Suyama, Naoya Negish, and Takahira Yamaguchi. Design and evaluation of an environment to automate the construction of inductive applications. In *Discovery Science*, page 75. Springer Berlin / Heidelberg, 1999.
- [Soh99] S. Y. Sohn. Meta analysis of classification algorithms for pattern recognition, 1999.
- [SPL04] Patrícia Maforte dos Santos, Ricardo B. C. Prudêncio, and Teresa Bernarda Ludermir. Seleção de modelos de previsão de séries temporais baseada em informações de performance. *Machine Learning*, 2004.

- [Tas00] L. Tashman. Out-of-sample tests of forecasting accuracy: an analysis and review. *International Journal of Forecasting*, 16(4):437–450, Dezembro 2000.
- [TD03] L. Todorovski and S. Dzeroski. Combining classifiers with meta decision trees. *Machine Learning*, pages 223–249, 2003.
- [TF92] Efraim Turban and Louis E. Frenzel. *Expert Systems and Applied Artificial Intelligence*. Prentice Hall Professional Technical Reference, 1992.
- [TW97] Kai Ming Ting and Ian H. Witten. Stacked generalization: when does it work? In *in Procs. International Joint Conference on Artificial Intelligence*, pages 866–871. Morgan Kaufmann, 1997.
- [VD02] Ricardo Vilalta and Youssef Drissi. A perspective view and survey of meta-learning. *Artificial Intelligence Review*, 18(2):77–95, dec 2002.
- [VFP96] Robert J. Vokurka, Benito E. Flores, and Stephen L. Pearce. Automatic feature identification and graphical support in rule-based forecasting: a comparison. *International Journal of Forecasting*, 12(4):495 – 512, 1996.
- [VGCB68] Ricardo Vilalta, Christophe Giraud-Carrier, and Pavel B. Brazdil. Meta-learning: Concepts and techniques. *Review of Educational Research*, 38(3):264–276, junho 1968.
- [VGcBS04] Ricardo Vilalta, Christophe Giraud-carrier, Pavel Brazdil, and Carlos Soares. Using meta-learning to support data mining. *Knowledge Acquisition*, 1(MI), 2004.
- [VS99] A. R. Venkatachalam and Jeffrey E. Sohl. An intelligent model selection and forecasting system. *Journal of Forecasting*, 18(3):167–180, May 1999.
- [WAN05] X. WANG. *Characteristic-Based forecasting for time series data*. PhD thesis, Monash University Australia, 2005.
- [Wol92] David H. Wolpert. Stacked generalization. *Neural Networks*, 5(2):241–259, 1992.
- [WSH06] Xiaozhe Wang, Kate Smith, and Rob Hyndman. Characteristic-based clustering for time series data. *Data Min. Knowl. Discov.*, 13(3):335–364, 2006.
- [WSMH09] Xiaozhe Wang, Kate Smith-Miles, and Rob Hyndman. Rule induction for forecasting method selection: Meta-learning the characteristics of univariate

time series. *Neurocomputing*, 72(10-12):2581 – 2594, 2009. Lattice Computing and Natural Computing (JCIS 2007) / Neural Networks in Intelligent Systems Design (ISDA 2007).