Elisângela Silva da Cunha Rodrigues

Teoria da Informação e Adaptatividade na Modelagem de Distribuição de Espécies

Tese apresentada à Escola Politécnica da Universidade de São Paulo para obtenção do Título de Doutor em Ciências.

São Paulo
2012

Elisângela Silva da Cunha Rodrigues

Teoria da Informação e Adaptatividade na Modelagem de Distribuição de Espécies

Tese apresentada à Escola Politécnica da Universidade de São Paulo para obtenção do Título de Doutor em Ciências.

Área de concentração:
Sistemas Digitais

Orientador:
Prof. Dr. Ricardo Luis de Azevedo da Rocha

São Paulo
2012

Aos meus pais, Eni e Erlei, e ao meu marido, Fabrício.

Agradecimentos

Agradeço ao meu marido, Fabrício, pelo companheirismo, incentivo e paciência. A experiência da convivência profissional, concomitantemente com a matrimonial, é algo indescritível, é a prova do amor pleno.

Agradeço aos meus pais, Eni e Erlei, pelo amor que sempre me dedicaram, por me ensinarem a ser perseverante e por me proporcionarem o bem mais valioso que qualquer ser humano pode ter, a educação.

Agradeço aos meus irmãos, Irlene e Guinter, pelo amor incondicional, pelo carinho e atenção que sempre me dedicaram.

Agradeço aos meus sobrinhos e afilhados, Arthur e Hiram, pelos raros momentos inocentes que me fazem esquecer as preocupações da vida adulta.

Agradeço ao meu orientador, professor Dr. Ricardo Luis de Azevedo da Rocha, pela dedicação e presteza na condução do trabalho.

Agradeço ao professor Dr. Pedro Luiz Pizzigatti Corrêa, pelo contato inicial que me proporcionou a oportunidade de participar do projeto *openModeller*.

Aos professores Dr. Antonio Mauro Saraiva e João José Neto, os meus agradecimentos pelas valiosas contribuições apresentadas durante o exame de qualificação.

Aos colegas do projeto *openModeller*, do Centro de Referência em Informação Ambiental (CRIA), do Instituto Nacional de Pesquisas Espaciais (INPE) e da Escola Politécnica da Universidade de São Paulo, os meus agradecimentos pelas discussões que proporcionaram o aprimoramento do trabalho. Em especial, agradeço à Dra. Marinez F. de Siqueira, pelos esclarecimentos de dúvidas, ao colega Renato De Giovanni, pela colaboração na implementação e à colega Tereza Cristina Giannini, pelo fornecimento de dados.

Agradeço ao professor Dr. André Riyuiti Hirakawa, pela orientação inicial do trabalho.

Agradeço aos professores das disciplinas cursadas, Dra. Liria M. Sato, Dr. Ricardo L. de A. da Rocha, Dr. Edson T. Midorikawa e Dr. Edson S. Gomi, pelos valiosos ensinamentos.

Agradeço aos colegas do LAA e do LTA, que de alguma forma contribuíram para o desenvolvimento deste trabalho.

Aos amigos Maria José, Rodiney, Débora e Gustavo, os meus agradecimentos pelo apoio fundamental na chegada a São Paulo.

Agradeço à Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), pelo apoio financeiro concedido para o desenvolvimento deste trabalho.

Agradeço aos funcionários das secretarias do PCS e da Pós-graduação (setorial e central), pelo apoio e presteza no atendimento.

Agradeço ao pesquisador Ph.D. Petri Kontkanen, da Universidade de Helsinki – Finlândia, pela concessão do código utilizado em parte dos experimentos deste trabalho.

Aos professores participantes da banca examinadora, Dra. Vera Lucia Imperatriz Fonseca, Dra. Maria Cristina Arias, Dr. Carlos Eduardo Cugnasca e Dr. Edson Satoshi Gomi, os meus agradecimentos pelas sugestões e críticas que contribuíram para o aprimoramento do trabalho.

Sobretudo, agradeço a Deus pela oportunidade da vida e pela fé.

Resumo

A modelagem de distribuição de espécies é uma técnica cuja finalidade é estimar modelos baseados em nichos ecológicos. Esses modelos podem auxiliar nos processos de tomadas de decisões, no planejamento e na realização de ações que visem a conservação e a preservação ambiental. Existem diversas ferramentas projetadas para modelagem de distribuição de espécies, dentre elas o *framework openModeller*, na qual este trabalho está inserido. Várias técnicas de Inteligência Artificial já foram utilizadas para desenvolver algoritmos de modelagem de distribuição de espécies, como Entropia Máxima. No entanto, as ferramentas estatísticas tradicionais não disponibilizam pacotes com o algoritmo de Entropia Máxima, o que é comum para outras técnicas. Além disso, apesar de existir um software gratuito específico para modelagem de distribuição de espécies com o algoritmo de Entropia Máxima, esse software não possui código aberto. Assim, a base deste trabalho é a investigação acerca da modelagem de distribuição de espécies utilizando Entropia Máxima. Desta forma, o objetivo principal é definir diferentes estratégias para o algoritmo de Entropia Máxima no contexto da modelagem de distribuição de espécies. Para atingir esse objetivo, foram estabelecidos um conjunto de alternativas possíveis a serem exploradas e um conjunto de métricas de avaliação e comparação das diferentes estratégias. Os resultados mais importantes desta pesquisa foram: um algoritmo adaptativo de Entropia Máxima, um algoritmo paralelo de Entropia Máxima, uma análise do parâmetro de regularização e um método de seleção de variáveis baseado no princípio da Descrição com Comprimento Mínimo (MDL – *Minimum Description Length*), que utiliza aprendizagem por compressão de dados.

Palavras-chave: Entropia Máxima. Modelagem de Distribuição de Espécies. Tecnologia Adaptativa. Descrição com Comprimento Mínimo.

Abstract

Species distribution modeling is a technique the purpose of which is to estimate models based on ecological niche. These models can assist decision making processes, planning and carrying out actions aiming at environmental conservation and preservation. There are several tools designed for species distribution modeling, such as the openModeller framework, in which this work is inserted. Several Artificial Intelligence techniques have been used to develop algorithms for species distribution modeling, such as Maximum Entropy. However, traditional statistical tools do not offer packages with the Maximum Entropy algorithm, which is common to other techniques. Furthermore, although there is specific free software for species distribution modeling with the Maximum Entropy algorithm, this software is not open source. The basis of this work is the investigation of the species distribution modeling using Maximum Entropy. Thus, its aim is to define different strategies for the Maximum Entropy algorithm in the context of the species distribution modeling. For this, a set of possible alternatives to be explored and a set of metrics for evaluation and comparison of the different strategies were established. The most important results were: an adaptive Maximum Entropy algorithm, a parallel Maximum Entropy algorithm, an analysis of the regularization parameter and a variable selection method based on the Minimum Description Length principle, which uses learning by data compression.

Keywords: Maximum Entropy. Species Distribution Modeling. Adaptive Technology. Minimum Description Length.

Sumário

Lista de Figuras

Lista de Tabelas

1 Introdução

A conservação e a preservação da biodiversidade e de recursos naturais, são temas que têm sido amplamente discutidos nas últimas décadas em todo o mundo. Antes de expor algumas razões que expressam a importância desses temas, é interessante comentar que o termo "preservação" refere-se ao ato de proteger a natureza integralmente, sem levar em consideração os interesses econômicos ou quaisquer utilidades. Todavia, o termo "conservação" refere-se ao ato de proteger a natureza visando algum objetivo, ou seja, permite o uso sustentável dos recursos naturais (ADAMS et al., 2004). Além disso, o termo "biodiversidade" não se refere apenas à diversidade do mundo biológico, como a palavra sugere, mas também se refere às interações complexas entre as diferentes formas de vida e as funções ecológicas desempenhadas por elas.

Existem várias razões para definir estratégias que ajudem na preservação e na conservação da biodiversidade e de recursos naturais. O valor econômico é a razão mais facilmente identificável para demonstrar a importância da conservação da biodiversidade. Economicamente, a biodiversidade fornece recursos diretos, tais como alimentos, remédios, energia etc, e, colabora indiretamente em diversos setores, como na indução da polinização de flores de frutas para melhorar a produção. No entanto, pensando eticamente, todas as espécies são inerentemente importantes. Além disso, cada espécie tem um papel diferente no ecossistema e todos eles são essenciais na natureza. Portanto, a gestão adequada de recursos para assegurar um desenvolvimento sustentável é muito útil.

A Organização das Nações Unidas estabeleceu oito metas para o desenvolvimento do milênio (ONU, 2011). Uma dessas metas é a sustentabilidade ambiental. Para alcançar essa meta foram definidos quatro objetivos, sendo um deles a redução da perda de biodiversidade. Os impactos causados pela perda de biodiversidade afetam diretamente os meios de subsistência do ser humano. Um exemplo disso é a redução de água potável. Desta forma, esse tema têm sido o foco de pesquisadores de diversas áreas da ciência.

Os efeitos da perda de biodiversidade são irreversíveis, pois muitas vezes causam

a extinção de determinadas espécies. No entanto, segundo Myers et al. (2000), é muito difícil a preservação de todas as espécies ameaçadas de extinção, pois o número de espécies a serem protegidas ultrapassa os recursos disponíveis para este fim. Assim, uma das maneiras de proteger a maior quantidade de espécies com o menor custo possível é a identificação de *hotspots*, ou seja, regiões cuja biodiversidade esteja significativamente ameaçada pela destruição e com alta concentração de espécies endêmicas. Por isso, a conservação e a identificação de áreas que podem ser preservadas representam grandes desafios atuais para a humanidade. Além disso, o incentivo e o apoio a pesquisas voltadas ao gerenciamento de recursos naturais e desenvolvimento sustentável são muito importantes para o planejamento e o auxílio em tomadas de decisões de políticas públicas.

Existem várias técnicas de Inteligência Artificial (IA) que estão sendo usadas como auxiliares na solução de problemas relacionados à conservação da biodiversidade e ao gerenciamento de recursos naturais (CHEN; JAKEMAN; NORTON, 2008). A Tabela 1.1 apresenta alguns exemplos de técnicas de IA e suas aplicações na área de conservação ambiental. As técnicas de IA têm grande potencial para resolver problemas complexos de inferência (RUSSELL; NORVIG, 2004).

Tabela 1.1: Exemplos de técnicas de IA e suas aplicações na área de conservação ambiental.

Técnica	Aplicações na área de conservação ambiental
Raciocínio Baseado em Casos	Gerenciamento de incêndios florestais (AVESANI; PERINI; RICCI, 2000) Predição da qualidade do ar em zonas urbanas (KALAPANIDAS; AVOURIS, 2001) Redução de impactos ambientais causados por processos químicos (KING; ALCÁNTARA; MANAN, 1999) Previsão de ciclones (PEDRO; BURSTEIN; SHARP, 2005)
Algoritmos Genéticos	Predições da qualidade do ar (KALAPANIDAS; AVOURIS, 2003) Calibração de modelos de qualidade da água (PELLETIER; CHAPRA; TAO, 2006) Gerenciamento da água na agricultura irrigada (INES et al., 2006)

continua

conclusão

Técnica	Aplicações na área de conservação ambiental
Redes Neurais Artificiais	Gerenciamento de recursos hídricos (ILIADIS; MARIS, 2007) Predição da qualidade da água (QING; STANLEY, 1997; POZDNYAKOV et al., 2005) Predição de concentrações de ozônio (SOUSA et al., 2007)
Sistemas Baseados em Regras	Diagnóstico de pragas (EL-AZHARY; HASSAN; RAFEA, 2000; MAHAMAN et al., 2002; MAHAMAN et al., 2003) Diagnóstico de doenças (ZETIAN et al., 2005)
Sistemas Multiagentes	Gerenciamento de recursos naturais (BOUSQUET; PAGE, 2004; BORRI; CAMARDA; LIDDO, 2005) Manejo florestal (PURNOMO et al., 2005)
Autômatos Celulares	Modelagem de migração de espécies (SCHÖNFISCH; KINDER, 2002) Modelagem de paisagem (EL-YACOUBI et al., 2003) Modelagem de atividades sísmicas (GEORGOUDAS et al., 2007)
Sistemas Fuzzy	Gestão da água (SCHLUTER; RUGER, 2007) Predição de erosão do solo (METTERNICHT; GONZALEZ, 2005) Estimativa de risco de incêndio florestal (ILIADIS, 2005)

A modelagem de distribuição de espécies, que será abordada em detalhes no Capítulo 2, é uma técnica que tem sido cuidadosamente estudada, cuja finalidade é estimar modelos baseados em nichos ecológicos. Esses modelos podem auxiliar nos processos de tomadas de decisões, no planejamento e na realização de ações que visem a conservação e a preservação ambiental. Alguns exemplos de utilização da modelagem de distribuição de espécies em problemas relacionados à conservação ambiental são: avaliação dos impactos de alterações climáticas (SIQUEIRA; PETERSON, 2003; THOMAS et al., 2004; FITZPATRICK et al., 2008), monitoramento da variação espaço-temporal na adequação de espécies ao habitat (BARTEL; SEXTON, 2009), estudo de delimitação de espécies (RAXWORTHY et al., 2007) e gerenciamento da propagação de espécies invasoras (PETERSON; VIEGLAIS, 2001; PETERSON; PAPES; KLUZA, 2003).

Várias técnicas de IA já foram utilizadas para desenvolver algoritmos de modelagem de distribuição de espécies, tais como Redes Neurais Artificiais (MAIER; DANDY; BURCH, 1998; RODRIGUES et al., 2009), *Support Vector Machines* (LORENA et al., 2008), Algoritmos Genéticos (PERSONA; CORRÊA; SARAIVA, 2003) e Entropia Máxima (PHILLIPS; ANDERSON; SCHAPIRE, 2006b). No entanto, algumas questões relativas à modelagem de distribuição de espécies, apresentadas na Seção 1.2, justificam o estudo de novas abordagens para esta tarefa ou de extensões das existentes.

Este trabalho está inserido no escopo do projeto *openModeller* (MUÑOZ et al., 2011), um *framework* para modelagem de distribuição de espécies com vários recursos disponíveis. O *openModeller* tem como principal objetivo oferecer uma ferramenta gratuita, de código aberto e portável a pessoas interessadas em modelagem de distribuição de espécies (SUTTON; GIOVANNI; SIQUEIRA, 2007).

O projeto *openModeller* foi financiado pela Fundação de Amparo à Pesquisa do Estado de São Paulo (Fapesp) e desenvolvido por três instituições de pesquisa: Centro de Referência em Informação Ambiental (CRIA), Escola Politécnica da Universidade de São Paulo (EPUSP) e Instituto Nacional de Pesquisas Espaciais (INPE).

1.1 Objetivos

O principal objetivo deste trabalho é definir diferentes estratégias para o algoritmo de Entropia Máxima no contexto da modelagem de distribuição de espécies. Desta forma, procura-se responder à seguinte pergunta: é possível inferir modelos de distribuição de espécies confiáveis através de extensões do algoritmo de Entropia Máxima ou de alguma outra técnica fundamentada em teoria da informação?

Para atingir esse objetivo, foram estabelecidos um conjunto de alternativas possíveis a serem exploradas e um conjunto de métricas de avaliação e comparação das diferentes estratégias. As alternativas determinadas foram: programação paralela e distribuída, tecnologia adaptativa e inferência indutiva por compressão de dados. As métricas de avaliação e comparação variaram de acordo com a técnica analisada e serão apresentadas no Capítulo 5.

Dentro de cada uma das alternativas estabelecidas, algumas metas foram traçadas. Na programação paralela e distribuída o objetivo foi desenvolver um algoritmo paralelo de Entropia Máxima com a finalidade de melhorar o tempo de execução da modelagem de distribuição de espécies. Na tecnologia adaptativa o alvo foi a formalização e desenvolvimento de um algoritmo adaptativo de Entropia Máxima visando atingir a taxa de convergência mais rapidamente. Outro objetivo foi analisar o ajuste de um

parâmetro do algoritmo de Entropia Máxima (parâmetro de regularização) e verificar a possibilidade de substituí-lo por uma estratégia adaptativa. Na inferência indutiva por compressão de dados o objetivo foi reduzir a quantidade de variáveis de entrada (etapa de pré-análise).

Tendo em vista os objetivos apresentados acima, os resultados teóricos e práticos esperados após a conclusão do trabalho foram:

- Integração de um algoritmo de Entropia Máxima à ferramenta *openModeller*;
- Formalização de uma estratégia adaptativa para o algoritmo de Entropia Máxima;
- Desenvolvimento de um algoritmo adaptativo de Entropia Máxima;
- Definição de uma estratégia de paralelização do algoritmo de Entropia Máxima;
- Desenvolvimento de um algoritmo paralelo de Entropia Máxima;
- Descrição de uma análise do ajuste do parâmetro de regularização do algoritmo de Entropia Máxima;
- Definição e análise de uma estratégia adaptativa para substituir o parâmetro de regularização do algoritmo de Entropia Máxima;
- Descrição de um método de seleção de variáveis utilizando o princípio da Descrição com Comprimento Mínimo.

1.2 Justificativa

A modelagem de distribuição de espécies é uma técnica que está sendo usada amplamente para auxiliar tomadas de decisões relacionadas à conservação e preservação ambiental. A finalidade da utilização da modelagem de distribuição de espécies torna fundamental as pesquisas científicas nesta área. Aliado a isso, o algoritmo de Entropia Máxima é um dos mais utilizados pela comunidade científica na tarefa de modelagem de distribuição de espécies. Mas por que investigar alternativas para esse algoritmo? Existem várias respostas para esta pergunta e algumas serão apresentadas a seguir.

Primeiramente, as ferramentas estatísticas tradicionais não disponibilizam pacotes com o algoritmo de Entropia Máxima, o que é comum para outras técnicas, como Redes Neurais Artificiais e Algoritmos Genéticos. Phillips, Anderson e Schapire (2006b) desenvolveram um software específico para modelagem de distribuição de espécies

com o algoritmo de Entropia Máxima. No entanto, apesar de ser gratuito, esse software não possui código aberto. Desta forma, a integração de um algoritmo de Entropia Máxima a uma ferramenta de modelagem de distribuição de espécies livre, de código aberto e que possui várias funcionalidades é um aspecto inovador.

A modelagem de sistemas ambientais é uma área de estudo que tem chamado a atenção nos últimos anos devido à sua contribuição na conservação da biodiversidade. No entanto, a maioria dos algoritmos desenvolvidos para essa tarefa requer um tempo excessivo de execução para a produção de resultados aceitáveis. Portanto, é de fundamental importância a pesquisa da aplicação de técnicas, conhecidamente bem sucedidas em outras áreas, na modelagem de distribuição de espécies. Desta forma, a utilização de técnicas de alto desempenho na construção de algoritmos de modelagem pode favorecer o tempo de resposta dos mesmos.

Os métodos baseados em Entropia Máxima já são aplicados há bastante tempo em outras áreas (JAYNES, 1982), mas só recentemente foram utilizados para modelagem de distribuição de espécies (PHILLIPS; ANDERSON; SCHAPIRE, 2006b). De acordo com uma análise comparativa realizada por Elith et al. (2006), os métodos baseados em Entropia Máxima apresentaram resultados promissores. Assim, as pesquisas acerca desse método podem produzir valiosos avanços na área de modelagem de distribuição de espécies.

Os sistemas adaptativos, cuja principal característica é a capacidade de modificarem sua própria estrutura sem interferências externas, compõem outro campo de pesquisa em expansão. A capacidade de automodificação favorece a aprendizagem dos sistemas adaptativos, propriedade fundamental dos sistemas considerados inteligentes. Os algoritmos de modelagem de distribuição de espécies podem se beneficiar dessa propriedade adaptativa para tornarem-se dinâmicos, adaptando-se ao tipo de dado disponível. Assim, o desenvolvimento de um algoritmo de Entropia Máxima adaptativo é uma importante inovação ao aplicar tecnologia adaptativa na seleção de distribuições de probabilidade.

Frequentemente, as bases de dados têm registros abundantes de presença de uma determinada espécie em uma região de estudo e raramente registros de ausência da espécie na mesma região estão disponíveis (ELITH et al., 2011). Isso acontece porque é muito difícil determinar a não existência de uma espécie, uma vez que alguns fatores tais como intervenção sazonal podem influenciar na observação da espécie. Do ponto de vista de aprendizagem de máquina, isso pode ser visto como a falta de exemplos negativos para treinar os algoritmos, dificultando o processo de aprendizagem. Ao mesmo tempo, a disponibilidade de dados de ocorrência de uma grande variedade de

espécies tem aumentado constantemente (SKARPAAS; STABBETORP, 2011). Essa discrepância de registros de presença e ausência de espécies estimula os estudos de métodos para tratar uma grande quantidade de dados contendo apenas exemplos positivos, como é o caso da Entropia Máxima.

Outro desafio na área de aprendizagem de máquina é o desenvolvimento de métodos que evitem o superajustamento – *overfitting*. O superajustamento é a memorização dos dados de treinamento, ou seja, o modelo descreve com precisão os exemplos usados para construí-lo, mas falha em aprender a generalizá-los. Geralmente, um modelo superajustado tem desempenho ruim com dados desconhecidos, por isso não pode ser usado para tarefas preditivas.

O princípio da Descrição com Comprimento Mínimo (*Minimum Description Length* – MDL) (GRÜNWALD, 2005) tem a propriedade de evitar automaticamente o superajustamento durante a aprendizagem dos parâmetros de um modelo. Outra propriedade interessante do princípio MDL é que não são necessários exemplos negativos na seleção do modelo. Assim, a utilização do princípio MDL pode contribuir com a modelagem de distribuição de espécies, evitando o superajuste de modelos e ampliando a oferta de algoritmos que só utilizam registros de presença de espécies.

A ideia principal do princípio MDL é a busca de um modelo com a menor descrição dos dados observados. Isso é feito encontrando regularidades nos dados que são usadas para compactá-los. O princípio MDL já foi aplicado com sucesso na estimativa de densidade de probabilidade (KONTKANEN; MYLLYMÄKI, 2007; CHAPEAU-BLONDEAU; ROUSSEAU, 2009), porém ainda não foi utilizado na área de modelagem de distribuição de espécies. Assim, o estudo de como o princípio MDL pode beneficiar essa área é um aspecto inovador. Além disso, não se tem conhecimento de outro grupo de pesquisa que atue na investigação do princípio MDL associado à modelagem de distribuição de espécies.

Este trabalho tem característica fundamentalmente interdisciplinar. Por isso, espera-se que as contribuições sejam relevantes principalmente para as seguintes áreas do conhecimento: teoria da informação, aprendizagem de máquina, modelagem de distribuição de espécies e tecnologia adaptativa.

1.3 Método do Trabalho

A primeira fase do trabalho foi a realização de uma pesquisa bibliográfica sobre o princípio de Entropia Máxima. Essa fase foi fundamental para a compreensão dos conceitos básicos. A consolidação desses conceitos ocorreu através de um estudo de

caso, o algoritmo de classificação ID3. O ID3 é um algoritmo de árvore de decisão que utiliza Entropia Máxima na escolha de seus nós.

Simultaneamente, foi realizada uma pesquisa bibliográfica sobre modelagem de distribuição de espécies. Como um dos objetivos era disponibilizar na ferramenta *openModeller* um algoritmo de modelagem de distribuição de espécies baseado em Entropia Máxima, a utilização de uma biblioteca foi a solução prática e rápida encontrada inicialmente. No entanto, por terem sido desenvolvidas para outras finalidades, as bibliotecas gratuitas compatíveis com o *openModeller* apresentavam algumas limitações. Por isso, a implementação de um algoritmo baseado em Entropia Máxima foi o método adotado posteriormente.

A tecnologia adaptativa foi estudada com detalhes para possibilitar a formalização de uma estratégia para o algoritmo de Entropia Máxima. Nesta fase do trabalho, foram realizadas pesquisas bibliográficas e um estudo de caso, o emparelhamento de cadeias. A definição de um algoritmo adaptativo de emparelhamento de cadeias foi essencial para o entendimento das características de um sistema automodificável.

Após o estudo de caso, a formalização e a implementação de um algoritmo adaptativo baseado em Entropia Máxima foram realizadas. Para validar a estratégia adaptativa proposta, alguns experimentos foram realizados com dados previamente selecionados. Os experimentos foram realizados com a versão tradicional e com a versão adaptativa do algoritmo com o objetivo de compará-las. As métricas de comparação foram tempo de execução de cada versão e acuidade dos modelos gerados.

Para definir a melhor estratégia de paralelização do algoritmo de Entropia Máxima foi realizado um estudo sobre programação paralela e distribuída. Nesta fase, além da pesquisa bibliográfica, a colaboração em um estudo de caso foi crucial para a definição da abordagem que seria adotada na paralelização do algoritmo de modelagem baseado em Entropia Máxima. Com a finalidade de verificar se houve aumento de desempenho do algoritmo, alguns experimentos foram realizados para comparar as versões sequencial e paralela. Os dados das espécies utilizados nesta fase, foram os mesmos dos experimentos realizados para testar a versão adaptativa do algoritmo de Entropia Máxima. Os critérios de comparação foram o tempo total de execução da aplicação e o tempo de execução da parte do código que estima os parâmetros do modelo com Entropia Máxima.

O algoritmo de Entropia Máxima possui um parâmetro especial para evitar o superajustamento dos modelos, o parâmetro de regularização. A definição de valores padrão para os parâmetros de um algoritmo, geralmente, é realizada através de uma quantidade exaustiva de experimentos, o que pode levar um longo tempo. Como não

foi encontrado na literatura um consenso sobre o valor padrão para este parâmetro, uma análise dos seus possíveis valores foi realizada. Esta análise levou em consideração uma série de experimentos com diferentes tamanhos de conjuntos de dados de várias espécies. Além disso, uma estratégia alternativa para substituir este parâmetro utilizando tecnologia adaptativa foi definida e analisada. Para as análises acerca do parâmetro de regularização foram considerados: o tamanho do conjunto de dados, a acuidade dos modelos, a AUC (área sob a curva ROC – *Receiver Operating Characteristic*) e o número de iterações utilizadas pelo algoritmo.

Considerando a possibilidade de que outras técnicas fundamentadas em teoria da informação pudessem contribuir com a evolução dos algoritmos de modelagem de distribuição de espécies, iniciou-se um estudo sobre o princípio MDL. Assim como nas outras áreas de conhecimento abordadas neste trabalho, após uma pesquisa bibliográfica, optou-se por um estudo de caso. No entanto, uma vez que a experiência com modelagem de distribuição de espécies já estava um pouco mais avançada, o estudo de caso foi feito em uma etapa anterior à construção dos modelos, ou seja, na seleção de variáveis.

A seleção de variáveis faz parte de uma etapa do processo de modelagem conhecida como pré-análise. Nesta etapa várias técnicas podem ser usadas para sugerir o melhor conjunto de variáveis a serem usadas para estimar o modelo. Neste trabalho, o princípio MDL foi usado para estimar a densidade das variáveis através de histogramas irregulares e agrupá-las de acordo com uma medida de complexidade.

1.4 Organização do Texto

Este trabalho está dividido em capítulos que procuram apresentar cada área do conhecimento abordada, bem como os resultados e contribuições alcançados. O Capítulo 2 apresenta os principais conceitos da modelagem de distribuição de espécies. O objetivo deste capítulo é descrever conceitos essenciais para a compreensão do trabalho. Neste capítulo, o conceito de nicho é discutido, os tipos de dados utilizados na modelagem são descritos, o processo completo da modelagem e a ferramenta *openModeller* são apresentados. Por fim, alguns algoritmos e aplicações da modelagem são indicados.

O Capítulo 3 apresenta dois princípios fundamentados em teoria da informação: o princípio da Entropia Máxima e o princípio da Descrição com Comprimento Mínimo. A Seção 3.1 mostra a teoria básica do princípio da Entropia Máxima, seu uso na modelagem de distribuição de espécies e algumas aplicações. A Seção 3.2 mostra os conceitos básicos do princípio da Descrição com Comprimento Mínimo e como ele

pode ser aplicado na prática. O propósito deste capítulo é elucidar os princípios de teoria da informação que foram abordados no decorrer do trabalho. Estes princípios também podem inspirar a aplicação de outras técnicas, com fundamentos semelhantes, na modelagem de distribuição de espécies.

No Capítulo 4 são apresentados os conceitos fundamentais da tecnologia adaptativa. As seções deste capítulo abrangem as características básicas dos dispositivos adaptativos, alguns exemplos de dispositivos adaptativos baseados em formalismos clássicos da computação, como autômatos adaptativos por exemplo, algumas aplicações da tecnologia adaptativa e um exemplo de aplicação da tecnologia adaptativa no emparelhamento de cadeias. A meta deste capítulo é explicar as propriedades básicas de um dispositivo adaptativo e exemplificar a aplicabilidade da tecnologia adaptativa.

O Capítulo 5 apresenta detalhadamente o desenvolvimento cronológico do trabalho. Em cada etapa, os passos intermediários são descritos, os experimentos realizados e os resultados obtidos são apresentados. A finalidade deste capítulo é mostrar a evolução do trabalho e a importância de cada etapa. No Capítulo 6, último capítulo do texto, são apresentadas as contribuições do trabalho, incluindo as produções bibliográficas. Ainda no mesmo capítulo, possíveis trabalhos futuros são esboçados e algumas conclusões são apresentadas.

Uma vez que este trabalho é interdisciplinar, foi criado um glossário com alguns termos comuns nas diferentes áreas do conhecimento abordadas. O objetivo é facilitar a leitura, evitando a explicação de termos técnicos ao longo do texto, e auxiliar a compreensão global do trabalho por pessoas de diversos campos de pesquisa.

2 Modelagem de Distribuição de Espécies

A modelagem consiste em construir uma representação de comportamento ou de características de um processo. A esta representação dá-se o nome de modelo (RUSSELL; NORVIG, 2004). Os modelos devem ser construídos a partir de uma amostra do processo, que representa um conhecimento incompleto sobre ele. O objetivo da modelagem é extrair a melhor explicação para um conjunto de dados e representá-la de forma precisa e compacta. Quando um modelo também consegue representar exemplos do processo que não foram usados na sua construção, o modelo tem alta capacidade de generalização. Nesse caso, os modelos podem ser usados posteriormente para fazer predições.

O processo de encontrar a melhor explicação para um conjunto de dados é conhecido como aprendizagem ou inferência indutiva (FORSYTH, 1988; CARBONELL, 1990; MITCHELL, 1997). A inferência pode ser vista como uma forma de selecionar um modelo que melhor represente a amostra do processo (DOWE, 2011). A aprendizagem por indução é um procedimento que pode ser caro computacionalmente (BERGER; PIETRA; PIETRA, 1996), por isso é recomendável fornecer um conjunto de heurísticas para facilitar a construção do modelo. As heurísticas representam a informação parcial sobre o processo.

Modelagem de distribuição de espécies é, portanto, a construção de um modelo que represente a distribuição geográfica de uma dada espécie. Distribuição geográfica é uma expressão utilizada para demarcar as regiões em que determinados eventos ocorrem, como tipos de vegetação, condições climáticas, ocorrências de populações, entre outros. Por simplicidade, neste trabalho foi adotada apenas a expressão “distribuição de espécies” ao invés de “distribuição geográfica de espécies”.

O objetivo deste capítulo é apresentar alguns conceitos relacionados à modelagem de distribuição de espécies. A Seção 2.1 contém uma discussão sobre nicho, um dos principais conceitos relacionados à modelagem de distribuição de espécies. A Seção 2.2 descreve os tipos de dados utilizados na modelagem. O processo de modelagem de distribuição de espécies é apresentado na Seção 2.3. Como este trabalho está no

contexto da ferramenta *openModeller*, a Seção 2.4 é usada para fornecer uma visão geral sobre ela. Em seguida, alguns algoritmos frequentemente utilizados na modelagem de distribuição de espécies são apresentados na Seção 2.5. Alguns exemplos de aplicações da modelagem de distribuição de espécies são mostrados na Seção 2.6. Por fim, a Seção 2.7 apresenta um resumo do capítulo.

2.1 O Conceito de Nicho

A distribuição de uma determinada espécie depende das condições ambientais da região. O nicho ecológico de uma espécie representa o conjunto de condições ambientais necessárias para uma espécie viver e se reproduzir. Assim, pode-se dizer que a distribuição de espécies está relacionada com o nicho ecológico. No entanto, o conjunto de condições ambientais necessárias para uma espécie viver e se reproduzir inclui tanto fatores bióticos, como predação e competição, quanto fatores abióticos, como temperatura e umidade.

Hutchinson (1957) formalizou o conceito de nicho ecológico através da definição de nicho fundamental. Ao definir nicho fundamental como um conjunto de estados do ambiente que permitiriam a existência indefinida de uma espécie, Hutchinson não fez distinção alguma entre condições biológicas e físicas. No entanto, ele ressalta a importância de considerar a competição entre as espécies e define como nicho realizado o subconjunto do nicho fundamental que exclui as situações de competição.

Segundo Soberón e Peterson (2005), além dos fatores bióticos e abióticos, a acessibilidade de uma região também é um fator importante a ser considerado na distribuição de espécies. Assim, eles elaboraram um diagrama para representar e discutir a interação destes três fatores, conforme ilustra a Figura 2.1, em que G representa o espaço geográfico de interesse, B representa a região em que os fatores bióticos são favoráveis para que uma espécie viva e se reproduza, A representa a região em que os fatores abióticos são favoráveis para a espécie e M representa a região acessível para a espécie. Nesta representação, a distribuição real da espécie encontra-se na região em que os três fatores são favoráveis à espécie, ou seja, $B \cap A \cap M$.

Soberón e Peterson (2005) consideram como nicho fundamental apenas o conjunto de fatores abióticos, ou seja, a região A é considerada a expressão geográfica que representa o nicho fundamental. Além disso, eles incluem outras interações entre as espécies no conceito de nicho realizado, como predação. Desta forma, o conceito de nicho realizado é representado pela expressão geográfica $A \cap B$. O nicho realizado pode ser visto como a união da área ocupada pela espécie e da área potencial (SOBE-

RÓN, 2007), ou seja, o nicho realizado é a região em G que possui tanto condições bióticas quanto abióticas adequadas para a espécie, independente da acessibilidade.

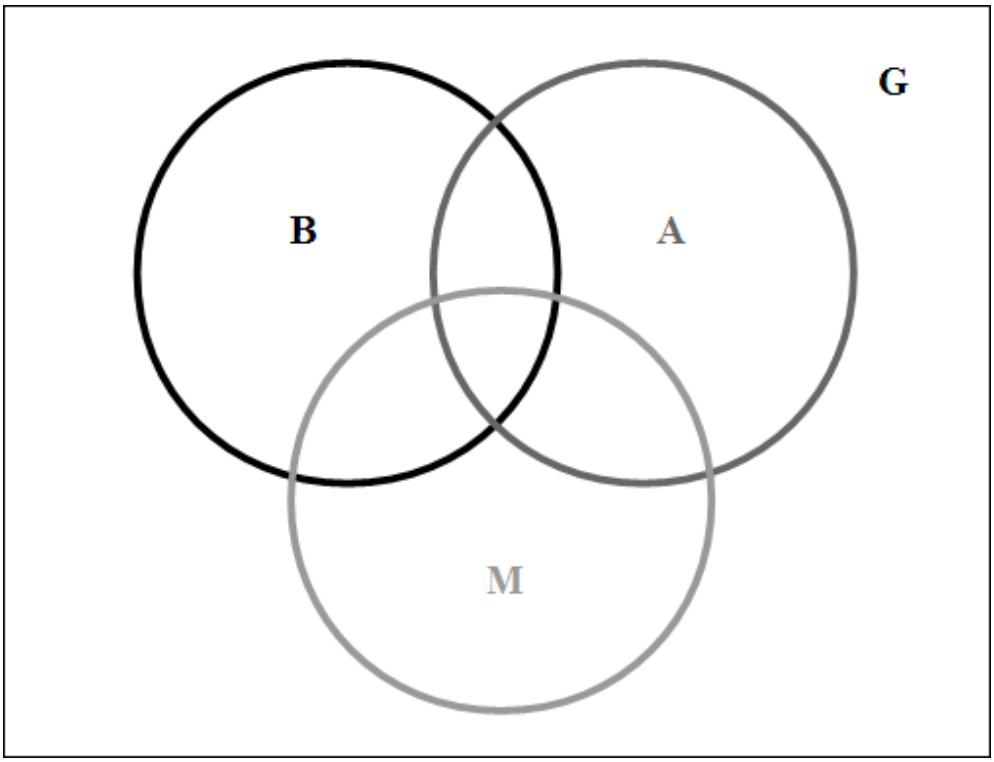


Figura 2.1: Diagrama de fatores que determinam a distribuição de espécies no espaço geográfico de interesse, G. B – fatores bióticos, A – fatores abióticos e M – região acessível. (Fonte: Soberón e Peterson (2005) e Soberón (2007)).

A análise da interação entre estes três fatores é essencial para a interpretação dos modelos estimados. No entanto, esta análise está além do escopo deste trabalho. Soberón e Peterson (2005) fazem uma discussão detalhada sobre a interpretabilidade dos modelos e sugerem alguns pontos cruciais para o desenvolvimento de modelos preditivos baseados em nicho ecológico.

Soberón (2007) sugere que o conceito de nicho seja dividido em duas classes: a classe Grinnelliana e a classe Eltoniana. Estas classes são definidas a partir do tipo de variáveis usadas, da escala espacial que estas variáveis são medidas e da capacidade de dispersão da espécie no ambiente. A classe Grinnelliana é baseada em condições ambientais e a classe Eltoniana é baseada em interações entre as espécies.

Esta visão geral sobre o conceito de nicho será importante para a compreensão do processo de modelagem que será explicado na Seção 2.3.

2.2 Dados

A constante evolução tecnológica tem aumentado a capacidade computacional de armazenamento e, com isso, a disponibilidade de dados na Internet também está crescendo. Da mesma forma, a capacidade computacional de processamento desses dados tem instigado o desenvolvimento de novas ferramentas para tratá-los e analisá-los. Os dados relacionados à biodiversidade que estão disponíveis na Internet são sobre a diversidade das espécies, taxonomia, variáveis ambientais, dentre outros (GRAHAM et al., 2004; SIQUEIRA, 2005).

Na modelagem de distribuição de espécies, são utilizados dados de ocorrência das espécies e dados de variáveis ambientais. Geralmente, as variáveis ambientais utilizadas representam fatores abióticos. As variáveis que representam os fatores bióticos raramente estão disponíveis e comumente são muito complexas para interpretar, por isso, elas não são muito usadas (SOBERÓN; PETERSON, 2005; SOBERÓN, 2007).

Os dados de ocorrência das espécies são pontos georreferenciados, ou seja, latitude e longitude, onde as espécies foram observadas e registradas. Esses dados podem ser divididos em pontos de presença e pontos de ausência. Os pontos de presença são as localidades em que a espécie foi registrada como presente e os pontos de ausência são as localidades em que a espécie foi registrada como ausente. A Figura 2.2 apresenta um exemplo de pontos de presença da espécie *Xylopia aromatica* registrados no estado de São Paulo.

Figura 2.2: Exemplo de pontos de presença da espécie *Xylopia aromatica* registrados no estado de São Paulo (Fonte: autora).

Frequentemente, existem somente dados de presença disponíveis nas bases de da-

dos, indicando a ocorrência das espécies (PHILLIPS; DUDÍK; SCHAPIRE, 2004). No entanto, alguns métodos de modelagem necessitam de informações sobre a ausência das espécies para gerar o modelo. Os dados de ausência das espécies só devem ser interpretados como uma ausência verdadeira, se a região estudada for muito bem explorada (SOBERÓN; PETERSON, 2005). Para resolver o problema da falta de dados sobre a ausência das espécies, pontos de pseudo-ausência podem ser gerados usando diferentes abordagens (GRAHAM et al., 2004).

As variáveis ambientais, também chamadas de camadas ambientais ou *layers*, também são georreferenciadas e devem pertencer à mesma região de estudo (PHILLIPS; ANDERSON; SCHAPIRE, 2006b). Os dados referentes às variáveis ambientais representam as informações disponíveis sobre o nicho ecológico da espécie na área de interesse. Alguns exemplos de variáveis ambientais são: temperatura, precipitação, altitude, tipo de vegetação, características do solo e radiação solar. A Figura 2.3 mostra um exemplo de variáveis ambientais para o estado de São Paulo.

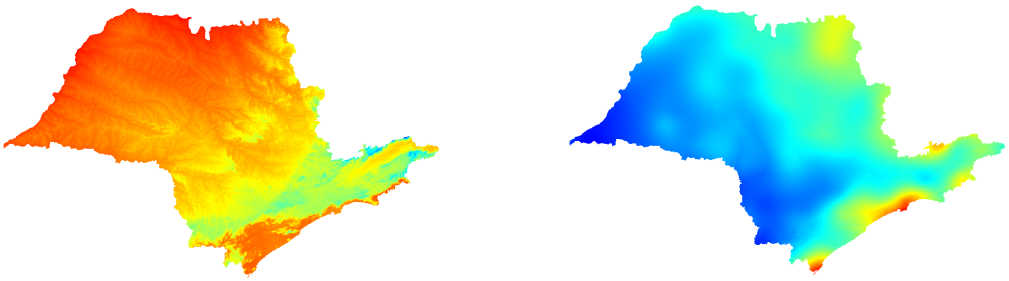


Figura 2.3: Exemplo de variáveis ambientais para o estado de São Paulo: temperatura e precipitação, respectivamente (Fonte: autora).

Qualquer modelo usado para fazer predições deve ser construído a partir de um conjunto confiável de dados. Marco-Júnior e Siqueira (2009) descrevem uma relação dos principais problemas encontrados nos dados disponíveis para modelagem de distribuição de espécies, bem como suas consequências. A seleção e o tratamento adequado dos dados, para modelagem de distribuição de espécies, pode interferir significativamente nos resultados obtidos, por isso, esta é uma área de pesquisa em expansão.

2.3 O Processo de Modelagem

O conceito de nicho ecológico, discutido na Seção 2.1, é o principal conceito relacionado à modelagem de distribuição de espécies. Isto porque, o processo de modelagem consiste em estimar um modelo que relacione as variáveis ambientais de uma determinada região com a ocorrência de espécies na mesma área estudada (SOBERÓN;

PETERSON, 2005) e, as variáveis ambientais representam as características conhecidas do nicho.

Conforme visto na Seção 2.2, a disponibilidade de variáveis ambientais que representam fatores abióticos é maior que a disponibilidade de variáveis ambientais que representam fatores bióticos. Por isso, os algoritmos de modelagem de distribuição de espécies procuram aproximar o modelo gerado do nicho fundamental, conforme definido por Soberón e Peterson (2005). Segundo Phillips, Anderson e Schapire (2006b), quanto maior a região estudada mais próxima do nicho fundamental será a modelagem.

Os pontos de ocorrência pertencem a uma área em que as espécies foram observadas, logo pode-se dizer que esta região contém condições ambientais adequadas para a sobrevivência da espécie. Assim, é mais apropriado dizer que os algoritmos de modelagem de distribuição de espécies procuram aproximar o modelo gerado do nicho realizado, conforme definido por Soberón e Peterson (2005), pois este representa a região potencial adequada para a espécie (SOBERÓN, 2007).

O processo de modelagem de distribuição de espécies pode ser visto como um método de auxílio à tomada de decisões. Desta forma, antes de iniciar o processo de modelagem, é necessário identificar o problema a ser resolvido. Impactos de mudanças climáticas, identificação de possíveis regiões adequadas para introduzir uma espécie e avaliação de possíveis rotas de propagação de doenças são alguns exemplos de problemas que podem ser auxiliados pela modelagem. As aplicações de modelagem serão discutidas na Seção 2.6.

Na especificação do problema devem ser definidas as espécies que serão estudadas, as variáveis ambientais que serão utilizadas, a região geográfica de interesse e as perguntas que devem ser respondidas pelo modelo (SANTANA et al., 2008). Como a escala espacial dos dados interfere na geração e na interpretação dos modelos (SOBERÓN; PETERSON, 2005), ela também deve ser definida com cuidado.

Computacionalmente, as etapas da modelagem de distribuição de espécies ocorrem conforme ilustrado na Figura 2.4. Após a definição do problema, os dados de ocorrência de espécies e variáveis ambientais são selecionados e tratados. O tratamento dos dados inclui remoção de erros de digitação, remoção de valores fora de um intervalo aceitável (por exemplo, temperaturas muito elevadas), verificação de limites dos pontos na região de interesse e transformação para algum padrão de coordenadas georreferenciadas. Em seguida, um algoritmo deve ser escolhido e seus parâmetros devem ser ajustados. A execução do algoritmo gera um modelo que deve ser validado por um especialista. A validação pode ser auxiliada por estatísticas fornecidas pela ferramenta de modelagem e pelo uso de dados não utilizados durante a geração do modelo.

Nesta etapa, o modelo estimado pode ser aceito ou pode ser necessário retornar para qualquer um dos níveis anteriores. Em (SANTANA et al., 2008) podem ser encontrados detalhes sobre cada etapa do processo de modelagem.

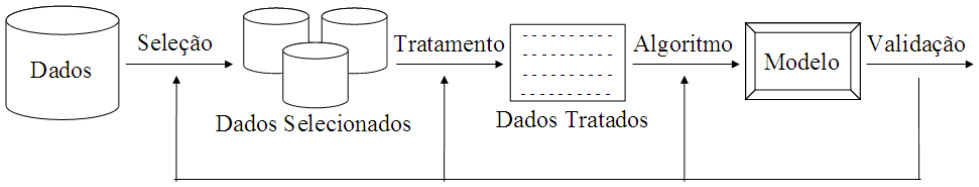


Figura 2.4: Etapas da modelagem de distribuição de espécies (Fonte: autora).

A quantidade de dados disponíveis e o método de modelagem escolhido interferem na qualidade do modelo estimado (SOBERÓN; PETERSON, 2005). Além disso, a qualidade dos dados selecionados também interfere no resultado da modelagem (MARCO-JÚNIOR; SIQUEIRA, 2009). Por isso, é aconselhável um conhecimento básico sobre os algoritmos de modelagem.

Uma estratégia interessante para aumentar a qualidade do modelo e melhorar o desempenho do estimador é o uso de métodos de pré-análise. Esses métodos são usados como ferramentas de pré-processamento e podem auxiliar, por exemplo, na escolha de um conjunto adequado de variáveis ambientais de acordo com a espécie estudada (REPORT-OM, 2006). O *Jackknife* (RODRIGUES et al., 2008), por exemplo, é um método de reamostragem que pode ser utilizado na pré-análise. Outro exemplo é o *Qui Quadrado* (LI; BIAN; YAN, 2006; WILSON; HILFERTY, 1931), um valor de dispersão para duas variáveis. O *Jackknife* também pode ser usado para avaliar a influência de cada ponto de ocorrência em conjuntos pequenos de dados (PEARSON et al., 2007) e o *Qui Quadrado* pode ser usado para relacionar a qualidade do modelo com o tamanho das amostras (PETERSON, 2001).

De maneira similar, após a estimativa dos modelos, técnicas de pós-análise podem ser usadas para auxiliar a validação do modelo estimado. Além disso, uma vez aceito o modelo, técnicas de pós-análise podem colaborar na elaboração e compreensão de respostas para as hipóteses formuladas na definição do problema (REPORT-OM, 2006). Exemplos de medidas de pós-análise comumente usadas são: acuidade, erro de omissão, erro de sobreprevisão, curva ROC (*Receiver Operating Characteristics*) e AUC (*Area Under the Curve*) (PRATI; BATISTA; MONARD, 2008). Todas estas medidas são calculadas a partir da matriz de confusão. A matriz de confusão é uma matriz bidimensional quadrada contendo o número de exemplos classificados corretamente e o número de classificações preditas para cada classe (MONARD; BARANAUSKAS, 2003a). Além destas, outras medidas podem ser derivadas a partir da matriz de confusão (FIEL-

DING; BELL, 1997).

A Figura 2.5 apresenta o processo de modelagem de maneira mais elucidativa. Os pontos georreferenciados de ocorrência das espécies, latitude e longitude, em conjunto com os pontos de nicho são utilizados como dados de entrada do algoritmo. Os pontos de nicho são os valores que cada variável ambiental selecionada assume em cada ponto de ocorrência. A partir dos dados de entrada, o algoritmo procura por relações entre as variáveis ambientais nos pontos de ocorrência, estabelecendo as condições ambientais adequadas para a espécie. Estas relações são expressas em forma de funções de probabilidade de ocorrência, produzindo, assim, um modelo baseado em nicho. O modelo estimado é, então, projetado em um determinado cenário, que pode ser referente às condições ambientais atuais, do passado ou do futuro. Esta projeção resulta em um mapa de distribuição potencial da espécie.

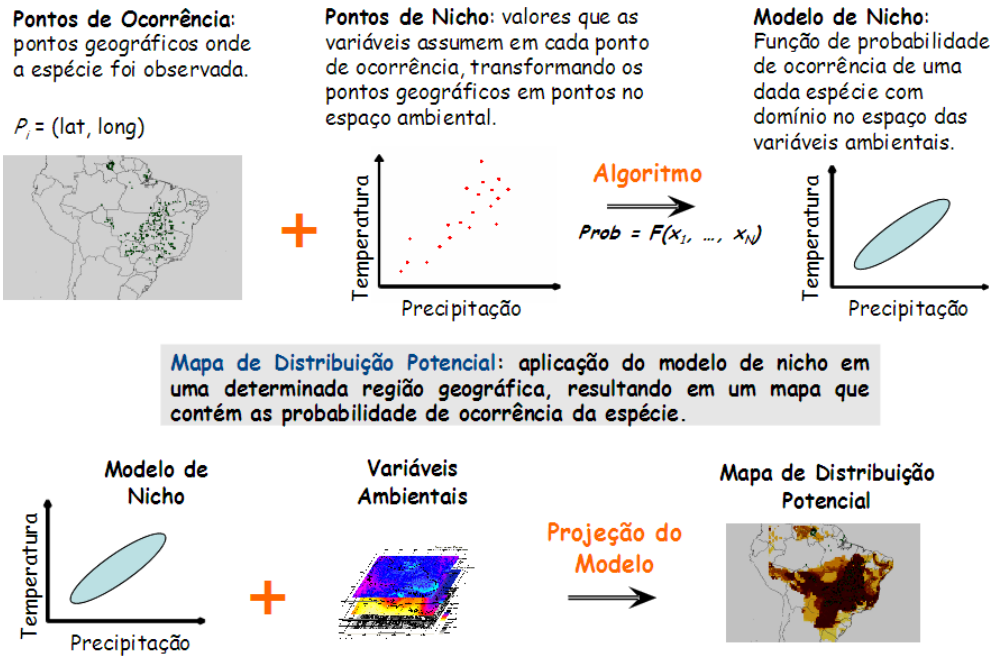


Figura 2.5: Processo de modelagem de distribuição de espécies (Fonte: Siqueira (2005)).

2.4 *openModeller*

Segundo Sutton, Giovanni e Siqueira (2007), geralmente, as ferramentas de modelagem de distribuição de espécies são desenvolvidas para executar apenas um algoritmo específico, como o *DesktopGarp* (PEREIRA, 2003) e o *MaxEnt* (PHILLIPS; ANDERSON; SCHAPIRE, 2006a). Além disso, outros requisitos, como diferentes preparações de dados de entrada e diferentes plataformas, dificultam o uso e a distribuição de tais ferramentas. Canhos et al. (2004) discutem detalhadamente a importância do desenvolvimento de ferramentas que auxiliem a execução de diversas tarefas relacionadas à

modelagem de distribuição de espécies em um ambiente integrado.

Com o objetivo de tratar os desafios relacionados à modelagem de distribuição de espécies, surgiu o projeto de pesquisa *openModeller* (MUÑOZ et al., 2011). O projeto *openModeller* foi financiado pela Fundação de Amparo à Pesquisa do Estado de São Paulo (Fapesp) e desenvolvido por três instituições de pesquisa: Centro de Referência em Informação Ambiental (CRIA), Escola Politécnica da Universidade de São Paulo (EPUSP) e Instituto Nacional de Pesquisas Espaciais (INPE).

O nome do projeto deu nome à ferramenta *openModeller* (CRIA; EPUSP; INPE, 2006), cujo principal objetivo é fornecer à comunidade científica um conjunto, robusto e flexível, de estratégias relacionadas à modelagem de distribuição de espécies (MUÑOZ et al., 2011). O *openModeller* é um software livre, gratuito, escrito em C++ e executa nas plataformas MS Windows, Mac OS X e GNU/Linux (SUTTON; GIOVANNI; SIQUEIRA, 2007).

Uma das vantagens da ferramenta *openModeller* sobre outras ferramentas é a união de vários algoritmos de modelagem em uma única arquitetura. Isto permite que os resultados de algoritmos diferentes sejam comparados mais facilmente, pois os mesmos dados de entrada podem ser usados para criar vários modelos e o formato de saída é comum a todos eles (SUTTON; GIOVANNI; SIQUEIRA, 2007). A Figura 2.6 apresenta a arquitetura simplificada da ferramenta *openModeller*. A arquitetura estendida pode ser encontrada em (MUÑOZ et al., 2011).

A ferramenta *openModeller* utiliza bibliotecas externas para ler e escrever dados georreferenciados. Os algoritmos são tratados como bibliotecas dinâmicas, ou seja, todos devem ter métodos específicos pré-estabelecidos que permitam a comunicação com o núcleo de modelagem, que se encarrega de realizar as tarefas comuns. Além disso, a ferramenta *openModeller* possui várias interfaces para o usuário (MUÑOZ et al., 2011).

O *openModeller Desktop* é uma versão da ferramenta que oferece um ambiente amigável aos usuários. Através de uma interface gráfica é possível selecionar os dados de entrada, executar o algoritmo desejado e visualizar os resultados obtidos na geração do modelo (SUTTON; GIOVANNI; SIQUEIRA, 2007). A interface *Console* possui um conjunto de funcionalidades que podem ser usadas por linha de comando. Esta interface facilita a execução de vários experimentos através do uso de *scripts* e também permite o uso da ferramenta *openModeller* como uma biblioteca de outras aplicações. Muñoz et al. (2011) descrevem maiores detalhes do funcionamento da ferramenta *openModeller*.

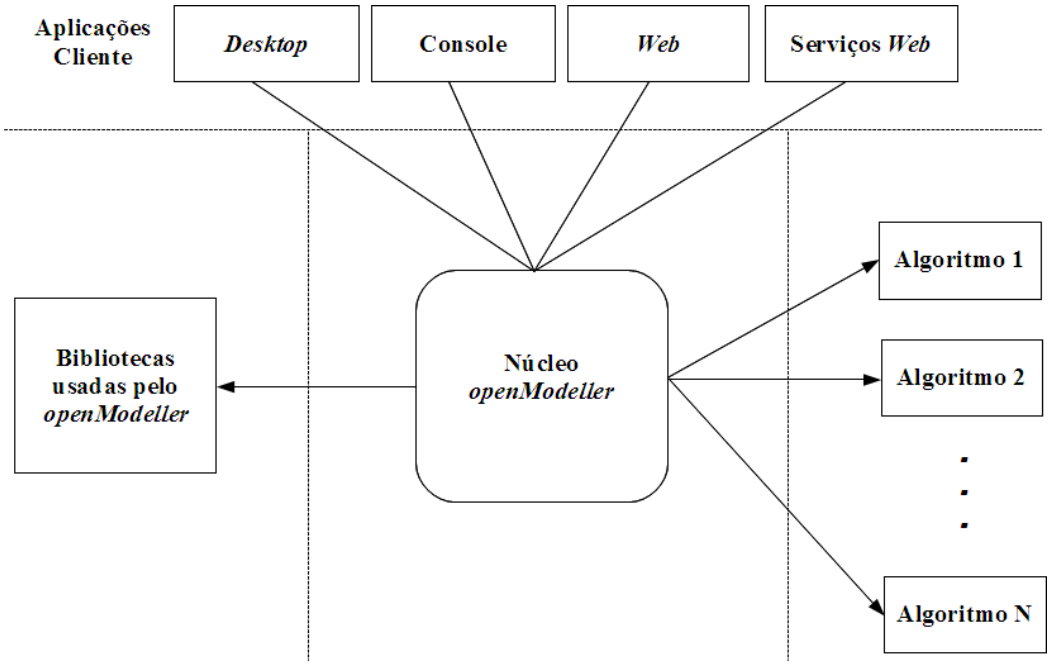


Figura 2.6: Arquitetura simplificada da ferramenta *openModeller* (Fonte: Muñoz et al. (2011)).

Por ser uma ferramenta com muitos recursos, o *openModeller* tem chamado a atenção de instituições como a NASA e está sendo usada por pesquisadores de muitos países, como Estados Unidos (Universidade do Kansas, Universidade da Califórnia em Berkeley), Inglaterra (Universidade de Oxford), Itália, Taiwan, dentre outros (REPORT-OM, 2008).

2.5 Algoritmos de Modelagem

A escolha do algoritmo de modelagem depende do problema a ser resolvido e dos dados selecionados. Segundo Soberón e Peterson (2005), a tarefa dos algoritmos de modelagem é encontrar regiões cujos valores das variáveis ambientais utilizadas como entrada se assemelham aos valores das variáveis nos pontos de ocorrência fornecidos. A qualidade do modelo depende de quão bem os pontos de ocorrência representam o nicho abiótico e da capacidade de extrapolação do algoritmo utilizado (SOBERÓN; PETERSON, 2005).

O algoritmo *Bioclim* (NIX, 1986) *apud* (SIQUEIRA, 2005) estima o modelo apenas com variáveis ambientais climáticas. Sua implementação é baseada em envelopes bioclimáticos, ou seja, um envelope (representado por um intervalo) para cada variável ambiental é calculado baseado nos valores de média, m, e desvio padrão, s, associados aos pontos de ocorrência. O envelope é definido por $[m - c * s, m + c * s]$, em que c é um parâmetro de entrada de corte. Além do envelope, cada variável ambiental é

limitada por seus valores máximo e mínimo extraídos a partir dos pontos de ocorrência. Um ponto é classificado como adequado para a espécie, se todos os valores das variáveis ambientais naquele ponto estão dentro dos respectivos envelopes. O ponto é classificado como marginal, se um ou mais valores das variáveis está fora do seu envelope, mas dentro dos limites máximo e mínimo. Por fim, o ponto é classificado como inadequado, se um ou mais valores das variáveis está fora dos limites máximo e mínimo.

O *AquaMaps* é um algoritmo para a modelagem de distribuição de organismos marinhos (KASCHNER et al., 2010) e se assemelha aos envelopes bioclimáticos, em que cada variável tem um intervalo associado e valores limites de aceitação. Este algoritmo calcula os valores dos intervalos a partir de percentis e requer um conjunto específico de variáveis ambientais: profundidade, temperatura da superfície do mar (para todas as espécies pelágicas), temperatura do fundo do mar (para todas as espécies não pelágicas), salinidade da superfície do mar (para todas as espécies pelágicas), salinidade do fundo do mar (para todas as espécies não pelágicas), produção primária, concentração de gelo no mar e distância até a costa.

ENFA (*Ecological-Niche Factor Analysis*) é um algoritmo que compara a distribuição dos pontos de ocorrência no espaço de variáveis ambientais com a distribuição de todo o conjunto de células (HIRZEL et al., 2002). A partir desta comparação, o algoritmo calcula dois fatores: a marginalidade, que é um fator baseado na diferença entre a média global dos valores da variável ambiental e a média dos valores da variável nos pontos de ocorrência; e a especialização, que é um fator baseado na diferença entre o desvio padrão da distribuição global e o desvio padrão da distribuição dos pontos de ocorrência.

Um algoritmo bastante utilizado em modelagem de distribuição de espécies é o GARP (*Genetic Algorithm for Rule-set Production*) (STOCKWELL; NOBLE, 1992). Este algoritmo é baseado em algoritmos genéticos e produz um conjunto de regras, representando a população, induzidas a partir dos dados de entrada. A cada iteração do algoritmo, a população é modificada por operadores de mutação e *crossover*, de tal forma que apenas as melhores regras façam parte do modelo (STOCKWELL; PETERS, 1999). Como o GARP é um algoritmo não determinístico, uma abordagem para tratar a variabilidade dos resultados é gerar vários modelos, selecionar os n melhores e compor um modelo final. No modelo final, cada ponto da área em estudo possui uma probabilidade de presença, que é igual a proporção de modelos que classificaram o ponto como presença. Esta abordagem é conhecida como GARP com melhores subconjuntos (GARP *with best subsets*) (ANDERSON; LEW; PETERSON, 2003).

SVM (*Support Vector Machines*) é um método de aprendizagem supervisionada que utiliza o hiperplano de separação ótima, ou seja, o hiperplano que maximiza a margem de separação entre as classes (VAPNIK, 1995). Embora o SVM tenha uma boa capacidade de generalização, suas principais desvantagens são a dificuldade de interpretar os resultados e a sua sensibilidade na escolha dos parâmetros (LORENA et al., 2011). Inicialmente, o algoritmo SVM trabalhava apenas com dados linearmente separáveis. Posteriormente, outras estratégias de vetores de suporte foram desenvolvidas para permitir a classificação de dados não linearmente separáveis e para melhorar o desempenho do algoritmo original (CORTES; VAPNIK, 1995; SCHÖLKOPF et al., 2000; SCHÖLKOPF et al., 2001).

As Redes Neurais Artificiais (ANN – *Artificial Neural Networks*) (HAYKIN, 2001) são inspiradas no funcionamento do cérebro humano e são formadas por unidades (nós) chamadas neurônios artificiais. Esses neurônios recebem estímulos de entrada, que podem ser valores que codificam padrões de entrada ou as saídas de outros neurônios. Uma ANN é formada pela conexão de vários neurônios. Cada neurônio recebe um ou mais valores de entrada e produz um sinal de saída, calculado a partir de uma função de ativação. Existem vários paradigmas de ANN, que diferem entre si principalmente pela topologia da rede e pelo método de aprendizagem. Na ferramenta *openModeller*, está implementada a estrutura clássica *Multilayer Perceptron* com o algoritmo *backpropagation* (RODRIGUES et al., 2009; RODRIGUES et al., 2010).

A ferramenta *openModeller* oferece diferentes algoritmos de modelagem de distribuição de espécies. Todos os algoritmos descritos acima estão disponíveis no *openModeller*. Além destes, outros algoritmos também disponíveis são: CSM (*Climate Space Model*) – baseado em componentes principais (ROBERTSON; CAITHNESS; VILLET, 2001); *Envelope Score* – variação do Bioclim (REPORT-OM, 2008); Distância Ambiental (*Environmental Distance*) – baseado em métricas de dissimilaridade ambiental (REPORT-OM, 2007); e Entropia Máxima, que será descrito em detalhes na Seção 3.1.

Vários estudos comparativos entre diferentes algoritmos de modelagem têm sido realizados no contexto da distribuição de espécies. Elith et al. (2006) compararam 16 métodos de modelagem de distribuição de espécies. Além de métodos clássicos, eles analisaram métodos pouco utilizados na modelagem e métodos que haviam sido implementados pouco tempo antes do estudo comparativo. Em todos os métodos analisados foram utilizados apenas dados de presença para estimar o modelo e dados de presença e ausência para avaliar os resultados. Na avaliação dos resultados foram encontradas evidências claras de que modelos que utilizam somente pontos de presença podem fazer predições suficientemente precisas para serem usadas em aplicações em que a estimativa da distribuição de espécies é relevante.

Lorena et al. (2008) compararam SVM, algoritmos genéticos e árvores de decisão na modelagem de distribuição de uma planta. Dentre as três técnicas comparadas, SVM foi a que estimou modelos melhores. Tognelli et al. (2009) avaliaram o desempenho de oito métodos de modelagem para a distribuição de dez espécies de insetos da Patagônia. Eles utilizaram cinco medidas de desempenho e classificaram os métodos em quatro grupos, sendo o melhor grupo composto por Entropia Máxima e ANN. Lorena et al. (2011) compararam nove estratégias de modelagem e identificaram as três melhores. Dentre estas três, destacam-se ANN e SVM. Vale salientar que nenhum algoritmo baseado em Entropia Máxima fez parte do estudo de Lorena et al. (2011).

Além dos algoritmos apresentados nesta seção, existem muitas outras técnicas de Inteligência Artificial que tem sido usadas para modelar sistemas ambientais. Alguns exemplos são: autômatos celulares, sistemas multiagentes, sistemas *fuzzy* e sistemas híbridos (CHEN; JAKEMAN; NORTON, 2008). No entanto, o objetivo desta seção foi apenas fornecer uma visão geral da variedade de abordagens possíveis para a modelagem de distribuição de espécies.

2.6 Aplicações da Modelagem de Distribuição de Espécies

A modelagem de distribuição de espécies têm sido usada em várias aplicações relacionadas à conservação da biodiversidade. Raxworthy et al. (2007) conduziram um estudo sobre a especiação de espécies de lagartixas. Eles integraram a modelagem de distribuição de espécies com dados morfológicos e dados de DNA mitocondrial para avaliar os limites geográficos das espécies. O trabalho realizado por Raxworthy et al. (2007) é um bom exemplo de como a modelagem de distribuição de espécies pode contribuir para o estudo filogenético de espécies.

As mudanças climáticas têm afetado a distribuição de várias espécies no mundo inteiro. Fitzpatrick et al. (2008) avaliaram os impactos das possíveis mudanças climáticas futuras na distribuição de 100 espécies do gênero *Banksia*, uma planta típica da Austrália. Dentre todos os cenários analisados, a expansão ou estabilidade de apenas 6% das espécies foram previstas. De acordo com o estudo, o restante das espécies se mostraram em declínio ou em extinção total.

Bartel e Sexton (2009) utilizaram a modelagem de distribuição de espécies, com um algoritmo de Entropia Máxima, associado a imagens de satélite para monitorar a variação espaço-temporal na adequabilidade do habitat de uma espécie de borboleta. A avaliação da dinâmica do habitat de espécies ameaçadas pode contribuir no plane-

jamento de recuperação das espécies através da identificação de áreas não habitadas, porém adequadas para a espécie.

Outra aplicação muito interessante é o uso de modelagem de distribuição de espécies na identificação de áreas adequadas para a produção de matéria-prima para biocombustíveis. Evans, Fletcher-Jr. e Alavalapati (2010) analisaram dois métodos de modelagem para duas espécies que servem como matéria-prima para biocombustíveis. Segundo Evans, Fletcher-Jr. e Alavalapati (2010), apesar de ser mais utilizada para problemas relacionados à conservação da biodiversidade, a modelagem de distribuição de espécies é uma poderosa ferramenta para determinar o potencial uso de uma região.

Existem muitas aplicações possíveis para a modelagem de distribuição de espécies. No entanto, por uma questão de escopo, apenas algumas foram descritas nesta seção. Elith et al. (2011) apresentam alguns exemplos de aplicações de modelagem de distribuição de espécies e mostram a variedade de organismos e regiões em que esta estratégia é aplicada. Além disso, Zimmermann et al. (2010) descrevem algumas tendências na área de modelagem de distribuição de espécies.

2.7 Epítome

O principal objetivo deste capítulo foi apresentar uma visão geral da modelagem de distribuição de espécies. Inicialmente, foram discutidos o conceito de nicho ecológico e sua formalização abstrata, nicho fundamental. Na discussão sobre o conceito de nicho, também foi apresentado o conceito de nicho realizado. O processo de modelagem consiste em encontrar relações entre as variáveis ambientais e os dados de ocorrência das espécies em um determinada região. O modelo gerado é, portanto, uma aproximação do nicho realizado.

Os dados de ocorrência podem ser divididos em pontos de presença, localidades onde a espécie foi registrada como presente, e pontos de ausência, localidades onde a espécie foi registrada como ausente. Os pontos de ausência raramente estão disponíveis, por isso alguns métodos de modelagem geram pontos de pseudo-ausência. As variáveis ambientais representam as informações disponíveis sobre o nicho ecológico da espécie em estudo. Todos os dados utilizados no processo de modelagem são georreferenciados e devem pertencer à mesma região geográfica.

As etapas do processo de modelagem podem ser resumidas em: definição do problema, seleção e tratamento de dados, seleção e execução do algoritmo de modelagem, e validação do modelo. Para melhorar a qualidade dos modelos, podem ser utilizadas técnicas de pré-análise e de pós-análise. O modelo resultante é projetado em uma

região, gerando um mapa de distribuição potencial da espécie.

Existem diversos algoritmos de modelagem baseados em diferentes estratégias e disponíveis em várias ferramentas. Este trabalho foi desenvolvido no contexto da ferramenta *openModeller*, uma ferramenta de código aberto, escrita em C++ e que pode ser executada em diferentes plataformas. Seu principal objetivo é fornecer, em um único ambiente, vários algoritmos de modelagem. Existem diversos algoritmos disponíveis no *openModeller*, como Entropia Máxima, ANN, GARP e SVM. Também é importante salientar a variedade de aplicações possíveis da modelagem de distribuição de espécies, como avaliação do impacto das mudanças climáticas, identificação da variabilidade do habitat de espécies e identificação de áreas potenciais para o desenvolvimento de uma espécie.

3 Métodos Fundamentados em Teoria da Informação para Modelagem de Distribuição de Espécies

O estudo de Teoria da Informação iniciou-se na década de 1940 quando Claude E. Shannon publicou o artigo "*A Mathematical Theory of Communication*" (SHANNON, 1948). A Teoria da Informação é comumente descrita como um ramo da Teoria das Comunicações. No entanto, a Teoria da Informação têm contribuições fundamentais em diversas áreas de pesquisa, conforme ilustrado na Figura 3.1.

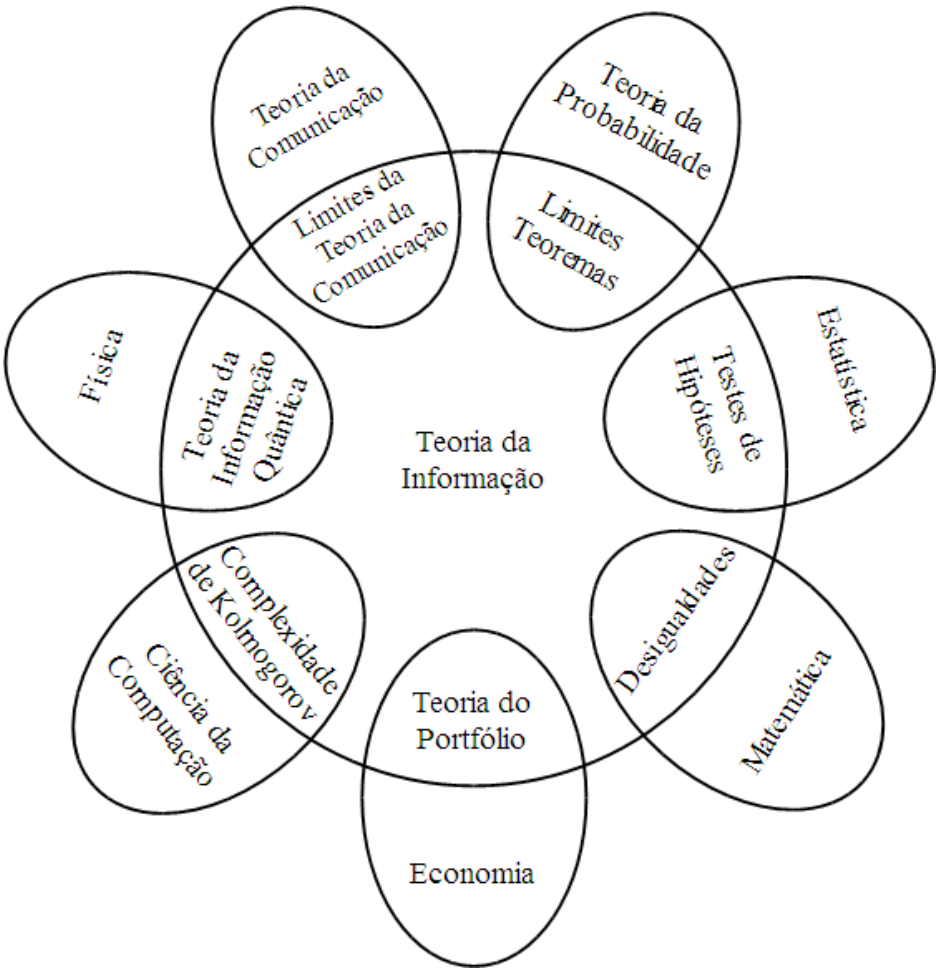


Figura 3.1: Relacionamento da Teoria da Informação com outras áreas de pesquisa (Fonte: Cover e Thomas (2006)).

A Figura 3.1 exibe apenas algumas possibilidades de contribuição da Teoria da Informação em outras áreas de pesquisa, mas não se limita aos exemplos mostrados. Neste capítulo, serão apresentados dois métodos fundamentados em Teoria da Informação que foram estudados durante o desenvolvimento deste trabalho e utilizados em alguma etapa do processo de modelagem de distribuição de espécies. A Seção 3.1 apresenta os fundamentos do princípio da Entropia Máxima, o seu uso na estimativa de modelos de distribuição de espécies e algumas aplicações. A Seção 3.2 mostra os conceitos básicos do princípio da Descrição com Comprimento Mínimo. Ao final do capítulo, a Seção 3.3 apresenta um resumo do mesmo.

3.1 Fundamentos de Entropia Máxima

A compreensão do conceito de informação é um requisito fundamental para o estudo de assuntos relacionados à Teoria da Informação. No entanto, não é raro encontrar na literatura uma confusão acerca do que entende-se por informação, pois esta se confunde tanto com conhecimento quanto com dado. A dificuldade em estabelecer uma definição para informação deve-se à amplidão de seu conceito. Desta forma, tornam-se necessárias algumas considerações sobre estas expressões.

Segundo Setzer (2001), dados são quaisquer sequências de símbolos quantificáveis, ou seja, podem ser representados formalmente por estruturas matemáticas. Pode-se dizer, então, que dados são exclusivamente sintáticos, isto é, sem qualquer significado associado. A informação não pode ser formalizada matematicamente, pois é uma abstração informal que representa alguma coisa significativa na mente de alguém. No entanto, a informação pode ser representada por dados. A principal diferença entre dado e informação é que a segunda contém semântica, isto é, um significado associado a ela.

Ainda de acordo com Setzer (2001, p. 245), o conhecimento pode ser entendido como uma "abstração interior, pessoal, de algo que foi experimentado, vivenciado, por alguém". Portanto, conhecimento, assim como informação, só pode ser caracterizado e não definido. Além disso, conhecimento não pode ser representado, porque depende da experiência pessoal de alguém. No entanto, o desenvolvimento de sistemas inteligentes requer a representação de fatos, ações e experiências.

A representação do conhecimento (RUSSELL; NORVIG, 2004; REZENDE, 2003) permite que os sistemas sejam capazes de armazenar o que sabem com a finalidade de "raciocinar" e "aprender", mesmo na presença de incerteza. Por isso, a representação do conhecimento é uma das vastas áreas de pesquisa que compõe a Inteligência

Artificial.

Embora a informação não seja quantificável, é desejável medir de alguma maneira a informação transmitida na ocorrência de um evento. A informação transmitida em um acontecimento está relacionada à probabilidade de ocorrência do mesmo. Uma observação inesperada transmite mais informação que uma observação esperada, ou seja, um acontecimento certo não transmitirá informação alguma (TAVARES, 2004).

Suponha que um sistema tenha o seguinte conjunto de configurações possíveis:

$$X = \{x_i \mid i = 1, 2, \ldots, n\}$$

e que p seja uma distribuição de probabilidade sobre X, de forma que cada uma das configurações x_i ocorra com a probabilidade $p_i = p(x_i)$, ou seja, p_i é a probabilidade de ocorrência da configuração x_i. A distribuição p atribui um valor de probabilidade não negativo a cada x_i, $p_i \geq 0$, e a soma das probabilidades de ocorrência de cada configuração de X é igual a 1 , conforme a Equação (3.1).

$$\sum_{i=1}^{n} p_i = 1. \tag{3.1}$$

Segundo Haykin (2001), quando, para alguma configuração x_i, $p_i = 1$, o que requer que $p_k = 0$, $\forall k \neq i$, então não há "surpresa" na ocorrência desta configuração. Nesta situação, nenhuma "informação" é transmitida, pois já se sabe qual configuração do sistema ocorrerá. No entanto, se as várias configurações podem ocorrer com diferentes probabilidades e, particularmente, a probabilidade p_i for baixa, então há mais "surpresa" e, portanto "informação" quando X assumir o valor x_i em vez de outro valor com maior probabilidade. Assim, antes da ocorrência de uma configuração, existe uma incerteza a seu respeito. Quando a configuração ocorre, há uma surpresa. Após a ocorrência da configuração há um aumento de informação. Essas três quantidades são obviamente as mesmas e a informação transmitida é inversamente proporcional à probabilidade de ocorrência do evento (HAYKIN, 2001).

Para qualquer distribuição de probabilidade p sobre X, existe uma quantidade associada chamada **entropia** (COVER; THOMAS, 2006), denotada por $H(p)$ e definida na Equação (3.2), que representa a medida de informação de X. Assim, pode-se dizer que a entropia é uma medida de incerteza, ou de imprevisibilidade, de um sistema e que está relacionada à probabilidade de ocorrência de suas possíveis configurações. No exemplo citado anteriormente, quando alguma configuração x_i, tem probabilidade $p_i = 1$, o que requer que $p_k = 0$, $\forall k \neq i$, então não há incerteza. Assim, a entropia é

limitada inferiormente por zero, ou seja, um modelo sem incerteza não tem entropia.

$$H(p) = -\sum_{i=1}^{n} p_i \ln p_i, \tag{3.2}$$

em que a base do logaritmo é arbitrária. Como a entropia geralmente é usada para medir a informação na troca de mensagens, usualmente a definição de entropia está na base 2, pois é medida em bits. No entanto, no estudo de fenômenos naturais, geralmente se usa o logaritmo natural. Portanto, neste trabalho foi adotado o logaritmo natural, ln.

Quando as várias configurações são igualmente prováveis, ou seja, têm a mesma probabilidade de ocorrência, então tem-se a **Entropia Máxima** (TAVARES, 2004). A entropia é máxima no estado de máxima desordem. Nesse caso, a falta de organização é total e a liberdade de escolha é completa, isto é, a distribuição de probabilidade é uniforme. Desta forma, a entropia é uma função côncava da distribuição, limitada superiormente por $\ln |X|$, em que $|X|$ é o número de elementos de X.

Para melhor compreensão, suponha que $X = \{a, b\}$. Se $p(a) = 1$ ou $p(b) = 1$, então não haverá incerteza, portanto $H(p) = 0$. Se $p(a) = \frac{1}{2}$, então $p(b) = \frac{1}{2}$ e a incerteza será máxima, o que corresponde a Entropia Máxima, $H(p) = 1$. A Figura 3.2 mostra a função de entropia para o exemplo dado.

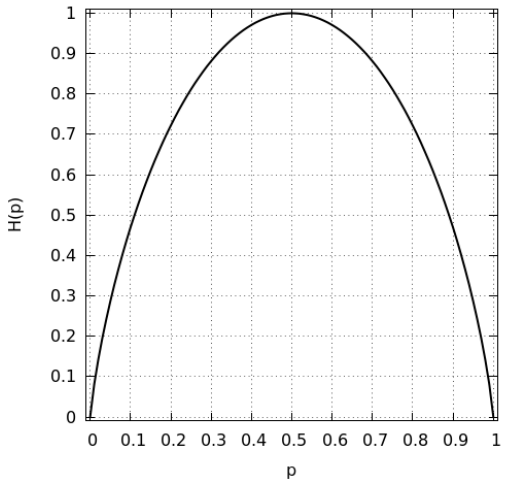


Figura 3.2: Função de entropia para um conjunto com dois elementos (Fonte: Cover e Thomas (2006)).

Todo sistema natural, quando deixado livre, evolui para um estado de máxima desordem, correspondente à Entropia Máxima (TAVARES, 2004). No caso dos algoritmos de modelagem de distribuição de espécies disponíveis até o momento, desordem é entendida como a falta de localização espacial.

A estimativa da distribuição de probabilidade com Entropia Máxima pode ser di-

vidida em duas etapas: seleção de características sobre o sistema a partir do conjunto de amostras e seleção de um modelo deste sistema (BERGER; PIETRA; PIETRA, 1996). Os termos "distribuição de probabilidade" e "modelo" serão usados como sinônimos para designar o resultado final da aplicação do princípio da Entropia Máxima.

A relação entre a ocorrência de um evento e as características do sistema, representadas por variáveis, pode ser bastante complexa. Assim, em muitas situações é desejável que as características do sistema sejam expandidas. Geralmente, esta expansão é realizada pela transformação das variáveis originais. Em Aprendizagem de Máquina, as variáveis originais e suas transformações são conhecidas como *features* (ELITH et al., 2011).

O modelo do sistema é estimado a partir de um conjunto de *features* que impõem restrições à distribuição de probabilidade. Normalmente, existe um número infinito de modelos que satisfazem as restrições. Portanto, o problema consiste em escolher um modelo probabilístico de acordo com este conhecimento prévio (HAYKIN, 2001).

O conjunto de *features* deve ser escolhido de acordo com as amostras disponíveis para treinamento. Segundo Jaynes (1990), utilizar somente informações sobre o que é conhecido do sistema, na estimativa da distribuição de probabilidade, é o que justifica o uso do modelo para predição. Esta escolha evita que hipóteses sobre fatos desconhecidos sejam assumidos como verdadeiros. Desta forma, um estimador de modelo pelo princípio da Entropia Máxima deve escolher a distribuição de probabilidade que maximiza a entropia e satisfaz todas as restrições impostas.

3.1.1 Análise do Princípio da Entropia Máxima

Para a melhor compreensão do princípio da Entropia Máxima, foi implementado o algoritmo de classificação ID3. O ID3 é um algoritmo de árvore de decisão que utiliza Entropia Máxima na escolha de seus nós. As árvores de decisão são representações simples do conhecimento cujo objetivo é classificar um número finito de categorias. A ideia é construir uma árvore a partir de um conjunto de registros com seus respectivos valores de classificação (MONARD; BARANAUSKAS, 2003b).

Cada registro do conjunto de dados é composto por pares de valores/atributos, sendo que um desses atributos representa uma classificação ou decisão. A árvore deve ser construída com base nas respostas de perguntas sobre os atributos para classificar corretamente os valores do atributo de decisão (MONARD; BARANAUSKAS, 2003b). A Tabela 3.1 apresenta um exemplo de registros e valores/atributos. Cada linha na tabela representa um registro e cada coluna representa um atributo. A última coluna é o

atributo de decisão.

Tabela 3.1: Exemplo de um conjunto de dados.

Prev_Tempo	Temperatura	Umidade	Vento	Classe
ensolarado	quente	alta	falso	não
ensolarado	quente	alta	verdadeiro	não
nublado	quente	alta	falso	sim
chuvoso	amena	alta	falso	sim
chuvoso	fria	normal	falso	sim
chuvoso	fria	normal	verdadeiro	não
nublado	fria	normal	verdadeiro	sim
ensolarado	amena	alta	falso	não
ensolarado	fria	normal	falso	sim
chuvoso	amena	normal	falso	sim
ensolarado	amena	normal	verdadeiro	sim
nublado	amena	alta	verdadeiro	sim
nublado	quente	normal	falso	sim
chuvoso	amena	alta	verdadeiro	não

Uma árvore de decisão é geralmente construída de maneira *top-down*, guiada pela frequência de informação nos exemplos, sem considerar a ordem em que esses exemplos aparecem (QUINLAN, 1986). A Figura 3.3 mostra um exemplo de uma árvore de decisão com seleção aleatória de atributos para os dados da Tabela 3.1.

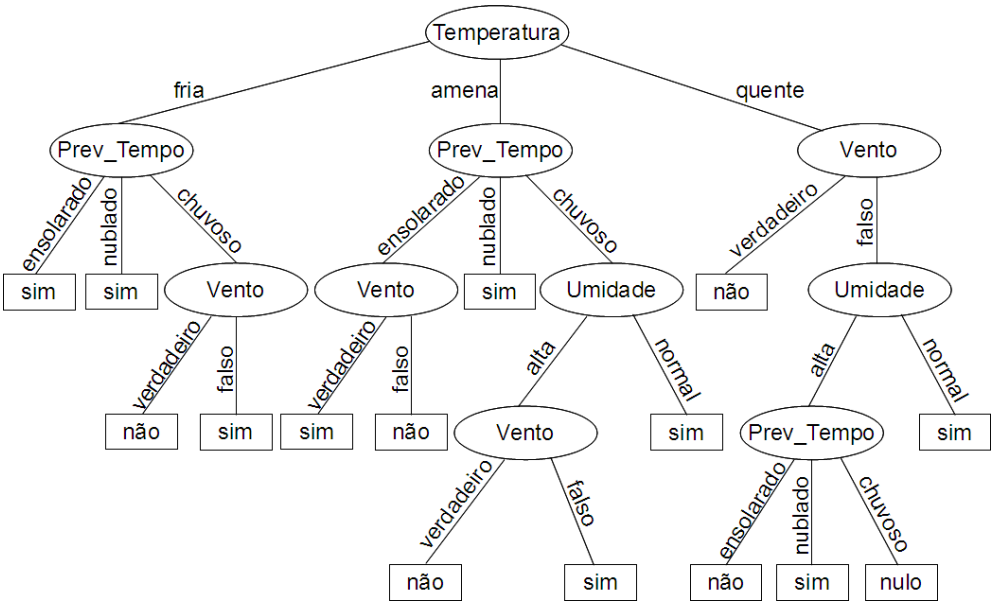


Figura 3.3: Exemplo de uma árvore de decisão com seleção aleatória de atributos (Fonte: Quinlan (1986)).

Os nós da árvore representam os atributos, os arcos representam os valores para os atributos e as folhas representam as classes. Cada folha especifica um valor para os exemplos descritos pelo caminho da raiz até a folha. Cada nó da árvore deve ser o atributo mais informativo do conjunto de dados ainda não incluído no caminho a partir

da raiz. A árvore da Figura 3.3 foi construída de maneira aleatória, ou seja, nenhum critério de seleção de atributos foi considerado na escolha dos nós da árvore.

O algoritmo ID3 foi implementado para a tarefa de indução de árvores de decisão. Este algoritmo foi desenvolvido por Quinlan (1986) e utiliza teoria da informação para a escolha do atributo mais informativo. Posteriormente, Quinlan (1993) aprimorou o algoritmo ID3, dando origem ao seu sucessor, o C4.5.

O algoritmo ID3 opera sobre um conjunto de exemplos T que estão classificados em uma das k classes do atributo de decisão $C = \{C_1, C_2, \ldots, C_k\}$. A árvore de decisão é induzida recursivamente através dos seguintes passos:

1) Se T é vazio, a árvore é um único nó com o valor *Falha*;

2) Se todos os exemplos de T pertencem a uma única classe C_j, tal que $j \in \{1, 2, \ldots, k\}$, então a árvore de decisão para T é um único nó com a classe C_j;

3) Se T contém exemplos que pertencem a classes diferentes, escolher um atributo, criar um nó com o nome do atributo escolhido e arcos rotulados com os valores desse atributo. Particionar o conjunto de dados em subconjuntos que contenham os exemplos com o mesmo valor para o atributo selecionado;

4) Repetir o procedimento para todos os subconjuntos.

A ordem de escolha dos atributos determina a complexidade da árvore construída. O algoritmo ID3 utiliza entropia para determinar o atributo mais informativo. Conforme descrito na Seção 3.1 do Capítulo 3, a informação transmitida na ocorrência de um evento é inversamente proporcional à probabilidade de ocorrência do mesmo.

Considerando um conjunto de exemplos T que estão classificados em uma das k classes do atributo de decisão $C = \{C_1, C_2, \ldots, C_k\}$, a quantidade de informação necessária para identificar cada classe é $Info(T) = H(p)$, em que p é a distribuição de probabilidade de C e $H(p)$ é a entropia de p.

No caso do exemplo da Tabela 3.1 e de acordo com a definição de entropia (3.2),

$$\begin{aligned} Info(T) &= H\left(\frac{9}{14}, \frac{5}{14}\right) \\ &= 0,94 \ , \end{aligned}$$

pois existem 9 exemplos classificados como “sim” e 5 exemplos classificados como “não”, totalizando 14 exemplos.

Para escolher o melhor atributo X (diferente do atributo de decisão) que será o nó da árvore, calcula-se a **entropia condicional**, $Info(T,X)$ (3.3), (COVER; THOMAS, 2006). A entropia condicional é uma medida da quantidade de informação necessária para identificar uma classe de um elemento de T dado o atributo X. Para isto, deve-se dividir T em subconjuntos $T_1, T_2, \ldots, T_n$, com base nos valores de X.

$$Info(T,X) = \sum_{i=1}^{n} \frac{\mid T_i \mid}{\mid T \mid} * Info(T_i) \tag{3.3}$$

em que $\mid T \mid$ representa a quantidade de elementos de T e $\mid T_i \mid$ representa a quantidade de elementos de T_i.

A diferença entre a informação necessária para classificar um elemento de T e a informação necessária para classificar um elemento de T dado X é o **ganho de informação** com relação ao atributo X, $Ganho(T,X)$ (3.4). O ganho de informação reduz a incerteza de um atributo e também é conhecido como **informação mútua** (COVER; THOMAS, 2006). A informação mútua representa uma medida de dependência entre duas variáveis.

$$Ganho(T,X) = Info(T) - Info(T,X). \tag{3.4}$$

Voltando ao exemplo da Tabela 3.1, para o atributo *Prev_Tempo*,

$$\begin{aligned} Info(T,Prev\_Tempo) &= \frac{5}{14} * H\left(\frac{2}{5},\frac{3}{5}\right) + \frac{4}{14} * H\left(\frac{4}{4},0\right) + \frac{5}{14} * H\left(\frac{3}{5},\frac{2}{5}\right) \\ &= 0,694 \end{aligned}$$

$$\begin{aligned} Ganho(T,Prev\_Tempo) &= Info(T) - Info(T,Prev\_Tempo) \\ &= 0,94 - 0,694 \\ &= 0,246. \end{aligned}$$

Para o atributo *Vento*, por exemplo, o ganho de informação é menor que o ganho obtido com o atributo *Prev_Tempo*, conforme os cálculos realizados da mesma forma:

$$\begin{aligned} Info(T,Vento) &= \frac{8}{14} * H\left(\frac{2}{8},\frac{6}{8}\right) + \frac{6}{14} * H\left(\frac{3}{6},\frac{3}{6}\right) \\ &= 0,892 \end{aligned}$$

$$\begin{aligned} Ganho\,(T, Vento) &= Info\,(T) - Info\,(T, Vento) \\ &= 0,94 - 0,892 \\ &= 0,048. \end{aligned}$$

Assim, o ganho é usado para ordenar os atributos e construir a árvore de decisão. Cada nó é escolhido de acordo com o maior ganho dentre os atributos que ainda não foram considerados naquele caminho a partir da raiz. A Figura 3.4 apresenta a árvore de decisão resultante para o exemplo da Tabela 3.1, considerando esse critério de seleção de atributos.

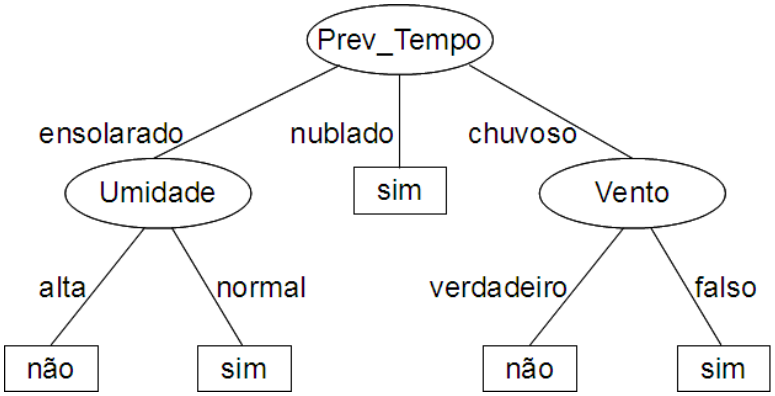


Figura 3.4: Exemplo de uma árvore de decisão utilizando informação mútua para seleção de atributos (Fonte: Quinlan (1986)).

Comparando as árvores das Figuras 3.3 e 3.4 é possível observar o quão importante é o critério de seleção de atributos. Uma árvore construída de forma aleatória, geralmente, é mais complexa que uma árvore construída a partir de algum critério mais informativo.

3.1.2 Entropia Máxima para Modelagem de Distribuição de Espécies

A crescente quantidade de dados de ocorrência de espécies disponíveis em bancos de dados na Internet (GRAHAM et al., 2004) incentiva o empenho em estudos de ferramentas para tratá-los. No entanto, enquanto dados de presença de espécies são abundantes, dados de ausência de espécies estão raramente disponíveis (PHILLIPS; ANDERSON; SCHAPIRE, 2006b). Quando tanto os dados de presença quanto os dados de ausência de espécies estão disponíveis para a estimativa dos modelos, métodos estatísticos tradicionais podem ser usados. Guisan e Zimmermmann (2000) fizeram uma revisão de vários métodos estatísticos para modelagem de distribuição de espécies. Todavia, fica evidente a necessidade do desenvolvimento de métodos de modelagem de distribuição de espécies que utilizem apenas dados de presença, como Entropia Máxima (PHILLIPS;

DUDÍK; SCHAPIRE, 2004; PHILLIPS; ANDERSON; SCHAPIRE, 2006b).

De acordo com Phillips, Anderson e Schapire (2006b), os métodos baseados em Entropia Máxima produzem modelos que não são funções de probabilidade de ocorrência das espécies, mas uma distribuição de probabilidade, isto é, o problema é estimar a distribuição com máxima entropia. Para isso, além dos dados de ocorrência das espécies, é necessário que estejam disponíveis dados de variáveis ambientais em toda a região geográfica de interesse. A principal vantagem da Entropia Máxima, como uma abordagem para modelagem de distribuição de espécies, é a sua capacidade de trabalhar apenas com pontos de presença (PHILLIPS; ANDERSON; SCHAPIRE, 2006b).

A Entropia Máxima é um método de propósito geral para fazer inferências a partir de informações incompletas (PHILLIPS; ANDERSON; SCHAPIRE, 2006b). A ideia da Entropia Máxima é estimar uma distribuição de probabilidade alvo encontrando a distribuição de probabilidade de Entropia Máxima, sujeita a um conjunto de restrições que representam a informação incompleta sobre a distribuição alvo.

Suponha que X é um conjunto finito de células de uma grade discreta, também conhecidos como pontos de *background*, representando a área geográfica de interesse e $x_1, x_2, \ldots, x_n$ é um conjunto de pontos em X, representando os registros dos pontos de ocorrência onde uma determinada espécie foi observada. Assume-se que os pontos de ocorrência foram escolhidos de forma independente de acordo com alguma distribuição de probabilidade desconhecida, p, cujo objetivo é estimá-la. Esta distribuição de probabilidade, p, representa a distribuição potencial da espécie (PHILLIPS; DUDÍK; SCHAPIRE, 2004). Portanto, a tarefa é estimar um modelo, $\hat{p}$, que se aproxime de p.

3.1.2.1 *Features* e Restrições

Na modelagem de distribuição de espécies, as *features* são as variáveis ambientais ou funções dessas variáveis. Alguns exemplos de *features* são:

- linear – são os próprios valores das variáveis;
- quadrática – quadrados dos valores das variáveis;
- produto – produtos de todas as combinações de pares de variáveis;
- limiar – função que dá uma determinada resposta se o valor da variável está acima de um limiar e outra resposta caso o valor da variável esteja abaixo do limiar;
- *hinge* – semelhante à função limiar, porém a resposta acima ou abaixo do valor

estabelecido como limiar é linear com um coeficiente positivo (*forward hinge*) ou negativo (*reverse hinge*);

- categórica – função binária que indica a pertinência de um exemplo à uma determinada classe. Se uma variável categórica possui K classes, então haverão K *features*, uma para cada classe.

A Figura 3.5 mostra graficamente as *features* citadas. No entanto, vale salientar que as *features* não estão restritas aos exemplos apresentados.

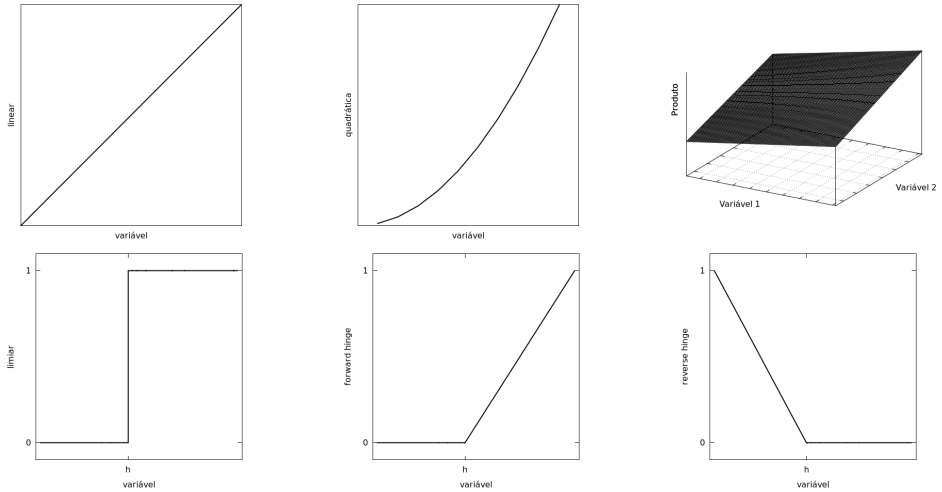


Figura 3.5: Gráficos de *features* obtidas a partir de variáveis contínuas. Da esquerda para a direita, os gráficos superiores representam as *features* linear, quadrática e produto; os gráficos inferiores representam as *features* limiar, *forward hinge* e *reverse hinge* (Fonte: autora).

Cada *feature* impõe uma restrição diferente aos modelos. As restrições indicam uma relação de proximidade entre os valores esperados para cada *feature* no modelo estimado e os valores da *feature* na amostra de dados. As *features* e as restrições representam a informação parcial disponível sobre o sistema (RATNAPARKHI, 1997). De acordo com as *features* apresentadas acima, as seguintes restrições são impostas ao modelo estimado:

- linear – média dos valores da variável;
- quadrática – variância, se usada em conjunto com a *feature* linear, e média, caso contrário;
- produto – covariância, se usada em conjunto com a *feature* linear, e média, caso contrário;
- limiar – proporção de exemplos cujos valores da variável estejam acima do limiar;

- *hinge* – média dos valores acima (*forward hinge*) ou abaixo (*reverse hinge*) do limiar;
- categórica – proporção de exemplos pertencentes à classe.

Suponha-se que os dados tenham *m features*, $f_1, f_2, \ldots, f_m$, em que f_j, $1 \leq j \leq m$, é uma função $f_j : X \to \mathbb{R}$. O valor esperado para cada *feature* no modelo p é representado por $p[f_j]$, conforme a Equação (3.5).

$$p[f_j] = \sum_{x \in X} p(x) * f_j(x). \tag{3.5}$$

em que $p(x)$ representa a distribuição de probabilidade sobre X.

Supondo que apenas um conjunto de *features* lineares seja considerado, como a tarefa é estimar um modelo, $\hat{p}$, que se aproxime de p, as restrições impostas pelas *features* seriam:

$$\hat{p}[f_j] = \tilde{p}[f_j], \quad \forall j, 1 \leq j \leq m \tag{3.6}$$

em que $\hat{p}[f_j]$ é o valor esperado de f_j no modelo $\hat{p}$ e $\tilde{p}[f_j]$ é a média empírica de f_j, conforme a Equação (3.7), isto é, a distribuição uniforme de f_j na amostra de dados.

$$\tilde{p}[f_j] = \frac{1}{n} \sum_{i=1}^{n} f_j(x_i). \tag{3.7}$$

Todos os modelos que satisfazem as restrições especificadas são considerados consistentes. Assim, uma vez estabelecidas as restrições, haverá um conjunto de possíveis modelos. Pelo princípio da Entropia Máxima, o melhor modelo é aquele que satisfaz todas as restrições especificadas e cuja entropia é máxima.

Em (PIETRA; PIETRA; LAFFERTY, 1997) pode ser encontrada a prova de que a distribuição de probabilidade com máxima entropia é única. De acordo com Penfield-Jr. (2003), quando o problema é simples, isto é, até três amostras e uma restrição, é possível calcular a entropia analiticamente. No entanto, à medida que a quantidade de amostras e de restrições aumenta, é necessário inserir um procedimento geral para procurar a distribuição com Entropia Máxima. Este é o caso da modelagem de distribuição de espécies. Para resolver esta questão, utilizam-se Multiplicadores de Lagrange (PENFIELD-JR., 2003).

3.1.2.2 Distribuição Paramétrica

O princípio da Entropia Máxima pode ser visto como um problema de otimização restrita, isto é, maximizar uma função satisfazendo algumas restrições. Segundo Berger,

Pietra e Pietra (1996), inserindo um parâmetro λ_j (Multiplicador de Lagrange) para cada *feature* f_j é possível caracterizar $\hat{p}$ de forma alternativa, considerando todas as distribuições da forma

$$p_\lambda(x) = \frac{e^{\sum_{j=1}^{m} \lambda_j f_j(x)}}{Z_\lambda} \tag{3.8}$$

em que $\lambda \in \mathbb{R}^m$ e Z_λ, Equação (3.9), é uma constante de normalização para garantir que $\sum_{i=1}^{n} p_\lambda(x_i) = 1$.

$$Z_\lambda = \sum_{i=1}^{n} e^{\sum_{j=1}^{m} \lambda_j f_j(x_i)}. \tag{3.9}$$

Cada parâmetro λ_j pode ser visto como um "peso" para a *feature* f_j (RATNAPARKHI, 1997). A distribuição (3.8) é conhecida como distribuição de *Gibbs* (PIETRA; PIETRA; LAFFERTY, 1997). Assim, com esta forma de caracterizar $\hat{p}$, estimar a distribuição de probabilidade com Entropia Máxima é o mesmo que estimar a distribuição de *Gibbs* com máxima verossimilhança (probabilidade) dos exemplos da amostra ou minimizar o *log* negativo normalizado da probabilidade (3.10) dos exemplos da amostra (DUDÍK; PHILLIPS; SCHAPIRE, 2004), também conhecido como *log loss*, $L_{\tilde{p}}(\lambda)$ (PIETRA; PIETRA; LAFFERTY, 1997).

$$L_{\tilde{p}}(\lambda) = -\frac{1}{n} \sum_{i=1}^{n} \ln\ p_\lambda(x_i) = -\tilde{p}\left[\ln\ p_\lambda(x)\right]. \tag{3.10}$$

As restrições impostas pelas *features* dificilmente serão satisfeitas, pois os valores estimados pelo modelo geralmente se aproximam dos valores esperados, mas não são iguais. Por isso, para evitar o superajustamento do modelo (*overfitting*), é possível suavizar as restrições, permitindo que os valores estimados pelo modelo apenas se aproximem dos valores esperados. Assim, supondo que somente a *feature* linear esteja sendo utilizada, ao invés de impor restrições do tipo (3.6), as restrições passam a ser

$$abs(\hat{p}[f_j] - \tilde{p}[f_j]) \leq \beta_j, \quad \forall j, 1 \leq j \leq m \tag{3.11}$$

em que cada β_j é uma constante que indica o quão próximo o valor esperado deve estar do valor estimado e $abs(\cdot)$ é o valor absoluto do seu parâmetro. Isto é chamado de regularização e modifica (3.10) para

$$L^{\beta}_{\tilde{p}}(\lambda) = L_{\tilde{p}}(\lambda) + \sum_{j=1}^{m} \beta_j\ abs(\lambda_j). \tag{3.12}$$

Chen e Rosenfeld (2000) apresentam uma análise de várias técnicas de suavização de restrições para modelos de Entropia Máxima. Para maiores detalhes sobre a regu-

larização apresentada acima, recomenda-se a leitura de (DUDÍK; PHILLIPS; SCHAPIRE, 2004).

3.1.2.3 Estimativa dos Parâmetros

Existem vários métodos que podem ser usados para estimar os parâmetros de uma distribuição de probabilidade. Uma forma muito comum de estimar os parâmetros de um modelo usando o gradiente de uma função é através do método de subida de encosta (RUSSELL; NORVIG, 2004). O gradiente de uma função é um vetor que indica a direção de maior crescimento de seus valores. Porém, o método de subida de encosta não possui uma taxa de convergência global muito boa, pois ele considera várias vezes as mesmas direções.

Os métodos de gradiente conjugado usam uma combinação linear da direção determinada pelo método de subida de encosta e da direção da busca anterior, ou seja, cada direção de busca é utilizada somente uma vez (MALOUF, 2002). Além desses métodos, também é possível utilizar métodos descendentes para estimar o modelo, como métodos de Newton e quasi-Newton (MALOUF, 2002; SALAKHUTDINOV; ROWEIS; GHAHRAMANI, 2003).

Darroch e Ratcliff (1972) desenvolveram um método geral de escalonamento iterativo para estimar os parâmetros de um modelo, conhecido como Escalonamento Iterativo Generalizado (GIS – *Generalized Iterative Scaling*). O algoritmo de escalonamento iterativo (IIS – *Improved Iterative Scaling*) disponível em (BERGER; PIETRA; PIETRA, 1996) foi adaptado de (DARROCH; RATCLIFF, 1972) para aumentar a velocidade de convergência e evitar o uso de uma *feature* de correção. Em (MALOUF, 2002) pode ser encontrada uma comparação de seis métodos de estimativa de parâmetros de uma distribuição de probabilidade com Entropia Máxima.

Esses métodos atualizam os pesos de todas as *features* em cada iteração, o que não é prático quando a quantidade de *features* é muito grande (DUDÍK; PHILLIPS; SCHAPIRE, 2004). Por isso, Phillips, Dudík e Schapire (2004) desenvolveram um algoritmo de atualização sequencial de pesos, seguindo a linha de algoritmos estudados por Collins, Schapire e Singer (2002). Nesse tipo de algoritmo, apenas o peso de uma *feature* é atualizado a cada iteração.

3.1.3 Aplicações de Entropia Máxima

As distribuições de probabilidade com Entropia Máxima pertencem a uma família de distribuições exponenciais em uma combinação linear de *features* (PHILLIPS; DUDÍK;

SCHAPIRE, 2004). Por isso, a capacidade de processamento dos computadores não permitia a aplicação do princípio da Entropia Máxima a problemas do mundo real até alguns anos atrás. No entanto, com a evolução tecnológica, a aplicação deste princípio a problemas práticos que envolvem análises estatísticas vem crescendo.

Em (BERGER; PIETRA; PIETRA, 1996) é apresentado um método de modelagem estatística para o processamento de linguagem natural. O trabalho apresenta uma descrição sobre como implementar de forma eficiente uma abordagem para estimar automaticamente modelos com Entropia Máxima. Os métodos estatísticos para o processamento de linguagem natural são muito usados em aplicações relacionadas ao reconhecimento de fala. Ratnaparkhi (1997) descreve o procedimento GIS e apresenta uma explicação simples para o princípio da Entropia Máxima aplicado ao processamento de linguagem natural. Em (GUPTA; FRIEDLANDER; GRAY, 2000) o princípio da Entropia Máxima foi utilizado para classificação supervisionada de palavras a partir de sinais da fala.

Segundo Chen, Harper e Huang (2006), a estimativa de unidades de sentenças, em tarefas de reconhecimento de fala, pode ser melhorada com a inclusão de dados visuais. Isto ocorre porque os seres humanos tendem a usar gestos, olhares e postura corporal como auxílio na comunicação. Desta forma, Chen, Harper e Huang (2006) propuseram uma abordagem baseada em Entropia Máxima para incorporar dados gestuais na estimativa de unidades de sentença. Ao comparar com outras técnicas, eles conseguiram mostrar que a abordagem proposta reduziu significativamente a taxa de erro.

O processamento e a mineração automática de informações a partir de textos é uma área que causa bastante interesse devido ao grande volume de documentos textuais disponíveis na Internet. Nigam, Lafferty e McCallum (1999) propuseram o uso do princípio da Entropia Máxima para classificar textos, um problema importante na tarefa de busca. Além disso, McCallum, Freitag e Pereira (2000) desenvolveram um método que combina o princípio da Entropia Máxima com modelos escondidos de Markov (HMM – *Hidden Markov Models*) para extrair informações de textos. Eles conseguiram mostrar que o método híbrido é melhor que usar os métodos separadamente para a tarefa de segmentação de textos.

Os modelos escondidos de Markov são muito utilizados para reconhecimento de padrões, principalmente em ferramentas de reconhecimento de fala. No entanto, muitos algoritmos de estimativa de parâmetros de um HMM resultam em superajustamento, ou seja, o modelo representa de forma satisfatória os exemplos de treinamento, mas falha na generalização. Em (WALDER; KOOTSOOKOS; LOVELL, 2003) é proposto

um método de estimativa de parâmetros de um HMM baseado em Entropia Máxima com o objetivo de superar o problema de superajustamento.

Assim como documentos textuais, a quantidade de imagens digitais disponíveis na Internet e em grandes bases de dados vem aumentando rapidamente. No entanto, muitas dessas imagens não possuem rótulos, o que dificulta a pesquisa dos usuários. Desta forma, Jeon e Manmatha (2004) utilizaram um método automático baseado em Entropia Máxima para rotular imagens, que se mostrou bastante promissor.

As pesquisas relacionadas à segurança em redes de computadores também se beneficiaram com o uso do princípio da Entropia Máxima. Gu, McCallum e Towsley (2005) desenvolveram uma técnica de detecção de anomalias em redes. Eles utilizaram uma distribuição de referência do tráfego da rede, estimada pelo princípio da Entropia Máxima. Esta distribuição de referência foi usada para comparar com o tráfego e verificar se algum ataque estava ocorrendo.

Neurociência é outra área onde o princípio da Entropia Máxima vem sendo explorado. Embora determinados padrões de propagação com interações coletivas dos neurônios sejam frequentemente observados durante o processamento de informações, eles ainda são raramente preditos por modelos que consideram as propagações de forma independente (TANG et al., 2008). Assim, Shlens et al. (2006) utilizaram o princípio da Entropia Máxima para fazer predições sobre os estados de uma rede de células da retina de primatas.

Schneidman et al. (2006) também utilizaram o princípio da Entropia Máxima para predizer correlações de estados de redes neuronais quando a correlação entre pares de neurônios é fraca. A partir dos trabalhos desenvolvidos por Shlens et al. (2006) e Schneidman et al. (2006), Tang et al. (2008) estudaram a generalidade desta abordagem e concluíram que os modelos baseados em Entropia Máxima estudados podem predizer com sucesso os estados correlacionados, mas não conseguem predizer a sua evolução ao longo do tempo.

3.2 Princípio da Descrição com Comprimento Mínimo

O princípio da Descrição com Comprimento Mínimo (MDL – *Minimum Description Length*) é um método de aprendizagem indutiva baseado em **compressão de dados** desenvolvido por Rissanen (1978). A ideia geral é que qualquer regularidade nos dados seja utilizada para descrevê-los com uma quantidade de símbolos menor que a quantidade necessária para listá-los literalmente. Desta forma, quanto maior a regularidade dos dados, mais eles podem ser comprimidos e, consequentemente, maior será a apren-

dizagem (GRÜNWALD, 2005). Como a seleção de modelos é um problema clássico da inferência indutiva, o princípio MDL é um método que pode ser usado para solucioná-lo. A seleção de modelos consiste em escolher, dentre um conjunto de modelos, aquele que melhor descreve os dados.

O princípio MDL possui várias propriedades interessantes para um método de aprendizagem indutiva (GRÜNWALD, 2005; BARRON; RISSANEN; YU, 1998), dentre as quais destacam-se:

- o modelo selecionado pelo princípio MDL procura equilibrar a generalização dos dados observados com a sua complexidade, ou seja, o princípio MDL pode ser visto como uma forma do princípio da navalha de Occam;
- os procedimentos do princípio MDL protegem automaticamente o modelo contra o superajustamento (*overfitting*);
- o princípio MDL não assume que existe um modelo "verdadeiro";
- não há necessidade de conhecimento sobre algum modelo a priori, ao contrário de outros métodos estatísticos;
- o MDL pode ser interpretado como um procedimento de busca por um modelo com boa capacidade preditiva em dados desconhecidos.

As subseções seguintes procuram apresentar as ideias fundamentais para a aplicação prática do princípio MDL. Por se tratar de um assunto extenso e complexo, recomenda-se a leitura das referências bibliográficas para o aprofundamento de algum tópico de interesse particular.

3.2.1 Aprendizagem por Compressão de Dados

Qualquer método de descrição consiste em mapear uma descrição D' de forma única a um conjunto de dados D, como as linguagens de programação. Assim, pode-se dizer que qualquer programa de computador cuja saída seja D e pare após a sua execução é uma descrição de D. Por exemplo, suponha que a sequência (3.13) seja uma repetição de 3500 ocorrências da cadeia 1000.

$$100010001000 \cdots 100010001000 \tag{3.13}$$

Essa sequência poderia ser representada pelo Algoritmo 3.1, cuja descrição é bem menor do que a sequência listando todos os seus elementos. Portanto, pode-se dizer que a sequência (3.13) é altamente compressível.

Por outro lado, se a sequência representar o resultado de 3500 lançamentos de uma moeda, o algoritmo para imprimi-la terá aproximadamente o mesmo tamanho que a sequência, pois não há qualquer padrão nos resultados dos lançamentos. Isso significa que a sequência não poderá ser comprimida. As sequências que apresentam alguma regularidade, sequências regulares, podem ser comprimidas através de uma descrição menor que a sequência original, enquanto as sequências que não apresentam regularidades, sequências aleatórias, são incompressíveis (GRÜNWALD, 2005).

Algoritmo 3.1 Algoritmo para imprimir 3500 vezes a sequência 1000.

for $i = 1$ **to** $i = 3500$ **do**

  print 1000;

end for

Ao contrário da sequência (3.13), que possui uma regularidade determinística, algumas sequências possuem regularidades estatísticas, como a sequência (3.14), em que a quantidade de 1s é aproximadamente quatro vezes a quantidade de 0s. Em ambos os casos, é possível assumir que as mesmas regularidades ocorrerão em dados posteriores e as representações dessas regularidades poderão ser usadas para predizer o comportamento futuro.

$$11100111110111\cdots110111101111101 \tag{3.14}$$

A formalização do princípio MDL é fundamentada na complexidade de *Kolmogorov* de uma sequência (LI; VITÁNYI, 2008). A complexidade de *Kolmogorov* é o tamanho do menor programa que imprime uma sequência e para. Portanto, quanto mais regularidades encontradas em uma determinada sequência, menor é a sua complexidade de *Kolmogorov*. Embora a definição da complexidade de *Kolmogorov* pareça depender da linguagem de programação usada ou do método escolhido para comprimir os dados D, a escolha da representação leva ao cálculo invariante da complexidade. Esta propriedade, chamada teorema da invariância, garante que os tamanhos dos menores programas que computam D em duas descrições distintas diferem apenas por uma constante e não depende do tamanho de D, desde que as descrições sejam de propósito geral (GRÜNWALD, 2005). Uma descrição de propósito geral é qualquer descrição que possa implementar uma Máquina de Turing universal (LI; VITÁNYI, 2008). Uma Máquina de Turing universal é capaz de simular qualquer outra Máquina de Turing a partir da sua descrição.

A complexidade de *Kolmogorov* foi desenvolvida nos anos 60 por Kolmogorov (KOLMOGOROV, 1965), Solomonoff (SOLOMONOFF, 1964a; SOLOMONOFF, 1964b) e

Chaitin (CHAITIN, 1969), de forma independente. Posteriormente, as ideias relacionadas à complexidade de *Kolmogorov* conduziram as pesquisas de diversos autores que idealizaram uma versão do princípio MDL. No entanto, esta versão idealizada não pode ser aplicada na prática porque é incomputável e, para conjuntos muito pequenos de dados, o princípio MDL idealizado pode depender de detalhes arbitrários da sintaxe da linguagem de programação considerada (GRÜNWALD, 2005).

Para tornar prático o uso do princípio MDL é necessário restringir os métodos descritivos, tornando-os menos expressivos que as linguagens de programação de propósito geral. Os métodos de descrição devem ser gerais o bastante para comprimir muitas regularidades e ao mesmo tempo restritos o suficiente para que sempre seja possível computar o tamanho da menor descrição dos dados. No entanto, isto faz com que o método não seja capaz de comprimir todas as regularidades dos dados. Assim, o princípio MDL une a aprendizagem indutiva com a compressão de dados, pois as regularidades encontradas nos dados são utilizadas para comprimi-los e a menor descrição dos dados é considerada o melhor modelo.

Supondo que deseja-se selecionar um modelo de acordo com um conjunto de dados, se o modelo selecionado for extremamente simples, ele não representará bem os dados, ou seja, não descreverá bem as regularidades encontradas nos dados. Por outro lado, se o modelo selecionado for extremamente complexo, ele se ajustará muito bem aos dados, mas não será capaz de generalizá-los. Isto significa que para um novo conjunto de dados o modelo escolhido previamente estará muito distante de representá-lo. Essa característica é conhecida como superajustamento (*overfitting*). O princípio MDL evita automaticamente o superajustamento porque escolhe um modelo que não é excessivamente simples nem extremamente complexo. O modelo selecionado é relativamente simples, com erro pequeno mas necessariamente diferente de zero (GRÜNWALD, 2005).

3.2.2 Códigos Livres de Prefixo

O princípio MDL, assim como a maioria dos algoritmos de compressão de dados, utiliza apenas códigos livres de prefixo (GRÜNWALD, 2005). Embora o termo "código de prefixo" seja o mais utilizado na literatura, neste texto será adotada a expressão "código livre de prefixo", como em (LI; VITÁNYI, 1997). Os códigos livres de prefixo são aqueles em que nenhuma palavra de código é um prefixo de alguma outra palavra de código (CORMEN et al., 2002).

Um **código** de X é uma função um para um, $C : X \rightarrow B^+$, em que cada elemento de X é representado por uma sequência binária finita única de tamanho maior ou igual

a 1 e X é um conjunto finito ou contável. Assim, $C(x)$, conhecida como **palavra de código**, denota um código C que codifica x e $B = \{0,1\}^+$. Como cada elemento de X é codificado de forma única, é possível identificar x a partir de sua codificação $C(x)$, ou seja, se $C(x) = z$, então $x = C^{-1}(z)$. A função C^{-1} é a função inversa de $C(x)$ e é conhecida como **função de decodificação**.

Para melhor compreensão, suponha que $X = \{a,b,c\}$, o código C_1 é definido por $C_1(a) = 0, C_1(b) = 10, C_1(c) = 11$ e o código C_2 é definido por $C_2(a) = 0, C_2(b) = 10$, $C_2(c) = 01$. Neste caso, C_1 é um código livre de prefixo, pois nenhuma palavra de código é prefixo de outra. No entanto, C_2 não é um código livre de prefixo, pois $C_2(a)$ é um prefixo de $C_2(c)$.

Os códigos livres de prefixo permitem que a codificação seja simplesmente a concatenação das palavras de código, o que também simplifica a decodificação. Uma vez que nenhuma palavra de código é um prefixo de outra, não há ambiguidade na decodificação. De acordo com Cormen et al. (2002), uma compressão de dados ótima sempre pode ser alcançada com um código livre de prefixo. Além disso, esse tipo de código evita a inserção de símbolos extras para delimitar palavras. Portanto, é possível utilizar apenas códigos livres de prefixo sem perda de generalidade.

Segundo Grünwald (2005), todo código induz uma **função tamanho de código**, denotada por $L_C : X \to \mathbb{N}$, em que $L_C(x)$ é o tamanho da palavra de código $C(x)$, ou seja, $L_C(x)$ é o número de símbolos necessários para codificar x usando o código C. Os códigos livres de prefixo devem satisfazer a desigualdade de *Kraft* (3.15) (COVER; THOMAS, 2006).

$$\sum_{x \in X} 2^{-L_C(x)} \leq 1. \tag{3.15}$$

A desigualdade de *Kraft* é uma condição necessária e suficiente para a existência de um código livre de prefixo para um determinado conjunto de tamanhos de palavras de código (COVER; THOMAS, 2006). Assim, dado um código qualquer, pode-se dizer que ele é um código livre de prefixo se os tamanhos das palavras que o compõe satisfazem a desigualdade de *Kraft*. Da mesma forma, dado um conjunto de tamanhos de palavras de código, se eles satisfizerem a desigualdade de *Kraft*, então é possível construir um código livre de prefixo com palavras que tenham esses tamanhos (veja a prova na Seção 5.2 de (COVER; THOMAS, 2006), p. 107–109).

De acordo com Rissanen (2005), as funções tamanho de código correspondem a distribuições de probabilidade. Assim, dado um código livre de prefixo C que codifica X com função tamanho de código L_C, a distribuição de probabilidade correspondente

é expressa pela Equação (3.16).

$$P(x) = \frac{2^{-L_C(x)}}{\sum_{z \in X} 2^{-L_C(z)}} \quad \forall x \in X. \tag{3.16}$$

Por outro lado, para qualquer distribuição de probabilidade P em X, é possível encontrar um código livre de prefixo com uma função tamanho de código conforme a Equação (3.17), em que para qualquer valor real y, $\lceil y \rceil$ é o menor número inteiro maior ou igual a y. Assim, uma função tamanho de código pode ser vista como uma forma diferente de expressar uma distribuição de probabilidade. Como as distribuições de probabilidade servem como um meio de expressar as características regulares nos dados, elas são consideradas modelos para os dados (BARRON; RISSANEN; YU, 1998).

$$L_C(x) = \lceil -logP(x) \rceil . \tag{3.17}$$

Essa correspondência entre distribuições de probabilidade e funções tamanho de código significa que quanto maior a probabilidade de x de acordo com P, menor será o tamanho da palavra de código que representa x no código C correspondente a P e vice-versa (GRÜNWALD, 2005). Em outras palavras, quanto maior a probabilidade de uma saída, menor deve ser o tamanho da sua descrição e, as saídas menos frequentes devem ser representadas por descrições mais longas.

3.2.3 MDL Prático

No contexto do princípio MDL, um **modelo** é uma distribuição de probabilidade paramétrica, ou seja, uma distribuição de probabilidade que pode ser representada por um vetor de parâmetros. Um conjunto de tais modelos é chamado de **classe de modelos** (RISSANEN, 2011). A ideia principal do princípio MDL é selecionar um **modelo universal** para representar uma determinada classe de modelos. Um modelo universal é capaz de simular o comportamento de qualquer modelo na classe e não precisa necessariamente pertencer à mesma classe de modelos que representa (KONTKANEN; MYLLYMÄKI, 2005).

Considere que Θ é um conjunto de vetores de parâmetros, ou seja, Θ é o **espaço de parâmetros**, tal que $\Theta \subseteq \mathbb{R}^d$ e $\mathbb{R}^d$ é o conjunto de todos os vetores com d números reais, em que d é um inteiro positivo. Os modelos P de uma classe de modelos $\mathscr{M}$ são indexados por algum vetor de parâmetros $\theta \in \Theta$ (GRÜNWALD, 2005). Assim, a classe de modelos $\mathscr{M}$ é definida como (KONTKANEN; MYLLYMÄKI, 2005)

$$\mathscr{M} = \{P(\cdot \mid \theta) : \theta \in \Theta\}. \tag{3.18}$$

A versão mais antiga e mais simples do princípio MDL é composta por duas partes. Suponha que $\mathscr{M}^{(1)}$, $\mathscr{M}^{(2)}$,... são classes de modelos e $x^n = (x_1, x_2, \ldots, x_n)$ é uma amostra de dados. Por simplicidade, será utilizado apenas o símbolo θ para representar um modelo $P(x^n \mid \theta)$. O melhor modelo $\theta \in \mathscr{M}^{(1)} \cup \mathscr{M}^{(2)} \cup \ldots$ para representar os dados x^n é aquele que minimiza a soma $L(\theta) + L(x^n \mid \theta)$, em que $L(\theta)$ é o tamanho da descrição do modelo θ e $L(x^n \mid \theta)$ é o tamanho da descrição dos dados x^n quando codificados pelo modelo θ. Assim, quanto mais o modelo se ajustar aos dados, menor será sua descrição. No entanto, é necessário definir um método de descrição preciso para tornar esse procedimento bem definido.

Na versão composta por duas partes, $L(x^n \mid \theta) = -logP(x^n \mid \theta)$, em que $P(x^n \mid \theta)$ é a distribuição de probabilidade de x^n de acordo com θ. Este método é conhecido como código de *Shannon-Fano* (COVER; THOMAS, 2006). Entretanto, encontrar um método de descrição para $L(\theta)$ é um problema, pois o tamanho da descrição pode ser muito grande em um método e muito pequeno em outro. Desta forma, a versão composta por duas partes pode se tornar arbitrária. Uma das soluções propostas por pesquisadores da área é usar o menor programa de computador que computa $P(x^n \mid \theta)$ quando x^n é apresentado como entrada. Todavia, esta solução é incomputável (Problema da Parada).

Para tornar o princípio MDL prático em muitas situações, a solução encontrada foi associar um modelo com a classe inteira, ou seja, selecionar um modelo que seja capaz de simular o comportamento de qualquer outro em uma determinada classe. Esses modelos são conhecidos como **modelos universais**. Na solução moderna do princípio MDL, o critério utilizado para a seleção de modelos é a **complexidade estocástica** (SC – *Stochastic Complexity*) dos dados dado uma classe de modelos (RISSANEN, 1986; RISSANEN, 1996), que pode ser ligeiramente definida como o tamanho da menor descrição para os dados x^n com relação a uma classe de modelos $\mathscr{M}$.

Segundo Rissanen (2011), não é possível obter uma descrição com tamanho menor que a complexidade estocástica dos dados. Por isso, quanto menor a complexidade estocástica, melhor a classe de modelos. A complexidade estocástica (KONTKANEN, 2009) pode ser definida formalmente como

$$SC(x^n \mid \mathscr{M}) = -\ log\ P_{NML}(x^n \mid \mathscr{M}). \tag{3.19}$$

Na Equação (3.19), $P_{NML}(x^n \mid \mathscr{M})$ é a distribuição de **máxima verossimilhança normalizada** (NML – *Normalized Maximum Likelihood*[1]). A distribuição NML é

[1] (SHTARKOV, 1987) *apud* (KONTKANEN, 2009); (GRÜNWALD, 2005).

definida por

$$P_{NML}(x^n \mid \mathcal{M}) = \frac{P(x^n \mid \hat{\theta}(x^n))}{\mathcal{R}_n(\mathcal{M})} \tag{3.20}$$

em que $\hat{\theta}(x^n)$ é o estimador de máxima verossimilhança dos dados x^n e $\mathcal{R}_n(\mathcal{M})$ é a **complexidade paramétrica** ou *regret* (GRÜNWALD, 2005) definida como

$$\mathcal{R}_n(\mathcal{M}) = \sum_{x^n \in X^n} P(x^n \mid \hat{\theta}(x^n)). \tag{3.21}$$

Assim, substituir a complexidade paramétrica (3.21) na equação da distribuição NML (3.20) resulta em

$$P_{NML}(x^n \mid \mathcal{M}) = \frac{P(x^n \mid \hat{\theta}(x^n))}{\sum_{x^n \in X^n} P(x^n \mid \hat{\theta}(x^n))}. \tag{3.22}$$

Similarmente, substituindo a distribuição NML resultante (3.22) na equação da complexidade estocástica (3.19) obtém-se

$$SC(x^n \mid \mathcal{M}) = -\ log\ P(x^n \mid \hat{\theta}(x^n))\ +\ log \sum_{x^n \in X^n} P(x^n \mid \hat{\theta}(x^n)). \tag{3.23}$$

O estimador de máxima verossimilhança $\hat{\theta}$ é uma função que mapeia x^n em $\theta \in \Theta$ de forma a maximizar a probabilidade $P(x^n \mid \theta)$, ou seja, o estimador pode ser visto como um algoritmo de aprendizagem que recebe uma amostra de dados x^n como entrada e retorna um vetor de parâmetros θ como saída (GRÜNWALD, 2005).

A complexidade paramétrica de uma classe indica a sua capacidade de se ajustar a dados aleatórios, ou seja, é uma medida de "riqueza" da classe de modelos (GRÜNWALD, 2005). A complexidade paramétrica pode ser interpretada como o logaritmo do número de distribuições essencialmente distintas dentro da classe de modelos. Assim, se duas distribuições atribuem altas probabilidades para a mesma amostra de dados, elas não devem ser contadas como diferentes, pois não contribuem para a complexidade da classe (KONTKANEN, 2009).

A versão moderna do princípio MDL associa classes de modelos paramétricos com uma complexidade inerente que não depende de qualquer método de descrição particular para os modelos. Desta forma, como as distribuições não são mais codificadas explicitamente, não existem arbitrariedades nessa codificação (GRÜNWALD, 2005).

3.2.4 Aplicações do Princípio MDL

Mehta, Rissanen e Agrawal (1995) desenvolveram um algoritmo baseado no princípio MDL para podas em árvores de decisão. Os algoritmos de poda são usados para reduzir o tamanho das árvores através da remoção de caminhos com pouco poder de classificação. O objetivo da poda é reduzir a complexidade do classificador e o superajustamento. O algoritmo baseado no princípio MDL foi comparado com outros três algoritmos de poda e os experimentos foram realizados com sete conjuntos de dados diferentes. Os resultados mostraram que o algoritmo de poda baseado no princípio MDL alcançou boas taxas de acuidade, árvores pequenas e rápido tempo de execução.

As linguagens naturais são exemplos bastante comuns de aplicações de técnicas de compressão de dados, pois o uso de programas de compactação tornou-se frequente com o aumento do volume de dados disponíveis. O princípio MDL já foi aplicado por Grünwald (1996) na inferência de gramáticas, em que o algoritmo de aprendizagem recebia como entrada um conjunto de sentenças de uma determinada linguagem e apresentava como saída um conjunto de regras gramaticais subjacentes à linguagem.

Em processamento de imagens, *wavelet thresholding* é um método que pode ser usado na remoção de ruídos e na compressão de imagens. Hansen e Yu (2000) associaram esse método com a seleção de modelos e propuseram um critério baseado no princípio MDL para estimar os coeficientes *wavelet*. O método proposto foi comparado com outros métodos clássicos e o princípio MDL mostrou-se superior aos demais.

Myung et al. (2001) utilizaram o princípio MDL para recuperar o modelo gerador de dados em duas aplicações de processamento cognitivo: integração da informação e categorização. Além disso, eles compararam o princípio MDL com outros dois métodos de seleção de modelos, o AIC (*Akaike Information Criterion*) e o BIC (*Bayesian Information Criterion*). Em média, o princípio MDL se mostrou melhor que os demais, especialmente nos casos em que a diferença de complexidade dos modelos era maior.

Geralmente, as técnicas de agrupamento e classificação são usadas separadamente em estudos genéticos. Os agrupamentos de genes são muito úteis para codificar características similares das espécies e a classificação de amostras é frequentemente usada para classificar as expressões gênicas. Visto que a separação dessas tarefas pode esconder estruturas importantes dos dados, Jörnsten e Yu (2003) desenvolveram um algoritmo baseado no princípio MDL para, simultaneamente, agrupar genes e selecionar um subconjunto de grupos de genes para classificação das amostras. O algoritmo foi testado em três conjuntos de dados e os resultados obtidos se mostraram promissores

nessa área de pesquisa.

A última instanciação da complexidade estocástica, baseada na distribuição NML, tem mostrado possuir muitas propriedades teóricas importantes. No entanto, as aplicações dessa versão do princípio MDL têm sido bastante raras devido a problemas de complexidade computacional (KONTKANEN, 2009). Mesmo com as dificuldades relacionadas à complexidade computacional, o princípio MDL tem sido explorado em áreas como modelagem cognitiva[2] (SU; MYUNG; PITT, 2005), psicologia (LEE; NAVARRO, 2005; NAVARRO; LEE, 2005) e construção de histogramas (WANG; SEVCIK, 2008).

3.3 Epítome

A meta deste capítulo foi a apresentação dos aspectos teóricos de dois princípios fundamentados em Teoria da Informação e que foram foco da pesquisa, Entropia Máxima e Descrição com Comprimento Mínimo. Estes princípios foram abordados visando suas aplicações na modelagem de distribuição de espécies.

A entropia pode ser vista como uma medida de informação associada às ocorrências de um evento. Quando as diversas configurações possíveis têm a mesma probabilidade de ocorrência, então a entropia é máxima. O problema consiste em selecionar a distribuição de probabilidade que satisfaz um conjunto de restrições impostas pelas *features*, que são as características do sistema. Como existem várias distribuições que satisfazem as restrições, a solução é encontrar a distribuição de probabilidade com Entropia Máxima.

No caso da modelagem de distribuição de espécies, as variáveis ambientais representam as características do sistema, ou seja, as *features*. Existem vários métodos de estimativa dos parâmetros da distribuição de probabilidade com Entropia Máxima, como escalonamento iterativo. Além disso, o princípio da Entropia Máxima vem sendo utilizado com sucesso em diversas aplicações, como processamento de linguagem natural, mineração de textos, segurança em redes de computadores, dentre outras.

No caso do princípio da Descrição com Comprimento Mínimo, a base é a compressão de dados. As regularidades encontradas nos dados são utilizadas para descrevê-los com uma quantidade de símbolos menor que a utilizada para listá-los literalmente. Assim, a compressão de dados pode ser equacionada com a aprendizagem. Os códigos livres de prefixo, ou seja, aqueles em que nenhuma palavra de código é um prefixo

[2] A modelagem cognitiva consiste em selecionar modelos de processos mentais, como identificação de objetos, integração de informações e retenção de informações sobre eventos aprendidos ao longo do tempo.

de outra palavra de código, são os mais adequados para algoritmos de compressão de dados.

Como as distribuições de probabilidade podem ser associadas aos tamanhos das palavras de código, a estratégia do princípio MDL é selecionar o modelo universal com a menor descrição possível. Um modelo universal é capaz de simular o comportamento de qualquer outro modelo de uma classe. Para não tornar o problema incomputável, é necessário restringir a busca a uma determinada classe de modelos. Desta forma, é possível aplicar o princípio MDL em diversas áreas, como processamento de imagens e modelagem cognitiva.

4 Tecnologia Adaptativa

A tecnologia adaptativa é a aplicação prática da adaptatividade na solução de problemas dinâmicos. Entende-se por adaptatividade, a capacidade de automodificação de um sistema (NETO, 2009). A computação é uma ciência essencialmente abstrata, cujos fundamentos são modelos matemáticos capazes de descrever todos os seus processos. Existem vários modelos equivalentes que representam a capacidade computacional, ou seja, a computabilidade. No entanto, para cada paradigma de programação existe um modelo mais adequado para representá-lo (SILVA; MELO, 2006). De maneira similar, existem vários formalismos capazes de representar as diferentes classes de linguagens[1], cada um mais apropriado para uma determinada classe (RAMOS; NETO; VEGA, 2009).

Qualquer formalismo computacional pode ser usado para implementar a adaptatividade. Assim, o objetivo deste capítulo é apresentar os principais conceitos relacionados à tecnologia adaptativa. Na Seção 4.1 são apresentados os conceitos fundamentais referentes aos dispositivos adaptativos. A Seção 4.2 cita alguns dispositivos adaptativos e seus respectivos formalismos subjacentes. Algumas aplicações são apresentadas na Seção 4.3. Um exemplo de aplicação da tecnologia adaptativa é mostrado na Seção 4.4. Por fim, a Seção 4.5 apresenta um resumo do capítulo.

4.1 Dispositivos Adaptativos – Conceitos Fundamentais

Os dispositivos tradicionais da teoria da computação possuem a característica de não poderem alterar a sua forma de proceder durante a sua execução, ou seja, as variações de comportamento são definidas durante a fase de configuração e permanecem estáticas. Contudo, os dispositivos adaptativos são capazes de se automodificarem durante a sua execução. Isto permite que o dispositivo se modifique conforme as suas necessidades de funcionamento.

Um dispositivo adaptativo é qualquer abstração, que representa um processo ou um fenômeno, em que um conjunto de regras define seu comportamento e sua ope-

[1] As linguagens são classificadas de acordo com sua complexidade relativa. Esta classificação é conhecida como **Hierarquia de Chomsky** (RAMOS; NETO; VEGA, 2009).

ração é caracterizada pela mudança sucessiva de configurações (NETO, 2009). Além disso, um dispositivo adaptativo deve ter como propriedade principal a adaptatividade, capacidade de modificar seu comportamento em resposta aos estímulos de entrada e seu histórico de operação, sem a interferência de agentes externos (NETO, 2007).

Por modificarem a sua própria estrutura durante seu funcionamento, os dispositivos adaptativos possuem a capacidade de aprendizagem (NETO, 2004). A aprendizagem é uma importante característica de dispositivos considerados inteligentes. Além disso, os dispositivos adaptativos são computacionalmente equivalentes às Máquinas de Turing (ROCHA; NETO, 2000).

Os dispositivos adaptativos são formados por duas camadas: um dispositivo não adaptativo, chamado de dispositivo subjacente, cujas regras de comportamento são estáticas; e um mecanismo adaptativo, que possibilita a automodificação do dispositivo quando associado a um dispositivo subjacente (NETO, 2002). O dispositivo subjacente pode ser qualquer formalismo descrito por um conjunto finito de regras, por exemplo: autômatos finitos, autômatos de pilha estruturados, tabelas de decisão, entre outros.

Segundo Neto (2002), um dispositivo adaptativo pode ser formalmente definido como um par $DA = (DS_k, MA)$, em que DS_k é um dispositivo subjacente, MA é um mecanismo adaptativo e $k \geq 0$ é um passo de adaptação. O dispositivo adaptativo DA segue o comportamento de seu dispositivo subjacente DS_k até que alguma função adaptativa não nula comece a ser executada. Nesse momento, o passo de adaptação k é incrementado e o dispositivo subjacente evolui para DS_{k+1} através da edição do conjunto de regras de DS_k. Em seguida, o dispositivo adaptativo DA segue o comportamento do dispositivo subjacente DS_{k+1} até que outra função adaptativa provoque outro passo de adaptação. Esse procedimento ocorre até que toda a cadeia de entrada tenha sido processada.

O dispositivo subjacente DS_k é uma 6-tupla: $DS_k = (C, R_k, \Sigma, c_k, A, \Gamma)$. Cada elemento do DS_k é definido da seguinte forma:

- C é o conjunto de configurações possíveis, em que uma configuração é uma representação de todos os elementos que definem o comportamento de um dispositivo;
- $R_k \subseteq C \times (\Sigma \cup \{\varepsilon\}) \times C \times (\Gamma \cup \{\varepsilon\})$ é o conjunto de regras que descreve o comportamento do dispositivo subjacente, em que ε representa um elemento nulo, isto é, um estímulo vazio;
- Σ é um conjunto finito com todos os símbolos válidos como estímulos de entrada;

- $c_k \in C$ é a configuração inicial do dispositivo no passo k;
- $A \subseteq C$ é o conjunto de configurações de aceitação;
- Γ é um conjunto finito de símbolos de saída.

A aplicação de uma regra $r = (c_i, s, c_{i+1}, z) \in R_k$ é denotada por $c_i \Rightarrow^s c_{i+1}$, em que c_i é a configuração corrente, $s \in (\Sigma \cup \{\varepsilon\})$ é o estímulo de entrada, c_{i+1} é a próxima configuração e $z \in (\Gamma \cup \{\varepsilon\})$ é o símbolo de saída. Uma configuração de aceitação é uma configuração que indica a aceitação da cadeia de entrada após seu processamento completo.

O mecanismo adaptativo *MA* é a combinação do dispositivo subjacente com funções que permitem alterar o próprio conjunto de regras do dispositivo. Essas funções são conhecidas como funções adaptativas. As funções adaptativas são unidas às regras do dispositivo subjacente para que possam ser executadas quando for necessário modificar o conjunto de regras (NETO, 2009).

As regras do dispositivo adaptativo *DA* são as regras do dispositivo subjacente com duas funções adaptativas associadas, uma para ser executada antes da mudança de configuração, *AA*, e outra para ser executada após a mudança de configuração, *AP*. As funções adaptativas podem ser nulas. Nesse caso, a execução da regra do dispositivo adaptativo é equivalente à execução da regra do dispositivo subjacente.

Sempre que houver alguma função adaptativa não nula associada a uma regra do dispositivo subjacente, o mecanismo adaptativo, $MA \subseteq AA \times R_k \times AP$, é aplicado com a regra. Enquanto as funções adaptativas não são executadas, o dispositivo adaptativo funciona conforme seu dispositivo subjacente. Se a execução de uma função adaptativa provocar a remoção da regra à qual ela está acoplada, a aplicação da regra deve ser descontinuada e um novo conjunto de regras deve ser selecionado para aplicação a partir do conjunto resultante. Os dispositivos só podem realizar mudanças em seu comportamento caso isto tenha sido previsto na estruturação inicial, através da implementação de funções adaptativas.

Toda vez que uma função adaptativa for executada, um conjunto de ações adaptativas elementares será aplicado ao conjunto de regras do dispositivo adaptativo. Existem três tipos de ações adaptativas elementares, com as quais devem ser implementadas as funções adaptativas e que são executadas na mesma ordem em que são apresentadas a seguir:

- **Ações de consulta**, denotada pelo prefixo "?": procuram por regras que casem com um padrão específico sem alterar o conjunto de regras;

- **Ações de remoção**, denotada pelo prefixo "$-$": removem do conjunto as regras que casam com um determinado padrão;

- **Ações de inserção**, denotada pelo prefixo "$+$": inserem no conjunto regras com um padrão específico.

As funções adaptativas devem ter um nome simbólico, para que este nome possa ser usado como um identificador de qual função será executada. Cada função também pode ter um conjunto de nomes simbólicos para representar os parâmetros, um conjunto de nomes simbólicos para as variáveis e um conjunto de nomes simbólicos para os geradores. Os parâmetros são instanciados pelos valores passados como argumentos na chamada da função; as variáveis armazenam os resultados das ações elementares de consulta; e os geradores representam os novos elementos do dispositivo. Uma vez preenchidos com algum valor, os parâmetros, as variáveis e os geradores não sofrem alterações durante a execução da função adaptativa. Além disso, uma função adaptativa possui uma codificação de todas as mudanças que devem ocorrer no conjunto de regras quando ela for chamada. Esta codificação forma o corpo da função adaptativa (NETO, 2002).

4.2 Dispositivos Adaptativos Baseados em Regras

Diversos formalismos clássicos da computação já foram utilizados para descrever os dispositivos subjacentes de dispositivos adaptativos. O formalismo mais utilizado, talvez por ter sido o primeiro a ser usado na descrição de dispositivos subjacentes, é o autômato de pilha estruturado. Este é o dispositivo subjacente para o autômato adaptativo (NETO, 1993).

Um autômato de pilha estruturado pode ser visto como um conjunto de submáquinas de estados finitos. Estas submáquinas executam transições internas para consumir símbolos de entrada e mudar de estado dentro da submáquina. Para chamar ou retornar de uma submáquina são executadas transições externas (NETO, 1994). Os autômatos adaptativos podem ser empregados como ferramentas com grande potencial para representar soluções elegantes, eficientes, compactas e práticas para muitos problemas complexos (NETO, 2001).

Os formalismos baseados na aplicação de regras de substituição são usados para descrever os dispositivos subjacentes das gramáticas adaptativas (NETO, 2007). Como as gramáticas são formalismos generativos, em uma gramática adaptativa, o conjunto de regras pode ser alterado à medida que as sentenças são derivadas (IWAI; NETO,

2000). Os conceitos fundamentais da adaptatividade são os mesmos para qualquer formalismo utilizado na descrição do dispositivo subjacente. A diferença está na forma de representação das regras que regem o comportamento de cada dispositivo.

As Redes de Markov são formalismos comumente utilizados para representar processos estocásticos. Esse tipo de formalismo é semelhante a um autômato finito, com estados e transições. No entanto, a função de transição depende das relações de probabilidade entre os estados. Uma Máquina de Markov é um método de produção de sentenças, de uma determinada linguagem, que utiliza Redes de Markov. As Máquinas de Markov são os dispositivos subjacentes das Máquinas de Markov Adaptativas (BASSETO, 2000). Como as Máquinas de Markov de primeira ordem geram linguagens regulares, as Máquinas de Markov Adaptativas foram criadas para gerar linguagens sensíveis ao contexto sem a complexidade das máquinas de ordem superior.

Os *statecharts* são descrições abstratas para representar visualmente as especificações que envolvem o desenvolvimento de um software. Os *statecharts* podem ser considerados uma forma mais abrangente de máquinas de estado finito, em que as transições entre os estados estão associadas a ocorrências de eventos, condições específicas de disparo e ações a serem executadas quando uma transição é disparada. Os *statecharts* são utilizados como dispositivos subjacentes dos *statecharts* adaptativos (ALMEIDA-JUNIOR, 1995). Os *statecharts* adaptativos foram criados para conceder a capacidade de alteração das regras de especificação do sistema em resposta aos estímulos de entrada.

As árvores de decisão (QUINLAN, 1986) são representações simples da informação cujo objetivo é classificar um número finito de classes. A ideia é construir uma árvore a partir de um conjunto de registros com seus respectivos valores de classificação. As árvores de decisão são os dispositivos subjacentes das árvores de decisão adaptativas (PISTORI; NETO, 2002). Estas permitem que dispositivos baseados em árvores de decisão alterem sua estrutura continuamente, facilitando a aprendizagem incremental.

Tabelas de decisão são representações, em forma de tabela, de condições necessárias para que um conjunto de ações seja executado. Estes são os dispositivos subjacentes das tabelas de decisão adaptativas (NETO, 2002). As tabelas de decisão possuem um conjunto de linhas que representam as condições, um conjunto de linhas que representam as ações e as colunas representam as regras, indicando quais condições são necessárias para determinadas ações. No caso das tabelas de decisão adaptativas, além dos elementos das tabelas convencionais, as tabelas possuem um conjunto de linhas para codificar as ações adaptativas a serem executadas antes e depois da aplicação de cada regra.

Esta seção apresentou apenas os formalismos computacionais usados com frequência como dispositivos subjacentes na implementação da adaptatividade. No entanto, vale salientar que a adaptatividade pode ser incorporada a qualquer formalismo com um conjunto bem definido de regras.

4.3 Aplicações da Tecnologia Adaptativa

A tecnologia adaptativa vem sendo aplicada com sucesso em diversas áreas de pesquisa. Pelegrini e Neto (2007) propuseram o uso da adaptatividade na construção de um sistema de encriptação rápido e seguro, proporcionando maior confiabilidade na transmissão de dados. A adaptatividade foi utilizada para esconder e modificar dinamicamente parte da lógica do algoritmo de codificação/decodificação. Para modelar o sistema proposto, Pelegrini e Neto (2007) utilizaram um transdutor de estados finitos adaptativos, cujo dispositivo subjacente é um formalismo similar aos autômatos finitos, chamado transdutor de estados finitos.

A robótica é uma área de pesquisa que envolve a automação de sistemas mecânicos. Assim, são necessários estudos relacionados a diversas capacidades idealizadas para um robô, como locomoção e detecção de objetos. Hirakawa, Saraiva e Cugnasca (2007) investigaram o uso de autômatos adaptativos em processos robóticos. Como exemplo prático, Hirakawa, Saraiva e Cugnasca (2007) apresentam o reconhecimento de formas e o mapeamento de obstáculos para a navegação de robôs.

Outra área de pesquisa em que a tecnologia adaptativa tem sido bastante aplicada é o processamento de linguagem natural. Os autômatos adaptativos podem ser aplicados em diferentes etapas do processamento computacional de linguagens naturais, por exemplo análise morfológica, análise sintática e tradução texto-voz (ROCHA, 2007). No entanto, vale ressaltar que outros formalismos adaptativos também podem ser utilizados no processamento de linguagens naturais.

Após a utilização de Redes de Markov Adaptativas no desenvolvimento de uma ferramenta para composição automática de música (BASSETO, 2000), a tecnologia adaptativa foi utilizada para auxiliar músicos na edição de músicas. Bellini e Tavella (2008) utilizaram autômatos adaptativos para desenvolver um algoritmo de conversão de partituras em tablaturas. Embora o algoritmo proposto não tenha apresentado resultados superiores aos dos algoritmos convencionais, a adaptatividade se mostrou uma estratégia interessante para o problema, com perspectivas de melhoria em pontos não explorados.

Alguns estudos sobre a contribuição da adaptatividade na área de modelagem de

distribuição de espécies já foram realizados. Bravo et al. (2007) implementaram um algoritmo de modelagem adaptativo a partir do GARP, o *AdaptGARP*, utilizando uma tabela de decisão adaptativa. Stange et al. (2009) realizaram um estudo comparativo entre o *AdaptGARP* e o GARP utilizando várias espécies do gênero *Peponapis* e *Cucurbita*.

Rodrigues, Rodrigues e Rocha (2009) propuseram um método de pós-análise baseado em um dispositivo adaptativo. Geralmente, a análise dos modelos de distribuição de espécies é realizada com base em medidas calculadas a partir da matriz de confusão, conforme explicado na Seção 2.3 do Capítulo 2. Métodos automatizados que colaborem na análise dos resultados são muito importantes no auxílio à tomada de decisões. O método proposto utiliza uma tabela de decisão adaptativa.

Na Seção 5.2 do Capítulo 5 será apresentado em detalhes o desenvolvimento de um algoritmo adaptativo baseado no princípio da Entropia Máxima. Este algoritmo é uma das contribuições desta tese e foi aplicado à modelagem de distribuição de espécies (RODRIGUES et al., 2010a).

4.4 Exemplo de Aplicação da Tecnologia Adaptativa

Conforme mostrado na Seção 4.3, a tecnologia adaptativa pode ser aplicada em diversas áreas. Esta seção apresenta um exemplo de aplicação de autômatos adaptativos para o problema do emparelhamento de cadeias. Este problema foi estudado e um autômato adaptativo foi implementado no início da pesquisa. O trabalho foi muito útil para a compreensão dos conceitos fundamentais da adaptatividade e resultou em uma publicação com os resultados do estudo (RODRIGUES; RODRIGUES; ROCHA, 2008b).

4.4.1 O Problema do Emparelhamento de Cadeias

O problema do emparelhamento de cadeias, também conhecido como casamento de padrões, ocorre quando uma sequência de caracteres é procurada em um texto. A tarefa é encontrar todas as ocorrências de uma determinada cadeia, sequência de caracteres, em um dado texto. Esta seção foi escrita com base na formalização do problema apresentada em (CORMEN et al., 2002).

Suponha que C e T são duas sequências de caracteres sobre o alfabeto finito Σ, em que C representa a cadeia procurada no texto T. Suponha também que m é o comprimento de C e n é o comprimento de T, com $m \leq n$. A cadeia C é encontrada no texto T a partir da posição $s+1$, se $0 \leq s \leq n-m$ e $T[s+1..s+m] = C[1..m]$.

De forma equivalente, diz-se que a cadeia C ocorre com deslocamento s em T. Nesse caso, s é dito um deslocamento válido. Caso contrário, s é dito um deslocamento inválido. Quando uma sequência de caracteres é procurada em um texto, o objetivo é encontrar todos os deslocamentos válidos desta sequência no texto. A Figura 4.1 mostra um exemplo de um deslocamento válido de uma cadeia em um texto, em que o alfabeto é $\Sigma = \{a, b, c\}$.

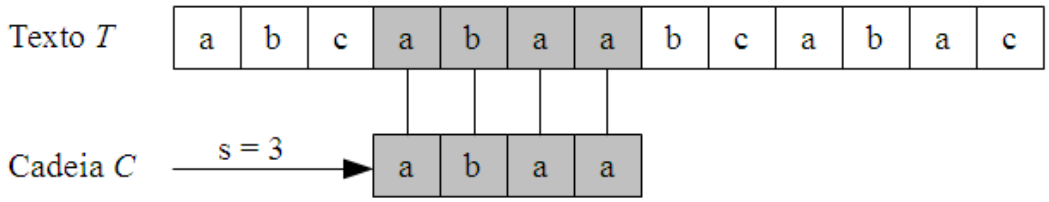


Figura 4.1: Exemplo de um deslocamento válido da cadeia *abaa* no texto T (Fonte: Cormen et al. (2002)).

As soluções para o emparelhamento de cadeias também podem ser usadas para encontrar, por exemplo, padrões específicos em sequências de DNA. Existem vários algoritmos para solucionar o problema do emparelhamento de cadeias na literatura e todos se baseiam nos conceitos apresentados acima.

4.4.2 Solução Baseada em Autômatos Adaptativos

A solução adaptativa para o problema do emparelhamento de cadeias é baseada nos autômatos adaptativos, cujos dispositivos subjacentes são os autômatos de pilha estruturados (NETO, 1994). Para cada cadeia a ser procurada em um texto, um autômato que reconheça somente aquela cadeia deve ser construído.

A busca por uma cadeia em um texto é feita em duas etapas. A primeira etapa é a construção do autômato que reconheça a cadeia a ser procurada no texto. Esta etapa pode ser considerada um pré-processamento. A segunda etapa é o uso do autômato construído para realizar a busca e encontrar todas as ocorrências da cadeia no texto.

Para facilitar a compreensão, neste trabalho o alfabeto foi limitado a apenas dois elementos. Suponha que o alfabeto seja $\Sigma = \{0, 1\}$ e que a cadeia a ser procurada no texto seja $C = 01\perp$. O símbolo especial no final da cadeia, $\perp$, é usado para delimitar o final da sequência de caracteres. A ideia geral é que a cada símbolo lido na cadeia, um novo estado seja criado e novas transições sejam inseridas para todos os símbolos do alfabeto. Para criar essas transições é necessário saber qual o estado destino para cada símbolo, pois é possível que haja sobreposição de símbolos de ocorrências sucessivas da cadeia no texto. Por exemplo, se a cadeia é 010 e o texto é 01010, então o último

zero da primeira ocorrência da cadeia é o primeiro zero da segunda ocorrência da cadeia no texto.

Com o objetivo de criar um autômato determinístico para ser usado na busca, é necessário saber, para qualquer estado, qual o próximo estado para cada símbolo do alfabeto. Considerando-se as sobreposições possíveis, é preciso determinar se um novo estado deve ser criado ou se a criação de uma transição para um estado já existente com cada símbolo do alfabeto é suficiente. Para solucionar este problema, é necessário encontrar o maior prefixo de C que é um sufixo da subcadeia que já é reconhecida pelo autômato, concatenado com o símbolo para o qual será criada a transição. Uma cadeia w é um prefixo de uma cadeia x, se $x = wy$ para alguma cadeia $w \in \Sigma^*$ e alguma cadeia $y \in \Sigma^*$, em que Σ^* é o conjunto de todas as cadeias formadas pelos símbolos de Σ, incluindo a cadeia vazia. Da mesma forma, uma cadeia w é um sufixo de uma cadeia x, se $x = yw$ para alguma cadeia $w \in \Sigma^*$ e alguma cadeia $y \in \Sigma^*$.

Para que o autômato conheça o comprimento do prefixo mais longo de C, que é um sufixo da subcadeia lida até o momento, o símbolo lido na cadeia é passado para uma função adaptativa responsável por criar um novo estado em uma submáquina, que irá reconhecer o padrão lido até o momento. Em seguida, outra submáquina empilha os símbolos anteriormente lidos em ordem invertida, que representam o prefixo de C, para fins de comparação com a subcadeia lida até o momento. Os símbolos são empilhados em ordem invertida porque é preciso verificar se este prefixo é um sufixo da cadeia lida até então.

Para separar o prefixo de C dos símbolos que ainda não foram lidos do padrão, é usado o símbolo especial #. A submáquina que reconhece o padrão lido até o momento é executada consumindo os símbolos empilhados até encontrar o símbolo #. À medida que os símbolos são consumidos e a submáquina muda de estado, um estado equivalente no autômato final aponta para o estado de destino de acordo com o símbolo que determina a transição a ser inserida. Portanto, cada vez que o sufixo é encontrado, uma submáquina com um ponteiro para o estado correspondente indica qual estado deve receber a transição.

Uma vez que a formalização da solução adaptativa para o problema do emparelhamento de cadeias está disponível no artigo (RODRIGUES; RODRIGUES; ROCHA, 2008b), optou-se por, neste texto, omitir as funções e apresentar apenas as definições das submáquinas que compõem a solução. De qualquer forma, a notação descrita na subseção seguinte é necessária para compreender, não só as submáquinas apresentadas como também, qualquer autômato adaptativo.

4.4.2.1 Notação

O objetivo desta subseção é descrever a notação utilizada na formalização de autômatos adaptativos, que foi o dispositivo utilizado na solução do problema do emparelhamento de cadeias. As transições das submáquinas são representadas por regras de substituição com a seguinte forma: $(\gamma g, s, \sigma\alpha), A :\rightarrow (\gamma g', s', \sigma'\alpha), B$, em que

- γ é um meta-símbolo que representa o conteúdo da pilha não considerado pelo autômato (não influencia na aplicação da regra);
- g e g' representam o estado armazenado no topo da pilha antes e depois da aplicação da transição, respectivamente;
- s e s' representam o estado corrente da transição e o próximo estado, respectivamente;
- σ é o símbolo do alfabeto que será consumido da cadeia e σ' é o símbolo do alfabeto que será empilhado (ambos podem ser iguais à cadeia vazia, representada pelo símbolo ε);
- α é um meta-símbolo que representa o restante da cadeia não considerada pelo autômato (não influencia na aplicação da regra);
- A e B são chamadas a funções adaptativas a serem executadas antes e depois da mudança de estados determinada pela regra. As funções adaptativas podem ter n parâmetros e assumem a seguinte forma: $A(a_1, a_2, \cdots, a_n)$, em que a_i, $1 \leq i \leq n$, são argumentos que assumirão valores a serem usados no corpo da função. As funções adaptativas são declaradas conforme descrito em (NETO, 1994).

Um autômato também pode ser representado graficamente, conforme a seguinte notação:

- Os círculos representam os estados;
- As setas representam as transições entre os estados;
- Uma seta sem origem apontando para um estado indica o estado inicial;
- Um círculo duplo indica o estado final;
- As transições com linhas mais espessas indicam uma chamada a outra submáquina;

- O rótulo de uma transição pode conter de um a três elementos separados por vírgula. Os elementos podem ser: uma função adaptativa a ser executada antes da mudança de estado, um símbolo do alfabeto que será consumido da cadeia e uma função adaptativa a ser executada após a mudança de estado determinada pela regra.

4.4.2.2 Exemplo de Funcionamento do Autômato Adaptativo

Considere a cadeia $C = 01\perp$ sobre o alfabeto $\Sigma = \{0,1\}$. O autômato adaptativo para o problema do emparelhamento de cadeias é composto por cinco submáquinas. A submáquina principal responsável pelo reconhecimento da cadeia no texto é Z. Nesta submáquina, Z_0 é o estado inicial e Z_F é o estado final. A definição de Z é mostrada abaixo.

$(\varepsilon, Z_0, \varepsilon) :\rightarrow (\varepsilon, Z_I, \varepsilon)$

$\{(\varepsilon, Z_I, \sigma\alpha) :\rightarrow (\varepsilon, Z_J, \alpha), B(\sigma) \forall \sigma \in \Sigma\}$

$(\varepsilon, Z_J, \varepsilon) :\rightarrow (\varepsilon, Z_K, \varepsilon)$

$(\gamma, Z_K, \varepsilon) :\rightarrow (\gamma Z_Q, PK_0, \varepsilon)$

$(\varepsilon, Z_Q, \varepsilon) :\rightarrow (\varepsilon, Z_X, \varepsilon), I(PK_1, PC_1)$

$(\varepsilon, Z_X, \varepsilon) :\rightarrow (\varepsilon, Z_U, \varepsilon), J()$

$(\gamma, Z_U, \varepsilon) :\rightarrow (\gamma Z_T, PQ_0, \varepsilon)$

$(\varepsilon, Z_T, \varepsilon) :\rightarrow (\varepsilon, Z_W, \varepsilon)$

$(\varepsilon, Z_W, \perp\alpha) :\rightarrow (\varepsilon, Z_V, \alpha), C()$

$(\varepsilon, Z_W, \varepsilon) :\rightarrow (\varepsilon, Z_I, \varepsilon)$

$(\varepsilon, Z_V, \varepsilon) :\rightarrow (\varepsilon, Z_F, \varepsilon)$

$(\varepsilon, P_Z, \varepsilon) :\rightarrow (\varepsilon, P_Z, \varepsilon)$

$(\varepsilon, P_O, \varepsilon) :\rightarrow (\varepsilon, P_O, \varepsilon)$

A Figura 4.2 mostra uma representação gráfica da submáquina Z antes de qualquer operação.

As submáquinas que representam, respectivamente, a subcadeia de C que já foi lida até o momento e que será usada para determinar o próximo estado de cada transição inserida, o prefixo de C que será inserido novamente na cadeia em ordem inversa, o contador de ocorrências do padrão no texto e o maior prefixo de C que é um sufixo da subcadeia lida, são PQ, PK, A e PC. Os estados iniciais de PQ, PK, A e PC são PQ_0,

PK_0, A_0 e PC_0, respectivamente. Da mesma forma, os estados finais são PQ_1, PK_2, A_1 e PC_2. Inicialmente todas as submáquinas estão no estado inicial. As definições das submáquinas são:

$$
\begin{aligned}
PQ : (\varepsilon, PQ_0, \varepsilon) &:\rightarrow (\varepsilon, PQ_1, \varepsilon) \\
(\varepsilon, PQ_0, \#\alpha) &:\rightarrow (\varepsilon, PQ_1, \alpha) \\
(\gamma\pi, PQ_1, \varepsilon) &:\rightarrow (\gamma, \pi, \varepsilon) \\
PK : (\varepsilon, PK_0, \alpha) &:\rightarrow (\varepsilon, PK_1, \#\alpha) \\
(\varepsilon, PK_1, \varepsilon) &:\rightarrow (\varepsilon, PK_2, \varepsilon) \\
(\gamma\pi, PK_2, \varepsilon) &:\rightarrow (\gamma, \pi, \varepsilon) \\
A : (\varepsilon, A_0, \varepsilon) &:\rightarrow (\varepsilon, A_1, \varepsilon) \\
(\gamma\pi, A_1, \varepsilon) &:\rightarrow (\gamma, \pi, \varepsilon) \\
PC : (\varepsilon, PC_0, \alpha) &:\rightarrow (\varepsilon, PC_1, \#\alpha) \\
(\varepsilon, PC_1, \varepsilon) &:\rightarrow (\varepsilon, PC_2, \varepsilon) \\
(\gamma\pi, PC_2, \varepsilon) &:\rightarrow (\gamma, \pi, \varepsilon)
\end{aligned}
$$

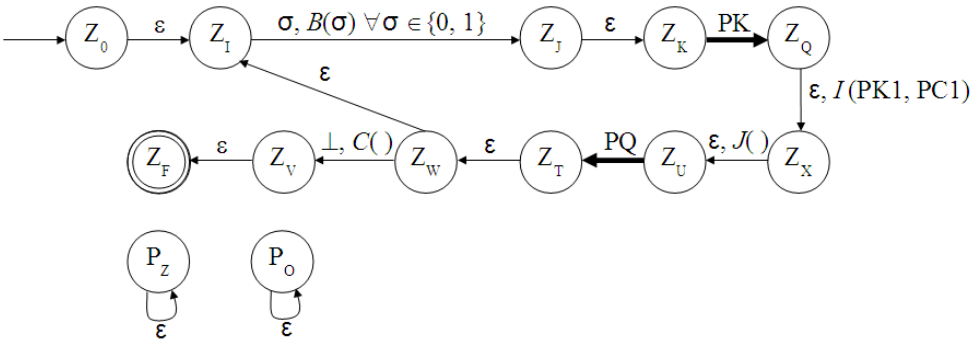


Figura 4.2: Representação gráfica da submáquina principal Z (Fonte: autora).

A solução apresentada no artigo (RODRIGUES; RODRIGUES; ROCHA, 2008b) foi criada para um alfabeto de apenas dois símbolos. No entanto, é possível ampliar esta solução para qualquer alfabeto. Para cada símbolo do alfabeto seria necessário incluir uma função adaptativa para inserir uma transição para os demais símbolos. Portanto, quanto maior o tamanho do alfabeto, mais funções adaptativas são necessárias.

Após o processamento de toda a cadeia, o símbolo $\perp$ é encontrado. Nesse ponto, uma função adaptativa é executada para remover as transições com a função que criava os estados e acrescentava as transições. Além disso, uma função adaptativa é inserida na transição que chega ao estado final. Esta ação será executada na etapa de busca para contar o número de ocorrências da cadeia no texto. A Figura 4.3 mostra o resultado final do autômato após o processamento da cadeia.

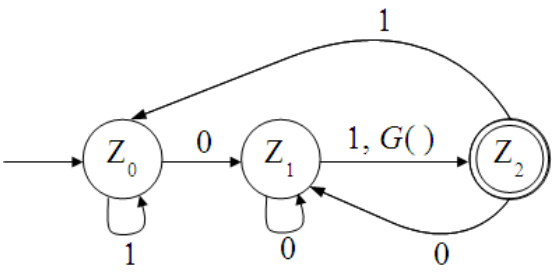


Figura 4.3: Representação gráfica da submáquina principal *Z* após o processamento da cadeia (Fonte: autora).

Uma vez construído o autômato, começa a etapa de busca no texto. A busca inicia no primeiro símbolo do texto e vai consumindo cada símbolo até atingir o final. O autômato muda de estado de acordo com cada símbolo lido no texto. Cada vez que o autômato alcança o estado final significa que a cadeia foi encontrada no texto. Para saber o deslocamento da cadeia no texto, basta subtrair o valor da posição do texto em que o padrão terminou pelo tamanho do padrão. A função adaptativa que é executada cada vez que um padrão é encontrado no texto, cria um autômato que representa o número de ocorrências da cadeia no texto.

4.5 Epítome

O foco deste capítulo foi a apresentação dos principais conceitos relacionados à tecnologia adaptativa, aplicação prática da adaptatividade na solução de problemas dinâmicos. A adaptatividade, capacidade de automodificação de um sistema durante a sua execução, é a característica fundamental dos dispositivos adaptativos. Os sistemas automodificáveis são capazes de mudar o seu comportamento em resposta aos estímulos de entrada e seu histórico de operação, sem a interferência de agentes externos.

Um dispositivo subjacente, com regras estáticas, e um mecanismo adaptativo formam um dispositivo adaptativo. Qualquer formalismo baseado em um conjunto finito de regras pode descrever um dispositivo subjacente. O mecanismo adaptativo é a combinação do dispositivo subjacente com funções adaptativas que permitem alterar o próprio conjunto de regras do dispositivo. Cada regra do dispositivo adaptativo é uma regra do dispositivo subjacente com duas funções adaptativas acopladas, uma para ser executada antes e outra após a mudança de configuração. As funções adaptativas devem ser implementadas com as seguintes ações adaptativas elementares: ações de consulta, ações de remoção e ações de inserção.

O dispositivo adaptativo mais utilizado é o autômato adaptativo, cujo dispositivo subjacente é o autômato de pilha estruturado. No entanto, vários formalismos adapta-

tivos já foram desenvolvidos, tais como: gramáticas adaptativas, Máquinas de Markov Adaptativas, *statecharts* adaptativos, árvores de decisão adaptativas e tabelas de decisão adaptativas.

A tecnologia adaptativa já foi aplicada em diversas áreas de pesquisa. Alguns exemplos de aplicação são: criptografia, robótica, processamento de linguagem natural, composição e edição musical, e modelagem de distribuição de espécies. Um exemplo prático de aplicação da adaptatividade foi apresentado em detalhes para o problema do emparelhamento de cadeias.

5 Desenvolvimentos, Experimentos e Resultados

As contribuições deste trabalho estão concentradas neste capítulo. Todos os desenvolvimentos, os experimentos realizados, as metodologias adotadas em cada etapa e os resultados obtidos serão apresentados nas próximas seções. O objetivo é descrever cada etapa do trabalho e apresentar os passos necessários para alcançar cada meta estabelecida. A seguir é apresentada a organização do capítulo.

A Seção 5.1 descreve a metodologia utilizada para tornar disponível um algoritmo de Entropia Máxima na ferramenta *openModeller* e apresenta o algoritmo implementado. Na Seção 5.2 é apresentado o algoritmo adaptativo de Entropia Máxima, os experimentos realizados para validá-lo e os resultados obtidos. O algoritmo paralelo de Entropia Máxima é mostrado na Seção 5.3, seguido dos experimentos realizados para legitimá-lo e dos resultados alcançados. A Seção 5.4 apresenta uma discussão sobre a necessidade de ajustar o parâmetro de regularização do algoritmo de Entropia Máxima, os experimentos realizados na tentativa de ajustá-lo e os resultados obtidos. A Seção 5.5 descreve como o princípio MDL pode ser aplicado na seleção de variáveis, alguns experimentos efetuados e os resultados alcançados. Por fim, a Seção 5.6 apresenta um resumo do capítulo.

5.1 Algoritmo de Entropia Máxima

Para atingir o objetivo principal de definir diferentes estratégias para o algoritmo de Entropia Máxima no contexto da modelagem de distribuição de espécies, foi necessário integrar um algoritmo de Entropia Máxima à ferramenta *openModeller*. Essa etapa inicial foi concluída em duas fases:

1. Inserção de uma biblioteca com o algoritmo de Entropia Máxima na ferramenta *openModeller*;

2. Implementação de um algoritmo de modelagem de distribuição de espécies baseado em Entropia Máxima e substituição da biblioteca inserida anteriormente por este algoritmo.

Cada uma dessas fases será discutida separadamente nas subseções seguintes.

5.1.1 Inserção de uma Biblioteca com o Algoritmo de Entropia Máxima na Ferramenta *openModeller*

A inserção de uma biblioteca foi uma solução prática e rápida para a disponibilização do algoritmo de Entropia Máxima na ferramenta *openModeller* (RODRIGUES; RODRIGUES; ROCHA, 2008c). A biblioteca inserida inicialmente chama-se *Maximum Entropy Modeling Toolkit* e foi escrita por Zhang Le (LE, 2004). Esta biblioteca oferece dois algoritmos para estimar os parâmetros da distribuição de probabilidade com Entropia Máxima, o GIS (*Generalized Iterative Scaling*) (DARROCH; RATCLIFF, 1972), que é um método de escalonamento iterativo, e o L-BFGS (*Limited-Memory Broyden-Fletcher-Goldfarb-Shanno*) (LIU; NOCEDAL, 1989).

A ideia geral dos métodos de escalonamento iterativo é que a partir de um modelo inicial $\lambda^{(0)}$, uma sequência de modelos $\lambda^{(1)}, \lambda^{(2)}, \ldots$ seja gerada, de tal forma que cada novo modelo $\lambda^{(t+1)}$ seja melhor que o modelo anterior $\lambda^{(t)}$. O procedimento GIS (DARROCH; RATCLIFF, 1972) requer que a soma das *features* de cada exemplo da amostra de dados seja igual a uma constante C (5.1). No entanto, isso nem sempre é verdadeiro.

$$C = \sum_{j=1}^{m} f_j(x). \tag{5.1}$$

Quando a soma das *features* de cada exemplo da amostra de dados não é uma constante, C é

$$C = max_{x \in X} \sum_{j=1}^{m} f_j(x) \tag{5.2}$$

e a seguinte *feature* de correção $f_c(x)$ é adicionada

$$f_c(x) = C - \sum_{j=1}^{m} f_j(x). \tag{5.3}$$

Entretanto, a inserção de uma *feature* de correção não adiciona informação alguma ao modelo, pois ela apenas depende das outras *features*. Por isso, segundo Goodman (2002), a *feature* de correção pode ser omitida. Isso pode ser feito atribuindo um peso igual a zero à *feature* de correção e nunca o atualizando. A convergência dessa simplificação do algoritmo GIS foi provada por Curran e Clark (2003). Assim, o procedimento GIS pode ser implementado conforme o Algoritmo 5.1.

Algoritmo 5.1 Algoritmo GIS.

Inicializar $\lambda_j^{(0)} = 0$, para $j = 1, 2, \ldots, m$;
$C = max_{x \in X} \sum_{j=1}^{m} f_j(x)$;
for $t = 0, \cdots$ **to** convergir **do**
    $soma(x_i) = 0$;
    $Z_\lambda = 0$;
    $p_\lambda[f_j] = 0$;
    for $i = 0$ **to** $i = n$ **do**
        for $j = 0$ **to** $j = m$ **do**
            $soma(x_i) = soma(x_i) + \lambda_j^{(t)} * f_j(x_i)$;
        end for
        $Z_\lambda = Z_\lambda + e^{soma(x_i)}$;
    end for
    for $j = 0$ **to** $j = m$ **do**
        for $i = 0$ **to** $i = n$ **do**
            $p_\lambda[f_j] = p_\lambda[f_j] + \frac{e^{soma(x_i)}}{Z_\lambda} * f_j(x_i)$;
        end for
    end for
    for $j = 1$ **to** $j = m$ **do**
        $\delta_j = \frac{1}{C} * log \frac{\tilde{p}[f_j]}{p_\lambda[f_j]}$;
        $\lambda_j^{(t+1)} = \lambda_j^{(t)} + \delta_j$;
    end for
end for

O L-BFGS (LIU; NOCEDAL, 1989) é um algoritmo da família dos métodos de otimização quasi-Newton que utiliza uma variação do método BFGS com memória limitada. O algoritmo L-BFGS é indicado para problemas com muitas variáveis, pois armazena apenas alguns vetores para representar uma aproximação da inversa da matriz Hessiana, enquanto o algoritmo original armazena toda a matriz. Os conceitos necessários para a compreensão do algoritmo L-BFGS estão fora do escopo deste trabalho, por isso ele não será apresentado aqui. No entanto, uma explicação detalhada deste algoritmo pode ser encontrada no trabalho de Liu e Nocedal (1989).

Segundo Goodman (2002), o algoritmo GIS é extremamente lento e de acordo com Liang, Yu e Taft (2004), as taxas de convergência deste algoritmo são exponenciais. Uma desvantagem do algoritmo GIS é que, na implementação disponível na biblioteca, ele só aceita variáveis binárias. Além disso, para a execução da biblioteca inserida era necessário instalar outras bibliotecas, das quais os algoritmos de estimativa de parâmetros dependiam. Como o algoritmo L-BFGS foi implementado em Fortran, também era necessária a instalação de um compilador Fortran para habilitar a execução deste algoritmo.

Os modelos gerados pelos algoritmos de estimativa de parâmetros da biblioteca inserida eram baseados em probabilidades condicionais. Os modelos condicionais de Entropia Máxima são mais usados em problemas de classificação, como em processamento de linguagem natural, por exemplo. No entanto, na modelagem de distribuição de espécies, a maioria dos dados são apenas de uma classe, que são os pontos de presença da espécie, conforme apresentado na Seção 2.2 do Capítulo 2. Desta forma, uma desvantagem dos algoritmos implementados na biblioteca era a necessidade de gerar pontos de pseudo-ausência.

As desvantagens apresentadas acima, fizeram com que surgisse a necessidade de implementação de um algoritmo de modelagem de distribuição de espécies baseado em Entropia Máxima específico para a ferramenta *openModeller*. Desta forma, a biblioteca inserida poderia ser substituída e os problemas relacionados a sua execução poderiam ser eliminados.

5.1.2 Implementação de um Algoritmo de Modelagem de Distribuição de Espécies Baseado em Entropia Máxima

Os dois algoritmos de estimativa de parâmetros disponíveis na biblioteca inserida inicialmente no *openModeller* atualizam os pesos de todas as *features* a cada iteração. O algoritmo de modelagem de distribuição de espécies baseado em Entropia Máxima implementado durante o desenvolvimento deste trabalho (CORRÊA et al., 2011) foi baseado no algoritmo sequencial de atualização de pesos de Phillips, Dudík e Schapire (2004).

O algoritmo de atualização sequencial de pesos escolhe apenas uma *feature* para atualizar o seu peso a cada iteração. Essa abordagem permite o uso de uma quantidade maior de *features*. Além disso, esse algoritmo utiliza uma abordagem incondicional, ideal para problemas com exemplos de uma única classe (PHILLIPS; ANDERSON; SCHAPIRE, 2006b).

O objetivo do algoritmo é encontrar o vetor de pesos λ que minimiza a função $L^{\beta}_{\tilde{p}}(\lambda)$ (3.12), conforme explicado na Seção 3.1.2.2 do Capítulo 3. O algoritmo atualiza o único peso λ_j que maximiza a diferença entre $L^{\beta}_{\tilde{p}}(\lambda')$ e $L^{\beta}_{\tilde{p}}(\lambda)$ (5.4) (PHILLIPS; DUDÍK; SCHAPIRE, 2004), em que λ' é o vetor λ com o j-ésimo peso atualizado.

$$L^{\beta}_{\tilde{p}}(\lambda') - L^{\beta}_{\tilde{p}}(\lambda) = -\alpha\tilde{p}[f_j] + log(1 + (e^{\alpha} - 1)p_{\lambda}[f_j]) + \beta_j(\left|\lambda_j + \alpha\right| - \left|\lambda_j\right|) \quad (5.4)$$

em que α é o valor que será somado ao peso da j-ésima *feature*. A sequência de derivações da equação (5.4) foi demonstrada no trabalho de Dudík, Phillips e Schapire (2004). Por simplicidade, a partir deste ponto, a equação (5.4) será denotada por $F_j(\lambda_j, \alpha)$. A ideia geral do algoritmo de atualização sequencial de pesos pode ser vista no Algoritmo 5.2.

Algoritmo 5.2 Algoritmo de atualização sequencial de pesos.

Entrada: X; $f_1, f_2, \ldots, f_m$ tal que $f_j : X \to \mathbb{R}$, para $j = 1, 2, \ldots, m$;

$x_1, x_2, \ldots, x_n$ tal que $x_i \in X$, para $i = 1, 2, \ldots, n$;

$\beta_1, \beta_2, \ldots, \beta_m$ tal que $\beta_j > 0$;

Saída: $\lambda_1, \lambda_2, \ldots, \lambda_m$ e $\hat{p}$;

Inicializar $\lambda_j = 0$;

Normalizar f_j para o intervalo $[0, 1]$;

for $t = 1$ **to** $t =$ iterações ou até convergir **do**

  $(j', \alpha) = arg\ max_{(j,\alpha)}\ F_j(\lambda_j, \alpha)$, em que $F_j(\lambda_j, \alpha)$ é a equação (5.4);

  if $j' = j$ **then**

    $\lambda_j = \lambda_j + \alpha$;

  else

    $\lambda_j = \lambda_j$;

  end if

end for

Em cada iteração do algoritmo, uma *feature* é escolhida para atualizar o seu peso. Esta escolha é feita de tal forma que o par (j', α) maximize a função $F_j(\lambda_j, \alpha)$ (5.4), em que j' é o índice da *feature* escolhida e α é o valor que será somado ao peso da *feature*. A seleção de α é computada pela tentativa de três possibilidades para cada *feature*. Assim, o valor de α deve ser $\alpha = -\lambda_j$, ou

$$\alpha = log\left(\frac{(\tilde{p}[f_j] - \beta_j)(1 - p_{\lambda}[f_j])}{(1 - \tilde{p}[f_j] + \beta_j)p_{\lambda}[f_j]}\right), \quad (5.5)$$

válido somente se $\lambda_j + \alpha \geq 0$, ou

$$\alpha = log\left(\frac{(\tilde{p}[f_j] + \beta_j)(1 - p_\lambda[f_j])}{(1 - \tilde{p}[f_j] - \beta_j)p_\lambda[f_j]}\right), \tag{5.6}$$

válido somente se $\lambda_j + \alpha \leq 0$.

Os três valores possíveis de α são calculados para todas as *features* em cada iteração. A partir do par (j', α) que maximiza (5.4), o peso da *feature* $f_{j'}$ é incrementado com α. A prova de que este algoritmo converge para a distribuição de probabilidade com Entropia Máxima pode ser encontrada no trabalho de Dudík, Phillips e Schapire (2004).

5.2 Algoritmo Adaptativo de Entropia Máxima

Esta seção apresenta uma abordagem adaptativa do algoritmo de Entropia Máxima para a modelagem de distribuição de espécies (RODRIGUES et al., 2011a). O principal objetivo desta abordagem é reduzir a quantidade de *features* usadas na estimativa dos modelos. A abordagem apresentada aqui é uma alternativa para o algoritmo clássico. Ao invés de usar o mesmo conjunto de *features* a cada iteração, a abordagem adaptativa tenta encontrar um novo conjunto de *features* que convirja mais rápido.

A definição e implementação de um autômato adaptativo para solucionar o problema do emparelhamento de cadeias, apresentado na Seção 4.4 do Capítulo 4, foi crucial para o entendimento dos conceitos fundamentais da tecnologia adaptativa. Embora naquele trabalho tenha sido usado um autômato adaptativo, as ideias que embasam o uso da tecnologia adaptativa são as mesmas para qualquer formalismo.

Até o presente momento, o único algoritmo de modelagem de distribuição de espécies que possui uma versão adaptativa, disponível no *openModeller*, é o GARP (BRAVO et al., 2007). No entanto, outros algoritmos de modelagem de distribuição de espécies, como o algoritmo de Entropia Máxima, também possuem características que se beneficiariam, de alguma maneira, da inserção da propriedade adaptativa em suas definições. Além disso, a adaptatividade pode ser explorada em outras etapas da modelagem, como na pós-análise por exemplo (RODRIGUES; RODRIGUES; ROCHA, 2009).

5.2.1 Desenvolvimento do Algoritmo Adaptativo de Entropia Máxima

Conforme apresentado na Seção 5.1.2, o algoritmo de Entropia Máxima implementado para a ferramenta *openModeller* escolhe uma *feature* de um conjunto a cada iteração

e atualiza seu peso. Esta escolha é baseada em uma função objetivo, em que todas as *features* são analisadas. Assim, as *features* influenciam diretamente a aprendizagem do algoritmo, isto é, inserir ou remover *features* do conjunto de treinamento pode melhorar o ajuste dos pesos e a convergência pode ser alcançada mais rapidamente. No entanto, no algoritmo implementado no *openModeller*, o conjunto de *features* utilizado durante o processo de aprendizagem é estático.

A abordagem aqui proposta insere ou remove *features* do conjunto de treinamento durante a execução do algoritmo de aprendizagem. Para representar esta característica dinâmica, foram utilizados os conceitos relacionados à tecnologia adaptativa, pois a ideia fundamental é modificar o conjunto de dados de treinamento sem interferências externas. O algoritmo proposto utiliza uma estratégia "gulosa" porque ele tenta descobrir um conjunto de *features* que parece ser o melhor em cada iteração, em vez de usar o mesmo conjunto de *features* durante todo o processo de aprendizagem do modelo.

A Figura 5.1 apresenta a visão geral do procedimento de treinamento. Conforme pode ser observado, os dados de pontos de ocorrência são selecionados apenas uma vez. No entanto, caso o algoritmo não tenha alcançado a taxa de convergência nem o número de iterações especificados pelo usuário, um novo conjunto de *features* pode ser selecionado.

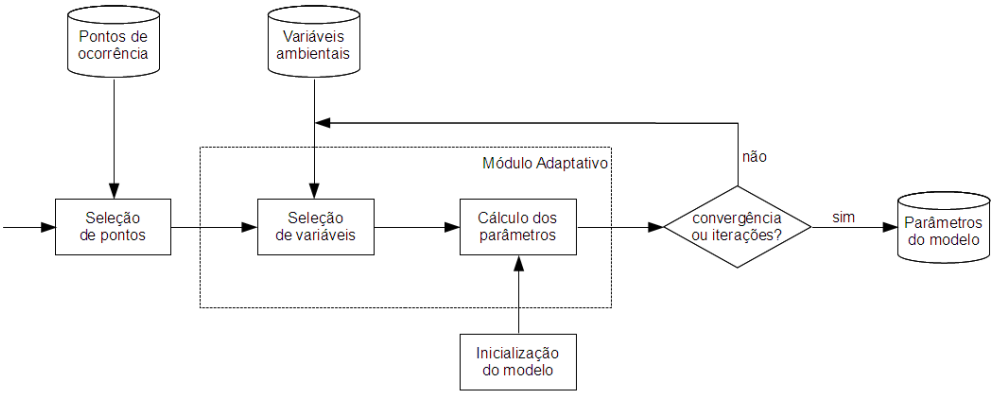


Figura 5.1: Visão geral do procedimento de treinamento (Fonte: autora).

O módulo adaptativo foi projetado para modificar apenas o número de *features* usadas no conjunto de treinamento em cada iteração. Portanto, este procedimento poderia ser aplicado em qualquer algoritmo de modelagem baseado em otimização disponível no *openModeller*. A Figura 5.2 mostra um esquema geral do módulo adaptativo proposto. O esquema apresenta o funcionamento de uma iteração, em que a função objetivo e o critério de seleção do melhor subconjunto de variáveis devem ser definidos de acordo com um determinado algoritmo.

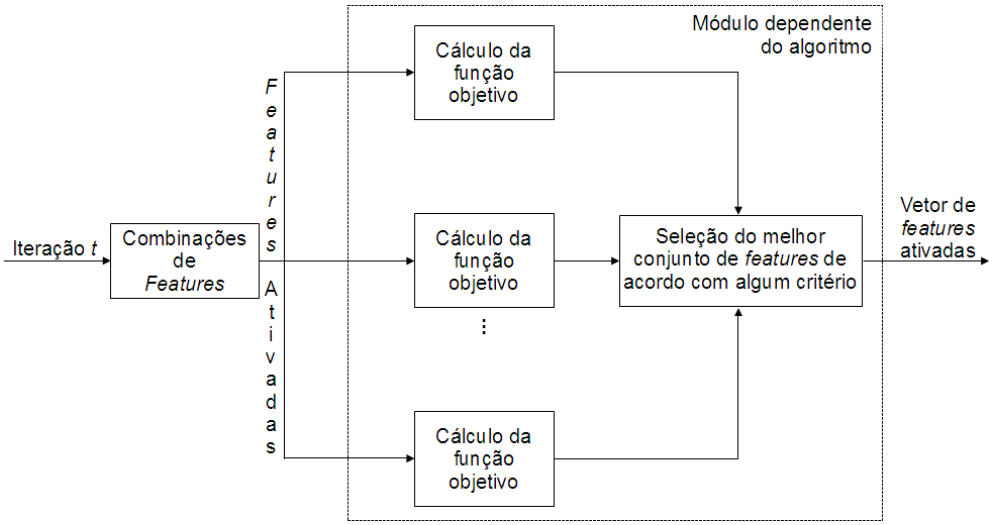


Figura 5.2: Esquema geral do módulo adaptativo proposto (Fonte: autora).

Para instanciar o esquema geral da Figura 5.2 para o algoritmo de Entropia Máxima, F_j, descrita na seção anterior, é usada como função objetivo e o *log loss* é usado como critério de escolha do melhor subconjunto de variáveis. Um esquema detalhado do módulo adaptativo do algoritmo de Entropia Máxima é apresentado na Figura 5.3. Inicialmente o usuário seleciona um conjunto de possíveis *features* para um determinado experimento. A partir do conjunto de *features* selecionado pelo usuário, o algoritmo cria subconjuntos para testar qual é a melhor combinação na iteração corrente.

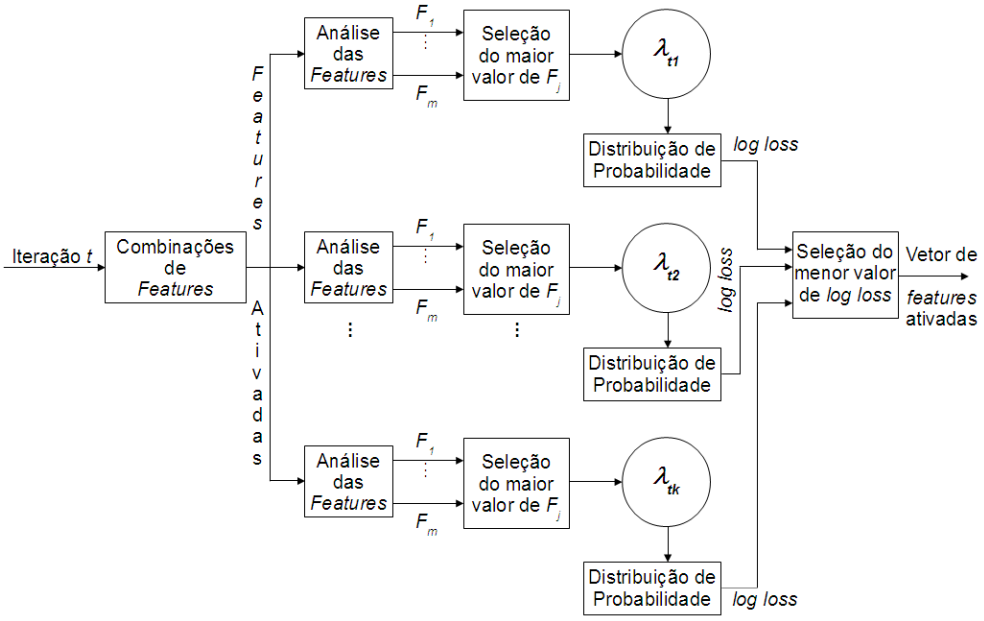


Figura 5.3: Esquema do módulo adaptativo do algoritmo de Entropia Máxima (Fonte: autora).

Para representar as *features* escolhidas, foi utilizado um vetor a com m posições, uma para cada *feature*. Se a *feature* correspondente à j-ésima posição do vetor está no subconjunto considerado melhor na iteração corrente, então a j-ésima posição é

ativada. Caso contrário, a j-ésima posição é desativada. O número 1 representa a ativação da *feature* e o número 0 representa a desativação. Para cada subconjunto, o valor da função objetivo é calculado, o peso da *feature* escolhida é atualizado e o valor do *log loss* é usado para determinar qual o melhor subconjunto. Assim, λ_{tk} é o valor do peso da *feature* atualizada na iteração t e que pertence ao subconjunto k, tal que $k \in \{1,2,\ldots,q\}$ e $q = 2^m - 1$.

O Algoritmo 5.3 apresenta o procedimento do algoritmo adaptativo de Entropia Máxima desenvolvido neste trabalho. Para facilitar a compreensão, a função adaptativa, chamada na linha 11, é apresentada no Algoritmo 5.4. Inicialmente todas as *features* estão desativadas.

Algoritmo 5.3 Algoritmo Adaptativo de Entropia Máxima.

1: **Entrada:** X; $f_1, f_2, \ldots, f_m$ tal que $f_j : X \to \mathbb{R}$, para $j = 1,2,\ldots,m$;
2: $x_1, x_2, \ldots, x_n$ tal que $x_i \in X$, para $i = 1,2,\ldots,n$;
3: $\beta_1, \beta_2, \ldots, \beta_m$ tal que $\beta_j > 0$;
4: **Saída:** $\lambda_1, \lambda_2, \ldots, \lambda_m$ e $\hat{p}$;
5: $a_1, a_2, \ldots, a_m$ tal que $a_j \in \{0,1\}$;
6: Inicializar $\lambda_j = 0$;
7: Normalizar f_j para o intervalo $[0,1]$;
8: Inicializar $a_j = 0$;
9: **for** $t = 1$ **to** $t =$ iterações ou até convergir **do**
10: **for** $j = 1$ **to** $j \leq m$ **do**
11: ***funcao_adaptativa***(0, m - i, 0, i);
12: **end for**
13: **if** $j' = j$ **then**
14: $\lambda_j = \lambda_j + \alpha$;
15: **else**
16: $\lambda_j = \lambda_j$;
17: **end if**
18: **end for**

Algoritmo 5.4 Função Adaptativa.

1: **begin** ***funcao_adaptativa***(inicio, fim, p, i)
2: **if** $(p+1) < i$ **then**
3: **for** $j = inicio$ **to** $j \leq fim$ **do**
4: **INSERE** f_j em C;
5: ***funcao_adaptativa***(j + 1, fim + 1, p + 1, i);
6: **REMOVE** f_j de C;
7: **end for**
8: **else**
9: **for** $j = inicio$ **to** $j \leq fim$ **do**
10: **INSERE** f_j em C;
11: $(k', \alpha_k) = arg\ max_{(k,\alpha)}\ F_k(\lambda_k, \alpha), \forall f_k \in C$;
12: **if** $\alpha_k < \alpha$ **then**
13: $\alpha = \alpha_k$;
14: $j' = k'$;
15: **for** $l = 1$ **to** $l \leq m$ **do**
16: **if** $f_l \in C$ **then**
17: $a_l = 1$;
18: **end if**
19: **end for**
20: **end if**
21: **REMOVE** f_j;
22: **end for**
23: **end if**
24: **end** ***funcao_adaptativa***

5.2.2 Experimentos

Os experimentos realizados tiveram como objetivo validar a abordagem adaptativa do algoritmo de Entropia Máxima. Para isso, foram medidos os tempos de execução do algoritmo de Entropia Máxima convencional e do algoritmo adaptativo de Entropia Máxima. Além disso, as precisões dos modelos estimados foram medidas para com-

parar a capacidade de aprendizagem dos algoritmos. Como o objetivo não era verificar o desempenho dos modelos e sim do algoritmo desenvolvido, não foram realizados testes independentes.

5.2.2.1 Dados Utilizados

Foram utilizados dados de duas espécies para validar o algoritmo adaptativo de Entropia Máxima. As espécies selecionadas foram: *Byrsonima intermedia* e *Xylopia aromatica*. A primeira é um arbusto da família *Malpighiaceae*. Ela é popularmente conhecida como *murici-pequeno* e é uma espécie medicinal nativa do cerrado brasileiro. A segunda é uma pequena árvore da família *Annonaceae*. Esta espécie é popularmente conhecida como *pimenta-de-macaco*, geralmente usada como lenha e também é encontrada no cerrado brasileiro. Os dados de ocorrência destas duas espécies são derivados do SinBiota - sistema de informação ambiental para o programa Biota/Fapesp (BIOTA, 2009).

Os experimentos foram realizados com 38 registros da espécie *Byrsonima intermedia* e 33 registros da espécie *Xylopia aromatica*. Tanto as espécies quanto as variáveis ambientais foram selecionadas por uma bióloga. Uma vez que todos os pontos de ocorrência das espécies usadas nos experimentos são do estado de São Paulo - Brasil, os modelos foram estimados para a mesma região. A Figura 5.4 apresenta os pontos utilizados nos experimentos.

Figura 5.4: Pontos de ocorrência das espécies *Byrsonima intermedia* (esquerda) e *Xylopia aromatica* (direita) (Fonte: autora).

Os valores das variáveis ambientais pertencem à mesma região dos pontos de ocorrência (estado de São Paulo) com resolução espacial de 30 arc-second (aproximadamente 1km^2). As variáveis ambientais foram fornecidas pelo *WorldClim - Global Climate Data* (HIJMANS et al., 2005). As variáveis ambientais usadas foram:

- temperatura média anual;
- escala média diurna (média mensal da diferença entre temperatura máxima e temperatura mínima);

- temperatura máxima do mês mais quente;
- temperatura mínima do mês mais frio;
- precipitação anual;
- precipitação do mês mais úmido; e
- precipitação do mês mais seco.

5.2.2.2 Metodologia

Todos os experimentos foram realizados em um computador com processador Intel Core 2 Duo de 1,66 GHz e 2 GB de RAM. O sistema operacional usado foi o Ubuntu 10.04, uma distribuição do Linux. Uma das métricas de avaliação do algoritmo foi a comparação entre o tempo de execução da versão convencional e o tempo de execução da versão adaptativa. Para medir o tempo de execução, de ambas as versões, foi utilizado o comando *time*, disponível no Linux.

O comando *time* mede o tempo de execução da aplicação, o tempo gasto pelas funções do sistema operacional durante a execução da aplicação, o tempo total do início até o final da execução, a porcentagem da CPU que a aplicação utilizou ((tempo da aplicação + tempo do sistema) / tempo total), o número de arquivos lidos e escritos pelo processo e o número de *page faults* durante a execução do processo. Existem outras opções de saída que podem ser ativadas por linha de comando. Neste trabalho, apenas o tempo total de execução da aplicação foi considerado.

Além do tempo de execução, a precisão de todos os modelos estimados foi considerada para avaliar a capacidade de aprendizagem da abordagem adaptativa em relação a abordagem clássica do algoritmo de Entropia Máxima. Ambas as versões foram executadas cinco vezes para cada espécie e os valores considerados foram as médias das precisões e dos tempos de execução registrados.

5.2.3 Resultados e Discussão

A Tabela 5.1 apresenta a média do tempo de execução do algoritmo adaptativo e do algoritmo convencional de Entropia Máxima para as duas espécies analisadas. A diferença entre os resultados de cada execução foi insignificante. Isto ocorreu porque o algoritmo de Entropia Máxima é determinístico, ou seja, sempre produz a mesma saída para uma determinada entrada. Assim, não é necessário executar os algoritmos mais de uma vez e calcular a média dos resultados. No entanto, os algoritmos foram executados cinco vezes para cada espécie porque a cada execução o algoritmo utiliza um

conjunto diferente de pontos de *background*. O uso de conjuntos distintos de pontos de *background* pode gerar pequenas diferenças nas estatísticas dos algoritmos.

Tabela 5.1: Tempo médio de execução.

	Byrsonima intermedia	*Xylopia aromatica*
algoritmo adaptativo	9,57 segundos	9,47 segundos
algoritmo convencional	14,38 segundos	14,25 segundos

Uma vez que o algoritmo adaptativo de Entropia Máxima testa todas as combinações possíveis de *features* para escolher a melhor em cada iteração, este é um algoritmo computacionalmente caro. Sua complexidade é exponencial porque o número de combinações cresce exponencialmente conforme o número de *features* aumenta. No entanto, o algoritmo adaptativo de Entropia Máxima busca a melhor combinação em cada iteração, inserindo ou removendo *features* no conjunto de dados. Por isso, o algoritmo adaptativo de Entropia Máxima converge mais rápido e, portanto, executa mais rapidamente que o algoritmo convencional, conforme pode ser observado na Tabela 5.1.

A precisão média dos modelos estimados pelo algoritmo adaptativo e pelo algoritmo convencional de Entropia Máxima são apresentados na Tabela 5.2. Uma redução significante no tempo total de execução dos algoritmos foi observada. Em média, o algoritmo adaptativo de Entropia Máxima foi 33,5% mais rápido que o algoritmo convencional. Além disso, conforme pode ser observado na Tabela 5.2, a precisão também melhorou. O algoritmo adaptativo de Entropia Máxima obteve uma precisão, em média, 40,19% melhor que o algoritmo convencional para a espécie *Byrsonima intermedia* e 6,7% melhor para a espécie *Xylopia aromatica*.

Tabela 5.2: Médias da precisão dos modelos estimados.

	Byrsonima intermedia	*Xylopia aromatica*
algoritmo adaptativo	56,31%	45,45%
algoritmo convencional	33,68%	42,42%

Para o algoritmo de Entropia Máxima, o aparente baixo índice de precisão não indica baixa capacidade de generalização. Como o algoritmo de Entropia Máxima para modelagem de distribuição de espécies utiliza somente pontos de presença em seu treinamento, mesmo com o baixo índice de precisão ele é capaz de diferenciar os pontos de presença dos pontos de *background*. Esta característica resulta em boas taxas preditivas em dados não utilizados durante o treinamento. Portanto, os resultados obtidos indicam que a abordagem proposta é adequada para a modelagem de distribuição de espécies.

5.3 Paralelização do Algoritmo de Entropia Máxima

Os algoritmos de modelagem de distribuição de espécies podem demandar um longo período de execução, dependendo de seus parâmetros e dos dados de entrada. Por isso, faz-se necessário o desenvolvimento de algoritmos mais rápidos, mas que mantenham ou melhorem sua capacidade de aprendizagem. Desta forma, uma das atividades realizadas durante o desenvolvimento deste trabalho foi a análise de técnicas de paralelismo que permitem a criação de algoritmos mais rápidos quando executados em máquinas com multiprocessadores.

A colaboração na implementação de uma versão paralela do algoritmo de pré-análise *Jackknife* (RODRIGUES et al., 2008), disponível na ferramenta *openModeller*, foi muito importante para a compreensão dos conceitos relacionados ao desenvolvimento de algoritmos de alto desempenho. O *Jackknife* é um algoritmo computacionalmente caro e a sua versão paralela apresentou um desempenho aproximadamente 98% melhor que a versão sequencial, com relação ao tempo total de execução.

Além do algoritmo *Jackknife*, outros algoritmos disponíveis na ferramenta *openModeller* também já foram paralelizados, como o GARP (SANTANA et al., 2006) e o algoritmo de projeção dos modelos (REPORT-OM, 2008). Como o algoritmo baseado em Entropia Máxima é um dos mais utilizados na comunidade de modelagem de distribuição de espécies, após a implementação de uma versão sequencial, uma versão paralela deste algoritmo foi implementada. O objetivo desta implementação foi reduzir o tempo de execução deste algoritmo e contribuir para a disponibilização de algoritmos de modelagem mais rápidos na ferramenta *openModeller*.

Vale salientar que os termos "sequencial" e "paralelo", nesta seção, não estão relacionados com a forma de atualização dos pesos das *features*. Nesta seção, esses termos são usados para identificar algoritmos cujos comandos são executados por um processador, algoritmo sequencial, ou por vários processadores simultaneamente, algoritmo paralelo.

5.3.1 Algoritmo Paralelo de Entropia Máxima

Na implementação paralela desenvolvida neste trabalho (RODRIGUES; RODRIGUES; ROCHA, 2008a; RODRIGUES et al., 2010b), foi utilizada a abordagem mestre-escravo apresentada na Figura 5.5. Nesta abordagem, o processo mestre distribui as tarefas para os processos escravos e, após o processamento, recebe o resultado de cada um deles. A estratégia adotada foi o escalonamento dinâmico das tarefas. No escalonamento dinâmico, a distribuição dos processos para os processadores é feita durante a execução do

programa de acordo com algum critério. O critério utilizado nesta implementação foi o escalonamento sob demanda, ou seja, quando um processo escravo termina a execução de uma tarefa, o processo mestre atribui uma nova tarefa para este processo escravo. Esse procedimento ocorre até que todas as tarefas sejam concluídas.

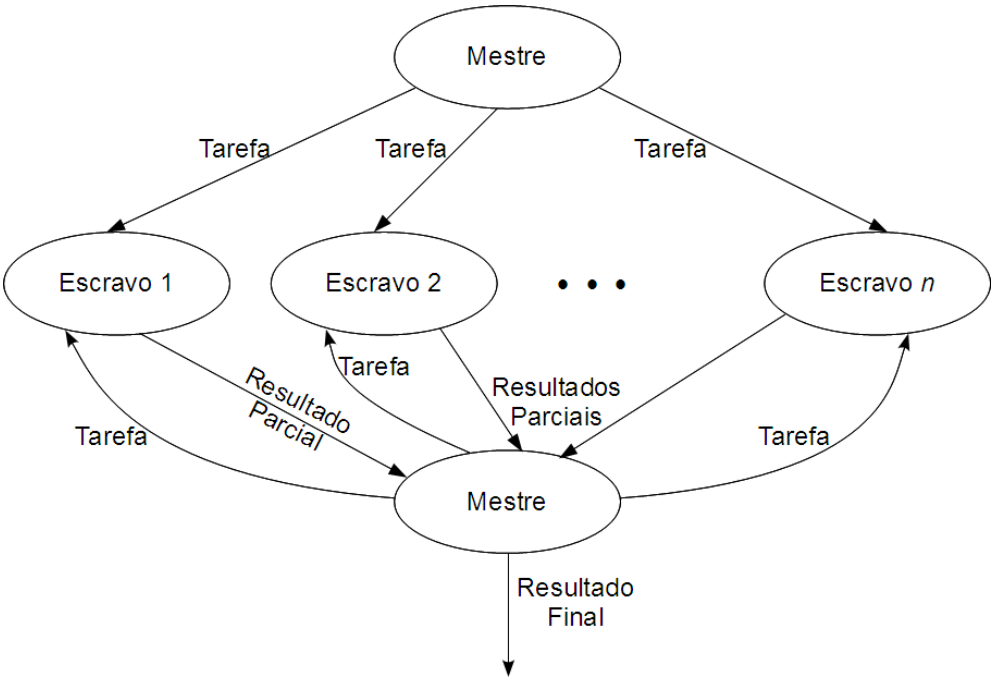


Figura 5.5: Abordagem mestre-escravo aplicada na implementação da versão paralela do algoritmo de Entropia Máxima (Fonte: Rodrigues et al. (2008)).

A versão paralela do algoritmo de Entropia Máxima foi implementada usando a biblioteca MPI (*Message Passing Interface*) (MCNALLY; SIEK, 1998), uma biblioteca de troca de mensagens com o objetivo de prover comunicação entre processos. Uma aplicação MPI adota um tipo de computação paralela conhecida como SPMD (*Single Program, Multiple Data*). Na abordagem de troca de mensagens, os dados devem ser enviados e recebidos explicitamente através de funções chamadas no código. A transferência de dados entre os processos demanda cooperação entre o emissor e o receptor, ou seja, uma mensagem de envio deve ter uma mensagem de recebimento associada. Assim, o programador é responsável por identificar os pontos de paralelismo.

Conforme apresentado na Seção 5.1.2, o comportamento do algoritmo de estimativa dos parâmetros de Entropia Máxima implementado no *openModeller* é essencialmente sequencial. No entanto, a cada iteração o algoritmo efetua o mesmo cálculo para todas as variáveis ambientais e escolhe uma delas para atualizar o peso. Desta forma, a paralelização foi implementada na maximização da função $F_j(\lambda_j, \alpha)$ (5.4).

O processo mestre é responsável por determinar de qual camada j cada processo escravo irá calcular a equação (5.4). Cada processo escravo retorna para o processo mestre os valores (j, α), em que α é o valor a ser somado ao peso da variável j, caso

esta seja escolhida para atualização. Quando o processo mestre recebe um resultado, ele verifica se o valor de α é o maior. Ao final de cada iteração, a variável com maior valor de α tem seu peso atualizado. O Algoritmo 5.5 apresenta a ideia geral do procedimento paralelo de Entropia Máxima.

O processo mestre distribui as tarefas para os processos escravos e recebe o resultado processado de cada um deles. Se o número de variáveis ambientais em um determinado experimento é maior do que o número de processos, quando o processo mestre recebe um resultado, ele envia um novo trabalho para o processo escravo do qual ele recebeu o resultado. Este procedimento é repetido até que todas as tarefas tenham sido concluídas.

5.3.2 Experimentos

Para verificar se o algoritmo paralelo de Entropia Máxima implementado obteve melhores resultados em termos de tempo de execução que a versão sequencial, foram realizados alguns experimentos. Os tempos de execução da versão paralela e da versão sequencial do algoritmo de Entropia Máxima foram medidos e comparados. Os experimentos, tanto da versão sequencial quanto da versão paralela, foram realizados sob a mesma arquitetura e dados de entrada.

Os dados, tanto das espécies quanto das variáveis ambientais, utilizados nos experimentos foram os mesmos usados na validação do algoritmo adaptativo de Entropia Máxima, descritos na Seção 5.2.2.1. Por isso, a descrição dos dados utilizados na avaliação do desempenho do algoritmo paralelo de Entropia Máxima foi omitida nesta seção.

5.3.2.1 Metodologia

A métrica de comparação da versão paralela com a versão sequencial do algoritmo de Entropia Máxima foi o tempo de execução. Duas medidas foram coletadas: o tempo total de execução da aplicação e o tempo de execução do algoritmo para estimar os parâmetros de Entropia Máxima. As execuções foram realizadas no *cluster* do projeto *openModeller*.

Algoritmo 5.5 Algoritmo Paralelo de Entropia Máxima.

Entrada: mesma do Algoritmo 5.2;

Saída: mesma do Algoritmo 5.2;

Inicializar $\lambda_j = 0$;

Normalizar f_j para o intervalo $[0,1]$;

for $t = 1$ **to** $t =$ iterações ou até convergir **do**

$j = 0$;

if Processo Mestre **then**

for $k = 1$ **to** $k <$ número de processadores **and** $k \leq m$ **do**

Envia o índice j para o processador k;

$j = j++$;

end for

while $j < m$ **do**

Recebe o resultado do melhor α calculado por um processo escravo;

Recebe, do mesmo processo escravo, o índice *ind* da *feature* correspondente ao α calculado;

Envia outro índice j para o mesmo processo escravo;

Armazena o melhor α até o momento, bem como o índice da sua *feature*;

$j = j++$;

end while

Recebe todos os resultados dos escravos e escolhe o melhor α;

else

Recebe um índice *ind* do processo mestre;

Calcula o melhor α para a f_{ind};

Envia, para o processo mestre, o melhor α calculado;

Envia, para o processo mestre, o índice *ind* da *feature* correspondente ao α calculado;

end if

end for

O *cluster* possui um sistema *SGI Altix XE 1300* composto por um nó de entrada *Altix XE 210* com dois processadores *Xeon quad Core* de 2 GHz, 8 GB de RAM, disco

rígido de 500 GB, 24-*port InfiniBand switch*, 24-*port Gigabit ethernet switch*, *SGI Propack 5*, SUSE Linux 10, e 10 nós *Altix XE 310*, cada um com dois processadores *Xeon quad Core* de 2 GHz, 8 GB de RAM e disco rígido de 250 GB, totalizando 80 núcleos.

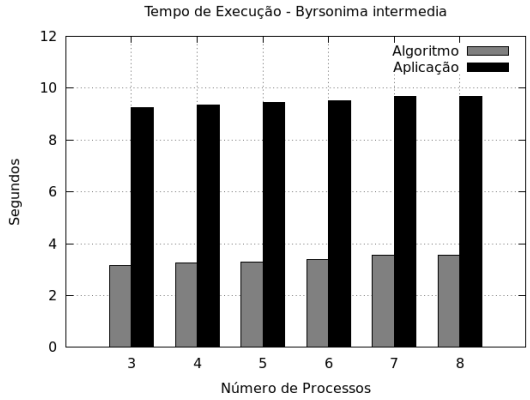
O tempo total de execução da aplicação foi medido com o comando *time*, disponível no Linux. O tempo de execução do algoritmo de estimativa dos parâmetros de Entropia Máxima foi medido por instrumentação de código. A versão sequencial do algoritmo foi executada 10 vezes para cada espécie e os valores considerados foram as médias das duas medidas. Na versão paralela do algoritmo, o número de processos variou de 3 a 8. Nesta versão, para cada espécie, 10 modelos foram estimados para cada número de processos e as médias dos tempos de execução foram consideradas.

O número de processos foi limitado superiormente de acordo com o número de variáveis, assim um processo executou como mestre e sete processos executaram como escravos, um para cada variável. Na abordagem mestre-escravo, a execução com dois processos se torna sequencial, pois apenas um dos processos atua como escravo. Além disso, é necessário um tempo adicional para as trocas de mensagem entre o processo mestre e o processo escravo. Por isso, o número de processos foi limitado inferiormente por três.

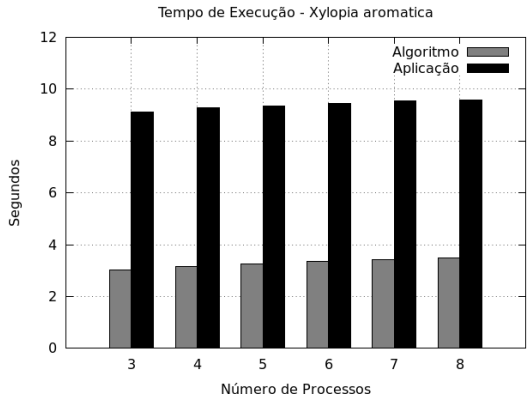
5.3.3 Resultados e Discussão

Na versão sequencial, a média do tempo total de execução da aplicação foi $11,18$ segundos para a modelagem da espécie *Xylopia aromatica* e $11,17$ segundos para a modelagem da espécie *Byrsonima intermedia*. O tempo de execução do algoritmo de estimativa dos parâmetros foi $5,03$ segundos para a modelagem da primeira espécie e $5,02$ segundos para a segunda. A Figura 5.6 mostra os tempos médios de execução da versão paralela para a modelagem das espécies *Xylopia aromatica* e *Byrsonima intermedia*. As colunas maiores representam o tempo de execução total da aplicação e as colunas menores representam o tempo de execução do algoritmo de estimativa dos parâmetros de Entropia Máxima.

O melhor tempo de execução foi alcançado quando o algoritmo foi executado com 3 processos, em ambos os casos. Para a espécie *Xylopia aromatica*, os melhores tempos foram $3,03$ e $9,13$ segundos. Para a espécie *Byrsonima intermedia*, os melhores tempos foram $3,15$ e $9,25$ segundos. O aumento no número de processos não reduziu o tempo de execução. Ao contrário, o tempo de execução aumentou à medida que o número de processos foi incrementado, embora a variação no tempo tenha sido muito pequena.



(a)



(b)

Figura 5.6: Tempo de execução do algoritmo paralelo de Entropia Máxima com diferentes números de processos para as espécies (a) *Byrsonima intermedia* e (b) *Xylopia aromatica* (Fonte: autora).

Apesar de o tempo de execução ter aumentado quando o número de processos foi incrementado, é importante ressaltar que houve um aumento de desempenho considerável da versão paralela em relação à versão sequencial, levando em consideração apenas o tempo de execução. O aumento de desempenho na modelagem da espécie *Xylopia aromatica* foi de aproximadamente 40% para o algoritmo de estimativa de parâmetros e cerca de 18% para a aplicação completa. Similarmente, o aumento de desempenho na modelagem da espécie *Byrsonima intermedia* foi de aproximadamente 37% para o algoritmo de estimativa de parâmetros e cerca de 17% para a aplicação completa.

Analisando os resultados obtidos, é possível observar que o procedimento paralelizado teve uma contribuição significativa para o aumento do desempenho do algoritmo de estimativa de parâmetros de Entropia Máxima, com relação ao tempo de execução. No entanto, o aumento de desempenho da aplicação completa não ocorreu na mesma proporção, indicando que outros procedimentos também podem ser paralelizados.

5.4 Parâmetro de Regularização

Os algoritmos de modelagem de distribuição de espécies requerem o ajuste dos valores dos seus parâmetros para que possam ser usados como valores *default*. O ajuste dos parâmetros pode levar um longo período de tempo e os valores *default* sempre são validados com um grande número de experimentos. No algoritmo de Entropia Máxima, o parâmetro de regularização é usado para prevenir o superajustamento do modelo, ou seja, esse parâmetro ajuda a evitar que o modelo perca sua habilidade de generalização.

A capacidade de generalização está diretamente ligada à habilidade de aprendizagem do algoritmo, por isso o superajustamento pode afetar severamente o seu desempenho. Assim, o objetivo desta etapa do trabalho foi examinar a relação entre o parâmetro de regularização e o tamanho das amostras de dados de ocorrência das espécies. Além disso, duas hipóteses foram analisadas: a remoção do parâmetro de regularização do algoritmo de Entropia Máxima e a substituição deste parâmetro por uma abordagem adaptativa, a mesma apresentada na Seção 5.2.

5.4.1 Experimentos

Segundo Phillips e Dudík (2008), a melhor escolha para os valores do parâmetro de regularização depende da classe de *features* e da quantidade de amostras de ocorrências das espécies. Apesar de a seleção dos melhores valores para β_j ainda ser uma área que requer muitos estudos, sabe-se que estes valores podem interferir seriamente no desempenho do algoritmo. Além disso, eles podem levar um longo período de tempo para serem ajustados. Por esse motivo, uma análise de diferentes valores para β_j com várias espécies foi realizada.

5.4.1.1 Dados Utilizados

Um conjunto de dados com espécies do gênero *Krameria* foi utilizado para avaliar o parâmetro de regularização. Tanto as espécies quanto as variáveis ambientais foram selecionadas por uma bióloga. O gênero *Krameria* da família *Krameriaceae* apresenta 18 espécies com distribuição geográfica abrangendo a região Neotrópica. Existem duas características peculiares deste gênero (VOGEL, 1974; SIMPSON, 1989):

1) as espécies do gênero *Krameria* são hemiparasitas - suas folhas são verdes e capazes de realizar sua própria fotossíntese, porém elas precisam de seus hospedeiros (outras plantas não específicas) para crescer;

2) suas flores apresentam glândulas especiais que produzem óleo floral - este óleo atrai abelhas, do gênero *Centris* da família *Apidae*, que polinizam as flores.

As espécies do gênero *Krameria* norte americanas ocorrem em desertos no México e nos Estados Unidos. As espécies do gênero *Krameria* sul americanas ocorrem em regiões sazonalmente secas (principalmente no cerrado e na caatinga brasileira) e em altitudes elevadas perto dos Andes (SIMPSON, 1989). O centro da diversidade das espécies está localizado no México e apresenta 11 espécies; a diversidade secundária está localizada no Brasil, com 5 espécies (SIMPSON, 1989).

Os pontos de ocorrência das 18 espécies do gênero *Krameria* foram obtidas de portais sobre biodiversidade disponíveis na Internet, principalmente no *Global Biodiversity Information Facility* (GBIF, 2010), *Southwest Environmental Information Network* (SEINET, 2010), *Comisión Nacional para el Conocimiento y uso de la Biodiversidad* (CONABIO, 2010) e no *speciesLink* (SPECIESLINK, 2010). Os trabalhos de Simpson (1989), Simpson et al. (2004) também foram usados. As coordenadas geográficas obtidas foram verificadas, convertidas e corrigidas, quando necessário, usando as ferramentas disponíveis no site do *speciesLink*. O conjunto de dados final contém 2742 pontos de ocorrência. A Tabela 5.3 apresenta os nomes das espécies utilizadas e a quantidade de pontos na amostra. A Figura 5.7 mostra a localização dos registros de ocorrência das espécies do gênero *Krameria*.

Tabela 5.3: Espécies utilizadas e quantidade de pontos na amostra.

Nome da espécie	Quantidade de pontos
K. paucifolia	4
K. bahiana	14
K. sonorae	19
K. cistoidea	19
K. spartioides	27
K. grandiflora	35
K. argentea	45
K. secundiflora	66
K. ramosissima	66
K. lappacea	72
K. revoluta	77
K. pauciflora	80
K. cytisoides	134
K. ixine	162
K. tomentosa	163
K. lanceolata	465
K. grayi	511
K. erecta	755

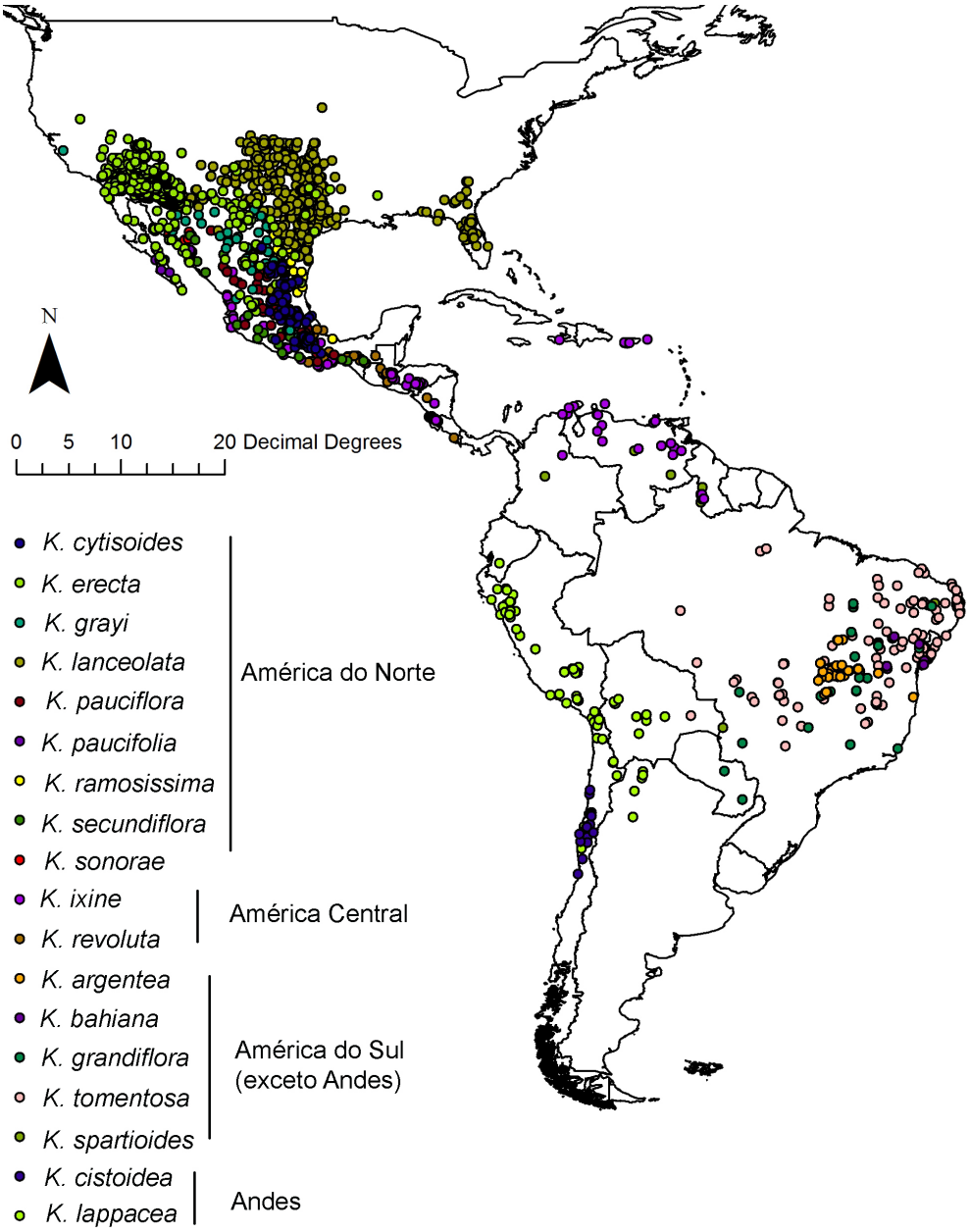


Figura 5.7: Pontos de ocorrência das espécies do gênero *Krameria*.

Para compor o conjunto de dados, foram selecionadas 37 variáveis ambientais com resolução espacial de 5 arc-min. Como os pontos de ocorrência pertenciam à região das Américas, os valores das variáveis ambientais pertenciam à mesma região. As variáveis usadas, todas fornecidas pelo *WorldClim - Global Climate Data*, foram:

- média da temperatura máxima para os doze meses do ano;
- média da temperatura mínima para os doze meses do ano;
- média da precipitação para os doze meses do ano; e
- altitude.

5.4.1.2 Metodologia

A arquitetura do computador usado nos experimentos foi a mesma utilizada na validação do algoritmo adaptativo, descrita na Seção 5.2.2.2. Uma vez que o objetivo desta parte do trabalho foi avaliar a influência da escolha do parâmetro de regularização do algoritmo de Entropia Máxima no processo de aprendizagem dos modelos, as medidas coletadas foram precisão, AUC e número de iterações.

Os valores para β_j foram variados de $0,00001$ a $1,0$ considerando apenas a variação do primeiro número diferente de zero em cada ordem de magnitude. A remoção do parâmetro de regularização também foi avaliada para o mesmo conjunto de espécies e uma abordagem adaptativa foi considerada como possível substituta deste parâmetro.

5.4.2 Resultados e Discussão

A Tabela 5.4 mostra os melhores resultados obtidos nos experimentos realizados para ajustar o parâmetro de regularização. A primeira coluna contém o nome de cada espécie. A segunda coluna contém o melhor valor do parâmetro de regularização para cada espécie considerando as estatísticas coletadas. As outras colunas contêm os melhores valores de cada medida considerada na avaliação do parâmetro de regularização.

Tabela 5.4: Melhores resultados do parâmetro de regularização.

Espécie	Parâmetro de regularização	Precisão (%)	AUC	Iterações
K. paucifolia	0,07	100	1,0	3752
K. bahiana	0,9	100	0,88	1036
K. sonorae	0,9	100	0,94	1250
K. cistoidea	0,05	100	1,0	9632
K. spartioides	0,9	100	0,93	1369
K. grandiflora	0,9	100	0,89	1408
K. argentea	1,0	100	0,93	1362
K. secundiflora	1,0	100	0,91	1257
K. ramosissima	0,9	100	0,98	1357
K. revoluta	1,0	89,61	0,95	1804
K. pauciflora	1,0	86,25	0,80	392
K. cytisoides	1,0	97,01	0,93	684
K. ixine	1,0	79,01	0,97	1816
K. tomentosa	0,07	68,71	0,95	5542
K. lanceolata	0,7	96,56	0,96	1614
K. grayi	0,9	89,04	0,98	1863
K. erecta	1,0	86,49	0,97	1978

A escolha do melhor valor para o parâmetro de regularização foi baseado nos melhores valores de precisão, AUC e número de iterações, respectivamente. A espécie *K.*

lappacea foi descartada do conjunto de dados porque a sua AUC foi menor que 0,75 e apenas os valores de AUC acima deste valor são considerados úteis (ELITH, 2002) *apud* (PHILLIPS; DUDÍK, 2008).

Os resultados apresentados na Tabela 5.4 indicam que o parâmetro de regularização não depende do número de amostras, nem do número de iterações no treinamento. Além disso, as espécies com o mesmo número de amostras ou com um número de amostras similar, se ajustaram melhor com diferentes valores para o parâmetro de regularização (diferentes em ordem de magnitude), tais como *K. cistoidea* e *K. sonorae*, *K. ixine* e *K. tomentosa*. Este comportamento reforça a dificuldade em ajustar o parâmetro de regularização.

Como uma alternativa, o parâmetro de regularização foi removido do algoritmo e o algoritmo adaptativo de Entropia Máxima foi considerado como um possível substituto para este parâmetro. Os melhores resultados são apresentados na Tabela 5.5.

Tabela 5.5: Melhores resultados do algoritmo de Entropia Máxima sem o parâmetro de regularização e com uma abordagem adaptativa.

Espécie	Sem regularização			Abordagem adaptativa		
	Precisão (%)	AUC	Iterações	Precisão (%)	AUC	Iterações
K. paucifolia	100	1,0	9007	100	1,0	5421
K. bahiana	78,57	0,97	2588	78,57	0,97	2410
K. sonorae	68,42	0,99	4290	68,42	0,99	3208
K. cistoidea	94,74	1,0	10000	94,74	1,0	10000
K. spartioides	62,96	0,97	1262	66,67	0,97	1408
K. grandiflora	54,29	0,94	3628	57,14	0,95	3341
K. argentea	37,78	0,94	200	37,78	0,94	213
K. secundiflora	27,66	0,74	49	27,66	0,74	54
K. ramosissima	96,97	0,99	1762	96,97	0,99	1854
K. revoluta	55,84	0,95	377	63,64	0,96	446
K. pauciflora	0,00	0,43	17	0,00	0,45	18
K. cytisoides	0,75	0,59	23	0,75	0,56	26
K. ixine	59,26	0,97	484	59,26	0,97	502
K. tomentosa	54,60	0,95	4055	57,67	0,94	4042
K. lanceolata	86,88	0,95	903	87,96	0,96	972
K. grayi	74,17	0,98	3155	74,17	0,98	2738
K. erecta	62,12	0,97	904	61,46	0,97	896

Os resultados apresentados na Tabela 5.5 mostram que a remoção do parâmetro de regularização causa um declínio nas estatísticas coletadas. A abordagem adaptativa apresentou quase os mesmos resultados que o algoritmo de Entropia Máxima sem o parâmetro de regularização. Portanto, a abordagem adaptativa não pode ser usada como substituta do parâmetro de regularização, embora seja bastante útil quando utilizada em conjunto com este parâmetro, conforme apresentado na Seção 5.2. Além disso, o parâmetro de regularização mostrou-se fundamental para o algoritmo.

A Figura 5.8 mostra um exemplo de um modelo estimado. As cores quentes representam condições mais adequadas para a espécie.

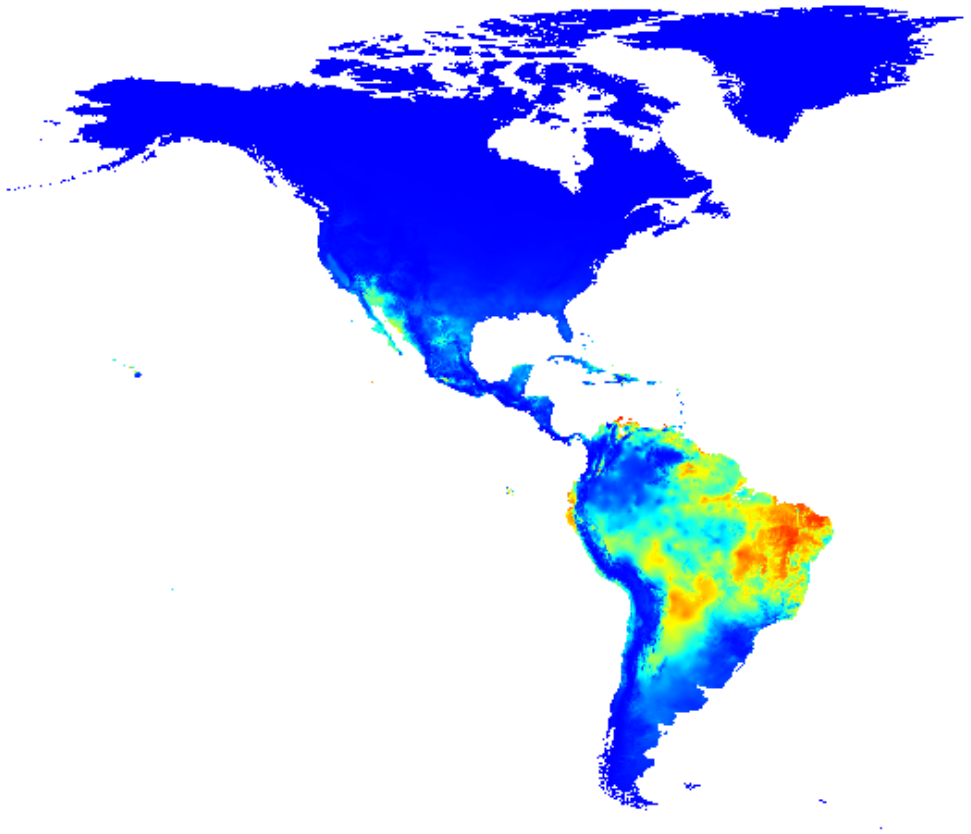


Figura 5.8: Melhor modelo para a espécie *K. tomentosa* (Fonte: autora).

5.5 MDL para Seleção de Variáveis

De acordo com a Seção 3.2 do Capítulo 3, o princípio MDL têm algumas propriedades interessantes que poderiam ser úteis na modelagem de distribuição de espécies. Dentre essas propriedades, destacam-se a sua habilidade de automaticamente evitar o superajustamento na aprendizagem dos parâmetros de um modelo e a sua capacidade de selecionar modelos sem a necessidade de exemplos negativos.

Até aqui, o termo modelo foi tratado como uma representação da distribuição de espécies em uma determinada região. No entanto, cabe lembrar que modelo é uma representação de comportamento ou de características de um processo, não necessariamente de uma distribuição de espécies, conforme visto no Capítulo 2. Assim, o objetivo desta seção é mostrar como o princípio MDL pode ser usado na seleção de variáveis. O princípio MDL com histogramas já foi usado com sucesso em estimativas de densidade de probabilidade (KONTKANEN; MYLLYMÄKI, 2007; CHAPEAU-BLONDEAU; ROUSSEAU, 2009).

Os histogramas são estratégias conceitualmente simples, mas capazes de criar modelos com propriedades complexas (KONTKANEN; MYLLYMÄKI, 2007). Geralmente, os histogramas com escaninhos de larguras iguais, também conhecidos como histogramas regulares, são usados para estimativa de densidade. Nesses casos, o objetivo dos métodos de construção de histogramas é determinar o número ótimo de escaninhos (CHAPEAU-BLONDEAU; ROUSSEAU, 2009). Todavia, existe um desperdício na representação do modelo quando os dados não estão distribuídos uniformemente. Isso acontece porque são necessários muitos escaninhos para representar os detalhes da parte dos dados com alta densidade. Desta forma, também haverão muitos escaninhos, sem necessidade, na parte dos dados com baixa densidade (KONTKANEN; MYLLYMÄKI, 2007).

Os histogramas irregulares, isto é, histogramas com larguras variáveis dos escaninhos, podem ser usados para evitar o desperdício causado pelos histogramas regulares. Nesse caso, além do número de escaninhos, o método de construção dos histogramas necessita encontrar a melhor largura para cada um, ou seja, o melhor conjunto de pontos de corte. Isto naturalmente dificulta o problema. No entanto, os conjuntos de pontos de corte podem ser tratados como modelos e o princípio MDL pode ser útil para selecionar o melhor modelo de acordo com a complexidade estocástica de cada um.

5.5.1 Estimativa da Densidade das Variáveis por Histogramas com MDL

A ideia geral desta etapa do trabalho foi calcular a distribuição NML (3.20) de cada variável para usar a complexidade estocástica como uma métrica de classificação (RODRIGUES et al., 2010c; RODRIGUES et al., 2011b). Assim, considere um conjunto de n valores $f_j^n = (f_{j1}, \ldots, f_{jn})$ para uma determinada variável f_j em um intervalo $[f_{j(min)}, f_{j(max)}]$, em que $f_{j(min)}$ e $f_{j(max)}$ são os valores mínimo e máximo de f_j, respectivamente.

Para simplificar a formulação matemática, os dados são armazenados em ordem crescente com uma precisão finita γ. Isto significa que cada elemento de f_j^n pertence ao conjunto Γ (5.7). Segundo Kontkanen e Myllymäki (2007), o efeito deste parâmetro na complexidade estocástica é uma constante que pode ser ignorada no processo de seleção do modelo.

$$\Gamma = \left\{ f_{j(min)} + t\gamma \mid t = 0, \ldots, \frac{f_{j(max)} - f_{j(min)}}{\gamma} \right\} \tag{5.7}$$

Para construir o histograma, o primeiro passo é escolher o conjunto de pontos de corte que serão considerados. Escolher todos os pontos de corte possíveis é a solução

mais simples nesta etapa, porém pode ser inviável na prática. Outra possibilidade seria escolher o conjunto de valores médios de todos os pares de valores consecutivos nos dados. No entanto, essa escolha não permite escaninhos vazios.

Uma escolha mais adequada é colocar dois pontos de corte entre todos os pares de valores consecutivos nos dados. Para que os escaninhos sejam os menores possíveis e a probabilidade para os dados seja a maior, os pontos de corte devem estar o mais próximo possível dos valores dos dados (KONTKANEN; MYLLYMÄKI, 2007). Desta forma, o conjunto de pontos de corte é

$$\begin{aligned}\mathscr{C} = \quad & \left(\left\{f_{ji} - \tfrac{\gamma}{2} \mid f_{ji} \in f_j^n\right\} \cup \left\{f_{ji} + \tfrac{\gamma}{2} \mid f_{ji} \in f_j^n\right\}\right) \\ & - \left\{f_{j(min)} - \tfrac{\gamma}{2}, f_{j(max)} + \tfrac{\gamma}{2}\right\}. \end{aligned} \tag{5.8}$$

Os pontos extremos são excluídos do conjunto porque estão automaticamente incluídos em todos os conjuntos de pontos de corte.

Uma vez definido o conjunto de pontos de corte, a tarefa é encontrar o melhor subconjunto C de $\mathscr{C}$ segundo algum critério de otimização. Aqui o critério é a complexidade estocástica (3.23), instanciada para o problema de estimar a densidade das variáveis por histogramas conforme a equação (5.9) e os subconjuntos de $\mathscr{C}$ são considerados modelos.

$$SC(f_j^n \mid C) = \sum_{k=1}^{K} -h_k(log(\gamma \cdot h_k) - log(L_k \cdot n)) + log\, \mathscr{R}_n(h_K). \tag{5.9}$$

em que K é o número de escaninhos do histograma, h_k é a quantidade de exemplos dos dados que estão no k-ésimo escaninho, L_k é a largura do k-ésimo escaninho e $\mathscr{R}_n(h_K)$ é a complexidade paramétrica do histograma com K escaninhos.

Segundo Kontkanen e Myllymäki (2007), em algumas aplicações práticas, além da complexidade estocástica, às vezes é necessário codificar de alguma forma o índice do modelo. Isso ocorre porque não existe um código que comprima os dados da melhor forma possível independentemente de quais sejam esses dados (GRÜNWALD, 2005). Portanto, é necessário indexar os modelos de alguma maneira e codificar seu índice, para que este possa ser usado na decodificação.

De acordo com Grünwald (2005), uma codificação baseada na distribuição uniforme é apropriada em muitos casos. No entanto, os conjuntos de pontos de corte de tamanhos diferentes podem produzir densidades com complexidades muito discrepantes. Por isso, o índice do modelo é codificado com uma distribuição uniforme sobre todos os conjuntos de pontos de corte de mesmo tamanho. Assim, supondo que E seja

o tamanho do conjunto de pontos de corte, existem

$$\begin{pmatrix} E \\ K-1 \end{pmatrix} \tag{5.10}$$

maneiras de escolher os pontos de corte para um histograma com K escaninhos, ou seja, a equação (5.10) é o número de combinações de $K-1$ pontos distintos escolhidos entre E pontos diferentes. Desta forma, o tamanho de código necessário para codificar os índices dos modelos de um histograma com K escaninhos é

$$log \begin{pmatrix} E \\ K-1 \end{pmatrix}. \tag{5.11}$$

Dada a necessidade de codificar o índice dos modelos, o critério final de comparação dos diferentes conjuntos de pontos de corte é

$$B(f_j^n \mid E,K,C) = SC(f_j^n \mid C) + log \begin{pmatrix} E \\ K-1 \end{pmatrix}. \tag{5.12}$$

Como existe uma quantidade exponencial de conjuntos de pontos de corte possíveis, uma busca exaustiva não é viável. No entanto, o conjunto ótimo pode ser encontrado recursivamente via programação dinâmica, ou seja, tabulando soluções parciais para o problema e usando essas soluções parciais para resolver o problema completo.

5.5.2 Experimentos

O objetivo dos experimentos realizados nesta etapa do trabalho foi mostrar a viabilidade da aplicação do princípio MDL na seleção de modelos. Como um histograma é uma representação gráfica da distribuição de frequência de uma variável, esta solução deve ser modificada se a intenção for selecionar um modelo de distribuição de espécies (RODRIGUES et al., 2010d). Assim, o princípio MDL foi usado para classificar as variáveis ambientais de acordo com uma determinada espécie. Para esta classificação a métrica usada foi a complexidade estocástica, conforme discutido na Seção 5.5.1.

5.5.2.1 Dados Utilizados

Nesta etapa do trabalho, apenas a espécie *Xylopia aromatica* foi utilizada nos experimentos, já descrita na Seção 5.2.2.1. No entanto, foram utilizados 65 pontos de ocorrência, dos quais 45 foram separados para treinamento e 20 para teste. Como o objetivo desta etapa do trabalho era classificar as variáveis ambientais, um conjunto maior de variáveis foi utilizado. Desta forma, foram utilizadas 55 variáveis, todas

com resolução espacial de 30 arc-second e os modelos foram projetados no Brasil. As variáveis selecionadas, todas fornecidas pelo WorldClim, foram:

- Bio_1 – temperatura média anual;
- Bio_2 – escala média diurna (média mensal da diferença entre temperatura máxima e temperatura mínima);
- Bio_3 – isotermalidade (Bio_2 / Bio_7 $*100$);
- Bio_4 – sazonalidade da temperatura (desvio padrão $*100$);
- Bio_5 – temperatura máxima do mês mais quente;
- Bio_6 – temperatura mínima do mês mais frio;
- Bio_7 – escala da temperatura anual (Bio_5 $-$ Bio_6);
- Bio_8 – temperatura média do trimestre mais úmido;
- Bio_9 – temperatura média do trimestre mais seco;
- Bio_10 – temperatura média do trimestre mais quente;
- Bio_11 – temperatura média do trimestre mais frio;
- Bio_12 – precipitação anual;
- Bio_13 – precipitação do mês mais quente;
- Bio_14 – precipitação do mês mais seco;
- Bio_15 – sazonalidade da precipitação (coeficiente de variação);
- Bio_16 – precipitação do trimestre mais úmido;
- Bio_17 – precipitação do trimestre mais seco;
- Bio_18 – precipitação do trimestre mais quente;
- Bio_19 – precipitação do trimestre mais frio;
- média da temperatura mínima para os doze meses do ano (Tmin_1 até Tmin_12);
- média da temperatura máxima para os doze meses do ano (Tmax_1 até Tmax_12);
- precipitação média para os doze meses do ano (Prec_1 até Prec_12).

Os valores das camadas Bio_4 e Bio_12 foram divididos por 10 para ficarem compatíveis com o limite da capacidade de computação do algoritmo que calcula a complexidade estocástica.

5.5.2.2 Metodologia

A arquitetura do computador usado nos experimentos foi a mesma utilizada na validação do algoritmo adaptativo, descrita na Seção 5.2.2.2. O primeiro passo deste conjunto de experimentos foi calcular a complexidade estocástica de cada variável ambiental conforme descrito na Seção 5.5.1. Para isso, foi utilizado um código fornecido por Petri Kontkanen. Dos 65 registros disponíveis, somente os valores das camadas ambientais correspondentes aos 45 pontos separados para treinamento foram utilizados para calcular a complexidade estocástica. A Figura 5.9 mostra um exemplo de histograma gerado durante o cálculo da complexidade estocástica.

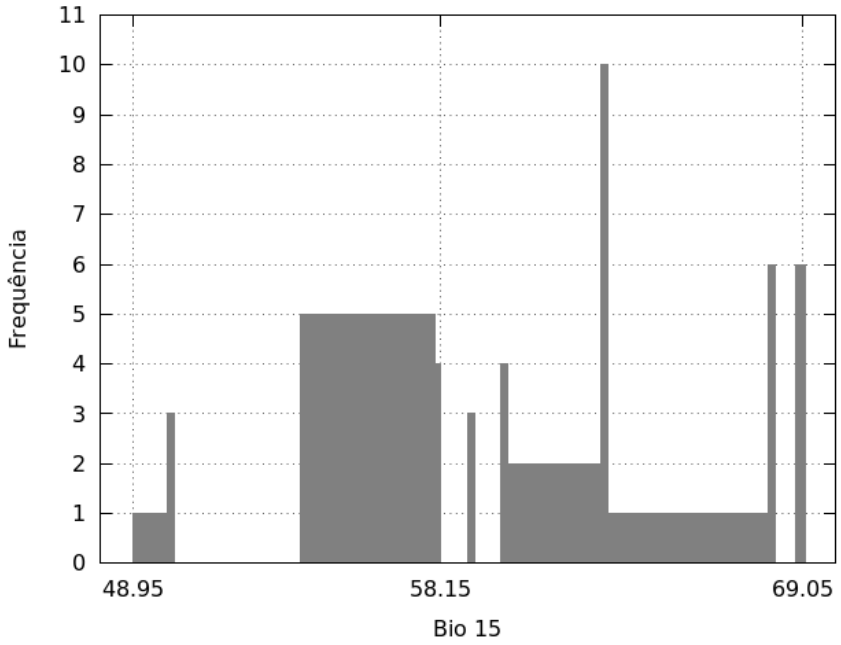


Figura 5.9: Histograma gerado para a variável Bio_15 (Fonte: autora).

Em seguida, as variáveis foram dispostas em ordem crescente de complexidade estocástica e divididas em três grupos. A escolha da divisão do conjunto de variáveis em três grupos foi aleatória. O objetivo desta divisão foi mostrar que a complexidade estocástica pode ser usada como um indicador de importância das variáveis. Desta forma, a primeira hipótese levantada foi a de que o grupo com menor complexidade pode estimar modelos melhores. Portanto, são necessários mais experimentos para definir a quantidade adequada de variáveis na estimativa da distribuição de espécies.

Após a divisão das camadas ambientais em grupos, cada conjunto foi usado para estimar a distribuição da espécie *Xylopia aromatica* com o algoritmo de Entropia Má-

xima disponível no *openModeller*. A métrica utilizada para avaliar a qualidade das distribuições estimadas foi a AUC.

5.5.3 Resultados e Discussão

Após o cálculo da complexidade estocástica, as variáveis foram agrupadas conforme a ordem crescente do valor calculado. Assim, os três grupos formados foram:

1) Bio_3, Bio_15, Prec_7, Prec_5, Prec_4, Prec_9, Bio_14, Prec_8, Bio_2, Tmin_1, Tmin_3, Tmin_2, Tmin_4, Tmin_12, Bio_8, Bio_10, Prec_6 e Bio_12;

2) Tmin_5, Tmin_6, Prec_10, Tmax_2, Bio_6, Tmin_7, Tmax_12, Tmin_11, Tmin_10, Tmax_1, Bio_7, Tmax_3, Prec_3, Bio_1, Tmin_9, Bio_11, Prec_12 e Tmin_8;

3) Tmax_4, Prec_2, Bio_9, Tmax_5, Prec_11, Tmax_6, Tmax_11, Tmax_10, Tmax_7, Bio_13, Prec_1, Bio_19, Tmax_8, Tmax_9, Bio_17, Bio_4, Bio_5, Bio_18 e Bio_16.

A Tabela 5.6 mostra os valores de AUC obtidos no treinamento e no teste com cada um dos grupos de variáveis ambientais apresentados acima.

Tabela 5.6: Resultados da modelagem usando cada um dos três subconjuntos de variáveis ambientais.

Grupo	AUC - treinamento	AUC - teste
1	0.91	0.92
2	0.80	0.90
3	0.28	0.27

Conforme pode ser observado na Tabela 5.6, os melhores resultados foram obtidos com o primeiro conjunto de camadas ambientais. O segundo grupo também alcançou bons resultados, mas declinou em relação ao primeiro. O algoritmo não conseguiu associar o último conjunto de camadas aos pontos de ocorrência apresentados, ou seja, não foi capaz de aprender e estimar um bom modelo. Por isso, o modelo também não foi capaz de generalizar.

Como o objetivo era utilizar o princípio MDL para determinar a importância das variáveis, pode-se observar que os resultados foram satisfatórios. No entanto, ainda são necessárias novas análises para determinar não apenas a importância de cada variável, mas a melhor quantidade de camadas para uma determinada espécie.

5.6 Epítome

O propósito deste capítulo foi explanar as contribuições deste trabalho, apresentando as diversas etapas necessárias para o cumprimento dos objetivos estabelecidos e os resultados obtidos. Como o trabalho é interdisciplinar, procurou-se abordar cada tema da forma mais objetiva possível.

O princípio da Entropia Máxima foi analisado primeiramente para a construção de uma árvore de decisão. Todo o raciocínio empregado na geração da árvore foi apresentado. O conhecimento adquirido foi essencial para a compreensão do princípio da Entropia Máxima. Para disponibilizar de forma rápida um algoritmo de Entropia Máxima na ferramenta *openModeller*, foi inserida uma biblioteca com dois algoritmos de atualização de pesos. Posteriormente, um algoritmo específico para a ferramenta foi implementado.

Um algoritmo adaptativo de Entropia Máxima para a modelagem de distribuição de espécies foi apresentado. O objetivo desta etapa foi reduzir a quantidade de *features* usadas na estimativa dos modelos. O algoritmo adaptativo de Entropia Máxima foi projetado a partir do algoritmo convencional de Entropia Máxima desenvolvido previamente. A abordagem proposta utiliza uma estratégia gulosa para encontrar o melhor conjunto de *features* a cada iteração. Os resultados alcançados com os experimentos de validação foram superiores aos do algoritmo convencional.

Além do algoritmo adaptativo, um algoritmo paralelo de Entropia Máxima foi apresentado. A meta era desenvolver um algoritmo mais rápido que o algoritmo convencional. No desenvolvimento do algoritmo paralelo foi utilizada a biblioteca MPI. A abordagem mestre-escravo foi utilizada com escalonamento dinâmico das tarefas. Os experimentos realizados na avaliação do algoritmo paralelo mostraram desempenho superior deste algoritmo em relação ao convencional, levando em consideração o tempo de execução total da aplicação e da parte paralelizada.

Outra etapa do trabalho foi a análise do parâmetro de regularização do algoritmo de Entropia Máxima. Nesta análise, foram testados diversos valores para o parâmetro de regularização. A hipótese inicialmente levantada, de que o valor deste parâmetro dependia da quantidade de exemplos nos dados, foi refutada pelos resultados obtidos nos experimentos. Também foram analisadas a remoção deste parâmetro e a possível substituição dele pela abordagem adaptativa proposta anteriormente. Ambas as hipóteses foram descartadas após a análise experimental.

A última etapa deste trabalho descrita neste capítulo foi o uso do princípio MDL na seleção de variáveis ambientais. A estratégia proposta foi testada com o algoritmo

de Entropia Máxima desenvolvido previamente. No entanto, ela poderá ser aplicada a qualquer outro algoritmo disponível no *openModeller*. Os resultados alcançados mostram que esta é uma promissora área de pesquisa.

6 Considerações Finais

Neste capítulo, as contribuições da pesquisa são expostas na Seção 6.1. Todos os resultados apresentados e discutidos no Capítulo 5 foram avaliados pela comunidade científica e publicados. Assim, a Seção 6.2 lista os trabalhos publicados. Na Seção 6.3 são apresentadas algumas sugestões de trabalhos futuros. Por fim, as conclusões do trabalho desenvolvido são apresentadas na Seção 6.4.

6.1 Contribuições da Pesquisa

O desenvolvimento deste trabalho gerou contribuições para diferentes áreas do conhecimento. A disponibilização de um algoritmo de Entropia Máxima na ferramenta *openModeller* foi a primeira contribuição produzida. Conforme apresentado na Seção 1.2 do Capítulo 1, as ferramentas estatísticas tradicionais não disponibilizam pacotes com o algoritmo de Entropia Máxima.

Para a comunidade científica interessada em modelagem de distribuição de espécies, a disponibilização de um algoritmo de Entropia Máxima na ferramenta *openModeller* traz outros benefícios. Como a ferramenta *openModeller* disponibiliza diversos algoritmos de modelagem de distribuição de espécies, é possível realizar vários experimentos com os mesmos dados de entrada, o que não ocorre com outras ferramentas. Além disso, existe um conjunto de estatísticas comum a todos os algoritmos, o que facilita a comparação dos resultados obtidos por diferentes abordagens.

A formalização de uma estratégia adaptativa para o algoritmo de Entropia Máxima contribuiu com a expansão da teoria computacional relacionada aos formalismos adaptativos e aprendizagem de máquina. Um novo princípio matemático foi utilizado como dispositivo subjacente, o princípio da Entropia Máxima. No entanto, a formalização proposta é geral o suficiente para beneficiar qualquer algoritmo de modelagem de distribuição de espécies baseado em otimização.

O desenvolvimento de um algoritmo adaptativo de Entropia Máxima baseado na formalização proposta é uma contribuição prática, tanto para a área de modelagem de distribuição de espécies, quanto para as áreas de tecnologia adaptativa e aprendiza-

gem de máquina. Os experimentos realizados mostraram a superioridade do algoritmo adaptativo de Entropia Máxima quando comparado ao algoritmo convencional.

A estratégia de paralelização definida para o algoritmo de Entropia Máxima, embora bastante utilizada na área de programação paralela e distribuída, apresenta uma contribuição para a modelagem de distribuição de espécies e para a área de aprendizagem de máquinas. Muitos algoritmos de modelagem de distribuição de espécies requerem um tempo excessivo de execução. O mesmo ocorre com algoritmos de aprendizagem cuja complexidade da função objetivo depende do tamanho do conjunto de entrada.

O desenvolvimento do algoritmo paralelo de Entropia Máxima constitui a aplicação da estratégia de paralelização definida. A disponibilização de algoritmos paralelos de modelagem de distribuição de espécies permite que os usuários obtenham respostas mais rápidas. Desta forma, as estratégias de conservação da biodiversidade e de gestão dos recursos naturais podem ser traçadas com mais eficiência. Os resultados obtidos durante os experimentos mostraram a preeminência da versão paralela em relação a versão sequencial do algoritmo de Entropia Máxima.

A análise realizada sobre o parâmetro de regularização é outra contribuição deste trabalho. Embora não tenha sido possível definir valores adequados para o parâmetro de regularização, os resultados da análise deste parâmetro permitiram avançar no conhecimento sobre o seu ajuste. Os valores para este parâmetro são definidos de forma empírica levando em consideração a quantidade de pontos de ocorrência. Desta forma, a contribuição da análise realizada neste trabalho está na conclusão de que esta não é uma boa estratégia de escolha.

Analisar o potencial do algoritmo adaptativo como substituto do parâmetro de regularização foi uma importante contribuição, tanto para a área de aprendizagem de máquina quanto para a de tecnologia adaptativa. Com esta análise, foi possível visualizar as funções disjuntas que a adaptatividade e o parâmetro de regularização exercem no algoritmo de Entropia Máxima. Por isso, os resultados só se mostraram vantajosos quando o algoritmo adaptativo foi usado em conjunto com o parâmetro de regularização.

A modelagem de distribuição de espécies constitui uma nova área de aplicação do princípio da Descrição com Comprimento Mínimo (MDL). Esta é uma contribuição para as pesquisas na área de teoria da informação, modelagem de distribuição de espécies e aprendizagem de máquina. As propriedades do princípio MDL não só se mostraram interessantes na seleção de variáveis, como também indicaram uma promissora linha de pesquisa, pois nem todas as qualidades inerentes deste princípio foram

exploradas.

A instanciação do princípio MDL para a seleção de variáveis para a modelagem de distribuição de espécies é algo completamente novo. Assim, esta parte do trabalho beneficiou a etapa de pré-análise do processo de modelagem e propiciou um novo ponto de vista acerca deste princípio.

Além da contribuição inerente da revisão bibliográfica de cada um dos temas envolvidos neste trabalho, o próprio texto do Capítulo 3 também é uma cooperação para o avanço da literatura sobre Entropia Máxima e MDL. Isto porque textos redigidos em língua portuguesa sobre Entropia Máxima e principalmente sobre o princípio MDL não são facilmente encontrados.

6.2 Trabalhos Publicados

Cada etapa do trabalho desenvolvido gerou a produção de um artigo ou resumo. Além disso, como o trabalho foi desenvolvido no escopo no projeto *openModeller*, algumas publicações resultaram da colaboração com outros pesquisadores, incluindo um capítulo de livro. A seguir, são apresentadas as publicações provenientes dos resultados alcançados e das cooperações realizadas.

- Rodrigues, E. S. C., Rodrigues, F. A. e Rocha, R. L. A. Autômatos Adaptativos para Emparelhamento de Cadeias. Memórias do WTA'2008: Segundo Workshop de Tecnologia Adaptativa, p. 27–30, 2008. São Paulo, SP, Brasil: Laboratório de Linguagens e Técnicas Adaptativas. ISBN 978–85–86686–46–7.

- Rodrigues, F. A., Rodrigues, E. S. C., Sato, L. M., Midorikawa, E. T., Corrêa, P. L. P. e Saraiva, A. M. Parallelization of the Jackknife Algorithm Applied to a Biodiversity Modeling System. In: Proceedings of the 7th International Information and Telecommunication Technologies Symposium - I2TS, p. 58–65, 2008. Foz do Iguaçu, PR, Brasil: Fundação Barddal de Educação e Cultura. ISBN 978–85–89264–09–9.

- Rodrigues, E. S. C., Rodrigues, F. A. e Rocha, R. L. A. Algoritmo Paralelo de Entropia Máxima Aplicado a Modelagem de Nicho Ecológico. In: I2TS'2008 7th International Information and Telecommunication Technologies Symposium, 2008. Foz do Iguaçu, PR, Brasil.

- Rodrigues, E. S. C., Rodrigues, F. A. e Rocha, R. L. A. Species Distribution Modeling with the Maximum Entropy Algorithm. In: CIC 2008 – 17th International

Conference on Computing, 2008. Mexico City, Mexico.

- Rodrigues, E. S. C., Rodrigues, F. A. e Rocha, R. L. A. Dispositivo Adaptativo na Análise de Modelos de Distribuição de Espécies. Memórias do WTA'2009: Terceiro Workshop de Tecnologia Adaptativa, p. 83–87, 2009. São Paulo, SP, Brasil: Laboratório de Linguagens e Técnicas Adaptativas. ISBN 978–85–86686–51–1.

- Rodrigues, F. A., Avilla, A. O., Rodrigues, E. S. C., Corrêa, P. L. P., Saraiva, A. M. e Rocha, R. L. A. Species Distribution Modeling with Neural Networks. In: e-Biosphere'09 International Conference on Biodiversity Informatics, p. 97, 2009. London.

- Saraiva, A. M., Corrêa, P. L. P., Sato, L. M., Rodrigues, F. A., Santana, F. S., Rodrigues, E. S. C., Stange, R. L., Murakami, E., Giovanni, R., Canhos, D. A. L. e Canhos, V. P. A service-based framework for species distribution modeling. In: e-Biosphere'09 International Conference on Biodiversity Informatics, 2009. London.

- Rodrigues, E. S. C., Rodrigues, F. A., Rocha, R. L. A. e Corrêa, P. L. P. An Adaptive Maximum Entropy Approach for Modeling of Species Distribution. Memórias do WTA'2010: Quarto Workshop de Tecnologia Adaptativa, p. 108–117, 2010. São Paulo, SP, Brasil: Laboratório de Linguagens e Técnicas Adaptativas. ISBN 978-85-86686-56-6.

- Rodrigues, F. A., Rodrigues, E. S. C., Corrêa, P. L. P., Rocha, R. L. A. e Saraiva, A. M. Modelagem da Biodiversidade Utilizando Redes Neurais Artificiais. II Workshop de Computação Aplicada à Gestão do Meio Ambiente e Recursos Naturais (WCAMA). XXX Congresso da Sociedade Brasileira de Computação - Computação Verde: Desafios Científicos e Tecnológicos, p. 585–594, 2010. Belo Horizonte, MG, Brasil.

- Rodrigues, E. S. C., Rodrigues, F. A., Rocha, R. L. A., Corrêa, P. L. P. e Giannini, T. C. Evaluation of different aspects of maximum entropy for niche-based modeling. ISEIS 2010 Ecological Informatics and Ecosystem Conservation, p. 1066–1077, 2010. Beijing, China: Elsevier.

- Rodrigues, E. S. C., Rodrigues, F. A., Rocha, R. L. A. e Corrêa, P. L. P. MDL-based Clustering for Modeling of Species Geographic Distribution. In: ISEI7 7th International Conference on Ecological Informatics, p. 178-179, 2010. Ghent, Belgium.

- Rodrigues, E. S. C., Rodrigues, F. A., Rocha, R. L. A. e Corrêa, P. L. P. Selection of niche-based models with minimum description lenght. In: 1st Conference on Computational Interdisciplinary Sciences (CCIS), 2010. São José dos Campos.

- Rodrigues, F. A., Rodrigues, E. S. C., Corrêa, P. L. P., Rocha, R. L. A. e Saraiva, A. M. Performance Analysis of Machine Learning Algorithms in Biodiversity Modeling. In: ISEI7 7th International Conference on Ecological Informatics, p. 174-175, 2010. Ghent, Belgium.

- Rodrigues, E. S. C., Rodrigues, F. A., Rocha, R. L. A. e Corrêa, P. L. P. Adaptive Approach for a Maximum Entropy Algorithm in Ecological Niche Modeling. Revista IEEE América Latina, v. 9, p. 331-338, 2011.

- Rodrigues, E. S. C., Rodrigues, F. A., Rocha, R. L. A. e Corrêa, P. L. P. Minimum description length principle to select environmental layers in modeling of species geographical distribution. Journal of Computational Interdisciplinary Sciences, v. 2, n. 2, p. 131–137, 2011. doi: 10.6062/jcis.2011.02.02.0040

- Corrêa, P. L. P., Carvalhaes, M. A., Saraiva, A. M., Rodrigues, F. A., Rodrigues, E. S. C. e Rocha, R. L. A. Computational Techniques for Biologic Species Distribution Modeling. In: Hércules Antonio Prado; Alfredo José Barreto Luiz; Homero Chaib Filho. (Org.). Computational Methods for Agricultural Research: Advances and Applications. 1ed. Hershey, PA: IGI Global, v. 1, p. 308-325, 2011.

6.3 Trabalhos Futuros

Durante o desenvolvimento de um trabalho, novas hipóteses surgem à medida que as pesquisas avançam. No entanto, por uma questão de escopo, não é possível explorar todas as ideias novas. Desta forma, algumas sugestões de trabalhos futuros são:

- Paralelizar o algoritmo adaptativo de Entropia Máxima. Conforme especificado na Seção 5.3.2.1 do Capítulo 5, o projeto *openModeller* possui um *cluster* com 80 núcleos. Assim, quanto maior a quantidade de algoritmos paralelos disponíveis na ferramenta *openModeller*, melhor aproveitado será o *cluster*. O desenvolvimento de algoritmos de modelagem de distribuição de espécies de alto desempenho pode tornar mais eficiente o processo de investigação de soluções para problemas relacionados a conservação da biodiversidade e gestão de recursos naturais. Uma vez que, tanto o algoritmo adaptativo quanto o algoritmo paralelo de Entropia Máxima, apresentados neste trabalho, mostraram resultados

superiores ao algoritmo convencional, é razoável pensar que um algoritmo que una as duas abordagens produza resultados ainda melhores.

- Desenvolver um algoritmo paralelo de estimativa de modelos com Entropia Máxima que atualize os pesos de todas as *features* a cada iteração. Atualizar os pesos de todas as *features* em cada iteração pode ser interessante quando a quantidade de *features* é relativamente pequena. No entanto, o desenvolvimento de um algoritmo paralelo que atualize todos os pesos em cada iteração e que seja executado no *cluster* pode ser mais eficiente para qualquer tamanho do conjunto de *features*.

- Desenvolver um método adaptativo para pós-análise dos modelos de distribuição de espécies estimados, conforme proposta já elaborada (RODRIGUES; RODRIGUES; ROCHA, 2009). A ideia geral é que uma tabela de decisão adaptativa seja usada para representar regras que serão utilizadas na análise da qualidade dos modelos estimados. Este método, aliado às estatísticas já disponíveis no *openModeller*, pode auxiliar no processo de tomada de decisão.

- Separar o módulo adaptativo do algoritmo de Entropia Máxima. Tornar o módulo adaptativo independente do algoritmo de Entropia Máxima permitirá que qualquer algoritmo de modelagem se beneficie da variabilidade do conjunto de dados de entrada.

- Analisar o comportamento do parâmetro de regularização no algoritmo de Entropia Máxima com um conjunto maior de espécies. O parâmetro de regularização ajuda a evitar o superajustamento dos modelos. Por isso, é importante conhecer a sua influência na estimativa dos modelos de acordo com alguma característica dos dados. Neste trabalho, foram usadas 18 espécies na análise deste parâmetro, mas todas pertenciam ao mesmo gênero. A proposta é utilizar espécies de gêneros diferentes e definir um método para melhor ajustá-lo.

- Integrar o método de classificação de variáveis baseado em MDL à ferramenta *openModeller*. O experimentos realizados neste trabalho tiveram o objetivo de mostrar a viabilidade do uso do princípio MDL na seleção de variáveis. No entanto, o método proposto ainda não foi incluído no *openModeller*. Esta integração permitiria que o usuário analisasse a importância de cada variável para um determinado conjunto de pontos de ocorrência de uma dada espécie.

- Definir uma métrica para auxiliar o usuário na escolha da quantidade de variáveis ambientais a serem usadas em um experimento. O princípio MDL foi usado para classificar as variáveis ambientais segundo a complexidade estocástica de cada

uma. Todavia, não se tem conhecimento de alguma medida que indique quantas variáveis devem ser usadas.

- Ampliar o uso do princípio MDL na classificação de variáveis para outros algoritmos. Como o método proposto é aplicado na pré-análise, sua usabilidade é independente do algoritmo de modelagem utilizado. Portanto, embora só tenha sido testado com o algoritmo de Entropia Máxima, o método pode ser ampliado para qualquer algoritmo disponível na ferramenta *openModeller*.

- Desenvolver um método paralelo do princípio MDL para seleção de variáveis. O método paralelo pode aumentar o desempenho da pré-análise e permitir o planejamento de experimentos mais apropriados ao problema a ser solucionado.

- Comparar o método de classificação de variáveis baseado em MDL com outros métodos, como o *Jackknife*. O método *Jackknife* gera o mesmo número de modelos que a quantidade de variáveis ambientais para indicar a importância de cada variável. O princípio MDL tem a vantagem de não precisar estimar modelo algum para classificar as variáveis. No entanto, é necessário avaliar o impacto das classificações dos métodos sobre os modelos estimados.

- Desenvolver um algoritmo de modelagem de distribuição de espécies baseado no princípio MDL. Conforme discutido na Seção 5.5.2 do Capítulo 5, um histograma é uma representação gráfica da distribuição de frequência de uma variável. Portanto, para desenvolver um algoritmo de estimativa de modelos baseado no princípio MDL, será necessário modificar a instanciação realizada na classificação de variáveis. Visto que a ideia principal do princípio MDL é comprimir juntos dados que apresentam alguma regularidade, pode-se optar por um método bastante conhecido na aprendizagem não supervisionada, o *clustering*. O objetivo do *clustering* é agrupar dados semelhantes. Desta forma, os dados do mesmo grupo podem ser comprimidos de acordo com algum padrão.

6.4 Conclusões

Melhorar o algoritmo de Entropia Máxima para modelagem de distribuição de espécies foi o principal objetivo deste trabalho. Essa tarefa não é considerada fácil, porque o princípio da Entropia Máxima tende a fornecer soluções ótimas. No entanto, considerando-se os diferentes elementos envolvidos na modelagem de distribuição de espécies, foi possível estabelecer alguns tópicos a serem explorados com a finalidade de atingir a meta do trabalho.

Os temas abordados nesta pesquisa como alternativas de melhorias para o algoritmo de Entropia Máxima foram: programação paralela e distribuída, tecnologia adaptativa e inferência indutiva por compressão de dados. Os temas foram explorados de forma independente, mas o sucesso obtido nos resultados fornecem indícios de que a combinação desses tópicos pode produzir resultados ainda melhores.

A disponibilização de um algoritmo de Entropia Máxima na ferramenta *openModeller* foi a base para o restante da pesquisa. A partir do algoritmo convencional de Entropia Máxima desenvolvido e integrado ao *openModeller*, todos os outros temas foram investigados. Inicialmente, optou-se pela inclusão de uma biblioteca de Entropia Máxima, mas esta logo revelou-se inadequada.

Como o princípio da Entropia Máxima já havia sido pesquisado especificamente para a tarefa de modelagem de distribuição de espécies, foi adotada a mesma lógica do algoritmo implementado no software cuja estratégia de modelagem é exclusivamente este princípio. No entanto, como o código deste software é fechado, muitos detalhes não são conhecidos, pois as referências bibliográficas apresentam apenas a ideia principal.

Mesmo com as dificuldades na fase de implementação do algoritmo de Entropia Máxima, o resultado alcançado foi satisfatório. Um dos obstáculos do algoritmo de Entropia Máxima é o ajuste do parâmetro de regularização. Embora essa questão não tenha sido solucionada, foi possível observar a importância desse parâmetro para o algoritmo. Desta forma, pode-se concluir que este é um fértil campo de pesquisa.

A identificação de características que poderiam ser melhoradas com o uso da tecnologia adaptativa, possibilitou o desenvolvimento de um algoritmo adaptativo de Entropia Máxima. Esse algoritmo apresentou resultados até 40% melhores que o algoritmo convencional. Uma característica muito interessante da abordagem proposta é que a adaptatividade do algoritmo está relacionada com os dados de entrada. Isso significa que o módulo adaptativo poderá ser desacoplado do algoritmo de entopia máxima e, devidamente modificado, poderá ser usado por qualquer outro algoritmo de modelagem baseado em otimização disponível na ferramenta.

Os avanços tecnológicos têm disponibilizado computadores com capacidade de processamento cada vez maior. No entanto, para aproveitar essa capacidade, as ferramentas precisam ser implementadas para usufruir deste benefício. Por isso, foi desenvolvido um algoritmo paralelo de Entropia Máxima. A atualização sequencial de pesos do algoritmo convencional não permite o sumo desempenho da abordagem. Ainda assim, o algoritmo desenvolvido apresentou resultados superiores ao algoritmo convencional em aproximadamente 40%.

A investigação da aplicação do princípio MDL na seleção de variáveis ambientais foi apenas o início de uma promissora área de pesquisa. As características deste princípio e os resultados obtidos nos experimentos apresentam evidências de que o MDL pode trazer grandes benefícios para a modelagem de distribuição de espécies. Além disso, este trabalho abre caminho para que outros métodos fundamentados em teoria da informação também sejam explorados.

Diante do exposto, conclui-se que este trabalho alcançou as metas estabelecidas inicialmente e proporcionou avanços significativos no conhecimento de diversas extensões da ciência, conforme apresentado na Seção 6.1.

Referências

ADAMS, W. M. et al. Biodiversity conservation and the eradication of poverty. *Science*, American Association for the Advancement of Science, v. 306, n. 5699, p. 1146–1149, 2004.

ALMEIDA-JUNIOR, J. R. *STAD - Uma Ferramenta para Representação e Simulação de Sistemas Através de Statecharts Adaptativos*. Tese (Doutorado) — Escola Politécnica da Universidade de São Paulo, 1995.

ANDERSON, R. P.; LEW, D.; PETERSON, A. T. Evaluating predictive models of species' distributions: criteria for selecting optimal models. *Ecological Modelling*, v. 162, p. 211–232, 2003.

AVESANI, P.; PERINI, A.; RICCI, F. Interactive case-based planning for forest fire management. *Applied Intelligence*, v. 13, p. 41–57, 2000.

BARRON, A.; RISSANEN, J.; YU, B. The minimum description principle in coding and modeling. *IEEE Transactions on Information Theory*, v. 44, n. 6, p. 2743–2760, 1998.

BARTEL, R. A.; SEXTON, J. O. Monitoring habitat dynamics for rare and endangered species using satellite images and niche-based models. *Ecography*, v. 32, p. 888–896, 2009.

BASSETO, B. A. *Um Sistema de Composição Musical Automatizada, Baseado em Gramáticas Sensíveis ao Contexto, Implementado com Formalismos Adaptativos*. Dissertação (Mestrado) — Escola Politécnica da Universidade de São Paulo, 2000.

BELLINI, D. J. S.; TAVELLA, A. C. B. Conversão de partituras para tablaturas usando algoritmo baseado em autômato adaptativo. *Memórias do WTA'2008: Segundo Workshop de Tecnologia Adaptativa*, Laboratório de Linguagens e Técnicas Adaptativas, São Paulo, SP, Brasil, p. 39–42, 2008.

BERGER, A. L.; PIETRA, S. A. D.; PIETRA, V. J. D. A maximum entropy approach to natural language processing. *Computational Linguistics*, v. 22, n. 1, p. 39–71, 1996.

BIOTA. *SinBiota*. 2009. World wide web. Acesso em: 15 nov. 2009. Disponível em: <http://sinbiota.cria.org.br/>.

BORRI, D.; CAMARDA, D.; LIDDO, A. D. Mobility in environmental planning: an integrated multi-agent approach. *Lecture Notes in Computer Science*, v. 3675, p. 119–129, 2005.

BOUSQUET, F.; PAGE, C. L. Multi-agent simulations and ecosystem management: a review. *Ecological Modelling*, v. 176, p. 313–332, 2004.

BRAVO, C. et al. Towards an adaptive implementation of genetic algorithms. *I Taller Latinoamericano de Informática para la Biodiversidad (INBI) - CLEI 2007*, San José, Costa Rica, 2007.

CANHOS, V. P. et al. Global biodiversity informatics: Setting the scene for a “new world” of ecological modeling. *Biodivesrity Informatics*, v. 1, p. 1–13, 2004.

CARBONELL, J. *Machine Learning: Paradigms and Methods*. New York, NY, USA: Elsevier North-Holland, Inc., 1990. ISBN 0-262-53088-0.

CHAITIN, G. J. On the length of programs for computing finite binary sequences: Statistical considerations. *Journal of ACM*, Buenos Aires, Argentina, v. 16, p. 145–159, 1969.

CHAPEAU-BLONDEAU, F.; ROUSSEAU, D. The minimum description length principle for probability density estimation by regular histograms. *Physica A*, v. 388, p. 3969–3984, 2009.

CHEN, L.; HARPER, M.; HUANG, Z. Using maximum entropy (me) model to incorporate gesture cues for su detection. In: *Proceedings of the 8th International Conference on Multimodal Interfaces*. New York, NY, USA: ACM, 2006. p. 185–192.

CHEN, S. F.; ROSENFELD, R. A survey of smoothing techniques for me models. *IEEE Transactions on Speech and Audio Processing*, v. 8, n. 1, p. 37–50, 2000.

CHEN, S. H.; JAKEMAN, A. J.; NORTON, J. P. Artificial intelligence techniques: An introduction to their use for modelling environmental systems. *Mathematics and Computers in Simulation*, v. 78, p. 379–400, 2008.

COLLINS, M.; SCHAPIRE, R. E.; SINGER, Y. Logistic regression, adaboost and bregman distances. *Machine Learning*, v. 48, n. 1, p. 253–285, 2002.

CONABIO. *Comisión Nacional para el Conocimiento y uso de la Biodiversidad*. 2010. World wide web. Acesso em: 15 mar 2010. Disponível em: <http://www-.conabio.gob.mx/>.

CORMEN, T. H. et al. *Algoritmos: teoria e prática*. Rio de Janeiro: Editora Campos – Elsevier, 2002. 717–737 p. ISBN 85–352–0926–3.

CORRÊA, P. L. P. et al. Computational techniques for biologic species distribution modeling. In: PRADO, H. A.; LUIZ, A. J. B.; FILHO, H. C. (Ed.). *Computational Methods for Agricultural Research: Advances and Applications*. 1. ed. Hershey, PA: IGI Global, 2011. v. 1, p. 308–325.

CORTES, C.; VAPNIK, V. Support-vector networks. *Machine Learning*, v. 20, p. 273–297, 1995.

COVER, T. M.; THOMAS, J. A. *Elements of Information Theory*. 2. ed. Hoboken, New Jersey: Willey-Insterscience, 2006.

CRIA; EPUSP; INPE. *openModeller*. 2006. World wide web. Acesso em: 15 nov. 2011. Disponível em: <http://openmodeller.sourceforge.net>.

CURRAN, J. R.; CLARK, S. Investigating gis and smoothing for maximum entropy taggers. *in Proceedings of the 11th Annual Meeting of the European Chapter of the Association for Computational Linguistics (EACL'03)*, Budapest, Hungary, p. 91–98, 2003.

DARROCH, J. N.; RATCLIFF, D. Generalized iterative scaling for log-linear models. *Annals of Mathematical Statistics*, v. 43, n. 5, p. 1470–1480, 1972.

DOWE, D. L. Mml, hybrid bayesian network graphical models, statistical consistency, invariance and uniqueness. In: BANDYOPADHYAY, P.; FORSTER, M. (Ed.). *Handbook of the Philosophy of Science - (HPS Volume 7) Philosophy of Statistics*. North-Holland: Elsevier, 2011. p. 901–982.

DUDÍK, M.; PHILLIPS, S. J.; SCHAPIRE, R. E. Performance guarantees for regularized maximum entropy density estimation. *Proceedings of the 17th Annual Conference on Computational Learning Theory*, 2004.

EL-AZHARY, E.; HASSAN, H. A.; RAFEA, A. Pest control expert system for tomato (pcest). *Knowledge and Information Systems*, v. 2, n. 2, p. 242–257, 2000.

EL-YACOUBI, S. et al. Lucas: an original tool for landscape modeling. *Environmental Modelling and Software*, v. 18, n. 5, p. 429–437, 2003.

ELITH, J. Quantitative methods for modeling species habitat: comparative performance and an application to australian plants. *Quantitative methods for conservation biology*, Springer, p. 39–58, 2002.

ELITH, J. et al. Novel methods improve prediction of species' distribution from occurrence data. *Ecography*, v. 29, n. 2, p. 129–151, 2006.

ELITH, J. et al. A statistical explanation of maxent for ecologists. *Diversity and Distributions*, v. 17, p. 43–57, 2011.

EVANS, J. M.; FLETCHER-JR., R. J.; ALAVALAPATI, J. Using species distribution models to identify suitable areas for biofuel feedstock production. *GCB Bioenergy*, 2010.

FIELDING, A. H.; BELL, J. F. A review of methods for the assessment of prediction errors in conservation presence/absence models. *Environmental Conservation*, v. 24, n. 1, p. 38–49, 1997.

FITZPATRICK, M. C. et al. Climate change, plant migration, and range collapse in a global biodiversity hotspot: the banksia (proteaceae) of western australia. *Global Change Biology*, v. 14, p. 1337–1352, 2008.

FORSYTH, R. *Machine Learning: Principles and Techniques*. London, UK, UK: Chapman & Hall, Ltd., 1988. ISBN 0-412-30580-1.

GBIF. *Global Biodiversity Information Facility*. 2010. World wide web. Acesso em: 15 mar 2010. Disponível em: <http://www.gbif.org/>.

GEORGOUDAS, I. G. et al. A cellular automaton simulation tool for modeling seismicity in the region of xanthi. *Environmental Modelling & Software*, v. 22, n. 10, p. 1455–1464, 2007.

GOODMAN, J. Sequential conditional generalized iterative scaling. *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL)*, Philadelphia, p. 9–16, 2002.

GRAHAM, C. H. et al. New developments in museum-based informatics and applications in biodiversity analysis. *Trends in Ecology & Evolution*, v. 19, n. 9, p. 497–503, 2004.

GRÜNWALD, P. A minimum description length approach to grammar inference. In: WERMTER, S.; RILLOF, E.; SCHELER, G. (Ed.). *Connectionist, Statistical, and Symbolic Approaches to Learning for Natural Language Processing*. Berlin; Heidelberg; New York; Barcelona; Budapest; Hong Kong; London; Milan; Paris; Santa Clara; Singapore; Tokyo: Springer – Lecture Notes in Artificial Intelligence, 1996. v. 1040, p. 203–216.

GRÜNWALD, P. Introducing the minimum description length principle. In: GRÜNWALD, P. D.; MYUNG, I. J.; PITT, M. A. (Ed.). *Advances in Minimum Description Length - Theory and Applications*. Cambridge, Massachusetts: The MIT Press, 2005. p. 3–21.

GU, Y.; MCCALLUM, A.; TOWSLEY, D. Detecting anomalies in network traffic using maximum entropy estimation. *Proceedings of the 5th ACM SIGCOMM Conference on Internet Measurement*, Berkeley, CA, USA, p. 345–350, 2005.

GUISAN, A.; ZIMMERMMANN, N. E. Predictive habitat distribution models in ecology. *Ecological Modelling*, v. 135, p. 147–186, 2000.

GUPTA, M.; FRIEDLANDER, M.; GRAY, R. M. Maximum entropy classification applied to speech. *Conference Record of the Thirty-Fourth Asilomar Conference on Signals, Systems and Computers*, Pacific Grove, CA, USA, v. 2, p. 1480–1483, 2000.

HANSEN, M.; YU, B. Wavelet thresholding via mdl for natural images. *IEEE Transactions on INformation Theory*, v. 46, p. 1778–1788, 2000.

HAYKIN, S. *Redes Neurais - Princípios e Prática*. 2. ed. Porto Alegre, RS, Brasil: Bookman, 2001.

HIJMANS, R. J. et al. Very high resolution interpolated climate surfaces for global land areas. *International Journal of Climatology*, v. 25, p. 1965–1978, 2005.

HIRAKAWA, A. R.; SARAIVA, A. M.; CUGNASCA, C. E. Autômatos adaptativos aplicados em automação e robótica (A4R). *IEEE Latin America Transactions*, v. 5, n. 7, p. 539–543, 2007.

HIRZEL, A. H. et al. Ecological-niche factor analysis: How to compute habitat-suitability maps without absence data? *Ecology*, v. 83, n. 7, p. 2027–2036, 2002.

HUTCHINSON, G. E. Concluding remarks. *Cold Spring Harbour Symposium on Quantitative Biology*, v. 22, p. 415–427, 1957.

ILIADIS, L. S. A decision support system applying an integrated fuzzy model for long-term forest fire risk estimation. *Environmental Modelling & Software*, v. 20, n. 5, p. 613–621, 2005.

ILIADIS, L. S.; MARIS, F. An artificial neural network model for mountainous water-resources management: The case of cyprus mountainous watersheds. *Environmental Modelling & Software*, v. 22, n. 7, p. 1066–1072, 2007.

INES, A. V. M. et al. Combining remote sensing-simulation modeling and genetic algorithm optimization to explore water management options in irrigated agriculture. *Agricultural Water Management*, v. 83, n. 3, p. 221–232, 2006.

IWAI, M. K.; NETO, J. J. *Introdução à Gramáticas Adaptativas*. 2000. Boletim Técnico, PCS-POLI-USP: São Paulo, SP, Brasil.

JAYNES, E. T. On the rationale of maximum-entropy methods. *Proceedings of the IEEE*, v. 70, n. 9, p. 939–952, 1982.

JAYNES, E. T. *Notes on Present Status and Future Prospects*. 1990.

JEON, J.; MANMATHA, R. Using maximum entropy for automatic image annotation. *Proceedings of the International Conference on Image and Video Retrieval*, p. 24–32, 2004.

JÖRNSTEN, R.; YU, B. Simultaneous gene clustering and subset selection for sample classification via mdl. *Bioinformatics*, v. 19, n. 9, 2003.

KALAPANIDAS, E.; AVOURIS, N. Short-term air quality prediction using a case-based classifier. *Environmental Modeling and Software*, v. 16, n. 3, p. 263–272, 2001.

KALAPANIDAS, E.; AVOURIS, N. Feature selection for air quality forecasting: a genetic algorithm approach. *AI Communications*, v. 16, n. 4, p. 235–251, 2003.

KASCHNER, K. et al. *AquaMaps: Predicted range maps for aquatic species*. 2010. World wide web electronic publication. Version 08/2010. Disponível em: <www.aquamaps.org>.

KING, J. M. P.; ALCÁNTARA, R. B. nares; MANAN, Z. A. Minimising environmental impact using cbr: An azeotropic distillation case study. *Environmental Modelling and Software*, v. 14, n. 5, p. 359–366, 1999.

KOLMOGOROV, A. Three approaches to the quantitative definition of information. *Problems of Information Transmission*, v. 1, n. 1, p. 3–11, 1965.

KONTKANEN, P. *Computationally Efficient Methods for MDL-Optimal Density Estimation and Data Clustering*. Tese (Doutorado) — University of Helsinki, Finland, 2009.

KONTKANEN, P.; MYLLYMÄKI, P. A fast normalized maximum likelihood algorithm for multinomial data. In: *Proceedings of the 19th International Joint Conference on Artificial Intelligence*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2005. p. 1613–1615.

KONTKANEN, P.; MYLLYMÄKI, P. Mdl histogram density estimation. *Proceedings of the Eleventh International Workshop on Artificial Intelligence and Statistics*, 2007.

LE, Z. *Maximum Entropy Modeling Toolkit for Python and C++*. 2004. Acesso em: 21 ago 2007. Disponível em: <http://homepages.inf.ed.ac.uk/lzhang10/maxent_toolkit-.html>.

LEE, M. D.; NAVARRO, D. J. Minimum description length and psychological clustering models. In: GRÜNWALD, P. D.; MYUNG, I. J.; PITT, M. A. (Ed.). *Advances in Minimum Description Length - Theory and Applications*. Cambridge, Massachusetts: The MIT Press, 2005. p. 355–384.

LI, L.; BIAN, L.; YAN, G. An integrated bayesian modelling approach for predicting mosquito larval habitats. *Summer 2006 Assembly*, Vancouver, Washington, USA, 2006.

LI, M.; VITÁNYI, P. M. B. *An Introduction to Kolmogorov Complexity and Its Applications*. Berlin/Heidelberg, Alemanha: Springer, 1997.

LI, M.; VITÁNYI, P. M. B. *An Introduction to Kolmogorov Complexity and Its Applications*. 3rd. ed. New York: Springer-Verlag, 2008.

LIANG, G.; YU, B.; TAFT, N. Maximum entropy models: Convergence rates and application in dynamic system monitoring. *IEEE International Symposium on Information Theory (ISIT)*, 2004.

LIU, D. C.; NOCEDAL, J. On the limited memory bfgs method for large scale optimization. *Mathematical Programming*, v. 45, p. 503–528, 1989.

LORENA, A. C. et al. Comparing machine learning classifiers in potential distribution modelling. *Expert Systems with Apllications*, v. 38, p. 5268–5275, 2011.

LORENA, A. C. et al. Potential distribution modelling using machine learning. In: *Proceedings of the 21st International Conference on Industrial, Engineering and Other Applications of Applied Intelligent Systems: New Frontiers in Applied Artificial Intelligence*. Berlin, Heidelberg: Springer-Verlag, 2008. (IEA/AIE '08), p. 255–264.

MAHAMAN, B. D. et al. A diagnostic expert system for honeybee pests. *Computers and Electronics in Agriculture*, v. 36, n. 1, p. 17–31, 2002.

MAHAMAN, B. D. et al. Diares-ipm: a diagnostic advisory rule-based expert system for integrated pest management in solanaceous crop systems. *Agricultural Systems*, v. 76, n. 3, p. 1119–1135, 2003.

MAIER, H. R.; DANDY, G. C.; BURCH, M. D. Use of artificial neural networks for modeling cyanobacteria anabaena spp. in the river murray, south australia. *Ecological Modeling*, v. 105, p. 257–272, 1998.

MALOUF, R. A comparison of algorithms for maximum entropy parameter estimation. *Proceedings of the Sixth Conference on Natural Language Learning (CoNLL-2002)*, p. 49–55, 2002.

MARCO-JÚNIOR, P.; SIQUEIRA, M. F. Como determinar a distribuição potencial de espécies sob uma abordagem conservacionista? *Megadiversidade*, v. 5, n. 1–2, p. 65–76, 2009.

MCCALLUM, A.; FREITAG, D.; PEREIRA, F. Maximum entropy markov models for information extraction and segmentation. *Proceedings of the 17th International Conference on Machine Learning*, Morgan Kaufmann, p. 591–598, 2000.

MCNALLY, M. D.; SIEK, J. *MPI Tutorial Part 1 - Introduction*. Bloomington, IN, USA, 1998. Acesso em: 15 abr. 2008. Disponível em: <http://www.lam-mpi.org-/tutorials/nd>.

MEHTA, M.; RISSANEN, J.; AGRAWAL, R. Mdl-based decision tree pruning. In: *Proceedings of the First International Conference on Knowledge Discovery and Data Mining (KDD'95)*. Montreal, Quebec, Canada: AAAI Press, 1995. p. 216–221.

METTERNICHT, G.; GONZALEZ, S. Fuero: foundations of a fuzzy exploratory model for soil erosion hazard prediction. *Environmental Modelling & Software*, v. 20, n. 6, p. 715–728, 2005.

MITCHELL, T. M. *Machine Learning*. New York: WCB – McGraw-Hill, 1997.

MONARD, M. C.; BARANAUSKAS, J. A. Conceitos sobre aprendizagem de máquina. In: REZENDE, S. O. (Ed.). *Sistemas Inteligentes – Fundamentos e Aplicações*. Barueri, SP: Manole, 2003. p. 89–114.

MONARD, M. C.; BARANAUSKAS, J. A. Indução de regras e Árvores de decisão. In: REZENDE, S. O. (Ed.). *Sistemas Inteligentes – Fundamentos e Aplicações*. Barueri, SP: Manole, 2003. p. 115–139.

MUÑOZ, M. E. S. et al. openmodeller: a generic approach to species' potential distribution modelling. *Geoinformatica*, v. 15, p. 111–135, 2011.

MYERS, N. et al. Biodiversity hotspots for conservation priorities. *Nature*, v. 403, p. 853–858, 2000.

MYUNG, I. J. et al. The use of mdl to select among computational models of cognition. In: LEEN, T. K.; DIETTERICH, T. G.; TRESP, V. (Ed.). *Advances in Neural Information Processing Systems 13*. Denver, CO, USA: MIT Press, 2001. p. 38–44.

NAVARRO, D. J.; LEE, M. D. An application of minimum description length clustering to partitioning learning curves. In: *Proceedings of the 2005 IEEE International Symposium on Information Theory*. Adelaide, Australia: IEEE, 2005. p. 587–591.

NETO, J. J. *Contribuições à Metodologia de Construção de Compiladores*. 1993. Tese de Livre Docência.

NETO, J. J. Adaptive automata for context-dependent languages. *SIGPLAN Notices*, ACM, New York, NY, USA, v. 29, p. 115–124, September 1994. ISSN 0362-1340.

NETO, J. J. Solving complex problems efficiently with adaptive automata. In: YU, S.; PAUN, A. (Ed.). *Implementation and Application of Automata*. Berlin, Heidelberg: Springer-Verlag, 2001, (Lecture Notes in Computer Science, v. 2088). p. 340–342.

NETO, J. J. Adaptive rule-driven devices - general formulation and case study. In: WATSON, B.; WOOD, D. (Ed.). *Implementation and Application of Automata, CIAA 2001*. Berlim, Heidelberg: Springer-Verlag, 2002. (Lecture Notes in Computer Science, v. 2494), p. 234–250.

NETO, J. J. *Tutorial sobre Autômatos Adaptativos*. 2004.

NETO, J. J. Um levantamento da evolução da adaptatividade e da tecnologia adaptativa. *IEEE Latin America Transactions*, v. 5, n. 7, p. 496–505, 2007.

NETO, J. J. Um glossário sobre adaptatividade. *Memórias do WTA'2009: Terceiro Workshop de Tecnologia Adaptativa*, Laboratório de Linguagens e Técnicas Adaptativas, São Paulo, SP, Brasil, p. II–VII, 2009.

NIGAM, K.; LAFFERTY, J.; MCCALLUM, A. Using maximum entropy for text classification. *In IJCAI-99 Workshop on Machine Learning for Information Filtering*, p. 61–67, 1999.

NIX, H. A. A biogeographic analysis of australian elapid snakes. In: LONGMORE, R. (Ed.). *Atlas of Australian Elapid Snakes*. Canberra: Australian Government Publishing Service, 1986. p. 4–15.

ONU. *United Nations Millenium Development Goals*. 2011. World wide web electronic publication. Acesso em: 15 nov. 2011. Disponível em: <www.un.org-/millenniumgoals/environ.shtml>.

PEARSON, R. G. et al. Predicting species distributions from small numbers of occurrence records: a test case using cryptic geckos in madagascar. *Journal of Biogeography*, v. 34, p. 102–117, 2007.

PEDRO, J. S.; BURSTEIN, F.; SHARP, A. A case-based fuzzy multicriteria decision support model for tropical cyclone forecasting. *European Journal of Operational Research*, v. 160, p. 308–324, 2005.

PELEGRINI, E. J.; NETO, J. J. Applying adaptive technology in data security. *Proceedings of the 6th Peruvian Computer Week - JPC 2007*, Trjillo, Peru, p. 31–40, 2007.

PELLETIER, G. J.; CHAPRA, S. C.; TAO, H. Qual2kw - a framework for modeling water quality in streams and rivers using a genetic algorithm for calibration. *Environmental Modelling & Software*, v. 21, p. 419–425, 2006.

PENFIELD-JR., P. *Information and Entropy*. 2003.

PEREIRA, R. S. *DesktopGarp*. 2003. World wide web electronic publication. Acesso em: 15 nov. 2011. Disponível em: <www.nhm.ku.edu/desktopgarp/>.

PERSONA, L.; CORRÊA, P. L. P.; SARAIVA, A. M. Modelagem de nicho ambiental em biodiversidade com algoritmos genéticos. *2nd International Information and Telecommunication Technologies Symposium. Proceedings of the IEEE - I2TS'2003*, Florianópolis, p. 1–5, 2003.

PETERSON, A. T. Predicting species' geographic distributions based on ecological niche modeling. *The Condor*, v. 103, p. 599–605, 2001.

PETERSON, A. T.; PAPES, M.; KLUZA, D. A. Predicting the potential invasive distributions of four alien plant species in north america. *Weed Science*, v. 51, p. 863–868, 2003.

PETERSON, A. T.; VIEGLAIS, D. A. Predicting species invasions using ecological niche modeling: New approaches from bioinformatics attack a pressing problem. *BioScience*, v. 51, n. 5, p. 363–372, 2001.

PHILLIPS, S. J.; ANDERSON, R. P.; SCHAPIRE, R. E. *MaxEnt Software*. 2006. World wide web electronic publication. Acesso em: 15 nov. 2011. Disponível em: <www.cs.princeton.edu/\~schapire/maxen>.

PHILLIPS, S. J.; ANDERSON, R. P.; SCHAPIRE, R. E. Maximum entropy modeling of species geographic distributions. *Ecological Modelling*, n. 190, p. 231–259, 2006.

PHILLIPS, S. J.; DUDÍK, M. Modeling of species distributions with maxent: new extensions and a comprehensive evaluation. *Ecography*, v. 31, p. 161–175, 2008.

PHILLIPS, S. J.; DUDÍK, M.; SCHAPIRE, R. E. A maximum entropy approach to species distribution modeling. *Proceedings of the 21st International Conference on Machine Learning*, ACM, New York, NY, USA, p. 83–90, 2004.

PIETRA, S. D.; PIETRA, V. D.; LAFFERTY, J. Inducing features of random fields. *IEEE Transactions Pattern Analysis and Machine Intelligence*, v. 19, n. 4, p. 1–13, 1997.

PISTORI, H.; NETO, J. J. Adaptree - proposta de um algoritmo para indução de Árvores de decisão baseado em técnicas adaptativas. *Anais da Conferência Latino Americana de Informática - CLEI*, Montevideo, Uruguai, 2002.

POZDNYAKOV, D. et al. Operational algorithm for the retrieval of water quality in the great lakes. *Remote Sensing of Environment*, v. 97, n. 3, p. 352–370, 2005.

PRATI, R. C.; BATISTA, G. E. A. P. A.; MONARD, M. C. Curvas roc para avaliação de classificadores. *Revista IEEE América Latina*, v. 6, n. 2, p. 215–222, 2008.

PURNOMO, H. et al. Developing multi-stakeholder forest management scenarios: a multi-agent system simulation approach applied in indonesia. *Forest Policy and Economics*, v. 7, n. 4, p. 475–491, 2005.

QING, Z.; STANLEY, S. J. Forecasting raw-water quality parameters for the north saskatchewan river by neural network modeling. *Water Research*, v. 31, n. 9, p. 2340–2350, 1997.

QUINLAN, J. R. Induction of decision trees. *Machine Learning*, v. 1, p. 81–106, 1986.

QUINLAN, J. R. *C4.5: Programs for Machine Learning*. San Mateo, California: Morgan Kaufmann, 1993.

RAMOS, M. V. M.; NETO, J. J.; VEGA, I. S. *Linguagens Formais: Teoria, Modelagem e Implementação*. Porto Alegre, RS, Brasil: Bookman, 2009. ISBN 978–85–7780–453–5.

RATNAPARKHI, A. *A Simple Introduction to Maximum Entropy Models for Natural Language Processing*. Philadelphia, PA, USA, 1997.

RAXWORTHY, C. J. et al. Applications of ecological niche modeling for species delimitation: A review and empirical evaluation using day geckos (phelsuma) from madagascar. *Systematic Biology*, v. 56, n. 6, p. 907–923, 2007. ISSN 1063-5157 print / 1076-836X online.

REPORT-OM. *openModeller – A framework for species modeling*. Campinas, São Paulo, Brazil, 2006.

REPORT-OM. *openModeller – A framework for species modeling*. Campinas, São Paulo, Brazil, 2007.

REPORT-OM. *openModeller – A framework for species modeling*. Campinas, São Paulo, Brazil, 2008.

REZENDE, S. O. Introdução. In: REZENDE, S. O. (Ed.). *Sistemas Inteligentes – Fundamentos e Aplicações*. Barueri, SP: Manole, 2003. p. 3–11.

RISSANEN, J. Modeling by shortest data description. *Automatica*, v. 14, p. 465–471, 1978.

RISSANEN, J. Stochastic complexity and modeling. *Annals of Statistics*, v. 14, p. 1080–1100, 1986.

RISSANEN, J. Fisher information and stochastic complexity. *IEEE Transactions on Information Theory*, v. 42, n. 1, p. 40–47, 1996.

RISSANEN, J. *Lectures on Statistical Modeling Theory*. 2005.

RISSANEN, J. *An Introduction to the MDL Principle*. 2011. World wide web electronic publication. Acesso em: 15 nov. 2011. Disponível em: <www.mdl-research.org/jorma.rissanen/pub/Intro.pdf>.

ROBERTSON, M. P.; CAITHNESS, N.; VILLET, M. H. A pca-based modelling technique for predicting environmental suitability for organisms from presence records. *Diversity and Distributions*, v. 7, p. 15–27, 2001.

ROCHA, R. L. A. Tecnologia adaptativa aplicada ao processamento computacional de língua natural. *IEEE Latin America Transactions*, v. 5, n. 7, p. 544–551, 2007.

ROCHA, R. L. A.; NETO, J. J. Autômato adaptativo, limites e complexidade em comparação com a máquina de turing. *Proceedings of the second Congress of Logic Applied to Technology - LAPTEC 2000*, São Paulo: Faculdade SENAC de Ciências Exatas e Tecnologia, p. 33–48, 2000.

RODRIGUES, E. S. C.; RODRIGUES, F. A.; ROCHA, R. L. A. Algoritmo paralelo de entropia máxima aplicado a modelagem de nicho ecológico. *Proceedings of the 7th International Information and Telecommunication Technologies Symposium - I2TS*, Foz do Iguaçu, PR, Brasil, 2008a.

RODRIGUES, E. S. C.; RODRIGUES, F. A.; ROCHA, R. L. A. Autômatos adaptativos para emparelhamento de cadeias. *Memórias do WTA'2008: Segundo Workshop de Tecnologia Adaptativa*, Laboratório de Linguagens e Técnicas Adaptativas, São Paulo, SP, Brasil, p. 27–30, 2008b.

RODRIGUES, E. S. C.; RODRIGUES, F. A.; ROCHA, R. L. A. Species distribution modeling with the maximum entropy algorithm. *In: CIC 2008 – 17th International Conference on Computing*, Mexico City, Mexico, 2008c.

RODRIGUES, E. S. C.; RODRIGUES, F. A.; ROCHA, R. L. A. Dispositivo adaptativo na análise de modelos de distribuição de espécies. *Memórias do WTA'2009: Terceiro Workshop de Tecnologia Adaptativa*, Laboratório de Linguagens e Técnicas Adaptativas, São Paulo, SP, Brasil, p. 83–87, 2009.

RODRIGUES, E. S. C. et al. An adaptive maximum entropy approach for modeling of species distribution. *Memórias do WTA'2010: Quarto Workshop de Tecnologia Adaptativa*, Laboratório de Linguagens e Técnicas Adaptativas, São Paulo, SP, Brasil, p. 108–117, 2010a.

RODRIGUES, E. S. C. et al. Evaluation of different aspects of maximum entropy for niche-based modeling. In: *ISEIS 2010 Ecological Informatics and Ecosystem Conservation*. Beijing, China: Elsevier, 2010b. p. 1066–1077.

RODRIGUES, E. S. C. et al. Mdl-based clustering for modeling of species geographic distribution. *In: ISEI7 7th International Conference on Ecological Informatics*, Ghent, Belgium, p. 178–179, 2010c.

RODRIGUES, E. S. C. et al. Selection of niche-based models with minimum description lenght. *In: 1st Conference on Computational Interdisciplinary Sciences (CCIS)*, São José dos Campos, SP, Brasil, 2010d.

RODRIGUES, E. S. C. et al. Adaptive approach for a maximum entropy algorithm in ecological niche modeling. *Revista IEEE América Latina*, v. 9, p. 331–338, 2011a.

RODRIGUES, E. S. C. et al. Minimum description length principle to select environmental layers in modeling of species geographical distribution. *Journal of Computational Interdisciplinary Sciences*, v. 2, n. 2, p. 131–137, 2011b.

RODRIGUES, F. A. et al. Species distribution modeling with neural networks. *e-Biosphere'09 International Conference on Biodiversity Informatics*, London, p. 97, 2009.

RODRIGUES, F. A. et al. Modelagem da biodiversidade utilizando redes neurais artificiais. *II Workshop de Computação Aplicada à Gestão do Meio Ambiente e Recursos Naturais (WCAMA). XXX Congresso da Sociedade Brasileira de Computação - Computação Verde: Desafios Científicos e Tecnológicos*, Belo Horizonte, MG, Brasil, p. 585–594, 2010.

RODRIGUES, F. A. et al. Parallelization of the jackknife algorithm applied to a biodiversity modeling system. *Proceedings of the 7th International Information and Telecommunication Technologies Symposium - I2TS*, Foz do Iguaçu, PR, Brasil, p. 58–65, 2008.

RUSSELL, S.; NORVIG, P. *Inteligência Artificial: tradução da segunda edição*. Rio de Janeiro, RJ, Brasil: Elsevier, 2004.

SALAKHUTDINOV, R.; ROWEIS, S.; GHAHRAMANI, Z. On the convergence of bound optimization algorithms. *Uncertainty in Artificial Intelligence*, v. 19, p. 509–516, 2003.

SANTANA, F. S. et al. P-garp (parallel genetic algorithm for rule-set production) for clusters applications. *Proceedings of 5th International Conference on Ecological Informatics - ISEI5*, Elsevier, 2006.

SANTANA, F. S. et al. A reference business process for ecological niche modelling. *Ecological Informatics*, v. 3, p. 75–86, 2008.

SCHLUTER, M.; RUGER, N. Application of a gis-based simulation tool to illustrate implications of uncertainties for water management in the amudarya river delta. *Environmental Modelling & Software*, v. 22, n. 2, p. 158–166, 2007.

SCHNEIDMAN, E. et al. Weak pairwise correlations imply strongly correlated network states in a neural population. *Nature*, v. 440, n. 7087, p. 1007–1012, 2006.

SCHÖLKOPF, B. et al. Estimating the support of a high-dimensional distribution. *Neural Computation*, v. 13, p. 1443–1471, 2001.

SCHÖLKOPF, B. et al. New support vector algorithms. *Neural Computation*, v. 12, p. 1207–1245, 2000.

SCHÖNFISCH, B.; KINDER, M. A fish migration model. *Lecture Notes in Computer Science*, v. 2493, p. 210–219, 2002.

SEINET. *Southwest Environmental Information Network*. 2010. World wide web. Acesso em: 15 mar 2010. Disponível em: <http://seinet.asu.edu/>.

SETZER, V. W. *Dado, Informação, Conhecimento e Competência*. 2. ed. São Paulo, SP, Brasil: Editora Escrituras, 2001. (Meios Eletrônicos: Uma visão alternativa, v. 10).

SHANNON, C. E. A mathematical theory of communication. *Bell System Technical Journal*, v. 27, 1948.

SHLENS, J. et al. The structure of multi-neuron firing patterns in primate retina. *The Journal of Neuroscience*, v. 26, n. 32, p. 8254–8266, 2006.

SHTARKOV, Y. M. Universal sequential coding of single messages. *Problems of Information Transmission*, v. 23, p. 3–17, 1987.

SILVA, F. S. C.; MELO, A. C. V. *Modelos Clássicos de Computação*. São Paulo: Thomson Learning, 2006. ISBN 85-221-0518-9.

SIMPSON, B. B. Krameriaceae. *Flora Neotropica Monograph*, n. 49, p. 1–109, 1989.

SIMPSON, B. B. et al. Species relationships in krameria (krameriaceae) based on its sequences and morphology: implications for character utility and biogeography. *Systematic Botany*, n. 29, p. 97–108, 2004.

SIQUEIRA, M. F. *Uso de modelagem de nicho fundamental na avaliação do padrão de distribuição geográfica de espécies vegetais*. Tese (Doutorado) — Escola de Engenharia de São Carlos da Universidade de São Paulo, São Carlos, SP, Brazil, 2005.

SIQUEIRA, M. F. de; PETERSON, A. T. Consequences of global climate change for geographic distributions of cerrado tree species. *Biota Neotropica*, v. 3, n. 2, 2003.

SKARPAAS, O.; STABBETORP, O. E. Population viability analysis with species occurrence data from museum collections. *Conservation Biology*, v. 25, n. 3, p. 577–586, 2011.

SOBERÓN, J. Grinnellian and eltonian niches and geographic distributions of species. *Ecology Letters*, v. 10, p. 1–9, 2007.

SOBERÓN, J.; PETERSON, A. T. Interpretation of models of fundamental ecological niches and species' distributional areas. *Biodiversity Informatics*, v. 2, p. 1–10, 2005.

SOLOMONOFF, R. J. A formal theory of inductive inference, part i. *Information and Control, Part I*, Academic Press Inc., v. 7, n. 1, p. 1–22, 1964.

SOLOMONOFF, R. J. A formal theory of inductive inference, part ii. *Information and Control, Part II*, Academic Press Inc., v. 7, n. 2, p. 224–254, 1964.

SOUSA, S. I. V. et al. Multiple linear regression and artificial neural networks based on principal components to predict ozone concentrations. *Environmental Modelling & Software*, v. 22, n. 1, p. 97–103, 2007.

SPECIESLINK. 2010. World wid web. Acesso em: 15 mar 2010. Disponível em: <http://splink.cria.org.br/>.

STANGE, R. L. et al. Estudo comparativo entre algoritmos genéticos adaptativos e não-adaptativos aplicados à modelagem ambiental de *Peponapis* e *Cucurbita*. *Memórias do WTA'2009: Terceiro Workshop de Tecnologia Adaptativa*, Laboratório de Linguagens e Técnicas Adaptativas, São Paulo, SP, Brasil, p. 120–128, 2009.

STOCKWELL, D.; PETERS, D. The garp modelling system: problems and solutions to automated spatial prediction. *International Journal of Geographical Information Science*, v. 13, n. 2, p. 143–158, 1999.

STOCKWELL, D. R. B.; NOBLE, I. R. Induction of sets of rules from animal distribution data: a robust and informative method of data analysis. *Mathematics and Computers in Simulation*, v. 33, p. 385–390, 1992.

SU, Y.; MYUNG, I. J.; PITT, M. A. Minimum description length and cognitive modeling. In: GRÜNWALD, P. D.; MYUNG, I. J.; PITT, M. A. (Ed.). *Advances in Minimum Description Length - Theory and Applications*. Cambridge, Massachusetts: The MIT Press, 2005. p. 411–433.

SUTTON, T.; GIOVANNI, R.; SIQUEIRA, M. F. Introducing openmodeller: A fundamental niche modelling framework. *OSGeo Journal*, v. 1, p. 1–6, 2007. ISSN 1994–1897.

TANG, A. et al. A maximum entropy model applied to spatial and temporal correlations from cortical networks in vitro. *The Journal of Neuroscience*, v. 28, n. 2, p. 505–518, 2008.

TAVARES, J. N. *Curso Livre sobre Teoria do Campo - Entropia, Medidas de Gibbs*. 2004. World wide web. Acesso em: 21 ago. 2007. Disponível em: <http://cmup.fc.up.pt/cmup/v2/frames/minicourses.htm>.

THOMAS, C. D. et al. Extinction risk from climate change. *Nature*, v. 427, p. 145–148, 2004.

TOGNELLI, M. F. et al. An evaluation of methods for modeling distribution of patagonian insects. *Revista Chilena de Historia Natural*, v. 82, p. 347–360, 2009.

VAPNIK, V. *The Nature of Statistical Learning Theory*. 2. ed. New York: SpringerVerlag, 1995.

VOGEL, S. Ölblumen und ölsammelnde bienen. *Tropische und Subtropische Pflanzenwelt*, n. 7, p. 285–547, 1974.

WALDER, C. J.; KOOTSOOKOS, P. J.; LOVELL, B. C. Towards a maximum entropy method for estimating hmm parameters. *Proceedings of the 2003 APRS Workshop on Digital Image Computing*, p. 45–49, 2003.

WANG, H.; SEVCIK, K. C. Histograms based on the minimum description length principle. *The VLDB Journal*, v. 17, p. 419–442, 2008.

WILSON, E. B.; HILFERTY, M. M. The distribution of chi-square. *Mathematics*, v. 17, p. 684–688, 1931.

ZETIAN, F. et al. Pig-vet: a web-based expert system for pig disease diagnosis. *Expert Systems with Applications*, v. 29, n. 1, p. 93–103, 2005.

ZIMMERMANN, N. E. et al. New trends in species distribution modelling. *Ecography*, v. 33, p. 985–989, 2010.

Glossário

Acuidade

Grau de proximidade entre valores estimados e valores reais 26

Algoritmo não determinístico

Algoritmo que produz saídas diferentes em cada execução 29

Aprendizagem supervisionada

Processo de inferência a partir de um conjunto de exemplos composto por pares, em que a primeira componente de cada par é um vetor de atributos e a segunda componente de cada par é o valor de saída esperado 30

AUC

Área abaixo da curva ROC 26

Comprimento de uma cadeia

Quantidade de símbolos que compõe a cadeia 65

Curva ROC

Representação gráfica da taxa de verdadeiros positivos contra a taxa de falsos positivos 26

Erro de omissão

Porcentagem de exemplos falsos negativos 26

Erro de sobreprevisão

Porcentagem de exemplos falsos positivos 26

Especiação

Processo evolutivo de formação das espécies 31

Espécies endêmicas

Espécies restritas a uma determinada região 11

Espécies pelágicas

Espécies que vivem em mar aberto, ou seja, não dependem do fundo marinho 29

Estímulo de entrada

Símbolo pertencente ao alfabeto de entrada capaz de modificar a configuração de um dispositivo. O alfabeto de entrada é o conjunto de todos os símbolos que podem ser utilizados por um dispositivo 59

Grade discreta

Conjunto contável de células de uma determinada área, em que cada célula está associada a um ponto georreferenciado 43

Gradiente

Alteração no valor de uma quantidade por unidade de espaço. 46

Hemiparasita

É um parasita que não depende completamente do seu hospedeiro, podendo inclusive viver sem ele em alguns casos. 91

Linguagens regulares

Qualquer linguagem reconhecida por um autômato finito 62

Linguagens sensíveis ao contexto

Qualquer linguagem reconhecida por um autômato linearmente limitado. Um autômato linearmente limitado é uma Máquina de Turing não determinística com fita limitada 62

Matriz Hessiana

A matriz Hessiana de uma função f é uma matriz quadrada das derivadas parciais de segunda ordem da função 74

Multiplicadores de Lagrange

Método para encontrar extremos de uma função cujas variáveis possuem restrições 45

Máquina de Markov de primeira ordem

Máquina em que o próximo estado depende apenas do estado atual 62

Máquina de Turing

Máquina abstrata de estados com uma fita de trabalho, onde os símbolos podem ser lidos ou escritos, e um cabeçote de leitura/gravação. Inicialmente, a fita contém a cadeia de entrada e o cabeçote está posicionado no primeiro símbolo da cadeia. Em cada mudança de estado, o cabeçote pode avançar uma posição para a direita ou retroceder uma posição para a esquerda 60

Page faults

É uma interrupção disparada pelo hardware quando um programa acessa uma página (bloco de memória) mapeada no espaço de memória virtual, mas que não foi carregada na memória física do computador. 83

Percentil

Valor que corresponde à frequência cumulativa da amostra. 29

Princípio da Navalha de Occam

Este princípio deve-se à *William of Ockham* e diz que dentre várias explicações possíveis para um mesmo fato, deve-se escolher sempre a mais simples 49

Problema da Parada

Dado qualquer programa de computador como entrada, nenhum outro programa é capaz de fornecer a saída **verdadeiro**, se o programa de entrada sempre para, ou **falso**, se o programa de entrada nunca para 54

Produção primária

Produção de compostos a partir de dióxido de carbono atmosférico ou aquático, principalmente através da fotossíntese 29

Script

Programas de computador escritos em linguagens de *script*, ou seja, linguagens de programação que servem para estender a funcionalidade de um programa e/ou controlá-lo. 28

Software livre

Qualquer programa de computador cujo código fonte está disponível para uso, cópia, estudo e redistribuição, de acordo com a licença à ele anexada. 27