UNIVERSIDADE DE SÃO PAULO
ESCOLA DE ARTES, CIÊNCIAS E HUMANIDADES
PROGRAMA DE PÓS-GRADUAÇÃO EM SISTEMAS DE INFORMAÇÃO

RODOLFO DA SILVA SIMÕES

**Técnicas de transferência de aprendizagem aplicadas a modelos QSAR para
regressão**

São Paulo

2018

RODOLFO DA SILVA SIMÕES

**Técnicas de transferência de aprendizagem aplicadas a modelos QSAR para
regressão**

Dissertação apresentada à Escola de Artes, Ciências e Humanidades da Universidade de São Paulo para obtenção do título de Mestre em Ciências pelo Programa de Pós-graduação em Sistemas de Informação.

Área de concentração: Metodologia e Técnicas da Computação

Versão corrigida contendo as alterações solicitadas pela comissão julgadora em 10 de abril de 2018. A versão original encontra-se em acervo reservado na Biblioteca da EACH-USP e na Biblioteca Digital de Teses e Dissertações da USP (BDTD), de acordo com a Resolução CoPGr 6018, de 13 de outubro de 2011.

Orientador: Profa. Dra. Patrícia Rufino Oliveira

São Paulo

2018

Autorizo a reprodução e divulgação total ou parcial deste trabalho, por qualquer meio convencional ou eletrônico, para fins de estudo e pesquisa, desde que citada a fonte.

CATALOGAÇÃO-NA-PUBLICAÇÃO

(Universidade de São Paulo. Escola de Artes, Ciências e Humanidades. Biblioteca)

CRB 8 - 4936

Simões, Rodolfo da Silva

Técnicas de transferência de aprendizagem aplicadas a modelos QSAR para regressão / Rodolfo da Silva Simões ; orientadora, Patrícia Rufino Oliveira. – 2018.

143 f. : il

Dissertação (Mestrado em Ciências) - Programa de Pós-Graduação em Sistemas de Informação, Escola de Artes, Ciências e Humanidades, Universidade de São Paulo.

Versão corrigida

1. Inteligência artificial. 2. Aprendizado computacional.
3. Relações quânticas entre estrutura química e atividade biológica. I. Oliveira, Patrícia Rufino, orient. II. Título

CDD 22.ed.– 006.3

Dissertação de autoria de Rodolfo da Silva Simões, sob o título “**Técnicas de transferência de aprendizagem aplicadas a modelos QSAR para regressão**”, apresentada à Escola de Artes, Ciências e Humanidades da Universidade de São Paulo, para obtenção do título de Mestre em Ciências pelo Programa de Pós-graduação em Sistemas de Informação, na área de concentração Metodologia e Técnicas da Computação, aprovada em 10 de abril de 2018 pela comissão julgadora constituída pelos doutores:

Profa. Dra. Patrícia Rufino Oliveira

Instituição: Universidade de São Paulo

Presidente

Prof. Dr. Fabrício Olivetti de França

Instituição: Universidade Federal do ABC

Prof. Dr. Fernando Fagundes Ferreira

Instituição: Universidade de São Paulo

Profa. Dra. Arianne Machado Lima

Instituição: Universidade de São Paulo

Dedico este trabalho à pessoa que guia o meu caminho diariamente, mesmo não presente em vida, mas em alma. Te amo mãe.

Agradecimentos

A Deus pelo meu caminho e por tudo conquistado até hoje. Aos professores Patrícia Rufino Oliveira (Orientadora), Káthia Maria Honório e Clodoaldo Aparecido de Moraes Lima por todas as discussões, revisões de conteúdo e contribuições dadas ao longo do mestrado, que foram fundamentais na concepção desse trabalho.

Ao meu pai Nelson que é um guerreiro e não mede esforços para trazer alegria para mim e para o meu irmão. Obrigado pelos ensinamentos e valores que carrego comigo até hoje. Ao meu irmão Rodrigo que dividiu todos os momentos difíceis comigo e sempre foi o meu melhor amigo.

A minha namorada Bruna pelo seu companheirismo e incentivo para o bem maior, a minha formação. Além de sua compreensão pela minha ausência.

Aos colegas de laboratório e programa de mestrado, aos amigos que terei pela vida (Gleydson, Thaís, Alex, Rafaela, Weslane e Gabriel) que auxiliaram no entendimento de diversos aspectos da área de aprendizagem de máquina e química medicinal, além das contribuições nesse trabalho e momentos de alegria e descontração.

A todos os professores do Programa de Pós-Graduação em Sistemas de Informação por toda conhecimento e aprendizado durante o período de mestrado.

A CAPES e a FAPESP pelo auxílio financeiro.

Resumo

SIMÕES, Rodolfo da Silva. **Técnicas de transferência de aprendizagem aplicadas a modelos QSAR para regressão**. 2018. 142 f. Dissertação (Mestrado em Ciências) – Escola de Artes, Ciências e Humanidades, Universidade de São Paulo, São Paulo, 2018.

Para desenvolver um novo medicamento, pesquisadores devem analisar os alvos biológicos de uma dada doença, descobrir e desenvolver candidatos a fármacos para este alvo biológico, realizando em paralelo, testes em laboratório para validar a eficiência e os efeitos colaterais da substância química. O estudo quantitativo da relação estrutura-atividade (QSAR) envolve a construção de modelos de regressão que relacionam um conjunto de descritores de um composto químico e a sua atividade biológica com relação a um ou mais alvos no organismo. Os conjuntos de dados manipulados pelos pesquisadores para análise QSAR são caracterizados geralmente por um número pequeno de instâncias e isso torna mais complexa a construção de modelos preditivos. Nesse contexto, a transferência de conhecimento utilizando informações de outros modelos QSAR's com mais dados disponíveis para o mesmo alvo biológico seria desejável, diminuindo o esforço e o custo do processo para gerar novos modelos de descritores de compostos químicos. Este trabalho apresenta uma abordagem de transferência de aprendizagem indutiva (por parâmetros), tal proposta baseia-se em uma variação do método de Regressão por Vetores Suporte adaptado para transferência de aprendizagem, a qual é alcançada ao aproximar os modelos gerados separadamente para cada tarefa em questão. Considera-se também um método de transferência de aprendizagem por instâncias, denominado de TrAdaBoost. Resultados experimentais mostram que as abordagens de transferência de aprendizagem apresentam bom desempenho quando aplicadas a conjuntos de dados de *benchmark* e a conjuntos de dados químicos.

Palavras-chaves: Aprendizagem de máquina; Transferência de aprendizagem; Regressão por vetores suporte; Modelos QSAR; Quimioinformática.

Abstract

SIMÕES, Rodolfo da Silva. **Transfer learning techniques applied to QSAR models for regression**. 2018. 142 p. Dissertation (Master of Science) – School of Arts, Sciences and Humanities, University of São Paulo, São Paulo, 2018.

To develop a new medicament, researches must analyze the biological targets of a given disease, discover and develop drug candidates for this biological target, performing in parallel, biological tests in laboratory to validate the effectiveness and side effects of the chemical substance. The quantitative study of structure-activity relationship (QSAR) involves building regression models that relate a set of descriptors of a chemical compound and its biological activity with respect to one or more targets in the organism. Datasets manipulated by researchers to QSAR analysis are generally characterized by a small number of instances and this makes it more complex to build predictive models. In this context, the transfer of knowledge using information other's QSAR models with more data available to the same biological target would be desirable, since it reduces the effort and cost to generate models of chemical descriptors. This work presents an inductive learning transfer approach (by parameters), such proposal is based on a variation of the Vector Regression method Adapted support for learning transfer, which is achieved by approaching the separately generated models for each task. It is also considered a method of learning transfer by instances, called TrAdaBoost. Experimental results show that learning transfer approaches perform well when applied to some datasets of benchmark and dataset chemical.

Keywords: Machine learning; Transfer learning; Support vector regression; QSAR models; Chemoinformatics.

Lista de figuras

Figura 1 – Visão geral de diferentes configurações de transferência de aprendizagem. Os retângulos em azul representam as categorias da técnica de transferência de aprendizagem; os círculos representam técnicas de aprendizagem de máquina que são relacionadas com cada categoria. Figura adaptada de (PAN; YANG, 2010)	37
Figura 2 – Hiperplano de separação ótimo (em vermelho), tendo como suporte dois outros hiperplanos (linhas tracejadas), os quais passam por alguns pontos para ambas as classes. Estes pontos são chamados de vetores suporte.	45
Figura 3 – Ilustração de vetores suporte para o problema de regressão com as variáveis de folga ξ e ξ^* nos dois lados do hiperplano.	46
Figura 4 – Funções de perda L1 e L2.	47
Figura 5 – Ilustração de vetores suporte para o problema de regressão. Figura adaptada de (ROSENBAUM et al., 2013).	48
Figura 6 – Ilustração do conhecimento transferido a partir da tarefa de origem, que contém uma grande quantidade de dados de treinamento (verde) para uma tarefa de destino, que contém poucos dados de treinamento (azul). Os tubos ϵ -sensível ao redor das funções de regressão $\mathbf{w}_s^T \mathbf{x}$ e $\mathbf{w}_t^T \mathbf{x}$ são descritos pelas linhas tracejadas.	50
Figura 7 – Objetivo do algoritmo TrAdaBoost.	54
Figura 8 – Processo de identificação de compostos químicos via QSAR.	57
Figura 9 – Grafo regularizado multitarefa (GRMT). Figura adaptada de (ROSENBAUM et al., 2013).	68
Figura 10 – Configuração dos experimentos com método SVR para dados químicos.	72
Figura 11 – Análise gráfica do desempenho dos métodos de regressão com <i>kernel</i> linear para o conjunto de dados <i>Kin-family</i>	76
Figura 12 – Análise gráfica do desempenho dos métodos de regressão com <i>kernel</i> linear para o conjunto de dados <i>Kin-family</i>	78
Figura 13 – Análise gráfica do desempenho dos métodos de regressão com <i>kernel</i> linear para o conjunto de dados <i>Earthquake</i>	80

Figura 14 – Análise gráfica do desempenho dos métodos de regressão com <i>kernel</i> polinomial para o conjunto de dados <i>Earthquake</i>	81
Figura 15 – Análise gráfica do desempenho dos métodos de regressão com <i>kernel</i> linear para o conjunto de dados <i>Fly</i>	83
Figura 16 – Análise gráfica do desempenho dos métodos de regressão com <i>kernel</i> polinomial para o conjunto de dados <i>Fly</i>	84
Figura 17 – Análise gráfica do desempenho dos métodos de regressão com <i>kernel</i> linear para o conjunto de dados ALK-5.	87
Figura 18 – Análise gráfica do desempenho dos métodos de regressão com <i>kernel</i> polinomial para o conjunto de dados ALK-5.	88
Figura 19 – Análise gráfica do desempenho dos métodos de regressão com <i>kernel</i> linear para o conjunto de dados PPAR- α	90
Figura 20 – Análise gráfica do desempenho dos métodos de regressão com <i>kernel</i> polinomial para o conjunto de dados PPAR- α	91
Figura 21 – Análise gráfica do desempenho dos métodos de regressão com <i>kernel</i> linear para o conjunto de dados PPAR- δ	93
Figura 22 – Análise gráfica do desempenho dos métodos de regressão com <i>kernel</i> polinomial para o conjunto de dados PPAR- δ	94
Figura 23 – Análise gráfica do desempenho dos métodos de regressão com <i>kernel</i> linear para o conjunto de dados DHFR.	96
Figura 24 – Análise gráfica do desempenho dos métodos de regressão com <i>kernel</i> polinomial para o conjunto de dados DHFR.	97
Figura 25 – Análise gráfica do desempenho dos métodos de regressão com <i>kernel</i> linear para o conjunto de dados COX2.	99
Figura 26 – Análise gráfica do desempenho dos métodos de regressão com <i>kernel</i> polinomial para o conjunto de dados COX2.	100
Figura 27 – Análise gráfica do desempenho dos métodos de regressão com <i>kernel</i> linear para o conjunto de dados ER.	102
Figura 28 – Análise gráfica do desempenho dos métodos de regressão com <i>kernel</i> polinomial para o conjunto de dados ER.	103
Figura 29 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.	118

Figura 30 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.	118
Figura 31 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.	119
Figura 32 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.	119
Figura 33 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.	119
Figura 34 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.	120
Figura 35 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.	120
Figura 36 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.	120
Figura 37 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.	121
Figura 38 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.	121
Figura 39 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.	121
Figura 40 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.	122
Figura 41 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.	122
Figura 42 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.	122
Figura 43 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.	123
Figura 44 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.	123
Figura 45 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.	123

Figura 46 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.	124
Figura 47 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.	124
Figura 48 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.	124
Figura 49 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.	125
Figura 50 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.	125
Figura 51 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.	125
Figura 52 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.	126
Figura 53 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.	126
Figura 54 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.	126
Figura 55 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.	127
Figura 56 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.	127
Figura 57 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.	127
Figura 58 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.	128
Figura 59 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.	128
Figura 60 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.	128
Figura 61 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.	129

Figura 62 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.	129
Figura 63 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.	129
Figura 64 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.	130
Figura 65 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.	130
Figura 66 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.	130
Figura 67 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.	131
Figura 68 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.	131
Figura 69 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.	131
Figura 70 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.	132
Figura 71 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.	132
Figura 72 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.	132
Figura 73 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.	133
Figura 74 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.	133
Figura 75 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.	133
Figura 76 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.	134
Figura 77 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.	134

Figura 78 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.	134
Figura 79 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.	135
Figura 80 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.	135
Figura 81 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.	135
Figura 82 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.	136
Figura 83 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.	136
Figura 84 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.	136
Figura 85 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.	137
Figura 86 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.	137
Figura 87 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.	137
Figura 88 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.	138
Figura 89 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.	138
Figura 90 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.	138
Figura 91 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.	139
Figura 92 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.	139
Figura 93 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.	139

Figura 94 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.	140
Figura 95 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.	140
Figura 96 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.	140
Figura 97 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.	141
Figura 98 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.	141
Figura 99 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.	141
Figura 100 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.	142

Lista de algoritmos

Algoritmo 1 – Algoritmo AdaBoost para regressão	52
Algoritmo 2 – Algoritmo TrAdaBoost para regressão	55

Lista de tabelas

Tabela 1 – Categorização de técnicas de transferência de aprendizagem. Tabela adaptada de (PAN; YANG, 2010).	36
Tabela 2 – Diferentes abordagens para transferir o conhecimento, que podem ser empregues em cada uma das categorias de transferência de aprendizagem. Tabela adaptada de (PAN; YANG, 2010).	39
Tabela 3 – Esquema para seleção de instâncias pelo método TrAdaBoost.	74
Tabela 4 – Resultados (em valores de EQM) obtidos pelos métodos de regressão com <i>kernel</i> linear para o conjunto de dados <i>Kin-family</i>	76
Tabela 5 – Resultados (em valores de EQM) obtidos pelos métodos de regressão com <i>kernel</i> polinomial para o conjunto de dados <i>Kin-family</i>	77
Tabela 6 – Resultados (em valores de EQM) obtidos pelos métodos de regressão com <i>kernel</i> linear para o conjunto de dados <i>Earthquake</i>	79
Tabela 7 – Resultados (em valores de EQM) obtidos pelos métodos de regressão com <i>kernel</i> polinomial para o conjunto de dados <i>Earthquake</i>	80
Tabela 8 – Resultados (em valores de EQM) obtidos pelos métodos de regressão com <i>kernel</i> linear para o conjunto de dados <i>Fly</i>	82
Tabela 9 – Resultados (em valores de EQM) obtidos pelos métodos de regressão com <i>kernel</i> polinomial para o conjunto de dados <i>Fly</i>	83
Tabela 10 – Resultados (em valores de EQM) obtidos pelos métodos de regressão com <i>kernel</i> linear para o conjunto de dados ALK-5.	86
Tabela 11 – Resultados (em valores de EQM) obtidos pelos métodos de regressão com <i>kernel</i> polinomial para o conjunto de dados ALK-5.	87
Tabela 12 – Resultados (em valores de EQM) obtidos pelos métodos de regressão com <i>kernel</i> linear para o conjunto de dados PPAR- α	90
Tabela 13 – Resultados (em valores de EQM) obtidos pelos métodos de regressão com <i>kernel</i> polinomial para o conjunto de dados PPAR- α	91
Tabela 14 – Resultados (em valores de EQM) obtidos pelos métodos de regressão com <i>kernel</i> linear para o conjunto de dados PPAR- δ	92
Tabela 15 – Resultados (em valores de EQM) obtidos pelos métodos de regressão com <i>kernel</i> polinomial para o conjunto de dados PPAR- δ	93

Tabela 16 – Resultados (em valores de EQM) obtidos pelos métodos de regressão com <i>kernel</i> linear para o conjunto de dados DHFR.	95
Tabela 17 – Resultados (em valores de EQM) obtidos pelos métodos de regressão com <i>kernel</i> polinomial para o conjunto de dados DHFR.	96
Tabela 18 – Resultados (em valores de EQM) obtidos pelos métodos de regressão com <i>kernel</i> linear para o conjunto de dados COX-2.	98
Tabela 19 – Resultados (em valores de EQM) obtidos pelos métodos de regressão com <i>kernel</i> polinomial para o conjunto de dados COX-2.	99
Tabela 20 – Resultados (em valores de EQM) obtidos pelos métodos de regressão com <i>kernel</i> linear para o conjunto de dados ER.	101
Tabela 21 – Resultados (em valores de EQM) obtidos pelos métodos de regressão com <i>kernel</i> polinomial para o conjunto de dados ER.	102
Tabela 22 – Resultados experimentais com <i>kernel</i> linear: conjunto <i>Kin-family</i> com 50 instâncias de treinamento na tarefa de destino.	118
Tabela 23 – Resultados experimentais com <i>kernel</i> polinomial: conjunto <i>Kin-family</i> com 50 instâncias de treinamento na tarefa de destino.	118
Tabela 24 – Resultados experimentais com <i>kernel</i> linear: conjunto <i>Kin-family</i> com 100 instâncias de treinamento na tarefa de destino.	119
Tabela 25 – Resultados experimentais com <i>kernel</i> polinomial: conjunto <i>Kin-family</i> com 100 instâncias de treinamento na tarefa de destino.	119
Tabela 26 – Resultados experimentais com <i>kernel</i> linear: conjunto <i>Kin-family</i> com 150 instâncias de treinamento na tarefa de destino.	119
Tabela 27 – Resultados experimentais com <i>kernel</i> polinomial: conjunto <i>Kin-family</i> com 150 instâncias de treinamento na tarefa de destino.	120
Tabela 28 – Resultados experimentais com <i>kernel</i> linear: conjunto <i>Kin-family</i> com 200 instâncias de treinamento na tarefa de destino.	120
Tabela 29 – Resultados experimentais com <i>kernel</i> polinomial: conjunto <i>Kin-family</i> com 200 instâncias de treinamento na tarefa de destino.	120
Tabela 30 – Resultados experimentais com <i>kernel</i> linear: conjunto <i>Earthquake</i> com 20 instâncias de treinamento na tarefa de destino.	121
Tabela 31 – Resultados experimentais com <i>kernel</i> polinomial: conjunto <i>Earthquake</i> com 20 instâncias de treinamento na tarefa de destino.	121

Tabela 32 – Resultados experimentais com <i>kernel</i> linear: conjunto <i>Earthquake</i> com 30 instâncias de treinamento na tarefa de destino.	121
Tabela 33 – Resultados experimentais com <i>kernel</i> polinomial: conjunto <i>Earthquake</i> com 30 instâncias de treinamento na tarefa de destino.	122
Tabela 34 – Resultados experimentais com <i>kernel</i> linear: conjunto <i>Earthquake</i> com 50 instâncias de treinamento na tarefa de destino.	122
Tabela 35 – Resultados experimentais com <i>kernel</i> polinomial: conjunto <i>Earthquake</i> com 50 instâncias de treinamento na tarefa de destino.	122
Tabela 36 – Resultados experimentais com <i>kernel</i> linear: conjunto <i>Earthquake</i> com 70 instâncias de treinamento na tarefa de destino.	123
Tabela 37 – Resultados experimentais com <i>kernel</i> polinomial: conjunto <i>Earthquake</i> com 70 instâncias de treinamento na tarefa de destino.	123
Tabela 38 – Resultados experimentais com <i>kernel</i> linear: conjunto <i>Fly</i> com 50 instâncias de treinamento na tarefa de destino.	123
Tabela 39 – Resultados experimentais com <i>kernel</i> polinomial: conjunto <i>Fly</i> com 50 instâncias treinamento na tarefa de destino.	124
Tabela 40 – Resultados experimentais com <i>kernel</i> linear: conjunto <i>Fly</i> com 200 instâncias de treinamento na tarefa de destino.	124
Tabela 41 – Resultados experimentais com <i>kernel</i> polinomial: conjunto <i>Fly</i> com 200 instâncias de treinamento na tarefa de destino.	124
Tabela 42 – Resultados experimentais com <i>kernel</i> linear: conjunto <i>Fly</i> com 400 instâncias de treinamento na tarefa de destino.	125
Tabela 43 – Resultados experimentais com <i>kernel</i> polinomial: conjunto <i>Fly</i> com 400 instâncias de treinamento na tarefa de destino.	125
Tabela 44 – Resultados experimentais com <i>kernel</i> linear: conjunto <i>Fly</i> com 800 instâncias de treinamento na tarefa de destino.	125
Tabela 45 – Resultados experimentais com <i>kernel</i> polinomial: conjunto <i>Fly</i> com 800 instâncias de treinamento na tarefa de destino.	126
Tabela 46 – Resultados experimentais com <i>kernel</i> linear: conjunto ALK-5 com 25% dos dados de treinamento na tarefa de destino.	126
Tabela 47 – Resultados experimentais com <i>kernel</i> polinomial: conjunto ALK-5 com 25% dos dados de treinamento na tarefa de destino.	126

Tabela 48 – Resultados experimentais com <i>kernel</i> linear: conjunto ALK-5 com 50% dos dados de treinamento na tarefa de destino.	127
Tabela 49 – Resultados experimentais com <i>kernel</i> polinomial: conjunto ALK-5 com 50% dos dados de treinamento na tarefa de destino.	127
Tabela 50 – Resultados experimentais com <i>kernel</i> linear: conjunto ALK-5 com 75% dos dados de treinamento na tarefa de destino.	127
Tabela 51 – Resultados experimentais com <i>kernel</i> polinomial: conjunto ALK-5 com 75% dos dados de treinamento na tarefa de destino.	128
Tabela 52 – Resultados experimentais com <i>kernel</i> linear: conjunto ALK-5 com 100% dos dados de treinamento na tarefa de destino.	128
Tabela 53 – Resultados experimentais com <i>kernel</i> polinomial: conjunto ALK-5 com 100% dos dados de treinamento na tarefa de destino.	128
Tabela 54 – Resultados experimentais com <i>kernel</i> linear: conjunto PPAR- α com 25% dos dados de treinamento na tarefa de destino.	129
Tabela 55 – Resultados experimentais: conjunto alpha - método TrAdaBoost variando o tamanho do conjunto de treinamento da tarefa de origem e com 25% dos dados de treinamento na tarefa de destino.	129
Tabela 56 – Resultados experimentais com <i>kernel</i> linear: conjunto PPAR- α com 50% dos dados de treinamento na tarefa de destino.	129
Tabela 57 – Resultados experimentais com <i>kernel</i> polinomial: conjunto PPAR- α com 50% dos dados de treinamento na tarefa de destino.	130
Tabela 58 – Resultados experimentais com <i>kernel</i> linear: conjunto PPAR- α com 75% dos dados de treinamento na tarefa de destino.	130
Tabela 59 – Resultados experimentais com <i>kernel</i> polinomial: conjunto PPAR- α com 75% dos dados de treinamento na tarefa de destino.	130
Tabela 60 – Resultados experimentais com <i>kernel</i> linear: conjunto PPAR- α com 100% dos dados de treinamento na tarefa de destino.	131
Tabela 61 – Resultados experimentais com <i>kernel</i> polinomial: conjunto PPAR- α com 100% dos dados de treinamento na tarefa de destino.	131
Tabela 62 – Resultados experimentais com <i>kernel</i> linear: conjunto PPAR- δ com 25% dos dados de treinamento na tarefa de destino.	131
Tabela 63 – Resultados experimentais com <i>kernel</i> polinomial: conjunto PPAR- δ com 25% dos dados de treinamento na tarefa de destino.	132

Tabela 64 – Resultados experimentais com <i>kernel</i> linear: conjunto PPAR- δ com 50% dos dados de treinamento na tarefa de destino.	132
Tabela 65 – Resultados experimentais com <i>kernel</i> polinomial: conjunto PPAR- δ com 50% dos dados de treinamento na tarefa de destino.	132
Tabela 66 – Resultados experimentais com <i>kernel</i> linear: conjunto PPAR- δ com 75% dos dados de treinamento na tarefa de destino.	133
Tabela 67 – Resultados experimentais com <i>kernel</i> polinomial: conjunto PPAR- δ com 75% dos dados de treinamento na tarefa de destino.	133
Tabela 68 – Resultados experimentais com <i>kernel</i> linear: conjunto PPAR- δ com 100% dos dados de treinamento na tarefa de destino.	133
Tabela 69 – Resultados experimentais com <i>kernel</i> polinomial: conjunto PPAR- δ com 100% dos dados de treinamento na tarefa de destino.	134
Tabela 70 – Resultados experimentais com <i>kernel</i> linear: conjunto DHFR com 25% dos dados de treinamento na tarefa de destino.	134
Tabela 71 – Resultados experimentais com <i>kernel</i> polinomial: conjunto DHFR com 25% dos dados de treinamento na tarefa de destino.	134
Tabela 72 – Resultados experimentais com <i>kernel</i> linear: conjunto DHFR com 50% dos dados de treinamento na tarefa de destino.	135
Tabela 73 – Resultados experimentais com <i>kernel</i> polinomial: conjunto DHFR com 50% dos dados de treinamento na tarefa de destino.	135
Tabela 74 – Resultados experimentais com <i>kernel</i> linear: conjunto DHFR com 75% dos dados de treinamento na tarefa de destino.	135
Tabela 75 – Resultados experimentais com <i>kernel</i> polinomial: conjunto DHFR com 75% dos dados de treinamento na tarefa de destino.	136
Tabela 76 – Resultados experimentais com <i>kernel</i> linear: conjunto DHFR com 100% dos dados de treinamento na tarefa de destino.	136
Tabela 77 – Resultados experimentais com <i>kernel</i> polinomial: conjunto DHFR com 100% dos dados de treinamento na tarefa de destino.	136
Tabela 78 – Resultados experimentais com <i>kernel</i> linear: conjunto COX2 com 25% dos dados de treinamento na tarefa de destino.	137
Tabela 79 – Resultados experimentais com <i>kernel</i> polinomial: conjunto COX2 com 25% dos dados de treinamento na tarefa de destino.	137

Tabela 80 – Resultados experimentais com <i>kernel</i> linear: conjunto COX2 com 50% dos dados de treinamento na tarefa de destino.	137
Tabela 81 – Resultados experimentais com <i>kernel</i> polinomial: conjunto COX2 com 50% dos dados de treinamento na tarefa de destino.	138
Tabela 82 – Resultados experimentais com <i>kernel</i> linear: conjunto COX2 com 75% dos dados de treinamento na tarefa de destino.	138
Tabela 83 – Resultados experimentais com <i>kernel</i> polinomial: conjunto COX2 com 75% dos dados de treinamento na tarefa de destino.	138
Tabela 84 – Resultados experimentais com <i>kernel</i> linear: conjunto COX2 com 100% dos dados de treinamento na tarefa de destino.	139
Tabela 85 – Resultados experimentais com <i>kernel</i> polinomial: conjunto COX2 com 100% dos dados de treinamento na tarefa de destino.	139
Tabela 86 – Resultados experimentais com <i>kernel</i> linear: conjunto ER com 25% dos dados de treinamento na tarefa de destino.	139
Tabela 87 – Resultados experimentais com <i>kernel</i> polinomial: conjunto ER com 25% dos dados de treinamento na tarefa de destino.	140
Tabela 88 – Resultados experimentais com <i>kernel</i> linear: conjunto ER com 50% dos dados de treinamento na tarefa de destino.	140
Tabela 89 – Resultados experimentais com <i>kernel</i> polinomial: conjunto ER com 50% dos dados de treinamento na tarefa de destino.	140
Tabela 90 – Resultados experimentais com <i>kernel</i> linear: conjunto ER com 75% dos dados de treinamento na tarefa de destino.	141
Tabela 91 – Resultados experimentais com <i>kernel</i> polinomial: conjunto ER com 75% dos dados de treinamento na tarefa de destino.	141
Tabela 92 – Resultados experimentais com <i>kernel</i> linear: conjunto ER com 100% dos dados de treinamento na tarefa de destino.	141
Tabela 93 – Resultados experimentais com <i>kernel</i> polinomial: conjunto ER com 100% dos dados de treinamento na tarefa de destino.	142

Lista de abreviaturas e siglas

AdaBoost	<i>Adaptative Boosting</i>
CDSC	Cross-Domain Classifier Spectral
DBT	Transferencia Baseado em Discriminabilidade
EMG	Eletromiografia
EQM	Erro Quadrático Médio
FSR	Remapeamento do Espaço de Características
GP	Processos Gaussianos
GRMT	Grafo Regularizado Multitarefa
KNN	K-Vizinhos Próximos
MCDTL	Multiple Cross-Domain Transfer Learning
MLP	Redes Neurais Perceptron Multicamadas
MLN	Redes Lógicas de Markov
MT-IVM	Máquinas de Vetores Informativos Multitarefa
MVTL-LM	Transferência de Aprendizagem com a abordagem Margem Grande
NIPS	<i>Neural Information Processing Systems</i>
OMS-TL	Transferência de Aprendizagem de Múltiplas Origens Online
OVact	Medida de Sobreposição de Atividade
QM	Química Medicinal
QSAR	Relação Quantitativa da Estrutura Atividade
SAR	Relação Estrutura Atividade
SVM	Máquinas de Vetores Suporte
SVR	Regressão por Máquinas de Vetores Suporte

TDMT Multitarefa Top-down

TrAdaBoost Transfer AdaBoost

TTL Transferência de Aprendizagem Transitiva

Lista de símbolos

Γ Letra grega Gama

α Letra grega alpha

δ Letra grega delta

χ Letra grega chi

$\in$ Pertence

$\mathbb{R}$ Números reais

Sumário

1	Introdução	28
1.1	<i>Contextualização e motivação</i>	29
1.2	<i>Hipóteses</i>	31
1.3	<i>Objetivos</i>	32
1.4	<i>Estrutura do trabalho</i>	32
2	Transferência de aprendizagem	33
2.1	<i>Notações e definições</i>	33
2.2	<i>Categorização de técnicas de transferência de aprendizagem</i>	34
2.3	<i>Transferência de aprendizagem indutiva</i>	39
2.3.1	Transferência de conhecimento baseado em instância	39
2.3.2	Transferência de conhecimento por representação de características	40
2.3.3	Transferência de conhecimento por parâmetros	41
2.3.4	Transferência de conhecimento relacional	41
2.4	<i>Aplicações de transferência de aprendizagem</i>	42
3	Transferência de aprendizagem em modelos de regressão por vetores suporte	44
3.1	<i>Máquinas de Vetores Suporte (SVM)</i>	44
3.2	<i>Regressão por Vetores Suporte (SVR)</i>	45
3.2.1	Formulação da técnica SVR	45
3.3	<i>Técnica SVR adaptada para transferência de aprendizagem (ATL)</i>	49
3.4	<i>Transferência de aprendizagem por instância</i>	51
3.4.1	Algoritmo TrAdaBoost para regressão	53
3.5	<i>Definição do problema</i>	56
3.5.1	Seleção de conjuntos de dados relacionados	58
3.5.2	Aplicação de transferência de aprendizagem em modelos QSAR para regressão	59
3.5.3	Esquema de avaliação de desempenho	60
4	Revisão bibliográfica	62

4.1	<i>Trabalhos relacionados na área de Química Medicinal</i>	66
5	Análises experimentais	70
5.1	<i>Considerações iniciais</i>	70
5.2	<i>Configuração dos experimentos com o método SVR</i>	71
5.3	<i>Configuração dos experimentos com transferência de aprendizagem</i>	72
5.3.1	Configuração dos experimentos com o método SVR-ATL	72
5.3.2	Configuração dos experimentos com o método TrAdaBoost	73
5.4	<i>Resultados experimentais com dados de benchmark</i>	74
5.4.1	Conjunto de dados <i>Kin-family</i>	75
5.4.2	Conjunto de dados <i>Earthquake</i>	78
5.4.3	Conjunto de dados <i>Fly</i>	81
5.5	<i>Resultados experimentais com dados químicos</i>	85
5.5.1	Conjunto de dados ALK-5	85
5.5.2	Conjunto de dados PPAR- α	88
5.5.3	Conjunto de dados PPAR- δ	92
5.5.4	Conjunto de dados DHFR	95
5.5.5	Conjunto de dados COX2	97
5.5.6	Conjunto de dados ER	100
5.6	<i>Considerações finais</i>	103
6	Conclusões	105
6.1	<i>Contribuições do trabalho</i>	106
6.2	<i>Trabalhos futuros</i>	107
	Referências¹	108
	Apêndice A – Problema de otimização SVR-ATL	114
	Apêndice B – Resultados TrAdaBoost	118
B.1	<i>Conjunto de dados Kin-family</i>	118
B.2	<i>Conjunto de dados Earthquake</i>	121
B.3	<i>Conjunto de dados Fly</i>	123

¹ De acordo com a Associação Brasileira de Normas Técnicas. NBR 6023.

<i>B.4</i>	<i>Conjunto de dados ALK-5</i>	126
<i>B.5</i>	<i>Conjunto de dados PPAR-α</i>	129
<i>B.6</i>	<i>Conjunto de dados PPAR-δ</i>	131
<i>B.7</i>	<i>Conjunto de dados DHFR</i>	134
<i>B.8</i>	<i>Conjunto de dados COX2</i>	137
<i>B.9</i>	<i>Conjunto de dados ER</i>	139

1 Introdução

Técnicas de mineração de dados e aprendizagem de máquina, incluindo métodos de classificação, regressão e agrupamento, têm alcançado sucesso significativo em muitas áreas do conhecimento. Entretanto, muitos desses métodos funcionam bem apenas sob a suposição de que os dados de treinamento e teste são oriundos do mesmo espaço de características e seguem a mesma distribuição de probabilidade (WANG; LI, 2014). Quando ocorrem alterações na distribuição dos dados, muitos modelos estatísticos precisam ser reconstruídos a partir da coleta de novos dados de treinamento. No entanto, em muitas aplicações do mundo real, é custoso ou impossível recoletar os dados necessários para reconstruir esses modelos. Em tais situações, a transferência de aprendizagem¹ (conhecida também como transferência de conhecimento) entre domínios seria desejável, uma vez que tal abordagem permite que domínios, tarefas e distribuições de probabilidade consideradas nas fases de treinamento e teste sejam diferentes (PAN; YANG, 2010).

De maneira geral, o desenvolvimento da transferência de aprendizagem foi motivado pelo fato de que é possível aplicar o conhecimento adquirido anteriormente para resolver novos problemas de forma mais rápida e com melhores soluções. Como exemplo, é possível considerar que aprender a reconhecer maçãs pode ajudar a reconhecer pêras ou que o conhecimento adquirido para tocar violão pode ajudar no processo de aprendizagem para tocar guitarra. Pode-se encontrar muitos outros exemplos em áreas de conhecimento em que a transferência de aprendizagem agrega benefícios. Destaca-se, por exemplo, o problema de classificação de sentimentos, em que a tarefa é classificar automaticamente comentários sobre um determinado produto, como uma câmera fotográfica, em positivo ou negativo. Para essa tarefa de classificação, inicialmente é preciso coletar muitos comentários sobre o produto e após isso, é necessário treinar um classificador para os dados coletados com seus respectivos rótulos. Uma vez que a distribuição de probabilidade dos comentários sobre outros produtos pode ser muito diferente, para o classificador manter um desempenho satisfatório, é necessário refazer o processo para cada produto. Para reduzir esse esforço, é possível adaptar um modelo de classificação que foi treinado em alguns produtos para ajudar a aprendizagem de modelos de classificação para outros.

De maneira geral, abordagens de transferência de aprendizagem têm se mostrado promissoras para combinar o conhecimento previamente obtido em tarefas relacionadas

¹ Do original, em inglês, *Transfer Learning*.

em um único modelo preditivo, seja este para classificação, regressão ou agrupamento (PAN; YANG, 2010). Em particular, pesquisas na área de Química Medicinal (QM) para descoberta de novos medicamentos têm se beneficiado com a transferência de aprendizagem, como pode ser visto em (ROSENBAUM et al., 2013) e (GIRSCHICK; RÜCKERT; KRAMER, 2012), e são o foco de aplicação do presente trabalho.

1.1 Contextualização e motivação

O desenvolvimento de medicamentos não é uma tarefa fácil, pois depende de vários fatores para o seu sucesso. Segundo Young (2009), alguns projetos são caros demais para serem viáveis ou, em alguns casos, os compostos podem ser tóxicos ou não apresentam atividade biológica necessária para a produção de um novo fármaco. Isso faz com que projetos de busca por novos medicamentos possam demorar cerca de dez anos para serem concluídos.

Para desenvolver um novo medicamento, pesquisadores devem analisar os alvos biológicos de uma dada doença, descobrir e desenvolver candidatos a fármacos para estes alvos, realizando em paralelo, testes biológicos em laboratório para validar a eficiência e os efeitos colaterais da substância química. Com o objetivo de evitar altos custos na produção e análise dessas substâncias, são gerados bancos de dados contendo informações de moléculas, e assim, modelos computacionais podem ser construídos a partir desses dados (YOUNG, 2009).

Dentre as diversas atividades existentes no processo de descoberta de novos medicamentos, há uma estratégia de análise de dados que propõe o estabelecimento de relações entre a estrutura química e a atividade biológica de uma série de moléculas (SAR)². O estudo quantitativo da relação estrutura-atividade (QSAR)³ envolve a construção de um modelo que descreve quantitativamente um conjunto de descritores de um composto químico e a sua atividade biológica com relação a um ou mais alvos no organismo (DEVILLERS; BALABAN, 2000).

Análises QSAR são amplamente utilizadas para estudar as propriedades químicas de um grupo de compostos com o objetivo de prever seus níveis de atividade biológica. Para essa finalidade, técnicas de aprendizagem de máquina podem ser aplicadas para

² Do original, em inglês, *Structure-Activity Relationships*.

³ Do original, em inglês, *Quantitative Structure-Activity Relationships*.

estimar valores que quantificam a atividade biológica (por meio de técnicas de regressão) ou valores de classe (por meio de técnicas de classificação). Neste último caso, pode ser necessário o auxílio de um especialista da área para determinar um limiar que permita classificar os compostos em ativos ou inativos. Caso o conjunto de dados disponível para o treinamento dos modelos apresente informações de rótulos (classes ou valores de atividades reais, para tarefas de classificação e regressão, respectivamente), é possível utilizar técnicas de aprendizagem supervisionadas. Destacam-se como métodos nessa categoria as Máquinas de Vetores Suporte (SVM)⁴ (CRISTIANINI; SHAW-TAYLOR, 2000), Árvores de Decisão e Redes Neurais Perceptron Multicamadas (MLP)⁵ (MITCHELL, 1997).

Michielan e Moro (2010) afirmam que os estudos QSAR são muito custosos, tendo em vista que a transformação dos descritores químicos em conjuntos de dados numéricos não é uma tarefa trivial. Além disso, Ning e Karypis (2011) discutem que grande parte dos compostos químicos não possui rótulos pré-definidos e que as amostras que possuem compostos rotulados são muito pequenas, o que pode gerar um modelo preditivo ineficiente. Em tais situações, seria desejável utilizar informações de outros conjuntos de dados químicos já existentes para melhorar o desempenho de um modelo QSAR relacionado e construído a partir de pouca informação. Em cenários como esses, técnicas de transferência de aprendizagem poderiam ser aplicadas para diminuir o esforço dos pesquisadores e o custo do processo para gerar novos conjuntos de descritores químicos. Dessa forma, é possível aproveitar o conhecimento de domínios de origem para ajudar a construir um modelo para um domínio de destino, em que os dados disponíveis são escassos (YAO; DORETTO, 2010).

Rosenbaum et al. (2013) afirmam que a transferência de aprendizagem geralmente é benéfica se aplicada a partir de um conjunto de dados com maior quantidade de instâncias disponíveis para um conjunto de dados com pouca quantidade de instâncias disponíveis. Entretanto, para uma transferência de aprendizagem válida, é necessário que os domínios de origem e de destino sejam relacionados (PAN; YANG, 2010). Se nenhuma relação direta for encontrada, a transferência forçada não vai funcionar, resultando em nenhuma melhora ou até mesmo piorando o desempenho no domínio de destino (FITZGERALD; THOMAZ, 2015).

⁴ Do original, em inglês, *Support Vector Machine*.

⁵ Do original, em inglês, *Multilayer Perceptron*.

Para uma seleção automática dos conjuntos de dados necessários para a aplicação de técnicas de transferência de aprendizagem, [Girschick, Rückert e Kramer \(2012\)](#) propuseram um método de seleção de conjuntos de forma dirigida, que se baseia no ranqueamento dos conjuntos relacionados, de acordo com a similaridade destes em relação ao conjunto de destino. Assim, o objetivo é selecionar o conjunto de origem mais semelhante ao de destino. Este método é considerado em alguns dos experimentos do presente trabalho.

De forma geral, este trabalho analisa a aplicação de duas técnicas de transferência de aprendizagem a modelos QSAR para regressão baseados no método de Regressão por Máquinas de Vetores Suporte (SVR)⁶.

1.2 Hipóteses

Este projeto de mestrado tomará como base as seguintes hipóteses:

- Para muitos algoritmos de aprendizagem de máquina, supõe-se que os dados de treinamento e teste devem pertencer ao mesmo espaço de características e devem seguir a mesma distribuição de probabilidade ([PAN; YANG, 2010](#)). No presente trabalho, espera-se que as abordagens de transferência de aprendizagem possam ser aplicadas em situações em que tais suposições não são válidas;
- Por meio do método de seleção de conjuntos de dados proposto por [Girschick, Rückert e Kramer \(2012\)](#) pretende-se selecionar conjuntos de dados relacionados ao conjunto de destino a partir de uma base de dados pública para a aplicação das técnicas de transferência de aprendizagem;
- O desempenho do método de regressão SVR aplicado somente ao conjunto de dados de destino deve ser inferior ao desempenho obtido com auxílio das abordagens de transferência de aprendizagem que utilizarão o conhecimento de modelos construídos a partir de conjuntos de dados de origem;
- A aplicação das técnicas de transferência de aprendizagem contribuirá para superação de limitações referentes à análise de dados de natureza química.

⁶ Do original, em inglês, *Support Vector Regression*.

1.3 Objetivos

O principal objetivo deste trabalho é avaliar a aplicabilidade de técnicas de transferência de aprendizagem na construção de modelos QSAR para regressão, em especial com o método SVR. Para isso, os seguintes objetivos específicos devem ser alcançados:

- Avaliar o desempenho do método SVR para construção de modelos QSAR para regressão;
- Avaliar a incorporação de duas técnicas de transferência de aprendizagem a modelos QSAR para regressão via SVR. O primeiro método consiste na transferência de aprendizagem por parâmetros, proposto no presente trabalho. O segundo método consiste na abordagem de transferência de aprendizagem por instâncias para problemas de regressão, desenvolvido por [Pardoe e Stone \(2010\)](#).
- Avaliar o benefício da transferência de aprendizagem para diversos cenários. Para isso, experimentos são conduzidos variando a quantidade de dados de treinamento disponíveis na tarefa de destino.

1.4 Estrutura do trabalho

O presente trabalho é organizado da forma como segue. O Capítulo 2 apresenta uma descrição da abordagem de transferência de aprendizagem, bem como definições, notações e aplicações da técnica. No Capítulo 3, apresenta-se o método SVR, a sua versão adaptada e proposta neste trabalho para transferência de aprendizagem e o algoritmo de transferência de aprendizagem por instâncias. A definição do problema, ou seja, a aplicação de técnicas de transferência de aprendizagem em modelos QSAR para regressão também é apresentada. Estão inclusos também nesse capítulo, o método que será utilizado para selecionar os conjuntos de dados de natureza química de forma dirigida e o esquema de avaliação empregue nos experimentos. O Capítulo 4 fornece uma revisão bibliográfica sobre técnicas de transferência de aprendizagem e suas aplicações na área de QM. O Capítulo 5 descreve as configurações experimentais, os conjuntos de dados utilizados nos experimentos, os resultados obtidos e discute o desempenho de cada método. Por fim, no Capítulo 6, são apresentadas as conclusões e contribuições do presente trabalho, além de sugestões para trabalhos futuros.

2 Transferência de aprendizagem

Define-se transferência de aprendizagem como sendo a habilidade de um sistema reconhecer e aplicar o conhecimento aprendido em tarefas anteriores para a solução de novos problemas relacionados (PAN; YANG, 2010). Portanto, o objetivo dessa abordagem consiste em extrair o conhecimento obtido por um modelo a partir de uma ou mais tarefas de origem e aplicá-lo para uma tarefa destino. Uma das premissas para a utilização da técnica de transferência de aprendizagem é que os domínios de origem e de destino devem ser relacionados. Tan et al. (2015) afirmam que essa relação pode ser concretizada na forma de instâncias (BICKEL; SAWADE; SCHEFFER, 2009) ou características (SATPAL; SARAWAGI, 2007). Se nenhuma relação direta for encontrada, a transferência forçada não vai funcionar, resultando em nenhuma melhora ou piorando o desempenho no domínio de destino (FITZGERALD; THOMAZ, 2015).

2.1 Notações e definições

Nesta seção, são apresentadas algumas notações e definições sobre os conceitos de domínio e tarefa que serão utilizadas neste trabalho. Um domínio D consiste de dois componentes: um espaço de características χ e uma distribuição de probabilidade marginal $P(\mathbf{X})$, em que $\mathbf{X} = \{\mathbf{x}_1, \dots, \mathbf{x}_n\} \in \chi$. Por exemplo, se o problema de aprendizagem é a classificação de documentos, e cada termo no documento é considerado uma característica binária, então χ é o espaço de todos os vetores de termos, $\mathbf{x}_i$ é o i -ésimo vetor de termos correpndente a algum documento e $\mathbf{X}$ é uma amostra em particular. Em geral, se dois domínios são diferentes, então estes possuem espaços de características ou distribuições de probabilidade marginais diferentes.

Dado um domínio específico, $D = \{\chi, P(\mathbf{X})\}$, uma tarefa (denotada por $T = \{Y, f(\cdot)\}$) consiste de dois componentes: um espaço de rótulos Y e uma função de predição $f(\cdot)$, que pode ser aprendida a partir dos dados de treinamento, que, por sua vez, consistem em pares $(\mathbf{x}_i, y_i)$, em que $\mathbf{x}_i \in \mathbf{X}$ e $y_i \in Y$. A função $f(\cdot)$ pode ser usada para prever o rótulo correspondente a uma nova instância $\mathbf{x}$, $(f(\mathbf{x}))$. Do ponto de vista probabilístico, $f(\mathbf{x})$ pode ser escrita como $P(y|\mathbf{x})$. No exemplo de classificação de documentos, Y é o

conjunto de todos os rótulos possíveis, podendo ser {“Verdadeiro”, “Falso”} para uma tarefa de classificação binária.

Para simplificar a notação, pode-se considerar o caso em que há apenas um domínio de origem D_S e um domínio de destino D_T . Mais especificamente, denota-se o domínio de origem como $D_S = \{(\mathbf{x}_{S1}, y_{S1}), \dots, (\mathbf{x}_{Sm}, y_{Sm})\}$, em que $\mathbf{x}_{Si} \in \chi_S$ é uma instância e $y_{Si} \in Y_S$ é o valor do rótulo correspondente. No exemplo de classificação de documentos, D_S pode ser um conjunto de vetores de termos com seus respectivos rótulos associados. Similarmente, $D_T = \{(\mathbf{x}_{T1}, y_{T1}), \dots, (\mathbf{x}_{Tn}, y_{Tn})\}$, em que $\mathbf{x}_{Ti} \in \chi_T$ é uma instância e $y_{Ti} \in Y_T$ é o valor do rótulo de classe correspondente. Em geral, $0 \leq n \ll m$, em que n e m são o número de instâncias no domínio de destino e de origem, respectivamente.

Dado um domínio D_S , uma tarefa de aprendizado T_S , um domínio de destino D_T e uma tarefa de aprendizado T_T , a transferência de aprendizagem tem o objetivo de ajudar a melhorar o aprendizado da função preditiva do destino $f_T(\cdot)$ no domínio D_T usando o conhecimento de D_S e T_S , em que $D_S \neq D_T$ ou $T_S \neq T_T$ (PAN; YANG, 2010). Com base nesta definição, domínio é um par $D = \{\chi, P(\mathbf{X})\}$. Assim, a condição $D_S \neq D_T$ implica que $\chi_S \neq \chi_T$ ou $P(\mathbf{X}_S) \neq P(\mathbf{X}_T)$. Similarmente, o conceito de tarefa é definido como um par $T = \{Y, P(Y|\mathbf{X})\}$. Portanto, a condição $T_S \neq T_T$ implica que $Y_S \neq Y_T$ ou $P(Y_S|\mathbf{X}_S) \neq P(Y_T|\mathbf{X}_T)$. Caso $D_S = D_T$ e $T_S = T_T$, o problema torna-se um problema tradicional de aprendizagem de máquina.

É importante ressaltar que técnicas de aprendizagem conhecidas como multitarefa estão intimamente relacionadas à transferência de aprendizagem, uma vez que tentam aprender várias tarefas simultaneamente, mesmo quando estas são diferentes. Díspar do método de aprendizagem multitarefa, que aprende as tarefas origem e destino simultaneamente, a transferência de aprendizagem foca apenas a aprendizagem na tarefa de destino.

2.2 Categorização de técnicas de transferência de aprendizagem

Pan e Yang (2010) categorizaram as técnicas de transferência de aprendizagem em: indutiva, transdutiva ou sem supervisão, com base nos possíveis cenários em que as mesmas podem ser aplicadas. A seguir, é apresentada uma breve descrição sobre cada categoria.

Na transferência de aprendizagem indutiva, a tarefa de destino é diferente da tarefa de origem e os domínios são os mesmos. Sendo $T_S \neq T_T$, então: 1) os espaços de rótulos são diferentes ($Y_S \neq Y_T$) ou 2) a distribuição de probabilidade condicional (função de predição) entre os domínios são diferentes ($P(Y_S|\mathbf{X}_S) \neq P(Y_T|\mathbf{X}_T)$, ou seja: $f_S(\cdot) \neq f_T(\cdot)$). Seguindo o exemplo de classificação de documentos utilizado na Seção 2.1, o caso 1 corresponde por exemplo, ao cenário em que o domínio de origem possui classificação binária e o domínio de destino possui classificação multiclasse. O caso 2 corresponde à situação em que a classificação dos documentos é desbalanceada em um dos domínios.

A transferência de aprendizagem indutiva é semelhante a abordagem aprendizagem multitarefa quando há dados rotulados disponíveis em ambos os domínios, porém o foco da aprendizagem indutiva é restrito apenas ao domínio de destino.

Na transferência de aprendizagem transdutiva, as tarefas de origem e destino são as mesmas e os domínios de origem e destino são diferentes ($\chi_S \neq \chi_T$ ou $P(\mathbf{X}_S) \neq P(\mathbf{X}_T)$). Geralmente, nessa categoria não há dados rotulados disponíveis no domínio de destino e há muitos dados rotulados disponíveis no domínio de origem. A título de exemplo:

- Os espaços de características entre os domínios de origem e destino são diferentes ($\chi_S \neq \chi_T$). Lembrando o exemplo sobre classificação de documentos descrito na Seção 2.1, pode-se considerar o caso em que as informações disponíveis para categorizar os documentos são diferentes, ou seja, para um documento específico mapeou-se mais características para o experimento;
- Os espaços de características entre os domínios são os mesmos, mas as distribuições de probabilidade marginal dos dados de entrada são diferentes ($P(\mathbf{X}_S) \neq P(\mathbf{X}_T)$). Seguindo o mesmo exemplo da classificação de documentos, isso pode corresponder ao caso em que os documentos presentes em cada domínio possuem foco em diferentes temas.

Por fim, a transferência de aprendizagem não supervisionada é similar à configuração de transferência de aprendizagem indutiva (tarefas de origem e destino são diferentes). Entretanto, essa abordagem centraliza-se no aprendizado não supervisionado no domínio de destino, tal como agrupamento, redução de dimensionalidade e estimação de densidade. Portanto, não há dados rotulados disponíveis no treinamento em ambos os domínios. A Tabela 1 apresenta as diferenças existentes entre as categorias discutidas nessa seção.

Tabela 1 – Categorização de técnicas de transferência de aprendizagem. Tabela adaptada de (PAN; YANG, 2010).

Definições de aprendizagem	Domínio de origem e destino	Tarefas de origem e destino
Transferência de aprendizagem indutiva	Os mesmos	Diferentes, mas relacionados
Transferência de aprendizagem transdutiva	Diferentes, mas relacionados	Os mesmos
Transferência de aprendizagem sem supervisão	Diferentes, mas relacionados	Diferentes, mas relacionados

Fonte: Rodolfo Simões, 2016.

Uma visão geral sobre as diferentes categorias de técnicas de transferência de aprendizagem pode ser vista na Figura 1. O caso 1 em destaque na Figura 1 condiz com a categoria de transferência de aprendizagem indutiva, com dados rotulados no domínio de origem, abordagem de transferência de aprendizagem empregue nos experimentos presente neste trabalho. Enfatiza-se também na figura a proximidade da categoria de transferência de aprendizagem indutiva com o método de aprendizagem multitarefa, pois essa relação será explorada pelo método de transferência de aprendizagem proposto neste trabalho e apresentado na Seção 3.3.

Figura 1 – Visão geral de diferentes configurações de transferência de aprendizagem. Os retângulos em azul representam as categorias da técnica de transferência de aprendizagem; os círculos representam técnicas de aprendizagem de máquina que são relacionadas com cada categoria. Figura adaptada de (PAN; YANG, 2010)

Fonte: (PAN; YANG, 2010)

Abordagens para a transferência de aprendizagem nas três categorias descritas podem ser resumidas em quatro casos baseados em qual parte do conhecimento transferir ou “*O que transferir*”. O primeiro caso refere-se à transferência baseada em instância, que assume que certa parte dos dados do domínio de origem pode ser reutilizada para a

aprendizagem no domínio de destino por meio de reponderação (*reweighting*). Reponderação de instância¹ e amostagem por importância² são as duas técnicas mais utilizadas nesse contexto (DAI et al., 2007c). O método TrAdaBoost, apresentado na Seção 3.4 baseia-se em reponderação de instâncias.

O segundo caso refere-se à abordagem de transferência por representação de características, que baseia-se na transformação dos dados de entrada para uma representação que possa ser efetivamente explorada por outros problemas. A ideia intuitiva por trás desse caso é aprender uma “boa” representação de características para utilização no domínio de destino (RAINA et al., 2007). Nesse caso, a transferência de aprendizagem é codificada na representação das novas características.

O terceiro caso refere-se à abordagem de transferência de parâmetros (LAWRENCE; PLATT, 2004), na qual assume-se que as tarefas de origem e as tarefas de destino compartilham alguns parâmetros ou distribuições de probabilidade a *priori* de hiperparâmetros dos modelos. Assim, descobrindo os parâmetros ou distribuição de probabilidade compartilhados a *priori*, o conhecimento pode ser transferido entre as tarefas. O método SVR adaptado para transferência de aprendizagem enquadra-se nessa abordagem. Por fim, o último caso refere-se ao problema de transferência de conhecimento relacional, que lida com a transferência de aprendizagem em domínios relacionados (MIHALKOVA; HUYNH; MOONEY, 2007). Com isso, o conhecimento pode ser transferido por meio de um mapeamento dos dados do domínio de origem para o domínio de destino. Recentemente, técnicas estatísticas de aprendizagem relacional dominam este contexto (MIHALKOVA; HUYNH; MOONEY, 2007), (DAVIS; DOMINGOS, 2009). A Tabela 2 apresenta os casos em que as diferentes abordagens podem ser utilizadas para cada categoria de transferência de aprendizagem.

¹ Do original, em inglês, *Instance reweighting*.

² Do original, em inglês, *Importance sampling*.

Tabela 2 – Diferentes abordagens para transferir o conhecimento, que podem ser empregues em cada uma das categorias de transferência de aprendizagem. Tabela adaptada de (PAN; YANG, 2010).

	Transferência de aprendizagem in- dutiva	Transferência de aprendizagem transdutiva	Transferência de aprendizagem sem supervisão
Transferência baseada em instância	X	X	
Transferência por repre- sentação de características	X	X	X
Transferência de parâmetros	X		
Transferência de conheci- mento relacional	X		

Fonte: (PAN; YANG, 2010)

Na Seção 2.3 serão apresentadas apenas as abordagens aplicáveis para a categoria de transferência de aprendizagem indutiva (abordagem considerada nos experimentos). As informações para as demais categorias, estão disponíveis em (PAN; YANG, 2010).

2.3 Transferência de aprendizagem indutiva

A seguir serão apresentadas as quatro abordagens para transferência de aprendizagem na categoria de transferência de aprendizagem indutiva, foco do presente trabalho.

2.3.1 Transferência de conhecimento baseado em instância

A abordagem de transferência de conhecimento baseado em instância para a configuração de transferência de aprendizagem indutiva é interessante, pois possui a premissa que os dados do domínio de origem não podem ser utilizados diretamente na transferência de aprendizagem, há apenas uma parte dos dados que pode ser reutilizada em conjunto a quantidade pequena de dados rotulados presentes no domínio de destino.

Dai et al. (2007c) propuseram o algoritmo *TrAdaBoost*, que é uma extensão do algoritmo *AdaBoost*, para resolver o problema de transferência de aprendizagem indutiva por instâncias. O algoritmo assume que os dados dos domínios de origem e destino utilizam exatamente o mesmo espaço de características e rótulos, porém, a distribuição de

probabilidade dos dados nos dois domínios são diferentes ($P(\mathbf{X}_S) \neq P(\mathbf{X}_T)$). Além disso, o algoritmo assume que devido a essa diferença entre os domínios, nem todos os dados do domínio de origem podem ser úteis na aprendizagem do domínio de destino, alguns desses podem inclusive ser prejudiciais. Diante disso, o algoritmo tenta iterativamente reponderar os dados do domínio de origem para reduzir o efeito dos dados “ruins”, enquanto encoraja a utilização dos dados “bons” para contribuir com o domínio destino. Em cada iteração, o algoritmo treina um classificador com os dados reponderados e com os dados do domínio de destino. Mais detalhes sobre a utilização desse algoritmo está disponível em (DAI et al., 2007c).

Jiang e Zhai (2007) propuseram um método heurístico para remover exemplos de treinamento “ruins” do domínio de origem por meio da diferença entre as distribuições de probabilidade dos dados ($P(\mathbf{X}_S) \neq P(\mathbf{X}_T)$). Em outro trabalho, Liao, Xue e Carin (2005) desenvolveram um novo método de aprendizagem que seleciona dados não rotulados no domínio de destino para serem rotulados com a ajuda dos dados do domínio de origem.

2.3.2 Transferência de conhecimento por representação de características

A abordagem de transferência por representação de características para o problema de transferência de aprendizagem indutiva possui o objetivo de encontrar uma representação adequada para as características, tal que, a divergência entre os domínios e o erro de classificação sejam minimizados (PAN; YANG, 2010).

Estratégias para solucionar esse problema variam de acordo com os dados do domínio de origem. Se há uma grande quantidade de dados rotulados disponível no domínio de origem, métodos de aprendizagem supervisionados podem ser utilizados para construir a representação de características. Isso torna-se similar ao aprendizado de características comuns³, desenvolvida na área de aprendizagem multitarefa (EVGENIOU; PONTIL, 2007). Evgeniou e Pontil (2007) propuseram um método de aprendizagem de características esparsas para a aprendizagem multitarefa. Se não houver dados rotulados disponíveis no domínio de origem, métodos de aprendizagem não supervisionados são propostos para representação de característica. Raina et al. (2007) propuseram a aplicação da codificação esparsa, que é um método de representação de características não supervisionado.

³ Do original, em inglês, *common feature learning*.

2.3.3 Transferência de conhecimento por parâmetros

Muitas abordagens de transferência de parâmetros para a categoria de transferência de aprendizagem indutiva assumem que modelos individuais para tarefas relacionadas podem compartilhar alguns parâmetros ou distribuições de probabilidade *a priori* de hiperparâmetros. A maioria das abordagens descritas nesta seção, incluindo um método baseado em regularização e um outro bayesiano foram projetados para trabalhar na aprendizagem multitarefa. Como mencionado anteriormente, a aprendizagem multitarefa é intimamente ligada ao método de transferência de aprendizagem, porém possui foco em treinar várias tarefas simultaneamente, enquanto a transferência de aprendizagem melhora apenas o desempenho do modelo no domínio de destino por meio da utilização de dados do domínio de origem.

Lawrence e Platt (2004) elaboraram um algoritmo conhecido como Máquinas de Vetores Informativos Multitarefa (MT-IVM)⁴, que é baseado em processos gaussianos (GP)⁵, para lidar com problemas de aprendizagem multitarefa. O algoritmo tenta aprender parâmetros de um GP sobre várias tarefas por meio do compartilhamento do modelo anterior. No mesmo caminho, Bonilla, Chai e Williams (2007) também investigaram a aprendizagem multitarefa no contexto de GP. Os autores propuseram a utilização de uma matriz de covariância para modelar as dependências entre as tarefas, na qual um GP anterior é utilizado para induzir as correlações entre as tarefas. Da mesma forma, Schwaighofer, Tresp e Yu (2004) desenvolveram uma abordagem bayesiana em conjunto com GP's para aprendizagem multitarefa. Além de transferir os parâmetros anteriores dos modelos GP, alguns pesquisadores propuseram a transferência de parâmetros do algoritmo SVM sob um método de regularização. Como, Evgeniou e Pontil (2004) que aproveitaram a ideia da abordagem bayesiana e adaptaram para a aprendizagem multitarefa utilizando o método SVM.

2.3.4 Transferência de conhecimento relacional

Diferente dos outros três contextos, a abordagem de transferência de conhecimento relacional lida com o problema de transferência de aprendizagem em domínios relacionados

⁴ Do original, em inglês, *Multi-task Informative Vector Machine*.

⁵ Do original, em inglês, *Gaussian Process*.

e que os dados extraídos de cada domínio não sejam independentes e identicamente distribuídos, como acontece tradicionalmente. É necessário, nesse caso, transferir a relação entre os dados do domínio de origem para um domínio de destino. Nesse contexto, são propostas técnicas de aprendizagem estatísticas relacionais para resolver esses problemas (PAN; YANG, 2010).

Mihalkova, Huynh e Mooney (2007) propuseram o algoritmo TAMAR que transfere o conhecimento relacional por meio de redes lógicas de Markov (MLNs)⁶ através de domínios relacionados. MLNs, por formalismo, combinam a expressividade compacta da lógica de primeira ordem com a flexibilidade da probabilidade para a aprendizagem estatística relacional (RICHARDSON; DOMINGOS, 2006). Em MLNs, entidades em um domínio relacional são representadas por predicados e suas relações são representadas na lógica de primeira ordem. O algoritmo TAMAR é motivado pelo fato que se dois domínios são relacionados entre si, podem existir mapeamentos para conectar entidades e seus relacionamentos a partir de um domínio de origem para um domínio de destino. Por exemplo, um professor pode ser relacionado em um domínio acadêmico com um gerente em um domínio industrial. Além disso, o relacionamento entre um professor e seus alunos é semelhante ao relacionamento entre um gerente e seus trabalhadores. Assim, pode existir um mapeamento de professor para gerente e outro que mapeia a relação de professor-aluno para gerente-trabalhador. Nesse ponto de vista, o algoritmo tenta utilizar uma MLN aprendida em um domínio de origem para auxiliar no aprendizado de uma MLN em um domínio alvo. Basicamente, TAMAR é um algoritmo com dois estágios. No primeiro estágio, um mapeamento é construído a partir de uma MLN fonte para o domínio de destino. No segundo estágio, a revisão é concluída para a estrutura mapeada no domínio de destino por meio de um algoritmo denominado algoritmo FORTE (RAMACHANDRAN; MOONEY, 1998), que é baseado em programação lógica indutiva para revisão de teorias de primeira ordem. A MLN revisada pode ser utilizada como um modelo relacional para inferência no domínio de destino.

2.4 Aplicações de transferência de aprendizagem

Recentemente, técnicas de transferência de aprendizagem têm sido aplicadas com sucesso em muitas situações do mundo real. Dai et al. (2007b), Dai et al. (2007a) propuseram

⁶ Do original, em inglês, *Markov Logic Networks*.

o uso de técnicas de transferência de aprendizagem para o problema de mineração de textos com o objetivo de reduzir esforços no processo de coleta e rotulação de dados. Em (BLITZER et al., 2007) a técnica de transferência de aprendizagem foi aplicada para resolver problemas de classificação de sentimentos. Em (YIN; YANG; NI, 2005), (PAN et al., 2007), (ZHENG et al., 2008), técnicas de transferência de aprendizagem foram desenvolvidas para extrair o conhecimento obtido por modelos de localização WiFi, considerando diferentes períodos de tempo, espaços e dispositivos móveis para melhorar o desempenho das tarefas de localização WiFi em outras configurações.

Em (LING et al., 2008b), foi elaborada uma abordagem de transferência de aprendizagem baseada na teoria da informação para resolver um problema de classificação de textos em mais de uma língua, conhecida como classificação translinguística⁷. Essa abordagem resolve o problema quando há, por exemplo, uma grande quantidade de dados rotulados de documentos de textos em inglês e há apenas um pequeno número de documentos de textos rotulados em chinês. Nesse trabalho, a transferência de aprendizagem entre os dois espaços de características foi alcançada por meio da construção de uma função de mapeamento adequada, que serviu como uma ponte de ligação.

⁷ Do original, em inglês, *Cross-language classification*.

3 Transferência de aprendizagem em modelos de regressão por vetores suporte

Neste capítulo serão apresentados os métodos de aprendizagem de máquina que são utilizados neste trabalho, a definição do problema, o método de seleção de conjuntos de dados relacionados e as métricas de avaliação. A Seção 3.1 descreve as Máquinas de Vetores Suporte, que servem como embasamento teórico para o método de Regressão por Vetores Suporte, que será apresentado na Seção 3.2. As Seções 3.3 e 3.4 apresentam dois métodos de transferência de aprendizagem que utilizam conhecimento de um conjunto de dados auxiliar: o método de SVR adaptado para a transferência de aprendizagem por parâmetros e o método TrAdaBoost, que realiza transferência de aprendizagem por instâncias, respectivamente.

3.1 Máquinas de Vetores Suporte (SVM)

Pode-se dizer que SVM's são modelos de classificação lineares que utilizam aprendizagem supervisionada. Por se tratar de um modelo linear, as classes analisadas por SVM's são separadas por uma reta. Para tal, busca-se maximizar a distância entre os pontos mais próximos da reta, em cada classe, e a própria reta. A esta distância dá-se o nome de margem (BURGES, 1998).

Para fins didáticos, o problema de classificação pode ser tomado como um problema de duas classes, sem perda de generalidade. Tal tarefa consiste, portanto, em separar duas classes por uma função que é induzida a partir das instâncias de treinamento com o objetivo de produzir um classificador que generaliza bem para dados desconhecidos (GUNN et al., 1998). Observando o exemplo da Figura 2, nota-se que há vários classificadores lineares possíveis que podem separar as instâncias disponíveis sem nenhum erro, mas há somente um que maximiza a margem, ou seja, maximiza a distância entre o classificador e a instância mais próxima de cada classe. Este classificador linear é chamado de hiperplano de separação ótimo, pois, ao menos intuitivamente, espera-se que este generalize melhor que os demais hiperplanos para dados desconhecidos (LIMA, 2004).

Figura 2 – Hiperplano de separação ótimo (em vermelho), tendo como suporte dois outros hiperplanos (linhas tracejadas), os quais passam por alguns pontos para ambas as classes. Estes pontos são chamados de vetores suporte.

Fonte: (LIMA, 2004)

3.2 Regressão por Vetores Suporte (SVR)

O método de vetores suporte pode ser aplicado ao problema de regressão, mantendo-se todas as principais propriedades que caracterizam o algoritmo de margem máxima (CRISTIANINI; SHAW-TAYLOR, 2000). A extensão de SVM para o tratamento de problemas de regressão se dá pela introdução de uma função de perda (SMOLA et al., 1996).

Para o problema de regressão, ocorre uma reversão de propósito quando comparado à formulação de problemas de classificação binária. Ou seja, em vez de forçar as instâncias a se posicionarem o mais distante possível do hiperplano ótimo, no espaço de características, o objetivo agora é forçá-las a se posicionarem o mais próximo possível do hiperplano ótimo, não importa de que lado do hiperplano a instância se encontre. Embora os propósitos da otimização sejam opostos, o objetivo de ambas as formulações é o mesmo: maximizar a capacidade de generalização do modelo preditivo (LIMA, 2004).

3.2.1 Formulação da técnica SVR

Dado um conjunto de dados de treinamento $\{(\mathbf{x}_i, y_i), i = 1, \dots, n\}$, em que $\mathbf{x}_i \in \mathbb{R}^d$ e $y_i \in \mathbb{R}$, o objetivo é aproximar o conjuntos de dados de treinamento com uma função linear:

$$f(\mathbf{x}) = (\mathbf{w} \cdot \mathbf{x}) + b. \quad (1)$$

Para isso, o método SVR linear encontra um modelo $\mathbf{w}$ tal que $\mathbf{w}^T \mathbf{x}_i$ é próximo do valor alvo y_i . E, isso é solucionado pela regularização do problema de otimização a seguir:

$$\min_{\mathbf{w}, \xi^*, \xi} = \frac{1}{2}(\mathbf{w}^T \mathbf{w}) + C(\sum_{i=1}^n \xi_i + \sum_{i=1}^n \xi_i^*). \quad (2)$$

Na Equação (2), o termo $(\mathbf{w}^T \mathbf{w})$ regulariza a complexidade do modelo, $C > 0$ é um parâmetro definido pelo usuário, ξ e ξ_i^* são variáveis de folga representando as restrições dos dois lados do hiperplano (veja Figura 3), conforme as equações abaixo:

$$\begin{cases} y_i - (\mathbf{w} \cdot \mathbf{x}_i) - b \leq \epsilon + \xi_i \\ (\mathbf{w} \cdot \mathbf{x}_i) + b - y_i \leq \epsilon + \xi_i^* \\ \xi_i, \xi_i^* \geq 0, \quad i = 1, \dots, n. \end{cases} \quad (3)$$

Figura 3 – Ilustração de vetores suporte para o problema de regressão com as variáveis de folga ξ e ξ^* nos dois lados do hiperplano.

Fonte: (LIMA, 2004)

A função de perda ϵ -insensível para o método SVR e associada a $(\mathbf{x}_i, y_i)$ é definida por:

$$L_{\epsilon\text{-insensível}}(f(\mathbf{x}) - y) = \begin{cases} 0 & \text{se } |f(\mathbf{x}) - y| < \epsilon \\ |f(\mathbf{x}) - y| - \epsilon & \text{caso contrário.} \end{cases} \quad (4)$$

A função de perda ϵ -quadrática para o método SVR e associada a $(\mathbf{x}_i, y_i)$ é definida por:

$$L_{\epsilon\text{-quadrática}}(f(\mathbf{x}) - y) = \begin{cases} 0 & \text{se } |f(\mathbf{x}) - y| < \epsilon \\ (|f(\mathbf{x}) - y| - \epsilon)^2 & \text{caso contrário.} \end{cases} \quad (5)$$

A função de perda garante que a perda é zero se $|\mathbf{w}^T \mathbf{x}_i - y_i| = |f(\mathbf{x}_i) - y_i| \leq \epsilon$. Isso significa que o valor alvo real y_i encontra-se dentro de um intervalo esperado em torno de $\mathbf{w}^T \mathbf{x}$. As duas funções apresentadas nas Equações (4 e 5) são também conhecidas como funções de perda L_1 e L_2 , respectivamente. Uma ilustração dessas funções é apresentada na Figura 4. Neste trabalho, será utilizada a medida de erro quadrático médio (EQM)¹ como função de erro, que é diretamente modelada pela função de perda L_2 . Assim, as equações ao longo deste trabalho assumem que a função de perda L_2 ou ϵ -quadrática é aplicada.

Figura 4 – Funções de perda L1 e L2.

Fonte: (HO; LIN, 2012)

A Figura 5 apresenta uma ilustração de uma função de regressão SVR representada por $\mathbf{w}^T \mathbf{x}$. O intervalo esperado em volta da função é indicado por linhas tracejadas. $f(\mathbf{x}_i) = \mathbf{w}^T \mathbf{x}_i$ é o valor alvo predito de $\mathbf{x}_i$ e y_i representa o valor alvo real. Os vetores suporte são indicados por uma borda vermelha.

¹ Do original, em inglês, *Mean Squared Error*.

Figura 5 – Ilustração de vetores suporte para o problema de regressão. Figura adaptada de (ROSENBAUM et al., 2013).

Fonte: (ROSENBAUM et al., 2013)

Utilizando a função de perda ϵ -quadrática, a formulação dual do modelo de regressão assume a seguinte configuração:

$$\max_{\alpha, \alpha^*} \left[-\frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n (\alpha_i^* - \alpha_i)(\alpha_j^* - \alpha_j)(\mathbf{x}_i \cdot \mathbf{x}_j) + \sum_{i=1}^n \alpha_i^*(y_i - \epsilon) - \alpha_i(y_i + \epsilon) \right] - \frac{1}{2C} \sum_{i=1}^n (\alpha_i^2 + (\alpha_i^*)^2). \quad (6)$$

Fazendo a seguinte substituição $\beta_i = \alpha_i - \alpha_i^*$, junto a formulação (6), viabiliza-se uma simplificação via condições de KKT, em que:

$$\alpha_i \alpha_i^* = 0 \quad i = 1, \dots, n, \quad (7)$$

o que implica em $\beta_i^2 = \alpha_i^2 + (\alpha_i^*)^2$ e $|\beta_i| = |\alpha_i - \alpha_i^*| = \alpha_i + \alpha_i^*$. Com isso, pode-se definir o problema dual considerando a função de perda L_2 como:

$$\min_{\beta} = \frac{1}{2} \sum_{i=1}^n \sum_{j=1}^n \beta_i \beta_j (\mathbf{x}_i \cdot \mathbf{x}_j) - \sum_{i=1}^n \beta_i y_i + \frac{1}{2C} \sum_{i=1}^n \beta_i^2 + \sum_{i=1}^n \epsilon |\beta_i|, \quad (8)$$

sujeito a:

$$\sum_{i=1}^n \beta_i = 0. \quad (9)$$

Assim, a função de regressão (1) pode ser reescrita da seguinte forma:

$$\mathbf{w} = \sum_{i=1}^n \beta_i \mathbf{x}_i \quad \text{e} \quad (10)$$

$$b = -\frac{1}{2}((\mathbf{w} \cdot \mathbf{x}_r) + (\mathbf{w} \cdot \mathbf{x}_s)). \quad (11)$$

A seguir, apresenta-se como o método SVR pode ser adaptado para a aplicação da técnica de transferência de aprendizagem por parâmetros.

3.3 Técnica SVR adaptada para transferência de aprendizagem (ATL)

Para modelos SVR lineares, uma função $f(\mathbf{x}) = \mathbf{w}^T \mathbf{x}$ é completamente determinada pelo seu vetor de peso $\mathbf{w}$. Na transferência de aprendizagem, o conhecimento de um domínio de origem s é transferido para um domínio de destino t semelhante, com poucos dados. E por meio da transferência de conhecimento, a função f_t presente no domínio de destino pode generalizar melhor dados desconhecidos (ROSENBAUM et al., 2013). Portanto, a técnica de transferência de aprendizagem pode ser melhor aplicada se uma tarefa de aprendizagem com poucas instâncias for semelhante a uma tarefa de aprendizagem com muitas instâncias.

Em um estudo pioneiro, Evgeniou e Pontil (2004) propuseram uma abordagem de aprendizagem multitarefa para o problema de otimização de Regressão por Vetores Suporte com base em grafos. Na abordagem, cada tarefa corresponde a um nó do grafo e a semelhança entre as tarefas é codificada nos pesos das arestas, armazenados em uma matriz de adjacência A , em que $A_{st} \geq 0$. A regularização resultante é a soma dos pesos de cada aresta e a adaptação para a abordagem multitarefa consiste na adição de um termo regularizador $J(\mathbf{w}_1, \dots, \mathbf{w}_t)$ ao problema de otimização primal (2), que contribui para que os t modelos $(\mathbf{w}_1, \dots, \mathbf{w}_t)$ sejam mais semelhantes entre si, e assim, facilitem a transferência de aprendizagem entre as t tarefas. O termo regularizador é definido por:

$$J(\mathbf{w}_1, \dots, \mathbf{w}_t) = \frac{1}{4} \sum_{s=1}^t \sum_{t=1}^t A_{st} \|\mathbf{w}_s - \mathbf{w}_t\|^2, \quad (12)$$

em que $\mathbf{w}_s$ e $\mathbf{w}_t$ são os vetores de pesos associados a cada tarefa específica para a abordagem multitarefa.

O termo regularizador torna os t modelos mais semelhantes entre si. Conseqüentemente, um modelo com um bom poder de generalização, obtido a partir de um conjunto de treinamento com um grande número de instâncias disponíveis, pode melhorar o desempenho de modelos ineficientes, construídos a partir de poucos dados. A partir do

estudo desses trabalhos, propõe-se aqui a adaptação dos problemas de otimização SVR direcionados ao cenário de aprendizagem multitarefa para a formulação do problema no contexto da técnica de transferência de aprendizagem. E, com base nas Equações (2) e (12), propõe-se a formulação de um problema de otimização primal para transferência de aprendizagem via método SVR:

$$\min_{\mathbf{w}_s, \mathbf{w}_t, \xi_s^*, \xi_t^*, \xi_s, \xi_t} = \frac{1}{2}(\mathbf{w}_s^T \mathbf{w}_s) + \frac{1}{2}(\mathbf{w}_t^T \mathbf{w}_t) + J(\mathbf{w}_s, \mathbf{w}_t) + \frac{c}{2} \left(\sum_{i=1}^n (\xi_{si})^2 + \sum_{i=1}^n (\xi_{si}^*)^2 \right) + \frac{c}{2} \left(\sum_{i=1}^n (\xi_{ti})^2 + \sum_{i=1}^n (\xi_{ti}^*)^2 \right). \quad (13)$$

A esse modelo deu-se o nome de Regressão por Vetores de Suporte Adaptado para Transferência de Aprendizagem (ATL), o qual considera apenas duas tarefas, com a notação s para a tarefa de origem e t para a tarefa de destino.

Na Equação (13), os termos $\|\mathbf{w}_s\|^2$, $\|\mathbf{w}_t\|^2$ controlam a complexidade do modelo da tarefa específica, como acontece para o método SVR padrão. A função $J(\mathbf{w}_s, \mathbf{w}_t)$ representa o termo de regularização adicional que aumenta a semelhança dos vetores de pesos das tarefas de origem e destino. Assim, a transferência de aprendizagem é alcançada ao impor que a função f_t seja mais semelhante a função f_s . Uma ilustração de como a técnica transfere o conhecimento entre as tarefas é apresentado na Figura 6.

Figura 6 – Ilustração do conhecimento transferido a partir da tarefa de origem, que contém uma grande quantidade de dados de treinamento (verde) para uma tarefa de destino, que contém poucos dados de treinamento (azul). Os tubos ϵ -sensível ao redor das funções de regressão $\mathbf{w}_s^T \mathbf{x}$ e $\mathbf{w}_t^T \mathbf{x}$ são descritos pelas linhas tracejadas.

Fonte: (ROSENBAUM et al., 2013)

Na Figura 6, o termo regularizador $J(\mathbf{w}_s, \mathbf{w}_t)$ torna o modelo da tarefa de destino ($\mathbf{w}_t$) mais semelhante ao modelo da tarefa de origem ($\mathbf{w}_s$). Com isso, um modelo $\mathbf{w}_t$ que é mais semelhante ao modelo $\mathbf{w}_s$ possui melhor poder de predição para dados desconhecidos, o que resulta em uma melhor generalização do modelo.

Como na Equação (6), utilizando a função de perda ϵ -quadrática, a formulação dual assume a seguinte configuração:

$$\begin{aligned} \min_{\mathbf{w}_s, \mathbf{w}_t, \xi_s^*, \xi_t^*, \xi_s, \xi_t} = & \frac{1}{2}(\mathbf{w}_s^T \mathbf{w}_s) + \frac{1}{2}(\mathbf{w}_t^T \mathbf{w}_t) + \frac{1}{4}(\mathbf{w}_s^T - \mathbf{w}_t^T)(\mathbf{w}_s - \mathbf{w}_t) + \frac{c}{2} \left(\sum_{i=1}^n (\xi_{si})^2 + \sum_{i=1}^n (\xi_{si}^*)^2 \right) \\ & + \frac{c}{2} \left(\sum_{i=1}^n (\xi_{ti})^2 + \sum_{i=1}^n (\xi_{ti}^*)^2 \right) + \sum_{i=1}^n \alpha_{si} (y_{si} - \mathbf{w}_s^T \mathbf{x}_{si} - \epsilon_s - \xi_{si}) + \sum_{i=1}^n \alpha_{ti} (y_{ti} - \mathbf{w}_t^T \mathbf{x}_{ti} - \epsilon_t - \xi_{ti}) + \\ & \sum_{i=1}^n \alpha_{si}^* (\mathbf{w}_s^T \mathbf{x}_{si} - y_{si} - \epsilon_s - \xi_{si}^*) + \sum_{i=1}^n \alpha_{ti}^* (\mathbf{w}_t^T \mathbf{x}_{ti} - y_{ti} - \epsilon_t - \xi_{ti}^*). \end{aligned} \quad (14)$$

Fazendo uma substituição, tal como na Equação (7), e calculando a derivada em relação a $\mathbf{w}_s, \mathbf{w}_t, \xi_{si}^*, \xi_{ti}^*, \xi_{si}, \xi_{ti}$, obtém-se a formulação dual do primal apresentado na Equação (13):

$$\begin{aligned} \min_{\beta_s, \beta_t} = & -\frac{3}{8} \sum_{i=1}^n \beta_{si} \beta_{sj} (\mathbf{x}_{si} \cdot \mathbf{x}_{sj}) - \frac{1}{4} \sum_{i=1}^n \beta_{si} \beta_{tj} (\mathbf{x}_{si} \cdot \mathbf{x}_{tj}) - \frac{3}{8} \sum_{i=1}^n \beta_{ti} \beta_{tj} \\ & (\mathbf{x}_{ti} \cdot \mathbf{x}_{tj}) - \sum_{i=1}^n \frac{\beta_{si}^2}{2C} - \sum_{i=1}^n \frac{\beta_{ti}^2}{2C} + \sum_{i=1}^n \beta_{si} y_{si} - \sum_{i=1}^n |\beta_{si}| \epsilon_s + \sum_{i=1}^n \beta_{ti} y_{ti} - \sum_{i=1}^n |\beta_{ti}| \epsilon_t. \end{aligned} \quad (15)$$

Portanto, pretende-se aqui, solucionar um problema de regressão utilizando o conhecimento de um modelo gerado a partir de uma tarefa relacionada, por meio da técnica de transferência de aprendizagem por parâmetros, tal como formulado na Equação (15). O Apêndice A apresenta os detalhes dos cálculos realizados para a obtenção do dual.

3.4 Transferência de aprendizagem por instância

Boosting é uma técnica de aprendizagem de máquina que combina diversos classificadores fracos com o objetivo de melhorar o desempenho de aprendizagem. Basicamente, o algoritmo atribui pesos iniciais às instâncias do conjunto de treinamento e atualiza esses valores iterativamente com o objetivo de construir um classificador adicional. Após um número determinado de iterações, o método combina diversos classificadores parciais e gera um classificador único, que possivelmente, possui um desempenho superior ao do classificador parcial (FREUND; SCHAPIRE et al., 1996). AdaBoost² (FREUND; SCHAPIRE,

² Do original, em inglês, *Adaptive Boosting*.

(1997) é o algoritmo mais famoso e utilizado para *Boosting*. Seu sucesso deve-se exatamente a sua capacidade de se adaptar aos classificadores-base. No algoritmo AdaBoost, os classificadores são gerados subsequentemente de forma a favorecer os exemplos erroneamente classificados pelos métodos anteriores. A versão do algoritmo AdaBoost para problemas de regressão foi proposta por Drucker (1997) e é apresentada no Algoritmo 1.

Algoritmo 1 Algoritmo AdaBoost para regressão

Entrada: Um conjunto de dados de treinamento $\mathbf{X}$ de tamanho n , um número máximo de iterações N e um algoritmo de aprendizagem **aprendiz**.

Inicialização: Vetor de pesos inicial $\mathbf{v}^1$, tal que $v_i^1 = 1/n$ para $1 \leq i \leq n$. O usuário deve especificar os valores iniciais de $\mathbf{v}^1$.

1: **procedure**

2: **for** $k = 1, \dots, N$ **do**

3: Passo 1) Chame o **aprendiz**, forneça o conjunto de treinamento T e a distribuição $\mathbf{v}^k$. Então, obtenha uma hipótese: $h_k : \mathbf{X} \rightarrow \mathbb{R}$.

4: Passo 2) Calcule o erro ajustado ϵ_i^k para cada instância:

$$\text{Seja } D_k = \max_{j=1}^n |y_j - h_k(x_j)|$$

$$\text{então } \epsilon_i^k = |y_i - h_k(x_i)|/D_k$$

5: Passo 3) Calcule o erro ajustado de h_k :

$$\epsilon_k = \sum_{i=1}^n \epsilon_i^k \cdot v_i^k; \text{ se } \epsilon_k \geq 0.5, \text{ pare e defina } N = k - 1.$$

6: Passo 4) Defina $\beta_k = \frac{\epsilon_k}{(1-\epsilon_k)}$.

7: Passo 5) Atualize o vetor de peso:

$$v_i^{k+1} = \frac{v_i^k \beta_k^{1-\epsilon_i^k}}{Z_k}, \quad Z_k \text{ é uma constante de normalização.}$$

Saída: a hipótese:

$$h_f(x) = \text{a média ponderada de } h_k(x) \text{ para } 1 \leq k \leq N,$$

$$\text{utilizando } \ln\left(\frac{1}{\beta_k}\right) \text{ como o peso para a hipótese } h_k.$$

Fonte: (DRUCKER, 1997)

Na adaptação do algoritmo AdaBoost para problemas de regressão, considera-se que a saída fornecida por uma hipótese h_k para uma instância $\mathbf{x}_i$ geralmente é incorreta, o que resulta em um valor de erro real considerável ou aceitável ($e_i = |y_i - h_k(\mathbf{x}_i)|$), que pode ser arbitrariamente grande. Com isso, torna-se necessário a adição de um método

para mapear um erro e_i para um erro ajustado e'_i que possa ser utilizado no processo de reponderação do AdaBoost. Além disso, no algoritmo, cada erro é definido em relação ao maior erro $D_k = \max_{i=1}^n |\epsilon_i|$, de tal forma que cada erro ajustado e'_i esteja no intervalo $[0, 1]$. Dessa forma, a condição na qual uma instância $\mathbf{x}_i$ é reponderada na iteração k depende de quão grande é D_k .

Dai et al. (2007c) apresentaram um método que estende o algoritmo AdaBoost (FREUND; SCHAPIRE, 1997) para problemas de transferência de aprendizagem, denominado TrAdaBoost³. A versão tradicional desse algoritmo é aplicada apenas para problemas de classificação. E, em 2010, houve uma adaptação do algoritmo TrAdaBoost para problemas de regressão (PARDOE; STONE, 2010). Esse método será considerado nesse projeto e apresentado na próxima seção.

3.4.1 Algoritmo TrAdaBoost para regressão

Pardoe e Stone (2010) categoriza os algoritmos de transferência de aprendizagem em duas categorias: os que fazem uso do modelo treinado nos dados de origem, como no caso do método SVR-ATL por exemplo, e os que utilizam os dados de origem diretamente como dados de treinamento, como o TrAdaBoost.

O algoritmo TrAdaBoost combina os dados de treinamento (rotulados) do domínio de origem e destino para formar um novo conjunto de dados de treinamento para construir um modelo preditivo de alta qualidade para os dados desconhecidos do conjunto de destino, quando esses dados não são suficientes para construir um modelo. Basicamente, o objetivo do algoritmo consiste em, selecionar algumas instâncias do conjunto de origem que sejam similares às instâncias do conjunto de destino e assim, formar um conjunto de dados de treinamento melhor. Para isso, considera-se um mecanismo de ponderação, em cada passo de *boosting*, o método atribui pesos as instâncias de destino e as instâncias de origem. De modo, a favorecer (aumenta-se o peso) as instâncias de destino (PARDOE; STONE, 2010). E, para problemas de regressão, o peso atribuído a cada instância de treinamento é em função do erro obtido. Uma ilustração do funcionamento do algoritmo é apresentado na Figura 7.

³ Abreviação de *Transfer AdaBoost*.

Figura 7 – Objetivo do algoritmo TrAdaBoost.

Fonte: Rodolfo Simões, 2016.

Formalmente, seja χ_S o espaço de características no conjunto de origem; χ_T o espaço de características no conjunto de destino; e $Y = \mathbb{R}$ o conjunto de rótulos reais. O conjunto de dados de teste é denotado por $\mathbf{T} = \{\mathbf{x}_i^t\}$, em que $\mathbf{x}_i^t \in \chi_T$ ($i = 1, \dots, b$). Assim, b é o tamanho do conjunto não rotulado $\mathbf{T}$. O conjunto de dados de treinamento $\mathbf{X}$ é particionado em dois conjuntos rotulados $\mathbf{X}_S$ e $\mathbf{X}_T$. O conjunto $\mathbf{X}_S$ representa as instâncias de treinamento de origem, tal que, $\mathbf{X}_S = \{(\mathbf{x}_i^s, y_i^s)\}$, em que $\mathbf{x}_i^s \in \chi_S$ ($i = 1, \dots, m$). O conjunto $\mathbf{X}_T$ representa o conjunto de treinamento de destino, tal que, $\mathbf{X}_T = \{(\mathbf{x}_i^t, y_i^t)\}$, em que $\mathbf{x}_i^t \in \chi_T$ ($i = 1, \dots, n$), m e n representam o tamanho de $\mathbf{X}_S$ e $\mathbf{X}_T$, respectivamente. O conjunto de treinamento combinado $\mathbf{X} = \{(\mathbf{x}_i, y_i)\}$ é definido como:

$$\mathbf{x}_i = \begin{cases} \mathbf{x}_i^s, i = 1, \dots, m; \\ \mathbf{x}_i^t, i = m + 1, \dots, m + n. \end{cases}$$

O problema a ser solucionado consiste em, dado um número pequeno de dados de treinamentos em $\mathbf{X}_T$, uma quantidade grande de dados de treinamentos em $\mathbf{X}_S$ e alguns dados não rotulados de teste em $\mathbf{T}$, objetiva-se treinar um classificador que minimiza o erro de predição no conjunto de dados $\mathbf{T}$.

Pardoe e Stone (2010) afirmam que, combinando o algoritmo AdaBoost de Drucker (1997) e o algoritmo TrAdaBoost de Dai et al. (2007c), é possível obter o algoritmo TrAdaBoost para regressão. De modo que, os passos de computação do erro devem seguir conforme descrito no Algoritmo (1), e que a principal modificação deve ser realizada na atualização dos vetores de pesos. Em vez de tratar igualmente todos os dados, no TrAdaBoost para regressão, deve-se aumentar os pesos das instâncias de destino por: $v_i^{k+1} = \frac{v_i^k \beta_k^{-\epsilon_i^k}}{Z_k}$ e diminuir os pesos das instâncias de origem por: $v_i^{k+1} = \frac{v_i^k \beta_k^{\epsilon_i^k}}{Z_k}$, em que $\beta = \frac{1}{(1 + \sqrt{\frac{2 \ln n}{N}})}$. Além disso, o algoritmo considera apenas a hipótese final $\lceil \frac{N}{2} \rceil$ quando se determina a saída, como considerado no algoritmo de Dai et al. (2007c). Com isso, Pardoe e Stone (2010) propuseram o algoritmo TrAdaBoost para regressão, apresentado

no Algoritmo 2, que é considerado neste trabalho, e será responsável pela transferência de aprendizagem por seleção de instâncias em modelos QSAR para regressão.

Algoritmo 2 Algoritmo TrAdaBoost para regressão

Entrada: Os dois conjuntos de dados rotulados $\mathbf{X}_S$ e $\mathbf{X}_T$, o conjunto de dados não rotulado $\mathbf{T}$ (teste), um algoritmo de aprendizagem **aprendiz** e um número máximo de iterações N .

Inicialização: Vetor de pesos inicial $\mathbf{v}^1 = (v_1^1, \dots, v_{m+n}^1)$. O usuário deve especificar os valores iniciais de $\mathbf{v}^1$.

1: **procedure**

2: **for** $k = 1, \dots, N$ **do**

3: Passo 1) Defina $\mathbf{p}^k = \frac{\mathbf{v}^k}{\sum_{i=1}^{m+n} v_i^k}$.

4: Passo 2) Chame o **aprendiz**, forneça o conjunto de treinamento combinado $\mathbf{X}$ com distribuição $\mathbf{p}^k$ sobre $\mathbf{X}$ e o conjunto de dados não rotulado $\mathbf{T}$. Então, obtenha uma hipótese: $h_k : \mathbf{X} \rightarrow \mathbb{R}$

5: Passo 3) Calcule o erro ajustado ϵ_i^k para cada instância:

$$\text{Seja } D_k = \max_{j=1}^{m+n} |y_j - h_k(x_j)|$$

$$\text{então } \epsilon_i^k = |y_i - h_k(x_i)| / D_k$$

6: Passo 4) Calcule o erro ajustado de h_k :

$$\epsilon_k = \sum_{i=1}^{m+n} \epsilon_i^k \cdot v_i^k; \text{ se } \epsilon_k \geq 0.5, \text{ pare e defina } N = k - 1.$$

7: Passo 5) Defina $\beta_k = \frac{\epsilon_k}{(1-\epsilon_k)}$ e $\beta = \frac{1}{(1+\sqrt{\frac{2 \ln n}{N}})}$.

8: Passo 6) Atualize os novos vetores de peso:

$$v_i^{k+1} = \begin{cases} \frac{v_i^k \beta \epsilon_i^k}{Z_k}, & 1 \leq i \leq m, \quad Z_k \text{ é uma constante de normalização} \\ \frac{v_i^k \beta_k^{-\epsilon_i^k}}{Z_k}, & m+1 \leq i \leq m+n, \quad Z_k \text{ é uma constante de normalização} \end{cases}$$

Saída: a hipótese:

$$h_f(x) = \begin{cases} \prod_{k=\lceil \frac{N}{2} \rceil}^N \beta_k^{-h_k(x)}, & \text{se } \prod_{k=\lceil \frac{N}{2} \rceil}^N \beta_k^{-h_k(x)} \geq \prod_{k=\lceil \frac{N}{2} \rceil}^N \beta_k^{-\frac{1}{2}} \\ \prod_{k=\lceil \frac{N}{2} \rceil}^N \beta_k^{-\frac{1}{2}}, & \text{caso contrário} \end{cases}$$

Fonte: (PARDOE; STONE, 2010)

Portanto, em cada iteração do algoritmo, diminui-se o peso das instâncias de origem em função do erro obtido, alterando o valor do seu peso atual por $\frac{v_i^k \beta \epsilon_i^k}{Z_k}$. Dessa forma, na próxima iteração, as instâncias de treinamento do conjunto de origem que possuem erros

grandes irão afetar menos o processo de aprendizagem do que na iteração atual. Após várias iterações, as instâncias com maiores pesos de treinamento são selecionadas para formar um novo conjunto de dados.

Na próxima seção, será apresentada a definição do problema de modelagem QSAR para regressão. Além disso, serão apresentados o resumo de cada método e a sua aplicação nos dados químicos e os esquemas de avaliação utilizados nos experimentos.

3.5 Definição do problema

A análise QSAR é fundamental para determinar grupos de substâncias químicas com um maior nível de atividade biológica responsáveis por uma determinada ação no organismo (PEREIRA et al., 2008). Para esse estudo, estão disponíveis conjuntos de dados com observações (moléculas) e atributos (descritores químicos) contendo informações sobre as propriedades de substâncias químicas, as quais são analisadas com o auxílio de métodos computacionais. A partir dessa identificação, o desenvolvimento de novos medicamentos torna-se viável, levando ao planejamento de substâncias com melhor atividade biológica e menos efeitos colaterais. Nesse contexto, a busca por novos medicamentos pode ser vista como um problema de regressão em que se pretende prever o valor real do nível de atividade biológica de um composto químico em relação a um determinado alvo biológico. A Figura 8 apresenta o processo de identificação dos compostos químicos (em ativos ou inativos) por meio da análise QSAR.

Figura 8 – Processo de identificação de compostos químicos via QSAR.

Fonte: (SOUZA, 2015)

Uma vez que é possível obter diversas variáveis capazes de descrever características estruturais e eletrônicas de um conjunto de compostos, um problema QSAR com um único alvo biológico consiste em um conjunto de descritores de moléculas rotulados $\{(\mathbf{x}_i, y_i), i = 1, \dots, n\}$, em que $\mathbf{x}_i \in \mathbb{R}^d$ é um vetor de descritores químicos e $y_i \in \mathbb{R}$ é um valor real, que define o nível de atividade biológica deste composto em relação ao alvo. A partir desses valores, é possível definir um limiar que permita classificar compostos em ativos ou inativos.

Ning e Karypis (2011) discutem que grande parte dos compostos químicos não possui rótulos (níveis de atividade biológica) pré-definidos e que as amostras que contêm compostos rotulados são muito pequenas, o que pode gerar um modelo preditivo ineficiente. Com isso, torna-se desejável utilizar informações de outros conjuntos de dados químicos já existentes para melhorar o desempenho de um modelo QSAR com pouca informação. Tal cenário torna propícia a aplicação de técnicas de transferência de aprendizagem. Entretanto, para uma transferência de aprendizagem válida, é necessário que os domínios de origem e de destino sejam relacionados (PAN; YANG, 2010). Na próxima seção, será apresentado o método para seleção dirigida dos conjuntos de dados químicos de origem que é utilizado neste trabalho.

3.5.1 Seleção de conjuntos de dados relacionados

Uma das premissas para a utilização da técnica de transferência de aprendizagem é que os domínios de origem e de destino devem ser relacionados. Tan et al. (2015) afirmam que essa relação pode ser concretizada na forma de instâncias ou características. Se nenhuma relação direta for encontrada, a transferência forçada não vai funcionar, resultando em nenhuma melhora ou piorando o desempenho de aprendizagem no domínio de destino (FITZGERALD; THOMAZ, 2015), fato conhecido como transferência negativa (WANG; LI, 2014).

Na maioria dos trabalhos correlatos, assume-se a suposição de que estão disponíveis dois conjuntos de dados relacionados e que a transferência de aprendizagem será aplicada a partir do conjunto de dados com maior volume para o conjunto de dados com menor quantidade de dados disponíveis. Porém, essa suposição em conjuntos de dados de natureza química nem sempre é suficiente, tornando necessária a opinião de um especialista da área para definir o(s) conjunto(s) de dados de origem.

Para superar essa limitação, Girschick, Rückert e Kramer (2012) propuseram uma abordagem para selecionar um conjunto de dados de origem de forma orientada a dados em um repositório que contém conjuntos relacionados ao de destino. Os autores utilizaram a base de dados *PubChem BioAssay*⁴, que contém mais de 500 mil conjuntos de dados obtidos a partir de experimentos toxicológicos e bioquímicos. Nessa base de dados, os compostos de um ensaio podem ser categorizados como ativos ou inativos com respeito a um alvo biológico (alguns também podem ser categorizados como inconclusivos, mas isto pode ser devido a problemas técnicos experimentais). Geralmente, a categorização é realizada por meio de um limiar definido por um especialista da área para valores reais de atividade biológica.

Deste modo, Girschick, Rückert e Kramer (2012) escolheram em seus experimentos, a partir da base de dados *PubChem BioAssay*, cinco conjuntos de dados de destino, um para cada problema (alvo específico), com o intuito de representar diferentes cenários de transferência de aprendizagem. Para cada um dos cinco conjuntos, foi calculada a medida de sobreposição de atividade (OV_{act})⁵ para alguns conjuntos com o mesmo alvo. Com isso, obteve-se um ranqueamento dos conjuntos relacionados de acordo com a similaridade

⁴ Disponível em: <https://pubchem.ncbi.nlm.nih.gov/>.

⁵ Do original, em inglês, *Activity Overlap Measure*.

destes com relação ao conjunto de destino. No trabalho, a similaridade de dois conjuntos A e B é calculada pelo coeficiente de Tanimoto, utilizando a categorização dos compostos (ativos ou inativos), conforme a equação:

$$OV_{act}(A, B) = \frac{act(A \wedge B)}{act(A) + act(B) - act(A \wedge B)}, \quad (16)$$

em que, $act(\cdot)$ retorna o número de compostos ativos em um conjunto.

A partir dos ranqueamentos obtidos, os autores selecionaram, para cada conjunto de dados de destino, o primeiro conjunto com tamanho suficientemente grande para ser utilizado como origem (tamanho > 100), com a restrição de que este não seja um superconjunto do conjunto de destino.

3.5.2 Aplicação de transferência de aprendizagem em modelos QSAR para regressão

Este trabalho analisa experimentalmente a aplicação de duas técnicas de transferência de aprendizagem a modelos QSAR para regressão via SVR. O primeiro método consiste na transferência de aprendizagem por parâmetros, proposto neste trabalho. O segundo método consiste na transferência de aprendizagem por instâncias para problemas de regressão (PARDOE; STONE, 2010). A seguir, serão apresentados mais detalhes sobre a aplicação desses métodos no contexto deste trabalho.

- Transferência de aprendizagem por parâmetros

Dado um conjunto de dados de treinamento $\{(\mathbf{x}_i, y_i), i = 1, \dots, n\}$, em que $\mathbf{x}_i \in \mathbb{R}^d$ e $y_i \in \mathbb{R}$, o método SVR linear encontra um modelo $\mathbf{w}$ tal que $\mathbf{w}^T \mathbf{x}_i$ é próximo do valor alvo y_i . Geralmente, o modelo obtido pelo método SVR depende fortemente do conjunto de dados de treinamento disponível. Se o conjunto de dados possui um número pequeno de instâncias, certamente esse modelo não irá prever de forma satisfatória valores para outros dados desconhecidos.

O método SVR-ATL (ver Seção 3.3) é utilizado neste trabalho para a aplicação da técnica de transferência de aprendizagem por parâmetros. Considera-se, aqui, que o conjunto de dados de treinamento de origem possui um bom número de instâncias e que o modelo obtido ($\mathbf{w}_s$) possui um bom poder de generalização. A adaptação do método

realizada neste trabalho consiste na adição do termo regularizador $J(\mathbf{w}_s, \mathbf{w}_t)$ ao problema de otimização primal do SVR, o que contribui para que os modelos obtidos $(\mathbf{w}_s, \mathbf{w}_t)$ por diferentes conjuntos de dados sejam mais semelhantes entre si, e assim, facilitem a transferência de aprendizagem entre as tarefas de origem e destino. Com isso, um modelo $\mathbf{w}_t$ (destino) que é mais semelhante ao modelo $\mathbf{w}_s$ (origem) possui melhor poder de predição para dados desconhecidos, o que resulta em uma melhor generalização do modelo.

- Transferência de aprendizagem por instâncias

A transferência de aprendizagem por instâncias é atingida ao utilizar os dados da tarefa de origem diretamente como dados de treinamento para construir um modelo preditivo de alta qualidade para os dados desconhecidos da tarefa de destino, quando esses dados não são suficientes para construir um modelo. O método TrAdaBoost para regressão, considerado aqui, seleciona instâncias dos domínios de origem e de destino para formar um novo conjunto. Para isso, considera-se um mecanismo de ponderação, em que a cada passo de *boosting*, o método atribui peso às instâncias em função do erro obtido em cada iteração (ver Algoritmo 2). Ao final do processo, seleciona-se um número arbitrário de instâncias (considerando os pesos resultantes) para formar um novo conjunto de dados de treinamento para a execução do método SVR.

3.5.3 Esquema de avaliação de desempenho

Após a etapa de implementação, os experimentos de regressão são realizados para avaliação dos métodos considerados. Para isso, considera-se, neste trabalho, a técnica de reamostragem de validação cruzada *k-fold* nos conjuntos da tarefa de destino, que consiste em dividir uma amostra de dados em k subconjuntos mutuamente exclusivos com aproximadamente o mesmo tamanho e, a partir disso, um subconjunto é utilizado para teste e os $k - 1$ restantes são utilizados para treinamento do modelo preditivo. Este processo é realizado k vezes alternando de forma circular o subconjunto de teste (mais detalhes em Arlot, Celisse et al. (2010)). Além disso, são consideradas amostras aleatórias de dados com diferentes tamanhos para analisar o *trade-off* entre a quantidade de dados de treinamento disponíveis na tarefa de destino e o desempenho obtido por cada método.

Como métrica de avaliação de desempenho dos modelos de regressão, é utilizada a medida de erro quadrático médio (EQM), definida por:

$$EQM = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2, \quad (17)$$

em que, y_i pode ser o valor real de atividade biológica e $\hat{y}_i$ é o valor predito para uma instância i .

O EQM é uma das métricas mais utilizadas em problemas de regressão. Essa métrica de avaliação calcula a média dos erros ao quadrado, ou seja, diferenças menores têm menos importância, enquanto diferenças maiores recebem mais peso.

4 Revisão bibliográfica

O estudo da técnica de transferência de aprendizagem foi incitado pelo fato de que pessoas inteligentes podem aplicar o conhecimento aprendido anteriormente para solucionar novos problemas de forma mais rápida e com melhores soluções. A motivação fundamental para transferência de aprendizagem, no campo de aprendizagem de máquina, foi discutida inicialmente no Workshop [NIPS-95](#)¹ intitulado “*Learning to Learn*”, no qual se debateu sobre a necessidade de métodos de aprendizagem de máquina reutilizarem o conhecimento aprendido anteriormente em problemas similares. Desde então, pesquisadores têm atraído mais e mais atenção para a utilização dessa técnica sob diferentes nomes: *learning to learn*, *life-long learning*, *knowledge transfer*, *inductive transfer*, *multitask learning* e *cumulative learning* ([PAN; YANG, 2010](#)).

O primeiro trabalho sobre transferência de aprendizagem é atribuído a Lorien Pratt, que formulou o algoritmo de transferência baseado em discriminabilidade (DBT)² ([PRATT et al., 1993](#)). Em 1997, o periódico *Machine Learning* publicou uma edição dedicada à transferência indutiva ([DIETTERICH; PRATT; THRUN, 1997](#)), e no mesmo ano, a aprendizagem multitarefa foi incluída como pertencente ao campo de pesquisa em transferência de aprendizagem, juntamente com uma análise de seus fundamentos teóricos([CARUANA, 1997](#)).

Inspirados pelos mecanismos humanos de inferência transitiva e capacidade de aprendizagem, em que dois conceitos aparentemente não relacionados podem ser ligados por intermédio de conceitos auxiliares, [Fitzgerald e Thomaz \(2015\)](#) apresentam a abordagem de transferência de aprendizagem transitiva (TTL)³. Tal abordagem visa reduzir as grandes distâncias entre os domínios, até mesmo quando os domínios de origem e destino são poucos relacionados. Para resolver o problema de TTL, os autores selecionam um ou mais domínios para atuar como ponte (conceitos auxiliares) entre o domínio de origem e o domínio de destino, e com isso, efetuam a transferência de conhecimento. A estrutura do mecanismo de transferência é composta de duas etapas. Na primeira delas, o objetivo é encontrar um domínio adequado para aproximar o domínio de origem ao domínio de destino. A segunda etapa consiste em fazer a transferência de conhecimento entre todos

¹ *Neural Information Processing Systems*.

² Do original, em inglês, *Discriminability-Based Transfer*.

³ Do original, em inglês, *Transitive Transfer Learning*.

os domínios (origem, “pontes” e destino). No primeiro passo, é proposto um modelo de probabilidade para selecionar domínios apropriados, com base nas suas características. No segundo passo é elaborado um algoritmo para executar a transferência de aprendizagem, que considera a relação e a distribuição transitiva entre os domínios. Na mesma direção, [Luo et al. \(2008\)](#) propuseram um método de regularização consensual para explorar as diferenças de distribuições de probabilidade entre vários domínios de origem e assim, melhorar o desempenho na aprendizagem do domínio de destino. Para isso, um classificador é treinado considerando apenas os dados disponíveis nos domínios de origem e a predição é realizada de forma consensual, considerando todos os domínios de origem. Para combinar os resultados dos classificadores, são compartilhados dados estatísticos, em vez dos rótulos obtidos pelos classificadores.

[Ge, Gao e Zhang \(2013\)](#) propuseram uma variação para o problema tradicional de transferência de aprendizagem a partir de vários domínios de origem para um domínio de destino. Tal trabalho foi motivado pelo fato de que a maioria dos métodos de transferência de aprendizagem trabalham de forma *offline*, sendo necessário armazenar os dados do domínio de destino antes da construção do modelo. Entretanto, em algumas aplicações de tempo real, nas quais os dados são sequenciais e em grande volume, essa abordagem pode não ser eficaz. Para superar essa limitação, foi proposto o método conhecido como Transferência de Aprendizagem de Múltiplas Origens *Online* (OMS-TL)⁴, que pode transferir conhecimento a partir de vários domínios de forma mais rápida, considerando grandes volumes de dados. [Ling et al. \(2008a\)](#) propuseram o uso da técnica de transferência de aprendizagem para dados rotulados a partir de um único domínio de origem visando à classificação de dados não rotulados em um domínio diferente. Os autores formularam o problema sob uma nova abordagem de classificação definida como *Cross-Domain Classifier Spectral* (CDSC), em que uma função de custo é introduzida para procurar coerência entre os dados supervisionados no domínio de origem e a estrutura intrínseca do domínio de destino. Por meio da otimização da função de custo, as informações dos dados rotulados no domínio de origem são efetivamente transferidas para melhorar a classificação de dados não rotulados do domínio de destino.

[Quanz e Huan \(2009\)](#) abordam especificamente problemas de transferência de aprendizagem transdutiva, nos quais há dados de treinamento rotulados disponíveis no domínio de origem e dados de teste não rotulados no domínio de destino, potencialmente

⁴ Do original, em inglês, *Online Multiple Source Transfer Learning*.

descritos por diferentes distribuições de probabilidade ($P(\mathbf{X}_S) \neq P(\mathbf{X}_T)$). O objetivo consiste em aproveitar os dados de treinamento rotulados para treinar um classificador que será testado em dados com diferentes distribuições de probabilidade. Para essa tarefa, é utilizado o método de classificação SVM para encontrar uma função de decisão de baixa complexidade que separe bem os dados. Com isso, o problema de transferência de aprendizagem objetiva maximizar uma margem de classificação, enquanto simultaneamente minimiza a distância entre as distribuições de treinamento e teste nos domínios de origem e destino, respectivamente. [Chattopadhyay et al. \(2012\)](#) propuseram dois métodos de transferência de aprendizagem, considerando vários domínios de origem com diferentes distribuições de probabilidade para o cenário de detecção da fadiga muscular. Nos métodos propostos, os dados de eletromiografia (EMG)⁵ de um sujeito representam um domínio, dados de vários indivíduos formam o conjunto de treinamento de múltiplos domínios de origem e os dados de teste formam o domínio de destino. Nesse cenário, os sinais de EMG possuem diferentes distribuições de probabilidade condicional. O primeiro método utiliza uma técnica de ponderação que aborda as diferenças de distribuições de probabilidade condicionais em vários domínios. O segundo método é uma abordagem de adaptação de domínios em dois estágios, em que primeiramente, combina-se os dados ponderados de várias fontes com base em suas diferenças de probabilidades marginais e na segunda etapa, são consideradas suas diferenças de probabilidades condicionais em relação aos dados do domínio de destino.

[Gong et al. \(2015\)](#) apresentam uma abordagem para resolver o problema da pequena quantidade de dados disponíveis para treinamento utilizando métodos de transferência de aprendizagem no cenário de desenvolvimento de modelos de risco para cirurgias cardíacas. É proposto um método para ponderar instâncias com base em suas similaridades e selecionar os exemplos úteis para a tarefa alvo. Para isso, utiliza-se a distribuição dos dados do domínio de origem com o objetivo de aumentar a distribuição dos dados do domínio de destino. Nesse trabalho, foi comprovado que a incorporação de dados apropriados a partir do domínio de origem pode levar à obtenção de melhores resultados do que utilizando somente os dados de treinamento do domínio de destino. [He, Liu e Lawrence \(2009\)](#) propõem um método de transferência de aprendizagem baseado em grafos. O algoritmo desenvolvido propaga as informações de rótulo a partir do domínio de origem para o domínio de destino por meio de um grafo tripartido (instâncias-características-instâncias), dando

⁵ Do original, em inglês, *ElectroMyoGraphy*.

maior ênfase aos poucos exemplos rotulados do domínio de destino via grafo bipartido (instâncias rotuladas-instâncias não rotuladas). Ao propagar as informações através desses grafos, é possível aprender uma função de classificação que prediz de forma satisfatória as instâncias não rotuladas no domínio de destino.

Ng, Wu e Ye (2012) propuseram uma nova estratégia de aprendizagem de máquina chamada *Co-Transfer Learning*. O foco dessa abordagem reside em utilizar dados rotulados de diferentes espaços de características para melhorar a classificação em diversas tarefas de aprendizagem simultaneamente. Para isso, o algoritmo proposto constrói diferentes relações para mapear e conectar os diferentes espaços de características, de modo que o conhecimento possa ser transferido. Ying et al. (2015) elaboraram o método *Multiple Cross-Domain Transfer learning* (MCDTL) como uma abordagem para a transferência de aprendizagem para domínios com alta variedade. A abordagem baseia-se na medida de Kolmogorov-Smirnov, sendo capaz de proporcionar uma boa estimativa para descobrir um conjunto de domínios de origem mais relacionados ao domínio de destino. Feuz e Cook (2015) desenvolveram uma técnica de transferência de aprendizagem para utilização em diferentes espaços de características, denominada de remapeamento do espaço de característica (FSR)⁶. A técnica é chamada de “remapeamento”, devido ao fato de que os dados originais estão mapeados em um espaço de características e o FSR remapeia os dados para um espaço de características diferente. O método relaciona as características em diferentes espaços por meio da construção de meta-características considerando apenas uma pequena quantidade de dados rotulados no domínio de destino para inferir as relações no domínio de origem. Opcionalmente, o método pode ser executado também sem a utilização de quaisquer dados rotulados no domínio de destino.

Em muitas aplicações do mundo real, os dados são frequentemente representados a partir de muitas perspectivas, que correspondem a múltiplos pontos de vistas. Tem sido comprovado que alavancar consistências entre os diferentes pontos de vista pode ajudar a melhorar a aprendizagem. No entanto, a maioria dos métodos existentes de transferência de aprendizagem não conseguem capturar a natureza de múltiplas visões e não podem ser adequados satisfatoriamente para essas situações. O objetivo da transferência de aprendizagem com múltiplas visões é construir um classificador preciso para o domínio de destino, utilizando instâncias rotuladas do domínio de origem, bem como a redundância presente nos múltiplos pontos de vista. Zhang et al. (2011) apresentaram o método de

⁶ Do original, em inglês, *Feature Space Remapping*.

transferência de aprendizagem com a abordagem de margem grande (MVTL-LM)⁷ para o melhor aproveitamento dessas instâncias rotuladas e das características com diferentes “visões”.

4.1 Trabalhos relacionados na área de Química Medicinal

Girschick, Rückert e Kramer (2012) propuseram um método de transferência adaptado, combinando aprendizagem por distância e transferência de aprendizagem indutiva. A abordagem é avaliada especificamente para problemas QSAR, nos quais um conjunto de dados contém n instâncias rotuladas $\mathbf{X} := \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)\}$, em que $\mathbf{x}_i \in \chi$ são dados arbitrários tomados a partir de um espaço de instâncias χ e $y_i \in \mathbb{R}$ são rótulos de valor real. O objetivo consiste em encontrar uma função de regressão: $r : \chi \rightarrow \mathbb{R}$ que prediz de forma satisfatória o rótulo alvo para dados desconhecidos. A construção do algoritmo é baseada em duas etapas. Para a primeira etapa, que se refere à aprendizagem por distância, foi utilizado o algoritmo dos K-Vizinhos mais Próximos (KNN)⁸ adaptado, que díspar da abordagem clássica, utiliza um limiar para selecionar os vizinhos mais próximos. Nesse caso, cada vizinho possui um peso associado, de modo que os mais semelhantes recebem maior importância. Na segunda etapa, a transferência de aprendizagem indutiva é adequada para situações em que há uma pequena quantidade de dados de treinamento com o objetivo de determinar um bom vetor de pesos para os vizinhos. Ao invés de aprender um novo vetor de pesos a partir dos dados limitados de treinamento, é considerado um conjunto de dados adicional semelhante ao conjunto de dados original. Além da abordagem clássica de transferência de aprendizagem, o trabalho apresenta outras duas abordagens: adaptação limitada⁹ e adaptação penalizada¹⁰, em que o vetor de pesos transferido é adaptado ao conjunto de dados de treinamento no domínio alvo para evitar uma grande diferença entre os dados. A adaptação é realizada adicionando o critério de que a dissimilaridade entre os dados deve ser menor que uma constante ϵ . Para as medidas de distância em dados químicos, os autores basearam-se na observação de Raymond e Willett (2002), de que as duas famílias de medidas de distância química, a baseada em *fingerprints* e as

⁷ Do original, em inglês, *Transfer Learning with a Large Margin Approach*.

⁸ Do original, em inglês, *K-nearest neighbors*.

⁹ Do original, em inglês, *Bounded Adaptation*.

¹⁰ Do original, em inglês, *Penalized adaptation*.

medidas baseadas em *Maximum common subgraph*, fornecem informações ortogonais sobre a semelhança química.

Considere a tarefa de predição de sensibilidade a drogas em pacientes com mieloma múltiplo, no qual há um número limitado de dados de treinamento, mas há uma quantidade suficiente de dados (dados auxiliares) para a tarefa de predição de sensibilidade a drogas em pacientes com outro tipo de câncer. Os dados auxiliares ou dados de origem estão em um espaço de característica diferente ou seguem uma distribuição diferente da tarefa alvo. Diante desse problema, [Turki, Wei e Wang \(2017\)](#) apresentaram duas abordagens que combinam os dados da tarefa de origem com os dados da tarefa de destino para melhorar o desempenho de predição nos dados de teste de destino.

O primeiro algoritmo Synthetic Minority Over-Sampling Technique (SMOTE) é um poderoso método de super-amostragem. Primeiro, o algoritmo seleciona todos os exemplos pertencentes a classe minoritária e então para cada exemplo selecionado, gera k exemplos sintéticos (k vizinhos mais próximos do exemplo) e armazena em um conjunto de dados temporário. Ao fim dessa etapa, um subconjunto é gerado a partir do conjunto temporário com o número de exemplos igual à diferença entre a classe majoritária e a classe minoritária. Assim, um algoritmo de aprendizagem de máquina deve ser empregue no conjunto final gerado. O Segundo algoritmo é o CUR Matrix Decomposition, proposto por ([MAHONEY; DRINEAS, 2009](#)), o algoritmo consiste na redução do número de colunas e linhas de um determinado conjunto de dados.

Para o propósito de melhorar o desempenho no conjunto de dados de teste de destino por meio dos dados de treinamento da tarefa de origem e destino, os autores primeiro utilizaram o algoritmo CUR para reduzir a dimensionalidade do conjunto de dados que possui um maior número de características disponíveis, sendo ele pertencente à tarefa de origem ou destino. Depois, aplicaram uma versão modificada do SMOTE nos conjuntos resultantes da etapa anterior, que seleciona b exemplos a partir do conjunto de origem e para cada exemplo seleciona os K vizinhos mais próximos no conjunto de destino. E assim, gera-se um conjunto de dados no qual um método de aprendizagem pode ser aplicado.

A seguir são apresentados alguns trabalhos utilizando a técnica de aprendizagem multitarefa para o problema QSAR, que podem ser adaptados para a utilização da técnica de transferência de aprendizagem. [Evgeniou e Pontil \(2004\)](#) proupuseram uma abordagem de aprendizagem multitarefa para o problema de otimização de Regressão por Vetores

Suporte baseada em grafo. Nessa abordagem, cada tarefa corresponde a um nó no grafo e a similaridade entre as tarefas é codificada pelos pesos das arestas em uma matriz de adjacência A , em que $A_{st} \geq 0$. A regularização resultante é a soma ponderada das distâncias de similaridade entre os vetores de pesos. A Figura 9 apresenta um exemplo para o problema com quatro tarefas, representadas por quatro nós em um grafo com seus vetores de pesos correspondentes, $\mathbf{w}_1, \dots, \mathbf{w}_4$. As tarefas são relacionadas por uma matriz de adjacência $\mathbf{A}$ de valores reais. O algoritmo treina uma tarefa de um modelo específico $\mathbf{w}_1, \dots, \mathbf{w}_4$ em um único passo, utilizando as instâncias de todas as tarefas.

Figura 9 – Grafo regularizado multitarefa (GRMT). Figura adaptada de (ROSENBAUM et al., 2013).

Fonte: (ROSENBAUM et al., 2013)

Widmer et al. (2012) desenvolveram uma formulação alternativa para o trabalho de Evgeniou e Pontil (2004), combinando os pesos de tarefas específicas $\mathbf{w}_1, \dots, \mathbf{w}_4$ em um único vetor de peso $\mathbf{w}$. Esta formulação utiliza a chamada “visão de vetores em bloco”¹¹.

Rosenbaum et al. (ROSENBAUM et al., 2013) apresentaram dois diferentes algoritmos multitarefa que podem explorar a semelhança entre vários alvos biológicos para transferir o conhecimento entre modelos QSAR específicos. Os algoritmos de regressão multitarefa foram baseados no algoritmo de classificação multitarefa de Widmer et al. (2012). O algoritmo Grafo Regularizado Multitarefa (GRMT)¹²-SVR resolve o problema de otimização por meio da abordagem padrão de regressão de vetores suporte e utiliza também a abordagem de “visão de vetores em bloco”. O segundo algoritmo, denominado de Multi-

¹¹ Do original, em inglês, *block vector view*.

¹² Do original, em inglês, *Graph-Regularized Multi-task*.

tarefa *Top-down* (TDMT)¹³-SVR explora o seguinte fato: se as tarefas de aprendizagem ou no caso, as proteínas alvo, são relacionadas por uma taxonomia τ , a estrutura hierárquica de τ pode ser explorada subsequentemente para treinar modelos mais especializados. A ideia de aprendizagem do algoritmo é treinar subsequentemente modelos para cada nó de uma dada taxonomia de forma *top-down*, assim, obtendo modelos mais especializados ao descer a taxonomia. A especialização sucessiva é alcançada pela minimização do erro de treinamento nas instâncias de treinamento das sub-árvores, mantendo a semelhança para o seu antecessor. Este algoritmo também utiliza a abordagem de SVR para o problema de otimização de um determinado nó de uma dada taxonomia.

¹³ Do original, em inglês, *Top-down Multi-task*.

5 Análises experimentais

Neste capítulo, são apresentados e discutidos os resultados dos experimentos realizados neste trabalho, que visaram a tratar o problema da falta de informação disponível para a construção de modelos preditivos eficientes, considerando dados de *benchmark* e de moléculas candidatas a novos medicamentos.

Primeiramente, são descritos os detalhes das configurações dos experimentos para cada método de regressão avaliado. Em seguida, são apresentados os resultados obtidos para cada conjunto de dados analisado, juntamente com uma análise comparativa para a discussão dos resultados alcançados.

5.1 Considerações iniciais

Hoje, devido a grande quantidade de dados produzidos em todas as áreas da ciência e indústria, existem muitos conjuntos de dados relacionados que permitem a aplicação das técnicas de transferência de aprendizagem. Diante desse fato, os pares de conjuntos de dados utilizados neste trabalho (conjuntos de dados de origem e destino) são apenas uma seleção pequena de uma ampla gama de possibilidades. Assume-se, ainda, que os conjuntos de dados são suficientemente relacionados para a aplicação das técnicas de transferência de aprendizagem.

Sendo assim, para cada par de conjuntos de dados analisado, investigou-se em quais situações a transferência de aprendizagem seria benéfica e capaz de melhorar o poder preditivo do modelo na tarefa de destino. Para tanto, a análise de desempenho foi realizada com conjuntos de teste da tarefa de destino.

Além disso, os experimentos de regressão apresentados a seguir possuem conjuntos de treinamento formados com tamanhos variados, com o objetivo de avaliar o *trade-off* entre o número de exemplos disponíveis na fase de treinamento e o desempenho obtido pelos métodos em cada cenário específico. Considera-se, nessas análises, o termo “cenário” como sendo a configuração experimental desenhada para comparar o desempenho dos métodos de regressão na tarefa de destino. Mais especificamente, essas configurações variam com relação ao número de instâncias disponíveis para a fase de treinamento na tarefa de destino e aos valores dos parâmetros utilizados na execução de cada método (*kernel* e custo).

Para a configuração dos experimentos conduzidos, foi considerada a técnica de reamostragem de validação cruzada *k-fold*, com $k=5$ e como métrica de avaliação de desempenho dos modelos de regressão foi utilizada a medida de erro quadrático médio. Além disso, os conjuntos de dados analisados foram submetidos, como etapa de pré-processamento, à padronização das suas variáveis, o que consistiu em subtrair de cada valor de uma determinada variável a sua respectiva média e dividir o resultado obtido pelo desvio padrão desta variável.

A seguir, apresenta-se a configuração experimental utilizada para os experimentos conduzidos com os métodos de regressão SVR, SVR-ATL e TrAdaBoost.

5.2 Configuração dos experimentos com o método SVR

Para a execução do método SVR, foi utilizada a biblioteca SVR (e1071)¹ do software R. Para os experimentos com *kernel* linear, a seguinte configuração de parâmetros foi empregada: *SVM-Type* = *eps-regression*, *SVM-Kernel* = *linear*, custo $C=(100, 10 \text{ e } 0.1)$. Por sua vez, os experimentos conduzidos com *kernel* polinomial foram configurados com os seguintes parâmetros: *SVM-Kernel* = *polinomial*, custo $C=(100, 10 \text{ e } 0.1)$, *degree*=2 e *coef.0*=1.

Como mencionado anteriormente, em cada experimento considerou-se subconjuntos com tamanhos arbitrários variados e extraídos dos conjuntos de treinamento da tarefa de destino. Para os conjuntos de dados de *benchmark*, utilizou-se valores específicos para o número de instâncias de treinamento considerado. Tal metodologia foi inspirada nos experimentos de transferência de aprendizagem desenvolvidos por Garcke e Vanck (2014).

Para os conjuntos de dados químicos, os experimentos foram realizados utilizando percentuais do total de instâncias disponíveis nos conjuntos de dados de treinamento de cada tarefa de destino (25%, 50%, 75% e 100%), seguindo a metodologia de Girschick, Rückert e Kramer (2012). Para cada subconjunto derivado dos conjuntos originais, aplicou-se a técnica de reamostragem de validação cruzada com $k=5$ *folds*. A Figura 10 apresenta a configuração dos experimentos conduzidos com o método SVR para os conjuntos de dados químicos.

¹ Disponível em: <https://cran.r-project.org/web/packages/e1071/index.html>.

Figura 10 – Configuração dos experimentos com método SVR para dados químicos.

Fonte: Rodolfo Simões, 2016.

5.3 Configuração dos experimentos com transferência de aprendizagem

Para um melhor entendimento dos experimentos com transferência de aprendizagem, considera-se um conjunto de treinamento de destino $\mathbf{X}_t = \{(\mathbf{x}_1, y_1), \dots, (\mathbf{x}_n, y_n)\}$ e um conjunto de teste de destino $\mathbf{T}_t = \{\mathbf{t}_1, \dots, \mathbf{t}_r\}$. Nesse caso, $\mathbf{x}_i \in \mathbb{R}^d$ é o i -ésimo exemplo de treinamento com d características; $y_i \in \mathbb{R}$ é o valor de $\mathbf{x}_i$ a ser previsto; $\mathbf{t}_i \in \mathbb{R}^d$ é a i -ésima instância do conjunto de teste de destino com d características, e n e r representam o número de instâncias de treinamento e de teste, respectivamente. Além disso, considera-se um conjunto de origem $\mathbf{X}_s = \{(\mathbf{s}_1; l_1), \dots, (\mathbf{s}_m; l_m)\}$, em que $\mathbf{s}_i \in \mathbb{R}^p$ é o i -ésimo exemplo com p características e $l_i \in \mathbb{R}$ é o valor de $\mathbf{s}_i$ a ser previsto. Nesse cenário, o objetivo é melhorar o desempenho de predição do modelo no conjunto de teste de destino ($\mathbf{T}_t$) via técnicas de transferência de aprendizagem, utilizando o conhecimento gerado a partir dos conjuntos de origem ($\mathbf{X}_s$) e de destino ($\mathbf{X}_t$).

Visto que os experimentos aqui conduzidos pertencem à abordagem de transferência de aprendizagem indutiva ($p = d$, conforme descrito na Seção 2), não é necessário realizar pré-processamento para redução de características em $\mathbf{X}_s$ ou $\mathbf{X}_t$.

5.3.1 Configuração dos experimentos com o método SVR-ATL

Durante a execução do método SVR-ATL, gera-se concomitantemente os modelos de origem e de destino, utilizando os conjuntos de treinamento de ambas as tarefas. Para isso, utilizou-se a mesma configuração aplicada nas análises do método SVR (ver Figura

10). Os conjuntos de treinamento de origem mantiveram o seu tamanho original (sem remoção de instâncias ou divisão do conjunto). Ao final da execução do método, obtém-se os modelos de origem e destino penalizados (ver Seção 3.3), no qual, apenas o modelo resultante da tarefa de destino é avaliado para os dados desconhecidos da tarefa de destino.

Para a configuração do método SVR-ATL, empregou-se a função de otimização quadrática *quadprog*² do Matlab. O problema de otimização dual da Eq.(15) foi solucionado via função quadrática para *kernel* linear, sendo que os valores a serem previstos nos modelos de origem e destino são dados como parâmetros, além da seguinte matriz:

$$KL = \begin{bmatrix} -3/4 * \mathbf{X}_s * \mathbf{X}_s' - eye(M) * 1/C & -1/4 * \mathbf{X}_s * \mathbf{X}_t' \\ -1/4 * \mathbf{X}_t * \mathbf{X}_s' & -3/4 * \mathbf{X}_t * \mathbf{X}_t' - eye(N) * 1/C \end{bmatrix}, \quad (18)$$

em que $\mathbf{X}_s$ é o conjunto de treinamento da tarefa de origem; M é o número de instâncias do conjunto de origem; $\mathbf{X}_t$ é o conjunto de treinamento da tarefa de destino experimentado e N é o número de instâncias do conjunto de destino. As seguintes variações no parâmetro de custo foram consideradas: $C = 100, 10$ e 0.1 .

No caso do *kernel* polinomial, a seguinte matriz é dada como parâmetro:

$$KP = \begin{bmatrix} -3/4 * (\mathbf{X}_s * \mathbf{X}_s' + 1)^2 - eye(M) * 1/C & -1/4 * (\mathbf{X}_s * \mathbf{X}_t' + 1)^2 \\ -1/4 * (\mathbf{X}_t * \mathbf{X}_s' + 1)^2 & -3/4 * (\mathbf{X}_t * \mathbf{X}_t' + 1)^2 - eye(N) * 1/C \end{bmatrix}, \quad (19)$$

em que $\mathbf{X}_s$ e $\mathbf{X}_t$ representam os conjuntos de treinamento de origem e de destino, respectivamente. Os números de instâncias dos conjuntos de treinamentos são representados por M e N e considera-se os valores do parâmetro custo $C = 100, 10$ e 0.1 . Além disso, a configuração utilizada para *kernel* polinomial é igual à considerada para o método SVR, com valores para os parâmetros *degree*=2 e *coef.0*=1.

5.3.2 Configuração dos experimentos com o método TrAdaBoost

O método TrAdaBoost foi avaliado da mesma forma que o método SVR-ATL, considerando diferentes tamanhos de conjuntos de treinamento de destino e o conjunto de treinamento de origem com seu tamanho original. Como mencionado anteriormente, em cada iteração do algoritmo TrAdaBoost, atribuí-se pesos às instâncias de destino e

² Disponível em: <https://www.mathworks.com/help/optim/ug/quadprog.html>.

de origem em função do erro obtido. Ao final da execução, um número arbitrário de instâncias com maior peso são selecionadas para formar um novo conjunto de dados de treinamento. Em seguida, o método SVR é aplicado ao conjunto formado, utilizando a mesma configuração experimental apresentada na Seção 5.2.

Como ainda não existe uma heurística para estimar a quantidade ótima de instâncias a serem selecionadas no final da execução do método TrAdaBoost, optou-se, neste trabalho, por um esquema que reside em aplicar a seleção para cinco diferentes proporções do conjunto resultante, como mostra a Tabela 3. Nesse caso, n representa o número de instâncias consideradas no conjunto de treinamento da tarefa de destino para o experimento em questão; m representa o número total de instâncias disponíveis no conjunto de treinamento da tarefa de origem; t é o número resultante de instâncias a ser selecionadas para formar o conjunto de treinamento final.

Tabela 3 – Esquema para seleção de instâncias pelo método TrAdaBoost.

Número t de instâncias do conjunto de treinamento final	
Experimento 1	$t = n + 20\%m$
Experimento 2	$t = n + 40\%m$
Experimento 3	$t = n + 60\%m$
Experimento 4	$t = n + 80\%m$
Experimento 5	$t = n + 100\%m$

Após a formação dos conjuntos de treinamento, o método SVR é aplicado em cada conjunto definido na estratégia, seguindo a mesma configuração experimental apresentada na Seção 5.2. Por fim, o melhor resultado para cada cenário é considerado na análise comparativa com os demais métodos de regressão. Os resultados de todos os experimentos com o método TrAdaBoost são apresentados no Apêndice B. A seguir, são apresentados e discutidos os resultados obtidos nas análises experimentais deste trabalho.

5.4 Resultados experimentais com dados de benchmark

A metodologia adotada neste trabalho para o desenho dos experimentos e para a escolha dos conjuntos de *benchmark* foram inspirados em (GARCKE; VANCK, 2014). Os autores consideraram, em seus experimentos, subconjuntos com tamanhos arbitrários variados e extraídos dos conjuntos de treinamento da tarefa de destino. Portanto, considera-se aqui, os mesmos tamanhos definidos para os subconjuntos.

5.4.1 Conjunto de dados *Kin-family*

Primeiramente, considerou-se a base de dados *Kin-family*³, pertencente ao repositório Delve. Os experimentos conduzidos com os conjuntos de dados de tal repositório possuem o objetivo de avaliar não apenas a medida de desempenho de um método de regressão específico, e sim o desempenho deste método considerando problemas com diferentes dimensionalidades, níveis de ruídos e grau de não linearidade de uma função. Para isso, disponibiliza-se variações de um mesmo conjunto experimental.

A base de dados *Kin-family* possui oito conjuntos de dados, sendo todos variações de um mesmo modelo experimental constituído de simulações reais com um braço robótico. A tarefa consiste em prever a distância do *end-effector*⁴ (dispositivo no final de um braço robótico, projetado para interagir com o meio ambiente) a um determinado alvo. Os dados foram especificamente gerados para o ambiente de análise Delve⁵ e, portanto, divergem quanto ao número de características, grau de não linearidade e a quantidade de ruído (moderada ou alta). Para os experimentos deste trabalho considerou-se os subconjuntos com 8 dimensões (número de entradas) e graus de não linearidade iguais a *nm* (variância não linear média) com alto ruído e *nh* (variância não linear alta) com ruído moderado.

Seguindo a metodologia utilizada nos experimentos realizados por Garcke e Vanck (2014), definiu-se o conjunto com variância não linear alta (com 8192 instâncias) como conjunto de destino; e o conjunto com variância não linear média (com 8192 instâncias) como conjunto de origem.

- Discussão dos resultados

Apresenta-se, na Tabela 4, os resultados obtidos pelos métodos SVR, SVR-ATL e TrAdaBoost com *kernel* linear considerando os conjuntos de teste da tarefa de destino (*Kin-family* nh). Os valores em destaque ressaltam o melhor resultado para cada cenário (número *n* de instâncias e parâmetro *C* custo). Nota-se que o método TrAdaBoost obteve melhor desempenho para todos os cenários considerados e que o método SVR-ATL também foi superior à abordagem tradicional em todos os experimentos com *kernel* linear.

³ Disponível no repositório delve: <http://www.cs.toronto.edu/~delve/data/kin/desc.html>.

⁴ Em português, efetor final.

⁵ Informações disponíveis em: <https://www.cs.toronto.edu/~delve/data/task-array.html>.

Tabela 4 – Resultados (em valores de EQM) obtidos pelos métodos de regressão com *kernel* linear para o conjunto de dados *Kin-family*.

Conjunto de destino	n	SVR			SVR-ATL			TrAdaBoost		
		$C=0.1$	$C=10$	$C=100$	$C=0.1$	$C=10$	$C=100$	$C=0.1$	$C=10$	$C=100$
<i>Kin-family</i> nh	50	7,36E-01	8,54E-01	8,53E-01	6,86E-01	7,92E-01	7,94E-01	5,64E-01	5,64E-01	5,64E-01
	100	8,05E-01	8,06E-01	8,06E-01	6,88E-01	7,06E-01	7,06E-01	6,73E-01	6,73E-01	6,74E-01
	150	6,76E-01	6,95E-01	6,95E-01	6,07E-01	6,17E-01	6,17E-01	5,90E-01	5,89E-01	5,90E-01
	200	6,34E-01	6,44E-01	6,44E-01	6,09E-01	6,10E-01	6,10E-01	6,00E-01	6,00E-01	6,00E-01

A Figura 11 apresenta uma análise gráfica para cada valor de custo considerado nos experimentos, a fim de ilustrar o impacto da transferência de aprendizagem em relação ao tamanho do conjunto de treinamento da tarefa de destino. Observa-se que o desempenho do método SVR melhorou à medida em que aumentou a quantidade de dados de treinamento, o que é esperado para uma técnica de aprendizagem de máquina tradicional. Nota-se também a partir da Figura 11, que o desempenho dos métodos se aproximaram no cenário com 200 instâncias disponíveis para treinamento na tarefa de destino. Contudo, as abordagens de transferência de aprendizagem ainda continuaram obtendo melhores resultados.

Figura 11 – Análise gráfica do desempenho dos métodos de regressão com *kernel* linear para o conjunto de dados *Kin-family*.

Os resultados obtidos para os métodos de regressão utilizando *kernel* polinomial são apresentados na Tabela 5. Em destaque, os menores valores de EQM obtidos para cada cenário, representando o melhor desempenho.

Tabela 5 – Resultados (em valores de EQM) obtidos pelos métodos de regressão com *kernel* polinomial para o conjunto de dados *Kin-family*.

Conjunto de destino	n	SVR			SVR-ATL			TrAdaBoost		
		$C=0.1$	$C=10$	$C=100$	$C=0.1$	$C=10$	$C=100$	$C=0.1$	$C=10$	$C=100$
<i>Kin-family</i> nh	50	1,14E+00	1,79E+00	1,79E+00	5,08E-01	3,34E+00	5,75E+00	4,12E-01	4,11E-01	4,07E-01
	100	1,50E+00	2,11E+00	2,14E+00	1,04E+00	1,62E+00	1,64E+00	6,53E-01	6,53E-01	6,56E-01
	150	8,11E-01	9,09E-01	1,01E+00	6,58E-01	7,56E-01	7,58E-01	5,25E-01	5,25E-01	5,24E-01
	200	8,21E-01	8,72E-01	9,05E-01	6,77E-01	7,29E-01	7,30E-01	5,42E-01	5,42E-01	5,38E-01

Assim como ocorrido nos experimentos com *kernel* linear, destaca-se aqui o desempenho obtido pelo método TrAdaBoost, que foi superior em todos os cenários. Comparando o desempenho dos métodos SVR e SVR-ATL, observa-se que o segundo obteve melhores resultados na maioria dos cenários.

A Figura 12 apresenta uma análise gráfica para cada valor de custo considerado nos experimentos, a fim de ilustrar o impacto da transferência de aprendizagem em relação ao tamanho do conjunto de treinamento da tarefa de destino. Observa-se que, como ocorrido nos experimentos com *kernel* linear, o desempenho dos métodos se aproximaram nos cenários com mais instâncias disponíveis para treinamento na tarefa de destino. Contudo, as abordagens de transferência de aprendizagem ainda continuaram obtendo melhores resultados.

Nos resultados experimentais analisados aqui, as abordagens de transferência de aprendizagem obtiveram melhores resultados do que a abordagem tradicional. Além disso, observa-se que o parâmetro custo influenciou no desempenho dos métodos SVR e SVR-ATL, uma vez que o aumento dos valores do parâmetro C reduziu o desempenho dos métodos. Outro ponto a ser destacado é que, em relação a análise com *kernel* linear, os métodos SVR e SVR-ATL tiveram desempenho inferior nos experimentos com *kernel* polinomial, ao contrário do método TrAdaBoost. De modo geral, os resultados apresentados indicam que a transferência de aprendizagem foi benéfica para todos os cenários experimentais.

Figura 12 – Análise gráfica do desempenho dos métodos de regressão com *kernel* linear para o conjunto de dados *Kin-family*.

5.4.2 Conjunto de dados *Earthquake*

O conjunto de dados *Earthquake* (GARCKE; VANCK, 2014) contém observações sobre terremotos ocorridos em diferentes países. Nesse contexto, um modelo preditivo gerado a partir dos dados medidos na Califórnia, por exemplo, seria adequado somente para fazer previsões futuras de terremotos para essa localidade, podendo não ser apropriado para obtenção de estimativas em outros países. No entanto, se os dados coletados no Japão, por exemplo, forem muito escassos, um modelo gerado apenas a partir desses dados locais pode não apresentar um bom poder de generalização. Com isso, embora as distribuições de probabilidade dos dados da Califórnia e do Japão possam diferir, em geral é razoável supor que é possível utilizar o modelo gerado a partir dos dados da Califórnia para aprimorar o modelo gerado a partir dos dados coletados no Japão. A tarefa de regressão, nesse caso, consiste em prever o valor de aceleração máxima do solo que ocorre durante o terremoto em um determinado local.

Seguindo a metodologia de Garcke e Vanck (2014), resolveu-se utilizar os dados coletados na Califórnia como conjunto de origem (com 841 instâncias) e os dados coletados

no Japão como conjunto de destino (com 3542 instâncias). Os conjuntos de testes utilizados nos experimentos são todos pertencentes aos dados coletados no Japão.

- Discussão dos resultados

Apresenta-se, na Tabela 6, os resultados obtidos pelos métodos SVR, SVR-ATL e TrAdaBoost nos experimentos conduzidos considerando os conjuntos de teste da tarefa de destino (Japão). Os valores em destaque ressaltam o melhor resultado atingido para cada cenário. Nota-se que em todos os cenários, pelo menos uma das técnicas de transferência de aprendizagem superou a abordagem tradicional.

Tabela 6 – Resultados (em valores de EQM) obtidos pelos métodos de regressão com *kernel* linear para o conjunto de dados *Earthquake*.

Conjunto de destino	n	SVR			SVR-ATL			TrAdaBoost		
		$C=0.1$	$C=10$	$C=100$	$C=0.1$	$C=10$	$C=100$	$C=0.1$	$C=10$	$C=100$
Japão	20	7,24E-01	7,01E-01	7,01E-01	6,10E-01	7,52E-01	7,67E-01	4,20E-01	3,67E-01	3,47E-01
	30	1,14E+00	1,49E+00	1,49E+00	9,47E-01	1,43E+00	1,46E+00	7,80E-01	8,09E-01	8,70E-01
	50	4,40E-01	5,58E-01	5,62E-01	4,19E-01	4,44E-01	4,45E-01	4,83E-01	4,89E-01	4,90E-01
	70	4,90E-01	5,57E-01	5,57E-01	4,64E-01	4,94E-01	4,95E-01	4,78E-01	4,81E-01	4,85E-01

A Figura 13 apresenta uma análise gráfica para cada valor do parâmetro custo considerado nos experimentos, a fim de ilustrar o impacto da transferência de aprendizagem em relação ao tamanho do conjunto de treinamento da tarefa de destino. Observa-se que, o método TrAdaBoost possui destaque principalmente para os experimentos com menor quantidades de dados de treinamento disponíveis, e que o desempenho do método SVR melhorou à medida em que aumentou a quantidade de dados de treinamento, de tal forma, que os desempenhos dos métodos se aproximaram nos experimentos com 50 instâncias disponíveis para treinamento na tarefa de destino. Contudo, as abordagens de transferência de aprendizagem tiveram melhores resultados com *kernel* linear.

Figura 13 – Análise gráfica do desempenho dos métodos de regressão com *kernel* linear para o conjunto de dados *Earthquake*.

Os resultados obtidos para os métodos de regressão utilizando *kernel* polinomial são apresentados na Tabela 7. Em destaque, os melhores resultados obtidos para cada cenário. Nota-se que o método TrAdaBoost obteve melhor desempenho para todos os cenários nos experimentos com *kernel* polinomial. Além disso, o método SVR-ATL também obteve resultados superiores aos obtidos pelo método SVR na maioria dos cenários. Ressalta-se também que o parâmetro custo, para os experimentos com menor quantidade de dados disponíveis na tarefa de destino, influenciou nos resultados obtidos pelos métodos SVR e SVR-ATL com *kernel* polinomial, uma vez que o aumento nos valores do parâmetro levou à redução do desempenho dos métodos.

Tabela 7 – Resultados (em valores de EQM) obtidos pelos métodos de regressão com *kernel* polinomial para o conjunto de dados *Earthquake*.

Conjunto de destino	n	SVR			SVR-ATL			TrAdaBoost		
		C=0.1	C=10	C=100	C=0.1	C=10	C=100	C=0.1	C=10	C=100
Japão	20	5,48E-01	1,33E+01	2,94E+01	5,41E-01	1,41E+00	1,59E+01	4,09E-01	4,03E-01	1,26E+00
	30	1,07E+00	5,12E+01	5,26E+01	8,81E-01	2,30E+01	8,90E+01	6,69E-01	6,56E-01	1,08E+00
	50	6,79E-01	1,38E+00	1,81E+00	4,57E-01	1,04E+00	1,30E+00	4,16E-01	4,74E-01	4,88E-01
	70	5,13E-01	6,78E-01	6,96E-01	4,40E-01	5,83E-01	6,02E-01	4,20E-01	4,73E-01	4,97E-01

A Figura 14 apresenta uma análise gráfica para cada valor do parâmetro custo considerado nos experimentos com *kernel* polinomial, a fim de ilustrar o impacto da

transferência de aprendizagem em relação ao tamanho do conjunto de treinamento da tarefa de destino. Nota-se que o método TrAdaBoost possui destaque principalmente para os experimentos com menos dados de treinamento disponíveis, e que os desempenhos dos métodos se aproximaram nos experimentos conduzidos com 70 instâncias de treinamento na tarefa de destino. Mas, de modo geral, observa-se o benefício da transferência de aprendizagem para os experimentos conduzidos com *kernel* polinomial, destacando-se o método TrAdaBoost.

Figura 14 – Análise gráfica do desempenho dos métodos de regressão com *kernel* polinomial para o conjunto de dados *Earthquake*.

Nos resultados experimentais analisados nesta seção, os métodos SVR-ATL e TrAdaBoost obtiveram melhores resultados que a abordagem tradicional, o que indica que a transferência de aprendizagem foi benéfica em tais experimentos.

5.4.3 Conjunto de dados *Fly*

O conjunto de dados *Fly*⁶ descreve os detalhes de chegada e partida para todos os voos comerciais nos Estados Unidos no período de outubro de 1987 a abril de 2008. Para os experimentos deste trabalho, utilizou-se os registros dos anos de 2007 e 2008. A tarefa

⁶ Conjunto de dados disponível em: <http://stat-computing.org/dataexpo/2009/>.

de regressão considerada nos experimentos, ou seja, o valor a ser predito é o atraso de um determinado voo.

Seguindo a metodologia utilizada por [Garcke e Vanck \(2014\)](#), resolveu-se utilizar os dados coletados no ano de 2007 como conjunto de origem (com 7453215 instâncias); e os dados coletados no ano de 2008 como conjunto de destino (com 7009728 instâncias). Os conjuntos de testes utilizados nos experimentos são todos pertencentes aos dados coletados no ano de 2008. A justificativa para utilização de transferência de aprendizagem nesse contexto deve-se à possibilidade de haver diferença na distribuições de probabilidade entre os dois conjuntos, ocasionada pela diferença no período de tempo da coleta dos dados.

- Discussão dos resultados

Apresenta-se, na Tabela 8, os resultados obtidos pelos métodos SVR, SVR-ATL e TrAdaBoost nos experimentos com *kernel* linear considerando os conjuntos de teste da tarefa de destino (2008). Os valores em destaque ressaltam o melhor resultado atingido para cada cenário.

Tabela 8 – Resultados (em valores de EQM) obtidos pelos métodos de regressão com *kernel* linear para o conjunto de dados *Fly*.

Conjunto de destino	n	SVR			SVR-ATL			TrAdaBoost		
		$C=0.1$	$C=10$	$C=100$	$C=0.1$	$C=10$	$C=100$	$C=0.1$	$C=10$	$C=100$
2008	50	5,38E-01	8,32E-02	8,32E-02	3,18E-01	8,22E-03	2,58E-04	1,06E-01	1,04E-01	1,07E-01
	200	1,07E-02	4,96E-03	4,96E-03	5,40E-02	6,49E-05	5,20E-06	1,57E-01	1,50E-01	1,48E-01
	400	5,80E-03	3,33E-03	3,33E-03	2,09E-02	2,09E-05	2,81E-07	1,61E-01	1,49E-01	1,51E-01
	800	2,72E-03	2,15E-03	2,15E-03	7,37E-03	3,97E-06	4,64E-08	1,04E-01	9,40E-02	1,03E-01

Observando os resultados apresentados na Tabela 8, nota-se que o método SVR-ATL obteve desempenho superior para todos os cenários com custo $C = 10$ e $C = 100$. Para os cenários com custo $C = 0.1$, o método TrAdaBoost foi superior apenas com $n = 50$. De modo geral, os melhores resultados foram obtidos pelo método SVR-ATL utilizando o parâmetro custo $C = 100$.

A Figura 15 apresenta uma análise gráfica para cada valor do parâmetro custo considerado nos experimentos, a fim de ilustrar o impacto da transferência de aprendizagem em relação ao tamanho do conjunto de treinamento da tarefa de destino. Analisando a Figura, percebe-se que o método SVR-ATL obteve destaque nos experimentos com o

parâmetro custo $C = 10$ e $C = 100$, e também que o método TrAdaBoost não atingiu desempenho satisfatório, comparando-o com o desempenho dos demais métodos.

Figura 15 – Análise gráfica do desempenho dos métodos de regressão com *kernel* linear para o conjunto de dados *Fly*.

Os resultados dos experimentos conduzidos com *kernel* polinomial são apresentados na Tabela 9. Em destaque, os melhores resultados para cada cenário. Nota-se que os métodos SVR-ATL e TrAdaBoost obtiveram os melhores resultados para a maioria dos cenários. Em um comparativo direto entre os desempenhos obtidos pelos métodos SVR-ATL e SVR, constata-se que o método SVR-ATL atingiu desempenho superior para todos os cenários, exceto no cenário com $n = 800$ e $C = 0.1$. Além disso, o método TrAdaBoost obteve desempenho superior ao método SVR na maioria dos cenários considerados nos experimentos com *kernel* polinomial.

Tabela 9 – Resultados (em valores de EQM) obtidos pelos métodos de regressão com *kernel* polinomial para o conjunto de dados *Fly*.

Conjunto de destino	n	SVR			SVR-ATL			TrAdaBoost		
		$C=0.1$	$C=10$	$C=100$	$C=0.1$	$C=10$	$C=100$	$C=0.1$	$C=10$	$C=100$
Voo 2008	50	1,59E+00	1,59E+00	1,59E+00	7,01E-01	1,12E+00	1,13E+00	1,16E-01	3,88E-01	3,21E-01
	200	2,48E-01	2,60E-01	2,60E-01	2,00E-01	8,52E-02	4,11E-02	1,48E-01	3,62E-01	2,43E-01
	400	3,83E-01	3,65E-01	3,65E-01	2,62E-01	9,00E-03	2,23E-03	1,87E-01	2,21E-01	1,38E+00
	800	1,30E-02	1,30E-02	1,30E-02	1,89E-02	4,53E-04	2,35E-04	1,61E-01	1,67E+00	9,90E-01

A Figura 16 apresenta uma análise gráfica para cada valor de custo considerado nos experimentos, a fim de ilustrar o impacto da transferência de aprendizagem em relação ao tamanho do conjunto de treinamento da tarefa de destino.

Figura 16 – Análise gráfica do desempenho dos métodos de regressão com *kernel* polinomial para o conjunto de dados *Fly*.

A partir da Figura 16, é possível observar que o método SVR-ATL atingiu desempenho superior aos demais métodos nos cenários analisados. Outro ponto a ser destacado é que o método SVR-ATL obteve ganho de desempenho à medida que a quantidade de dados treinamento e o valor do parâmetro custo aumentaram, ou seja, melhor desempenho atingido no cenário com $n = 800$ e parâmetro custo $C = 100$.

Nos resultados experimentais analisados nesta seção, destaca-se o desempenho obtido pelo método SVR-ATL, sendo que a configuração experimental que obteve melhor desempenho para os cenários analisados, foi obtida com o *kernel* linear com custo $C = 100$. Portanto, nos experimentos aqui conduzidos, a transferência de aprendizagem mostrou-se benéfica ao processo de aprendizagem da tarefa de destino.

5.5 Resultados experimentais com dados químicos

Como discutido anteriormente, o processo de descoberta de medicamentos envolve o uso de metodologias híbridas para o planejamento de novas substâncias bioativas, que podem ser candidatas a novos fármacos. Uma estratégia valiosa nesse contexto, é a predição da atividade biológica de substâncias químicas a partir de um conjunto de descritores atômicos e moleculares, conhecido como estudos QSAR. Portanto, as metodologias QSAR são usadas para desenvolver modelos estatísticos que relacionam a estrutura de compostos químicos e a sua atividade biológica com relação a um alvo no organismo (GERTRUDES et al., 2012).

Nesta seção, são discutidos os experimentos conduzidos neste trabalho e que visaram a tratar o problema da falta de informação disponível para a construção de modelos preditivos, por meio dos estudos QSAR. A metodologia adotada para o desenho dos experimentos com conjuntos de dados químicos foi inspirada em (GIRSCHICK; RÜCKERT; KRAMER, 2012). Esses autores realizaram experimentos de regressão com proporções variadas do conjunto de treinamento disponível para avaliar o *trade-off* entre o número de exemplos disponíveis na fase de treinamento e o desempenho obtido por diferentes métodos. Mais especificamente, foram considerados nesses experimentos, conjuntos de destino equivalentes a 25%, 50%, 75% e 100% do tamanho do conjunto de treinamento original. Além disso, definiu-se como conjunto de origem o conjunto que possui o maior número de instâncias disponíveis, dada a motivação de que a transferência de aprendizagem é benéfica se aplicada a partir de um conjunto de dados com um grande número de instâncias disponíveis para outro com poucas instâncias disponíveis na fase de treinamento (ROSENBAUM et al., 2013).

5.5.1 Conjunto de dados ALK-5

Segundo Jorde (2004), o câncer é considerado uma das maiores causas de morte no mundo e é definido como uma doença que surge como consequência de alterações cumulativas no material genético (DNA) de células normais, as quais podem sofrer transformações até se tornarem malignas. Um dos principais alvos da doença é o receptor tipo 1 do fator β de transformação do crescimento celular (TGF β R1)⁷. A proteína

⁷ Do original, em Inglês, *TransformingGrowthFactor- β receptor type 1*.

conhecida como *Activin-Like Kinase 5* (ALK-5), membro da superfamília TGF β , controla a proliferação e diferenciação celular, associada também a doenças cardíacas, diabetes e síndrome de Marfan. Além disso, essa proteína é fundamental na cicatrização de feridas e no controle do sistema imunológico (BUDD; HOLMES, 2012).

Portanto, para o primeiro par de conjuntos químicos analisado neste trabalho, tem-se como alvo biológico a proteína ALK-5. Nesse caso, a tarefa de regressão consiste em prever o nível de inibição dos compostos com relação ao alvo, medido pelo valor IC_{50} . Os conjuntos de dados analisados pertencem a estudos realizados pelo grupo de pesquisa GrIA⁸ e estão publicados em (ARAUJO et al., 2015) e (ALMEIDA et al., 2016).

Nos experimentos realizados com o alvo ALK-5, considerou-se o conjunto de Araujo et al. (2015) com apenas 59 instâncias, como conjunto de destino; e o conjunto disponível em (ALMEIDA et al., 2016) com 117 instâncias, como conjunto de origem.

• Discussão dos resultados

Apresenta-se, na Tabela 10, os resultados obtidos pelos métodos SVR, SVR-ATL e TrAdaBoost nos experimentos conduzidos considerando os conjuntos de teste da tarefa de destino. Os valores em destaque ressaltam o melhor resultado para cada cenário (percentual n de instâncias no conjunto de destino e parâmetro C custo).

Tabela 10 – Resultados (em valores de EQM) obtidos pelos métodos de regressão com *kernel* linear para o conjunto de dados ALK-5.

Conjunto de destino	n	SVR			SVR-ATL			TrAdaBoost		
		$C=0.1$	$C=10$	$C=100$	$C=0.1$	$C=10$	$C=100$	$C=0.1$	$C=10$	$C=100$
ALK-5	25%	4,69E+00	8,89E+00	5,86E+01	1,01E+00	4,59E+00	1,12E+01	2,52E+00	2,96E+00	2,99E+00
	50%	1,02E+00	1,31E+00	8,39E+00	7,22E-01	1,17E+00	2,05E+00	1,21E+00	1,31E+00	1,32E+00
	75%	8,05E-01	7,53E-01	7,60E-01	7,23E-01	8,38E-01	9,09E-01	1,00E+00	9,16E-01	9,14E-01
	100%	1,08E+00	1,09E+00	1,05E+00	7,71E-01	7,99E-01	9,14E-01	1,06E+00	1,03E+00	1,01E+00

Observando-se os resultados apresentados na Tabela 10, destaca-se o desempenho obtido pelo método SVR-ATL, que obteve os melhores resultados com a configuração com parâmetro custo $C = 0.1$. Além disso, o método TrAdaBoost obteve resultados superiores ao método SVR na maioria dos cenários.

A Figura 17 apresenta uma análise gráfica para cada valor do parâmetro custo considerado nos experimentos, a fim de ilustrar o impacto da transferência de aprendizagem

⁸ Grupo de Inteligência Artificial. (<http://www.each.usp.br/gria/>)

em relação ao tamanho do conjunto de treinamento da tarefa de destino. Analisando a figura, destaca-se o desempenho do método SVR-ATL no cenário com número limitado de instâncias para a fase de treinamento ($n = 25\%$). Além disso nota-se que o desempenho dos métodos de regressão aproximaram-se a partir dos experimentos conduzidos com 50%. Contudo, o método SVR-ATL obteve desempenho superior.

Figura 17 – Análise gráfica do desempenho dos métodos de regressão com *kernel* linear para o conjunto de dados ALK-5.

Os resultados obtidos para os métodos de regressão utilizando *kernel* polinomial são apresentados na Tabela 11, em destaque, os melhores resultados para cada cenário. Nota-se, que para todos os cenários experimentais, os métodos SVR-ATL e TrAdaBoost obtiveram desempenho superior ao método SVR.

Tabela 11 – Resultados (em valores de EQM) obtidos pelos métodos de regressão com *kernel* polinomial para o conjunto de dados ALK-5.

Conjunto de destino	n	SVR			SVR-ATL			TrAdaBoost		
		$C=0.1$	$C=10$	$C=100$	$C=0.1$	$C=10$	$C=100$	$C=0.1$	$C=10$	$C=100$
ALK-5	25%	3,78E+03	3,79E+03	3,79E+03	9,06E-01	5,06E-01	6,50E-01	3,35E+00	6,22E+00	6,22E+00
	50%	1,12E+02	5,63E+02	5,63E+02	6,27E-01	2,27E+00	3,74E+00	1,25E+00	1,01E+00	1,01E+00
	75%	4,48E+02	2,04E+03	2,05E+03	6,29E-01	3,61E+00	9,22E+00	9,14E-01	1,23E+00	1,25E+00
	100%	2,70E+02	1,51E+04	2,06E+04	8,71E-01	7,86E+00	7,95E+01	1,24E+00	1,72E+00	1,75E+00

A Figura 18 apresenta uma análise gráfica dos resultados para cada valor de custo considerado nos experimentos com *kernel* polinomial, a fim de ilustrar o impacto da

transferência de aprendizagem em relação ao tamanho do conjunto de treinamento na tarefa de destino.

Figura 18 – Análise gráfica do desempenho dos métodos de regressão com *kernel* polinomial para o conjunto de dados ALK-5.

Analisando a Figura 18, percebe-se que o método SVR não obteve desempenho satisfatório nos experimentos com *kernel* polinomial, possivelmente ocasionado pelo número limitado de instâncias para a fase de treinamento na tarefa de destino. Em tal situação, nota-se o desempenho superior e o benefício da aplicação de técnicas de transferência de aprendizagem.

Para os resultados experimentais apresentados nesta seção, destaca-se o desempenho obtido pelo método SVR-ATL com parâmetro custo $C = 0.1$ com *kernel* linear e polinomial. Portanto, nos experimentos conduzidos com os conjuntos de dados químicos com o alvo ALK-5, a transferência de aprendizagem mostrou-se benéfica para melhorar o desempenho na tarefa de destino.

5.5.2 Conjunto de dados PPAR- α

Os conjuntos considerados nesta seção são utilizados para a geração de modelos estatísticos capazes de descobrir moléculas candidatas a fármacos para o tratamento da

diabetes mellitus (DM) e síndrome metabólica (SM). A DM é uma doença caracterizada como distúrbio crônico do metabolismo de carboidratos, lipídios e proteínas. Outra característica dessa enfermidade é a utilização inadequada de glicose pelo organismo, ocasionando a hiperglicemia (quantidade elevada de glicose no sangue). A SM é um transtorno do metabolismo e está relacionada a um conjunto de fatores de riscos cardiovasculares e obesidade. Estão ligadas a este distúrbio características como resistência à insulina (o que também causa a hiperglicemia), alto índice de ácidos graxos livres, hipertensão arterial e outros distúrbios cardiovasculares (GERTRUDES, 2013). Por possuírem características semelhantes, a DM e a SM possuem o mesmo alvo biológico, o receptor PPAR⁹, responsável pelo controle dos níveis de glicose e lipídeos (MALTAROLLO; HONORIO, 2012). Além disso, três tipos de PPAR foram identificados: PPAR- α , PPAR- γ e PPAR- δ . O par de conjunto de dados analisado aqui é o PPAR- α , expresso no fígado. A tarefa de regressão nesse contexto, consiste em prever o nível de inibição dos compostos com relação ao alvo biológico, medido pelo valor IC_{50} .

Nos experimentos realizados com o alvo PPAR- α , considerou-se o conjunto disponível em (MALTAROLLO; HONORIO, 2012) como conjunto de destino (com 70 instâncias). Além disso, utilizou-se a abordagem de seleção de conjuntos de dados dirigida (Seção 3.5.1), a partir do repositório Pubchem¹⁰ para realizar os experimentos de transferência de aprendizagem. O conjunto de dados selecionado foi o conjunto que atingiu um maior índice de similaridade com o conjunto de dados de destino, no caso, o conjunto identificado no repositório como AID 279988, com 89 instâncias.

• Discussão dos resultados

A Tabela 12 apresenta os resultados obtidos pelos métodos SVR, SVR-ATL e TrAdaBoost nos experimentos com *kernel* linear considerando os conjuntos de teste da tarefa de destino. Os valores em destaque ressaltam o melhor desempenho para cada cenário. Nota-se que as abordagens de transferência de aprendizagem obtiveram desempenho superior em relação ao desempenho obtido pelo método SVR em todos os cenários com $n = 25\%$, $n = 50\%$ e $n = 75\%$. No cenário com $n = 100\%$ de informação disponível para treinamento, o método SVR obteve desempenho superior. O que evidencia a importância

⁹ Do original, em inglês, *Peroxisome Proliferator-Activated Receptor*.

¹⁰ <https://pubchem.ncbi.nlm.nih.gov/>.

das técnicas de transferência de aprendizagem em cenários com menor quantidade de dados disponíveis para a fase de treinamento na tarefa de destino.

Tabela 12 – Resultados (em valores de EQM) obtidos pelos métodos de regressão com *kernel* linear para o conjunto de dados PPAR- α .

Conjunto de destino	n	SVR			SVR-ATL			TrAdaBoost		
		$C=0.1$	$C=10$	$C=100$	$C=0.1$	$C=10$	$C=100$	$C=0.1$	$C=10$	$C=100$
PPAR- α	25%	1,34E+00	8,59E+00	6,81E+00	6,48E-01	4,44E+00	5,97E+00	5,17E-01	5,56E-01	5,59E-01
	50%	8,83E-01	1,41E+00	1,26E+00	6,80E-01	1,25E+00	1,31E+00	5,85E-01	5,89E-01	5,89E-01
	75%	5,60E-01	6,63E-01	6,94E-01	4,93E-01	5,52E-01	5,85E-01	5,48E-01	5,61E-01	5,63E-01
	100%	5,36E-01	4,84E-01	4,85E-01	5,49E-01	6,34E-01	6,38E-01	5,38E-01	5,39E-01	5,33E-01

A Figura 19 apresenta uma análise gráfica para cada valor do parâmetro custo considerado nos experimentos, a fim de ilustrar o impacto da transferência de aprendizagem em relação ao tamanho do conjunto de treinamento da tarefa de destino. Destaca-se o desempenho do método TrAdaBoost nos cenários com $n = 25\%$ e $n = 50\%$.

Figura 19 – Análise gráfica do desempenho dos métodos de regressão com *kernel* linear para o conjunto de dados PPAR- α .

Os resultados obtidos para os métodos de regressão utilizando *kernel* polinomial são apresentados na Tabela 13. Destaca-se o desempenho obtido pelo método TrAdaBoost. O método obteve os melhores resultados para quase todos os cenários. Além disso, o método SVR-ATL também obteve desempenho superior em relação ao método SVR.

Tabela 13 – Resultados (em valores de EQM) obtidos pelos métodos de regressão com *kernel* polinomial para o conjunto de dados PPAR- α .

Conjunto de destino	n	SVR			SVR-ATL			TrAdaBoost		
		$C=0.1$	$C=10$	$C=100$	$C=0.1$	$C=10$	$C=100$	$C=0.1$	$C=10$	$C=100$
PPAR- α	25%	3,64E+00	1,07E+01	1,12E+01	1,03E+00	3,83E+00	1,12E+01	5,02E-01	6,04E-01	5,98E-01
	50%	2,66E+00	3,36E+00	7,41E+01	9,73E-01	1,61E+01	3,71E+01	4,40E-01	5,87E-01	7,49E-01
	75%	6,69E-01	1,66E+00	2,63E+00	5,51E-01	9,81E-01	1,02E+00	5,22E-01	5,70E-01	6,00E-01
	100%	4,98E+00	2,30E+01	3,95E+01	8,54E-01	5,49E+00	1,01E+01	8,64E-01	2,36E+00	2,24E+00

A Figura 20 apresenta uma análise gráfica para cada valor do parâmetro custo considerado nos experimentos, a fim de ilustrar o impacto da transferência de aprendizagem em relação ao tamanho do conjunto de treinamento da tarefa de destino. Analisando a Figura, destaca-se o desempenho do método TrAdaBoost nos cenários com menor quantidade de dados.

Figura 20 – Análise gráfica do desempenho dos métodos de regressão com *kernel* polinomial para o conjunto de dados PPAR- α .

Para os resultados apresentados nesta seção, nota-se que os métodos SVR-ATL e TrAdaBoost obtiveram destaque principalmente nos cenários com menos dados de treinamento disponíveis na tarefa de destino, o que ressalta a importância da utilização das técnicas de transferência de aprendizagem em tais situações.

5.5.3 Conjunto de dados PPAR- δ

O par de conjunto de dados analisado nesta seção contém moléculas possíveis candidatas a fármacos para o alvo PPAR- δ , que pertence também ao estudo PPAR (descrito na Seção 5.5.2) e regula o metabolismo lipídico (incluindo absorção e transporte de lipídios), resistência à insulina e redução de peso, e é principalmente expresso em vários tecidos e células. A tarefa de regressão também consiste em prever o nível de inibição dos compostos com o alvo, medido pelo valor IC_{50} .

Nos experimentos realizados com o alvo PPAR- δ , considerou-se o conjunto disponível em (MALTAROLLO; HONORIO, 2012) como conjunto de destino (com 70 instâncias). Além disso, utilizou-se a abordagem de seleção de conjuntos de dados dirigida, a partir do repositório Pubchem para realizar os experimentos de transferência de aprendizagem. O conjunto de dados selecionado foi o conjunto que atingiu um maior índice de similaridade com o conjunto de dados de destino, no caso, o conjunto identificado no repositório como AID 279989, com 89 instâncias.

• Discussão dos resultados

Apresenta-se, na Tabela 14, os resultados obtidos pelos métodos SVR, SVR-ATL e TrAdaBoost com *kernel* linear nos experimentos conduzidos considerando os conjuntos de teste da tarefa de destino. Nota-se que em todos os cenários, pelo menos uma das técnicas de transferência de aprendizagem superou a abordagem tradicional.

Tabela 14 – Resultados (em valores de EQM) obtidos pelos métodos de regressão com *kernel* linear para o conjunto de dados PPAR- δ .

Conjunto de destino	n	SVR			SVR-ATL			TrAdaBoost		
		$C=0.1$	$C=10$	$C=100$	$C=0.1$	$C=10$	$C=100$	$C=0.1$	$C=10$	$C=100$
PPAR- δ	25%	1,09E+00	1,42E+00	1,87E+00	8,51E-01	1,41E+00	1,58E+00	7,18E-01	6,16E-01	6,15E-01
	50%	1,00E+00	1,16E+00	1,19E+00	9,63E-01	1,20E+00	1,22E+00	9,59E-01	9,68E-01	9,68E-01
	75%	1,09E+00	1,10E+00	1,10E+00	1,04E+00	1,21E+00	1,21E+00	1,10E+00	1,10E+00	1,09E+00
	100%	9,32E-01	9,72E-01	9,73E-01	8,90E-01	9,63E-01	9,67E-01	9,59E-01	9,91E-01	1,00E+00

A Figura 21 apresenta uma análise gráfica para cada valor de custo considerado nos experimentos, a fim de ilustrar o impacto da transferência de aprendizagem em relação ao tamanho do conjunto de treinamento da tarefa de destino.

Figura 21 – Análise gráfica do desempenho dos métodos de regressão com *kernel* linear para o conjunto de dados PPAR- δ .

A partir da Figura 21, destaca-se o desempenho do método TrAdaBoost nos cenários com menor quantidade de dados de treinamento disponíveis. Nota-se também, que o desempenho do método SVR melhorou à medida que aumentou a quantidade de dados de treinamento, contudo, o método SVR-ATL obteve desempenho superior em tais cenários.

Os resultados obtidos para os métodos de regressão com *kernel* polinomial são apresentados na Tabela 15. Destaca-se o método TrAdaBoost que obteve os melhores resultados na maioria dos cenários analisados. Além disso, o método SVR-ATL também obteve desempenho superior em relação ao desempenho obtido pelo método SVR na maioria dos cenários analisados.

Tabela 15 – Resultados (em valores de EQM) obtidos pelos métodos de regressão com *kernel* polinomial para o conjunto de dados PPAR- δ

Conjunto de destino	n	SVR			SVR-ATL			TrAdaBoost		
		C=0.1	C=10	C=100	C=0.1	C=10	C=100	C=0.1	C=10	C=100
PPAR- δ	25%	1,99E+00	4,49E+00	4,49E+00	7,52E-01	1,61E+00	2,62E+00	4,44E-01	3,99E-01	3,71E-01
	50%	1,07E+00	1,02E+01	1,64E+01	1,38E+00	6,65E+00	1,08E+01	2,32E+00	1,23E+00	1,51E+00
	75%	3,46E+00	7,05E+00	1,23E+01	2,39E+00	5,94E+00	7,36E+00	1,23E+00	1,58E+00	1,74E+00
	100%	1,22E+00	3,29E+00	4,95E+00	1,24E+00	2,41E+00	3,35E+00	1,45E+00	2,17E+00	2,01E+00

A Figura 22 apresenta uma análise gráfica para cada valor de custo considerado nos experimentos, a fim de ilustrar o impacto da transferência de aprendizagem em relação ao tamanho do conjunto de treinamento da tarefa de destino. É possível destacar o desempenho do método TrAdaBoost nos cenários com menos dados de treinamento. De modo geral, destaca-se as abordagens de transferência de aprendizagem que obtiveram melhor desempenho na maioria dos cenários analisados.

Figura 22 – Análise gráfica do desempenho dos métodos de regressão com *kernel* polinomial para o conjunto de dados PPAR- δ .

Nos resultados experimentais analisados nesta seção, os métodos SVR-ATL e TrAdaBoost obtiveram melhores resultados do que o método SVR na maioria dos cenários analisados, destaca-se o desempenho do método TrAdaBoost nos cenários em que a tarefa de destino possui menos dados de treinamento à disposição. O que indica que abordagens de transferência de aprendizagem podem ser utilizadas para melhorar o desempenho da tarefa de destino em experimentos de regressão considerando conjuntos de dados químicos referentes ao alvo PPAR- δ .

5.5.4 Conjunto de dados DHFR

O Dihidrofolato redutase (DHFR)¹¹, membro da família da enzima redutase, é uma enzima que reduz o ácido dihidrofolato, pois converte dihidrofolato em tetrahydrofolato. É também a única enzima que reduz o ácido fólico, uma vitamina sintética não encontrada na natureza, para dihidrofolato (MENDES et al., 2010). O objetivo dos experimentos de regressão para esse conjunto é prever o nível de inibição dos compostos com a enzima, medido pelo valor IC_{50} .

O par de conjuntos de dados para o alvo DHFR utilizado nos experimentos deste trabalho, também foi considerado nos experimentos de Girschick, Rückert e Kramer (2012). Considerou-se o conjunto DHFR_S (SUTHERLAND; O'BRIEN; WEAVER, 2003) com 391 instâncias, como conjunto de destino; e o DHFR_4q (SUTHERLAND; O'BRIEN; WEAVER, 2004) com 739 instâncias como conjunto de origem.

- Discussão dos resultados

Apresenta-se, na Tabela 16, os resultados obtidos pelos métodos SVR, SVR-ATL e TrAdaBoost nos experimentos com *kernel* linear considerando os conjuntos de teste da tarefa de destino (DHFR_S). Nota-se que o método SVR-ATL obteve melhor desempenho para todos os cenários considerados.

Tabela 16 – Resultados (em valores de EQM) obtidos pelos métodos de regressão com *kernel* linear para o conjunto de dados DHFR.

Conjunto de destino	n	SVR			SVR-ATL			TrAdaBoost		
		$C=0.1$	$C=10$	$C=100$	$C=0.1$	$C=10$	$C=100$	$C=0.1$	$C=10$	$C=100$
DHFR_S	25%	6,71E-01	6,32E-01	6,19E-01	6,15E-01	5,42E-01	5,31E-01	9,15E-01	9,30E-01	9,33E-01
	50%	5,38E-01	5,33E-01	5,35E-01	5,27E-01	4,98E-01	4,98E-01	6,70E-01	6,69E-01	6,69E-01
	75%	5,35E-01	5,32E-01	5,32E-01	5,28E-01	5,01E-01	5,00E-01	6,54E-01	6,54E-01	6,59E-01
	100%	5,58E-01	5,73E-01	5,74E-01	5,50E-01	5,37E-01	5,38E-01	6,52E-01	6,59E-01	6,66E-01

A Figura 23 apresenta uma análise gráfica para cada valor de custo considerado nos experimentos, a fim de ilustrar o impacto da transferência de aprendizagem em relação ao tamanho do conjunto de treinamento da tarefa de destino. Nota-se, que o método SVR obteve desempenho próximo ao desempenho do método SVR-ATL nos experimentos com custo $C = 0.1$.

¹¹ Do original, em inglês, *Dihydrofolate Reductase*.

Figura 23 – Análise gráfica do desempenho dos métodos de regressão com *kernel* linear para o conjunto de dados DHFR.

Os resultados obtidos para os métodos de regressão utilizando *kernel* polinomial são apresentados na Tabela 17. Nota-se que em todos os cenários, pelo menos uma das técnicas de transferência de aprendizagem superou a abordagem tradicional.

Tabela 17 – Resultados (em valores de EQM) obtidos pelos métodos de regressão com *kernel* polinomial para o conjunto de dados DHFR.

Conjunto de destino	n	SVR			SVR-ATL			TrAdaBoost		
		$C=0.1$	$C=10$	$C=100$	$C=0.1$	$C=10$	$C=100$	$C=0.1$	$C=10$	$C=100$
DHFR_S	25%	8,83E-01	2,03E+00	2,95E+00	7,67E-01	1,63E+00	2,26E+00	6,16E-01	6,28E-01	6,40E-01
	50%	5,93E-01	7,16E-01	7,08E-01	5,13E-01	6,55E-01	6,52E-01	4,84E-01	5,18E-01	5,30E-01
	75%	5,53E-01	6,29E-01	6,65E-01	5,35E-01	4,93E-01	4,72E-01	5,01E-01	5,56E-01	5,78E-01
	100%	4,77E-01	4,48E-01	4,42E-01	4,49E-01	4,41E-01	4,38E-01	4,83E-01	4,79E-01	4,82E-01

A Figura 24 apresenta uma análise gráfica para cada valor de custo considerado nos experimentos, a fim de ilustrar o impacto da transferência de aprendizagem em relação ao tamanho do conjunto de treinamento da tarefa de destino. Observa-se que o desempenho do método SVR melhorou à medida em que aumentou a quantidade de dados de treinamento, o que é esperado para uma técnica de aprendizagem de máquina tradicional. De tal forma, que os resultados se aproximaram nos experimentos com $n = 100\%$. Contudo, as abordagens de transferência de aprendizagem ainda continuaram obtendo melhores resultados, destaca-se

o método TrAdaBoost para os cenários com menos dados de treinamento e nos demais cenários, destaca-se o método SVR-ATL.

Figura 24 – Análise gráfica do desempenho dos métodos de regressão com *kernel* polinomial para o conjunto de dados DHFR.

Nos resultados experimentais analisados nesta seção, as abordagens de transferência de aprendizagem obtiveram melhores resultados do que a abordagem tradicional. De modo geral, os resultados indicam que as técnicas de transferência de aprendizagem podem ser utilizadas para melhorar o desempenho na tarefa de destino em experimentos de regressão considerando conjuntos de dados químicos referentes ao alvo DHFR.

5.5.5 Conjunto de dados COX2

Quando ocorre uma lesão na membrana celular, que é constituída fundamentalmente por fosfolípeos, a enzima fosfolipase A2, presente nos leucócitos e plaquetas, é ativada por citocinas pró-inflamatórias. Essa enzima leva à degradação dos fosfolípeos, resultando na produção de ácido araquidônico. Este, ao ser metabolizado, forma os leucotrienos, pela ação da enzima lipooxigenase, e as prostaglandinas, as prostaciclina e os tromboxanos,

pela ação da enzima ciclooxygenase (COX)¹² uma enzima responsável pela inflamação e dor (HILÁRIO; TERRERI; LEN, 2006).

Considerando esse cenário, um avanço importante na terapêutica anti-inflamatória foi a descoberta de duas isoformas da COX. A COX-1 que está presente em quase todos os tecidos (vasos sanguíneos, plaquetas, estômago, intestino, rins) e é, por isso, denominada de enzima constitutiva. Em contraste, a COX-2 está presente nos locais de inflamação, sendo, por isso, denominada de enzima indutiva. Esta é expressa primariamente por células envolvidas no processo inflamatório. Portanto, considera-se nos experimentos conduzidos aqui, os conjuntos de dados químicos para a enzima COX-2. A tarefa de regressão consiste em prever o nível de inibição dos compostos com o alvo, medido pelo valor IC_{50} .

O par de conjuntos de dados para o alvo COX-2 utilizado nos experimentos deste trabalho, também foi considerado nos experimentos de Girschick, Rückert e Kramer (2012). Considerou-se o conjunto COX2_S (SUTHERLAND; O'BRIEN; WEAVER, 2004) com 317 instâncias, como conjunto de destino; e o COX2_4q (SUTHERLAND; O'BRIEN; WEAVER, 2003) com 431 instâncias como conjunto de origem.

• Discussão dos resultados

Apresenta-se, na Tabela 18, os resultados obtidos pelos métodos SVR, SVR-ATL e TrAdaBoost nos experimentos com *kernel* linear considerando os conjuntos de teste da tarefa de destino (COX2_S). Nota-se que o método SVR-ATL obteve melhor desempenho para todos os cenários analisados.

Tabela 18 – Resultados (em valores de EQM) obtidos pelos métodos de regressão com *kernel* linear para o conjunto de dados COX-2.

Conjunto de destino	n	SVR			SVR-ATL			TrAdaBoost		
		$C=0.1$	$C=10$	$C=100$	$C=0.1$	$C=10$	$C=100$	$C=0.1$	$C=10$	$C=100$
COX2.S	25%	7,49E-01	7,35E-01	7,61E-01	6,57E-01	6,73E-01	6,79E-01	7,67E-01	7,77E-01	7,78E-01
	50%	1,05E+00	1,07E+00	1,07E+00	9,36E-01	9,29E-01	9,30E-01	1,08E+00	1,08E+00	1,08E+00
	75%	9,19E-01	9,19E-01	9,18E-01	8,86E-01	8,91E-01	8,91E-01	9,42E-01	9,43E-01	9,43E-01
	100%	9,39E-01	9,47E-01	9,49E-01	8,69E-01	8,84E-01	8,86E-01	9,49E-01	9,50E-01	9,50E-01

A Figura 25 apresenta uma análise gráfica para cada valor de custo considerado nos experimentos, a fim de ilustrar o impacto da transferência de aprendizagem em relação ao

¹² Do original, em inglês, *Cyclooxygenase-2*

tamanho do conjunto de treinamento da tarefa de destino. Destaca-se, o desempenho do método SVR-ATL.

Figura 25 – Análise gráfica do desempenho dos métodos de regressão com *kernel* linear para o conjunto de dados COX2.

A Tabela 19 apresenta os resultados obtidos pelos métodos de regressão com *kernel* polinomial. Nota-se que o método TrAdaBoost obteve melhor desempenho para todos os cenários considerados e que o método SVR-ATL também foi superior à abordagem tradicional em todos os experimentos com *kernel* polinomial.

Tabela 19 – Resultados (em valores de EQM) obtidos pelos métodos de regressão com *kernel* polinomial para o conjunto de dados COX-2.

Conjunto de destino	n	SVR			SVR-ATL			TrAdaBoost		
		$C=0.1$	$C=10$	$C=100$	$C=0.1$	$C=10$	$C=100$	$C=0.1$	$C=10$	$C=100$
COX2.S	25%	1,33E+00	4,57E+00	1,06E+01	9,24E-01	2,13E+00	3,41E+00	6,41E-01	6,47E-01	6,69E-01
	50%	9,98E-01	1,51E+00	1,68E+00	8,85E-01	1,25E+00	1,41E+00	7,99E-01	7,96E-01	8,08E-01
	75%	8,93E-01	1,04E+00	1,13E+00	7,88E-01	9,56E-01	9,89E-01	7,43E-01	7,44E-01	7,64E-01
	100%	8,97E-01	9,74E-01	9,80E-01	7,79E-01	8,52E-01	8,67E-01	7,42E-01	7,30E-01	7,65E-01

A Figura 26 apresenta uma análise gráfica para cada valor de custo considerado nos experimentos, a fim de ilustrar o impacto da transferência de aprendizagem em relação ao tamanho do conjunto de treinamento da tarefa de destino. Observa-se que o desempenho do método SVR melhorou à medida que aumentou a quantidade de dados de treinamento,

o que é esperado para uma técnica de aprendizado de máquina tradicional. Contudo, as abordagens de transferência de aprendizagem continuaram obtendo melhores resultados.

Figura 26 – Análise gráfica do desempenho dos métodos de regressão com *kernel* polinomial para o conjunto de dados COX2.

Nos resultados experimentais analisados nesta seção, as abordagens de transferência de aprendizagem obtiveram melhores resultados que a abordagem tradicional para todos os cenários analisados. Nos cenários analisados com *kernel* linear destaca-se o método SVR-ATL, nos cenários com *kernel* polinomial destaca-se o método TrAdaBoost. De modo geral, os resultados apresentados indicam que a transferência de aprendizagem foi benéfica para todos os cenários experimentais.

5.5.6 Conjunto de dados ER

O hormônio β -estradiol é o principal hormônio sexual feminino e é importante na regulação do ciclo menstrual, sendo essencial para o desenvolvimento e manutenção dos tecidos reprodutivos femininos. Os conjuntos de dados aqui analisados pertencem aos receptores do estrogênio (ER)¹³, um grupo de proteínas encontradas dentro das células que são ativadas pelo hormônio estrogênio (β -estradiol). A tarefa de regressão consiste em

¹³ Do inglês, *Estrogen receptors*.

medir a afinidade dos compostos químicos ao receptor estrogênico, por meio do valor de IC_{50} .

O par de conjuntos de dados para o alvo ER utilizado nos experimentos deste trabalho, também foi considerado nos experimentos de Girschick, Rückert e Kramer (2012). Considerou-se o conjunto ER_TOX (SUTHERLAND; O'BRIEN; WEAVER, 2003) com 323 instâncias, como conjunto de destino; e o ER_LIT (SUTHERLAND; O'BRIEN; WEAVER, 2003) 380 instâncias como conjunto de origem.

- Discussão dos resultados

Apresenta-se na Tabela 20, os resultados obtidos pelos métodos SVR, SVR-ATL e TrAdaBoost nos experimentos conduzidos com *kernel* linear considerando os conjuntos de teste da tarefa de destino (ER_TOX). Observa-se, que o método SVR-ATL obteve os melhores resultados na maioria dos cenários. Além disso, o método SVR atingiu resultados superiores em relação ao método TrAdaBoost na maioria dos cenários.

Tabela 20 – Resultados (em valores de EQM) obtidos pelos métodos de regressão com *kernel* linear para o conjunto de dados ER.

Conjunto de destino	n	SVR			SVR-ATL			TrAdaBoost		
		$C=0.1$	$C=10$	$C=100$	$C=0.1$	$C=10$	$C=100$	$C=0.1$	$C=10$	$C=100$
ER_TOX	25%	7,26E-01	8,07E-01	8,23E-01	7,09E-01	7,20E-01	7,330E-01	9,27E-01	8,84E-01	8,82E-01
	50%	9,10E-01	8,90E-01	8,56E-01	8,35E-01	1,27E+00	1,72E+00	1,01E+00	1,03E+00	1,04E+00
	75%	9,06E-01	9,18E-01	9,04E-01	8,56E-01	8,52E-01	8,63E-01	9,45E-01	8,67E-01	8,66E-01
	100%	7,94E-01	7,72E-01	7,63E-01	7,79E-01	7,57E-01	7,53E-01	8,66E-01	8,60E-01	8,60E-01

A Figura 27 apresenta uma análise gráfica dos resultados de cada método para cada valor do parâmetro custo considerado nos experimentos, a fim de ilustrar o impacto da transferência de aprendizagem em relação ao tamanho do conjunto de treinamento da tarefa de destino. Nota-se, que o método SVR-ATL obteve destaque nos experimentos com o parâmetro custo $C = 0.1$. Além disso, observa-se que os métodos não tiveram um ganho de desempenho significativo à medida em que aumentou a quantidade de dados disponíveis para treinamento na tarefa de destino.

Figura 27 – Análise gráfica do desempenho dos métodos de regressão com *kernel* linear para o conjunto de dados ER.

Os resultados obtidos para os métodos de regressão utilizando *kernel* polinomial são apresentados na Tabela 21. Nota-se que os desempenhos dos métodos de regressão com o *kernel* polinomial foram inferiores em relação aos resultados apresentados considerando o *kernel* linear. Além disso, observa-se que a configuração que obteve melhor desempenho para tais experimentos, foi obtida pelo método SVR-ATL com parâmetro custo $C = 0.1$.

Tabela 21 – Resultados (em valores de EQM) obtidos pelos métodos de regressão com *kernel* polinomial para o conjunto de dados ER.

Conjunto de destino	n	SVR			SVR-ATL			TrAdaBoost		
		$C=0.1$	$C=10$	$C=100$	$C=0.1$	$C=10$	$C=100$	$C=0.1$	$C=10$	$C=100$
ER.TOX	25%	2,29E+00	1,28E+01	4,08E+01	1,21E+00	6,28E+00	1,59E+01	1,25E+00	1,34E+00	1,21E+00
	50%	1,17E+00	2,71E+00	4,01E+00	1,12E+00	2,63E+00	4,32E+00	1,33E+00	2,55E+02	7,89E+02
	75%	1,50E+00	1,73E+00	2,17E+00	1,20E+00	2,11E+00	2,62E+00	1,04E+00	5,76E+00	9,43E+00
	100%	1,06E+00	1,12E+00	2,86E+00	9,21E-01	1,15E+00	1,47E+00	9,55E-01	4,07E+01	3,22E+01

A Figura 28 apresenta uma análise gráfica dos resultados de cada método para cada valor do parâmetro custo considerado nos experimentos, a fim de ilustrar o impacto da transferência de aprendizagem em relação ao tamanho do conjunto de treinamento da tarefa de destino.

Figura 28 – Análise gráfica do desempenho dos métodos de regressão com *kernel* polinomial para o conjunto de dados ER.

Observando a Figura 28, é possível constatar que o desempenho do método SVR melhorou à medida que aumentou a quantidade de dados de treinamento na tarefa de destino. Entretanto, o método SVR-ATL atingiu os melhores resultados no cenário com a quantidade total dos dados disponíveis para treinamento na tarefa de destino. Além disso, nota-se que o método TrAdaBoost apresentou desempenho insatisfatório em alguns dos experimentos realizados.

Para os resultados experimentais analisados nesta seção, os melhores resultados obtidos foram nos experimentos com *kernel* linear, destacando-se o método SVR-ATL, que obteve desempenho superior na maioria dos cenários analisados. De modo geral, os resultados apresentados indicam que a transferência de aprendizagem foi benéfica para os cenários analisados.

5.6 Considerações finais

Neste capítulo apresentou-se os resultados dos experimentos conduzidos com os métodos SVR, SVR-ATL e TrAdaBoost considerando conjuntos de *benchmark* e conjuntos de dados químicos gerados para estudos QSAR. As abordagens de transferência

de aprendizagem, na qual, utilizaram conhecimento a partir de um conjunto de dados relacionado obtiveram melhores resultados na maioria dos cenários analisados. Destaca-se, principalmente, os resultados obtidos em cenários com poucos dados de treinamento disponíveis para a tarefa de destino. Portanto, pelos resultados apresentados, as abordagens de transferência de aprendizagem melhoraram o processo de aprendizagem da tarefa de destino, sendo para tarefas específicas de regressão em dados de *benchmark* ou para a predição do nível de atividade de novos compostos químicos.

6 Conclusões

De modo geral, os dados manipulados pelos pesquisadores para análise QSAR são caracterizados por não possuir um nível de atividade biológica pré-definida (rótulos) para os compostos químicos (instâncias). Mesmo quando essa informação está presente, o tamanho da amostra, na maioria dos casos, é pequeno. Isso torna mais complexa a modelagem de fenômenos moleculares que constituem os processos químicos, como por exemplo, a construção de modelos preditivos. Nesse contexto, transferir conhecimento, aproveitando informações de outros conjuntos de dados químicos com mais informações disponíveis para o mesmo alvo biológico, torna-se desejável. Espera-se, assim, diminuir o esforço e o custo do processo para gerar novos modelos a partir de descritores de compostos químicos, utilizando o conhecimento de modelos relacionados já existentes.

No presente trabalho, foram aplicadas técnicas de transferência de aprendizagem com o objetivo de avaliar o impacto de tal abordagem no processo de aprendizagem da tarefa de destino. Para isso, analisou-se o *trade-off* entre o número de exemplos disponíveis na fase de treinamento na tarefa de destino e o desempenho obtido pelos métodos de transferência de aprendizagem e pelo método tradicional. Assim, foi possível analisar em quais situações o conhecimento gerado a partir de conjuntos de dados relacionados torna-se benéfico à tarefa de destino.

Em experimentos com conjuntos de dados de *benchmark*, considerou-se subconjuntos com tamanhos arbitrários variados e extraídos dos conjuntos de treinamento originais. E, para os experimentos com conjuntos de dados químicos, utilizou-se proporções variadas do conjunto original. O desempenho dos métodos de regressão foram avaliados pela medida de erro quadrático médio.

Nos resultados experimentais analisados aqui, as abordagens de transferência de aprendizagem obtiveram melhores resultados que a abordagem tradicional na maioria dos cenários analisados. Dessa forma, é possível concluir que as técnicas de transferência de aprendizagem podem ser utilizadas para melhorar o processo de aprendizagem na tarefa de destino em experimentos de regressão em que se pretende predizer o valor real do nível de atividade biológica de um composto químico em relação a um determinado alvo biológico, principalmente quando não existem dados suficientes para construção de modelos pelas técnicas de aprendizagem de máquina convencionais.

6.1 Contribuições do trabalho

O desenvolvimento do presente trabalho resultou, até o momento, na produção dos seguintes trabalhos:

- de Souza, J. C. S., Claudino, S. G., da Silva Simões, R., Oliveira, P. R., Honório, K. M. (2016). Recent advances for handling imbalance and uncertainty in labelling in medicinal chemistry data analysis. In SAI Computing Conference (SAI), 2016 (pp. 217-222), IEEE.

O trabalho discute os desafios encontrados atualmente para a análise de dados na área de QM. Mais especificamente, esses desafios estão relacionados a dificuldades encontradas no processo de classificação de dados de natureza química, ocasionadas pelo desbalanceamento de classes e incerteza na rotulação de dados.

- Simões, R., Oliveira, P. R., Lima C. A. D. M. (2017). Transferência de aprendizagem via método de regressão por vetores suporte. Anais do Encontro Nacional de Inteligência Artificial e Computacional (ENIAC 2017).

O trabalho apresenta o método SVR adaptado a transferência de aprendizagem, além de uma análise comparativa com o método SVR para dados de *benchmark*.

- Simões, R., Oliveira, P. R., Honorio, K. M., Lima C. A. D. M. (2018). Applying transfer learning to QSAR regression models.

Artigo aceito para publicação na 15th International Conference on Information Technology : New Generations¹ à ser publicado pelo Springer. O trabalho apresenta o método SVR adaptado a transferência de aprendizagem aplicado a construção de modelos de regressão QSAR e realiza um comparativo com o método SVR.

- Simões, R., Maltarollo, V. G., Oliveira, P. R., Honorio, K. M. (2018). Transfer and Multi-task Learning in QSAR Modeling: Advances and Challenges. *Frontiers in Pharmacology*, 9, 74.

O artigo apresenta uma revisão dos principais estudos relacionados à transferência de aprendizagem e aprendizagem multitarefa aplicados à construção de modelos QSAR. Além disso, o trabalho discute os avanços e desafios atuais mais relevantes nessa área de aplicação.

¹ Disponível em: <http://www.itng.info/>.

- Projeto FAPESP²: Aplicação de técnicas de transferência de aprendizagem e ontologias a modelos QSAR para regressão³.

A proposta apresentada no exame de qualificação do presente trabalho baseou um projeto de auxílio à pesquisa que foi submetido e aceito pela FAPESP na linha de fomento Acordos de Cooperação / IBM Brasil / IBM Brasil - PITE - Computação Cognitiva. Tal projeto está em vigência desde Agosto de 2017.

6.2 Trabalhos futuros

As propostas de continuidade deste trabalho derivam das limitações encontradas no decorrer desta pesquisa e são descritas a seguir:

- Conforme mencionado em (GIRSCHICK; RÜCKERT; KRAMER, 2012), um problema comum em dados de natureza química é a falta de metadados (por exemplo, dados de ontologia) para relacionar as variáveis em diferentes conjuntos de dados. Portanto, sugere-se incorporar ao processo de aprendizado uma metodologia que possa explorar os metadados existentes em ontologias para inferir a relação entre conjuntos de dados químicos e assim, aprimorar os experimentos com tais conjuntos de dados;
- Aponta-se como sugestão o desenvolvimento de uma heurística para determinar o número de instâncias ótimo para formar o novo conjunto de treinamento a partir da execução do método TrAdaBoost;
- Com o objetivo de aprofundar as análises comparativas realizadas neste trabalho, sugere-se a implementação das abordagens desenvolvidas por (GARCKE; VANCK, 2014) e (GIRSCHICK; RÜCKERT; KRAMER, 2012) e a execução de experimentos adicionais que permitam uma análise comparativa estatisticamente consistente.

² Sigla de Fundação de Amparo à Pesquisa do Estado de São Paulo.

³ Disponível em: (<http://www.bv.fapesp.br/pt/auxilios/96324/aplicacao-de-tecnicas-de-transferencia-de-aprendizagem>).

Referências⁴

- ALMEIDA, M. O. et al. In silico studies on the interaction between bioactive ligands and alk5, a biological target related to the cancer treatment. *Journal of Biomolecular Structure and Dynamics*, Taylor & Francis, p. 1–9, 2016. Citado na página 86.
- ARAÚJO, S. C. et al. Alk-5 inhibition: a molecular interpretation of the main physicochemical properties related to bioactive ligands. *Journal of the Brazilian Chemical Society*, SciELO Brasil, v. 26, n. 9, p. 1936–1946, 2015. Citado na página 86.
- ARLOT, S.; CELISSE, A. et al. A survey of cross-validation procedures for model selection. *Statistics surveys*, The author, under a Creative Commons Attribution License, v. 4, p. 40–79, 2010. Citado na página 60.
- BICKEL, S.; SAWADE, C.; SCHEFFER, T. Transfer learning by distribution matching for targeted advertising. In: *Advances in neural information processing systems*. [S.l.: s.n.], 2009. p. 145–152. Citado na página 33.
- BLITZER, J. et al. Biographies, bollywood, boom-boxes and blenders: Domain adaptation for sentiment classification. In: *Proc. 45th Ann. Meeting of the Assoc. Computacional Linguistics*. [S.l.]: ACL, 2007. v. 7, p. 440–447. Citado na página 43.
- BONILLA, E. V.; CHAI, K. M.; WILLIAMS, C. Multi-task gaussian process prediction. In: *Advances in neural information processing systems*. [S.l.: s.n.], 2007. p. 153–160. Citado na página 41.
- BUDD, D. C.; HOLMES, A. M. Targeting $\text{tgf}\beta$ superfamily ligand accessory proteins as novel therapeutics for chronic lung disorders. *Pharmacology & therapeutics*, Elsevier, v. 135, n. 3, p. 279–291, 2012. Citado na página 86.
- BURGES, C. J. A tutorial on support vector machines for pattern recognition. *Data mining and knowledge discovery*, Springer, v. 2, n. 2, p. 121–167, 1998. Citado na página 44.
- CARUANA, R. Multitask learning. *Machine learning*, Springer, v. 28, n. 1, p. 41–75, 1997. Citado na página 62.
- CHATTOPADHYAY, R. et al. Multisource domain adaptation and its application to early detection of fatigue. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, ACM, v. 6, n. 4, p. 18, 2012. Citado na página 64.
- CRISTIANINI, N.; SHAWE-TAYLOR, J. *An introduction to support vector machines*. [S.l.]: Cambridge University Press Cambridge, United Kingdom:, 2000. Citado 2 vezes nas páginas 30 e 45.
- DAI, W. et al. Co-clustering based classification for out-of-domain documents. In: *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*. [S.l.]: ACM, 2007. p. 210–219. Citado na página 42.

⁴ De acordo com a Associação Brasileira de Normas Técnicas. NBR 6023.

DAI, W. et al. Transferring naive bayes classifiers for text classification. In: *Proceedings of the national conference on artificial intelligence*. [S.l.]: Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2007. v. 22, n. 1, p. 540–545. Citado na página [42](#).

DAI, W. et al. Boosting for transfer learning. In: *Proceedings of the 24th international conference on Machine learning*. [S.l.]: ACM, 2007. p. 193–200. Citado 5 vezes nas páginas [38](#), [39](#), [40](#), [53](#) e [54](#).

DAVIS, J.; DOMINGOS, P. Deep transfer via second-order markov logic. In: *Proceedings of the 26th annual international conference on machine learning*. [S.l.]: ACM, 2009. p. 217–224. Citado na página [38](#).

DEVILLERS, J.; BALABAN, A. T. *Topological indices and related descriptors in QSAR and QSPAR*. [S.l.]: CRC Press, 2000. Citado na página [29](#).

DIETTERICH, T. G.; PRATT, L.; THRUN, S. Special issue on inductive transfer. *Machine Learning*, Kluwer Academic Publishers, v. 28, n. 1, 1997. Citado na página [62](#).

DRUCKER, H. Improving regressors using boosting techniques. In: *ICML*. [S.l.: s.n.], 1997. v. 97, p. 107–115. Citado 2 vezes nas páginas [52](#) e [54](#).

EVGENIOU, A.; PONTIL, M. Multi-task feature learning. *Advances in neural information processing systems*, v. 19, p. 41–48, 2007. Citado na página [40](#).

EVGENIOU, T.; PONTIL, M. Regularized multi-task learning. In: *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*. [S.l.]: ACM, 2004. p. 109–117. Citado 4 vezes nas páginas [41](#), [49](#), [67](#) e [68](#).

FEUZ, K. D.; COOK, D. J. Transfer learning across feature-rich heterogeneous feature spaces via feature-space remapping (fsr). *ACM Transactions on Intelligent Systems and Technology (TIST)*, ACM, v. 6, n. 1, p. 3, 2015. Citado na página [65](#).

FITZGERALD, T.; THOMAZ, A. L. Skill demonstration transfer for learning from demonstration. In: *Proceedings of the Tenth Annual ACM/IEEE International Conference on Human-Robot Interaction Extended Abstracts*. [S.l.]: ACM, 2015. p. 187–188. Citado 4 vezes nas páginas [30](#), [33](#), [58](#) e [62](#).

FREUND, Y.; SCHAPIRE, R. E. A decision-theoretic generalization of on-line learning and an application to boosting. *Journal of Computer and System Sciences*, v. 55, n. 1-2, p. 119–139, 1997. Citado 2 vezes nas páginas [52](#) e [53](#).

FREUND, Y.; SCHAPIRE, R. E. et al. Experiments with a new boosting algorithm. In: *Icml*. [S.l.: s.n.], 1996. v. 96, p. 148–156. Citado na página [51](#).

GARCKE, J.; VANCK, T. Importance weighted inductive transfer learning for regression. In: SPRINGER. *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. [S.l.], 2014. p. 466–481. Citado 6 vezes nas páginas [71](#), [74](#), [75](#), [78](#), [82](#) e [107](#).

GE, L.; GAO, J.; ZHANG, A. Oms-tl: A framework of online multiple source transfer learning. In: *Proceedings of the 22nd ACM international conference on Conference on information & knowledge management*. [S.l.]: ACM, 2013. p. 2423–2428. Citado na página [63](#).

GERTRUDES, J. et al. Machine learning techniques and drug design. *Current medicinal chemistry*, Bentham Science Publishers, v. 19, n. 25, p. 4289–4297, 2012. Citado na página 85.

GERTRUDES, J. C. *Emprego de técnicas de análise exploratória de dados utilizados em Química Medicinal*. Tese (Doutorado) — Universidade de São Paulo, 2013. Citado na página 89.

GIRSCHICK, T.; RÜCKERT, U.; KRAMER, S. Adapted transfer of distance measures for quantitative structure-activity relationships and data-driven selection of source datasets. *The Computer Journal*, Br Computer Soc, p. bxs092, 2012. Citado 10 vezes nas páginas 29, 31, 58, 66, 71, 85, 95, 98, 101 e 107.

GONG, J. J. et al. Instance weighting for patient-specific risk stratification models. In: *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. [S.l.]: ACM, 2015. p. 369–378. Citado na página 64.

GUNN, S. R. et al. Support vector machines for classification and regression. *ISIS technical report*, v. 14, 1998. Citado na página 44.

HE, J.; LIU, Y.; LAWRENCE, R. Graph-based transfer learning. In: *Proceedings of the 18th ACM conference on Information and knowledge management*. [S.l.]: ACM, 2009. p. 937–946. Citado na página 64.

HILÁRIO, M. O. E.; TERRERI, M. T. R. A.; LEN, C. A. Antiinflamatórios não-hormonais: inibidores da ciclooxigenase 2. *Jornal de Pediatria*, Sociedade Brasileira de Pediatria, 2006. Citado na página 98.

HO, C.-H.; LIN, C.-J. Large-scale linear support vector regression. *The Journal of Machine Learning Research*, JMLR. org, v. 13, n. 1, p. 3323–3348, 2012. Citado na página 47.

JIANG, J.; ZHAI, C. Instance weighting for domain adaptation in nlp. In: *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*. [S.l.]: ACL, 2007. p. 264–271. Citado na página 40.

JORDE, L. B. *Genética médica*. [S.l.]: Elsevier Brasil, 2004. Citado na página 85.

LAWRENCE, N. D.; PLATT, J. C. Learning to learn with the informative vector machine. In: *Proceedings of the twenty-first international conference on Machine learning*. [S.l.]: ACM, 2004. p. 65. Citado 2 vezes nas páginas 38 e 41.

LIAO, X.; XUE, Y.; CARIN, L. Logistic regression with an auxiliary data source. In: *Proceedings of the 22nd international conference on Machine learning*. [S.l.]: ACM, 2005. p. 505–512. Citado na página 40.

LIMA, C. A. d. M. *Comitê de Máquinas: uma abordagem unificada empregando máquinas de vetores-suporte*. Tese (Doutorado) — Universidade Estadual de Campinas, 2004. Citado 3 vezes nas páginas 44, 45 e 46.

LING, X. et al. Spectral domain-transfer learning. In: *Proceedings of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining*. [S.l.]: ACM, 2008. p. 488–496. Citado na página 63.

- LING, X. et al. Can chinese web pages be classified with english data source? In: *Proceedings of the 17th international conference on World Wide Web*. [S.l.]: ACM, 2008. p. 969–978. Citado na página [43](#).
- LUO, P. et al. Transfer learning from multiple source domains via consensus regularization. In: *Proceedings of the 17th ACM conference on Information and knowledge management*. [S.l.]: ACM, 2008. p. 103–112. Citado na página [63](#).
- MAHONEY, M. W.; DRINEAS, P. Cur matrix decompositions for improved data analysis. *Proceedings of the National Academy of Sciences*, National Acad Sciences, v. 106, n. 3, p. 697–702, 2009. Citado na página [67](#).
- MALTAROLLO, V. G.; HONORIO, K. M. Ligand-and structure-based drug design strategies and ppar δ / α selectivity. *Chemical biology & drug design*, Wiley Online Library, v. 80, n. 4, p. 533–544, 2012. Citado 2 vezes nas páginas [89](#) e [92](#).
- MENDES, C. C. et al. 19-base pair deletion polymorphism of the dihydrofolate reductase (dhfr) gene: maternal risk of down syndrome and folate metabolism. *Sao Paulo Medical Journal*, SciELO Brasil, v. 128, n. 4, p. 215–218, 2010. Citado na página [95](#).
- MICHIELAN, L.; MORO, S. Pharmaceutical perspectives of nonlinear qsar strategies. *Journal of chemical information and modeling*, ACS Publications, v. 50, n. 6, p. 961–978, 2010. Citado na página [30](#).
- MIHALKOVA, L.; HUYNH, T.; MOONEY, R. J. Mapping and revising markov logic networks for transfer learning. In: *Proc. 22nd Assoc. for the advancement of Artificial Intelligence (AAAI) Conf. Artificial Intelligence*. [S.l.]: AAAI, 2007. p. 608–614. Citado 2 vezes nas páginas [38](#) e [42](#).
- MITCHELL, T. M. *Machine learning*. [S.l.]: McGraw-Hill, 1997. Citado na página [30](#).
- NG, M. K.; WU, Q.; YE, Y. Co-transfer learning via joint transition probability graph based method. In: *Proceedings of the 1st International Workshop on Cross Domain Knowledge Discovery in Web and Social Network Mining*. [S.l.]: ACM, 2012. p. 1–9. Citado na página [65](#).
- NING, X.; KARYPIS, G. In silico structure-activity-relationship (sar) models from machine learning: a review. *Drug Development Research*, Wiley Online Library, v. 72, n. 2, p. 138–146, 2011. Citado 2 vezes nas páginas [30](#) e [57](#).
- PAN, S. J. et al. Adaptive localization in a dynamic wifi environment through multi-view learning. In: *Proceedings of the national conference on artificial Intelligence*. [S.l.]: Menlo Park, CA; Cambridge, MA; London; AAAI Press; MIT Press; 1999, 2007. v. 22, n. 2, p. 1108. Citado na página [43](#).
- PAN, S. J.; YANG, Q. A survey on transfer learning. *IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING*, IEEE Computer Society, v. 22, n. 10, p. 1345–1359, 2010. Citado 15 vezes nas páginas [8](#), [16](#), [28](#), [29](#), [30](#), [31](#), [33](#), [34](#), [36](#), [37](#), [39](#), [40](#), [42](#), [57](#) e [62](#).
- PARDOE, D.; STONE, P. Boosting for regression transfer. In: *Proceedings of the 27th international conference on Machine learning (ICML-10)*. [S.l.: s.n.], 2010. p. 863–870. Citado 5 vezes nas páginas [32](#), [53](#), [54](#), [55](#) e [59](#).

- PEREIRA, M. A. et al. Study of the structure-activity relationship for theoretical molecular descriptors using density functional theory and chemometric methods in cannabinoid metabolites. *International Journal of Quantum Chemistry*, Wiley Online Library, v. 108, n. 13, p. 2530–2539, 2008. Citado na página 56.
- PRATT, L. Y. et al. Discriminability-based transfer between neural networks. In: *Advances in Neural Information Processing Systems 5*. [S.l.]: Citeseer, 1993. Citado na página 62.
- QUANZ, B.; HUAN, J. Large margin transductive transfer learning. In: *Proceedings of the 18th ACM conference on Information and knowledge management*. [S.l.]: ACM, 2009. p. 1327–1336. Citado na página 63.
- RAINA, R. et al. Self-taught learning: transfer learning from unlabeled data. In: *Proceedings of the 24th international conference on Machine learning*. [S.l.]: ACM, 2007. p. 759–766. Citado 2 vezes nas páginas 38 e 40.
- RAMACHANDRAN, S.; MOONEY, R. J. *Theory refinement of bayesian networks with hidden variables*. Tese (Doutorado) — Citeseer, 1998. Citado na página 42.
- RAYMOND, J. W.; WILLETT, P. Effectiveness of graph-based and fingerprint-based similarity measures for virtual screening of 2d chemical structure databases. *Journal of computer-aided molecular design*, Springer, v. 16, n. 1, p. 59–71, 2002. Citado na página 66.
- RICHARDSON, M.; DOMINGOS, P. Markov logic networks. *Machine learning*, Springer, v. 62, n. 1-2, p. 107–136, 2006. Citado na página 42.
- ROSENBAUM, L. et al. Inferring multi-target qsar models with taxonomy-based multi-task learning. *J. Cheminformatics*, Journal of Cheminformatics, v. 5, n. 1-2, p. 33, 2013. Citado 8 vezes nas páginas 8, 29, 30, 48, 49, 50, 68 e 85.
- SATPAL, S.; SARAWAGI, S. Domain adaptation of conditional probability models via feature subsetting. In: *Knowledge Discovery in Databases: PKDD 2007*. [S.l.]: Springer, 2007. p. 224–235. Citado na página 33.
- SCHWAIGHOFER, A.; TRESP, V.; YU, K. Learning gaussian process kernels via hierarchical bayes. In: *Advances in Neural Information Processing Systems*. [S.l.: s.n.], 2004. p. 1209–1216. Citado na página 41.
- SMOLA, A. J. et al. Regression estimation with support vector learning machines. *Master's thesis, Technische Universit at M unchen*, 1996. Citado na página 45.
- SOUZA, J. C. S. *Aprendizado de dados positivos e não rotulados para o tratamento de incerteza na rotulação de dados de química medicinal*. Tese (Doutorado) — USP, 2015. Citado na página 57.
- SUTHERLAND, J. J.; O'BRIEN, L. A.; WEAVER, D. F. Spline-fitting with a genetic algorithm: A method for developing classification structure-activity relationships. *Journal of chemical information and computer sciences*, ACS Publications, v. 43, n. 6, p. 1906–1915, 2003. Citado 3 vezes nas páginas 95, 98 e 101.

SUTHERLAND, J. J.; O'BRIEN, L. A.; WEAVER, D. F. A comparison of methods for modeling quantitative structure-activity relationships. *Journal of medicinal chemistry*, ACS Publications, v. 47, n. 22, p. 5541–5554, 2004. Citado 2 vezes nas páginas 95 e 98.

TAN, B. et al. Transitive transfer learning. In: *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. [S.l.]: ACM, 2015. p. 1155–1164. Citado 2 vezes nas páginas 33 e 58.

TURKI, T.; WEI, Z.; WANG, J. T. Transfer learning approaches to improve drug sensitivity prediction in multiple myeloma patients. *IEEE Access*, IEEE, 2017. Citado na página 67.

WANG, S.; LI, Z. A new transfer learning boosting approach based on distribution measure with an application on facial expression recognition. In: *Neural Networks (IJCNN), 2014 International Joint Conference on*. [S.l.]: IEEE, 2014. p. 432–439. Citado 2 vezes nas páginas 28 e 58.

WIDMER, C. et al. Efficient training of graph-regularized multitask svms. In: *Machine Learning and Knowledge Discovery in Databases*. [S.l.]: Springer, 2012. p. 633–647. Citado na página 68.

YAO, Y.; DORETTO, G. Boosting for transfer learning with multiple sources. In: *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*. [S.l.]: IEEE, 2010. p. 1855–1862. Citado na página 30.

YIN, J.; YANG, Q.; NI, L. Adaptive temporal radio maps for indoor location estimation. In: *Pervasive Computing and Communications, 2005. PerCom 2005. Third IEEE International Conference on*. [S.l.]: IEEE, 2005. p. 85–94. Citado na página 43.

YING, J. J.-C. et al. Transfer learning on high variety domains for activity recognition. In: *Proceedings of the ASE BigData & SocialInformatics 2015*. [S.l.]: ACM, 2015. p. 37. Citado na página 65.

YOUNG, D. C. *Computacional Drug Design - A Guide for Computational and Medicinal Chemists*. [S.l.]: John Wiley & Sons, Inc, 2009. Citado na página 29.

ZHANG, D. et al. Multi-view transfer learning with a large margin approach. In: *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. [S.l.]: ACM, 2011. p. 1208–1216. Citado na página 65.

ZHENG, V. W. et al. Transferring localization models over time. In: *Proc. 23rd Assoc. for the Advancement of Artificial Intelligence (AAAI) Conf. Artificial Intelligence*. [S.l.: s.n.], 2008. p. 1421–1426. Citado na página 43.

Apêndice A – Problema de otimização SVR-ATL

Detalha-se aqui, os cálculos realizados para a obtenção do problema de otimização dual do método SVR-ATL (15). Primeiramente, é apresentado o problema de otimização primal do método SVR:

$$\min_{\mathbf{w}, \xi^*, \xi} = \frac{1}{2}(\mathbf{w}^T \mathbf{w}) + C \left(\sum_{i=1}^l \xi_i + \sum_{i=1}^l \xi_i^* \right). \quad (20)$$

Adequa-se a Equação (20) acima ao problema de transferência de aprendizagem, em que, considera-se duas tarefas: origem (s) e destino (t). A função primal de regressão é dada pelo funcional:

$$\begin{aligned} \min_{\mathbf{w}_s, \mathbf{w}_t, \xi_s^*, \xi_t^*, \xi_s, \xi_t} &= \frac{1}{2}(\mathbf{w}_s^T \mathbf{w}_s) + \frac{1}{2}(\mathbf{w}_t^T \mathbf{w}_t) + \frac{1}{4}(\mathbf{w}_s^T - \mathbf{w}_t^T)(\mathbf{w}_s - \mathbf{w}_t) + \frac{c}{2} \left(\sum_{i=1}^l (\xi_{si})^2 + \sum_{i=1}^l (\xi_{si}^*)^2 \right) + \\ &\frac{c}{2} \left(\sum_{i=1}^l (\xi_{ti})^2 + \sum_{i=1}^l (\xi_{ti}^*)^2 \right) + \sum_{i=1}^l \alpha_{si} (y_{si} - \mathbf{w}_s^T \mathbf{x}_{si} - \epsilon_s - \xi_{si}) + \sum_{i=1}^l \alpha_{ti} (y_{ti} - \mathbf{w}_t^T \mathbf{x}_{ti} - \epsilon_t - \xi_{ti}) + \\ &\sum_{i=1}^l \alpha_{si}^* (\mathbf{w}_s^T \mathbf{x}_{si} - y_{si} - \epsilon_s - \xi_{si}^*) + \sum_{i=1}^l \alpha_{ti}^* (\mathbf{w}_t^T \mathbf{x}_{ti} - y_{ti} - \epsilon_t - \xi_{ti}^*), \end{aligned} \quad (21)$$

realizando a derivada em função de: $\mathbf{w}_s, \mathbf{w}_t, \xi_{si}^*, \xi_{ti}^*, \xi_{si}, \xi_{ti}$ temos:

$$\begin{aligned} \frac{\partial \delta}{\partial \mathbf{w}_s} &= \frac{3}{4} \sum_{i=1}^l (\alpha_{si} - \alpha_{si}^*) \mathbf{x}_{si} + \frac{1}{4} \sum_{i=1}^l (\alpha_{ti} - \alpha_{ti}^*) \mathbf{x}_{ti}, & \frac{\partial \delta}{\partial \mathbf{w}_t} &= \frac{3}{4} \sum_{i=1}^l (\alpha_{ti} - \alpha_{ti}^*) \mathbf{x}_{ti} + \frac{1}{4} \sum_{i=1}^l (\alpha_{si} - \alpha_{si}^*) \mathbf{x}_{si}, \\ \frac{\partial \delta}{\partial \xi_{si}^*} &= \frac{\alpha_{si}^*}{C}, & \frac{\partial \delta}{\partial \xi_{ti}^*} &= \frac{\alpha_{ti}^*}{C}, & \frac{\partial \delta}{\partial \xi_{si}} &= \frac{\alpha_{si}}{C} & \text{e} & \frac{\partial \delta}{\partial \xi_{ti}} = \frac{\alpha_{ti}}{C}. \end{aligned} \quad (22)$$

Para a obtenção do problema dual a partir do primal do método SVR-ATL (14). Considere:

- $\beta_s = (\alpha_{si} - \alpha_{si}^*)$;
- $\beta_t = (\alpha_{ti} - \alpha_{ti}^*)$;
- $|\beta_s| = |(\alpha_{si} - \alpha_{si}^*)| = (\alpha_{si} + \alpha_{si}^*)$;
- $|\beta_t| = |(\alpha_{ti} - \alpha_{ti}^*)| = (\alpha_{ti} + \alpha_{ti}^*)$.

$$\begin{aligned} \text{Seja } (I) &= \frac{1}{2}(\mathbf{w}_s^T \mathbf{w}_s) + \frac{1}{2}(\mathbf{w}_t^T \mathbf{w}_t) + \frac{1}{4}(\mathbf{w}_s^T - \mathbf{w}_t^T)(\mathbf{w}_s - \mathbf{w}_t) = \\ &\frac{1}{2}(\mathbf{w}_s^T \mathbf{w}_s) + \frac{1}{2}(\mathbf{w}_t^T \mathbf{w}_t) + \frac{1}{4}(\mathbf{w}_s^T \mathbf{w}_s) - \frac{1}{4}(\mathbf{w}_s^T \mathbf{w}_t) - \frac{1}{4}(\mathbf{w}_t^T \mathbf{w}_s) + \frac{1}{4}(\mathbf{w}_t^T \mathbf{w}_t) = \\ &\frac{3}{4}(\mathbf{w}_s^T \mathbf{w}_s) + \frac{3}{4}(\mathbf{w}_t^T \mathbf{w}_t) - \frac{1}{2}(\mathbf{w}_s^T \mathbf{w}_t). \end{aligned}$$

Substituindo os valores de $\mathbf{w}_s$ e $\mathbf{w}_t$ em (I):

$$\begin{aligned} & \frac{3}{4} \left(\frac{3}{4} \sum_{i=1}^l \beta_{si} \mathbf{x}_{si} + \frac{1}{4} \sum_{i=1}^l \beta_{ti} \mathbf{x}_{ti} \right)^T \left(\frac{3}{4} \sum_{j=1}^l \beta_{sj} \mathbf{x}_{sj} + \frac{1}{4} \sum_{j=1}^l \beta_{tj} \mathbf{x}_{tj} \right) + \\ & \frac{3}{4} \left(\frac{3}{4} \sum_{i=1}^l \beta_{ti} \mathbf{x}_{ti} + \frac{1}{4} \sum_{i=1}^l \beta_{si} \mathbf{x}_{si} \right)^T \left(\frac{3}{4} \sum_{j=1}^l \beta_{tj} \mathbf{x}_{tj} + \frac{1}{4} \sum_{j=1}^l \beta_{sj} \mathbf{x}_{sj} \right) - \\ & \frac{1}{2} \left(\frac{3}{4} \sum_{i=1}^l \beta_{si} \mathbf{x}_{si} + \frac{1}{4} \sum_{i=1}^l \beta_{ti} \mathbf{x}_{ti} \right)^T \left(\frac{3}{4} \sum_{j=1}^l \beta_{tj} \mathbf{x}_{tj} + \frac{1}{4} \sum_{j=1}^l \beta_{sj} \mathbf{x}_{sj} \right), \end{aligned}$$

realizando as operações de multiplicação obtemos:

$$\begin{aligned} & \frac{3}{4} \left(\frac{9}{16} \sum_{i=1}^l \sum_{j=1}^l \beta_{si} \beta_{sj} \mathbf{x}_{si}^T \mathbf{x}_{sj} + \frac{6}{16} \sum_{i=1}^l \sum_{j=1}^l \beta_{si} \beta_{tj} \mathbf{x}_{si}^T \mathbf{x}_{tj} + \frac{1}{16} \sum_{i=1}^l \sum_{j=1}^l \beta_{ti} \beta_{tj} \mathbf{x}_{ti}^T \mathbf{x}_{tj} \right) + \\ & \frac{3}{4} \left(\frac{9}{16} \sum_{i=1}^l \sum_{j=1}^l \beta_{ti} \beta_{tj} \mathbf{x}_{ti}^T \mathbf{x}_{tj} + \frac{6}{16} \sum_{i=1}^l \sum_{j=1}^l \beta_{ti} \beta_{sj} \mathbf{x}_{ti}^T \mathbf{x}_{sj} + \frac{1}{16} \sum_{i=1}^l \sum_{j=1}^l \beta_{si} \beta_{sj} \mathbf{x}_{si}^T \mathbf{x}_{sj} \right) - \\ & \frac{1}{2} \left(\frac{10}{16} \sum_{i=1}^l \sum_{j=1}^l \beta_{si} \beta_{tj} \mathbf{x}_{si}^T \mathbf{x}_{tj} + \frac{3}{16} \sum_{i=1}^l \sum_{j=1}^l \beta_{si} \beta_{sj} \mathbf{x}_{si}^T \mathbf{x}_{sj} + \frac{3}{16} \sum_{i=1}^l \sum_{j=1}^l \beta_{ti} \beta_{tj} \mathbf{x}_{ti}^T \mathbf{x}_{tj} \right), \end{aligned}$$

realizando novamente as operações de multiplicação obtemos:

$$\begin{aligned} & \frac{27}{64} \sum_{i=1}^l \sum_{j=1}^l \beta_{si} \beta_{sj} \mathbf{x}_{si}^T \mathbf{x}_{sj} + \frac{18}{64} \sum_{i=1}^l \sum_{j=1}^l \beta_{si} \beta_{tj} \mathbf{x}_{si}^T \mathbf{x}_{tj} + \frac{3}{64} \sum_{i=1}^l \sum_{j=1}^l \beta_{ti} \beta_{tj} \mathbf{x}_{ti}^T \mathbf{x}_{tj} + \\ & \frac{27}{64} \sum_{i=1}^l \sum_{j=1}^l \beta_{ti} \beta_{tj} \mathbf{x}_{ti}^T \mathbf{x}_{tj} + \frac{18}{64} \sum_{i=1}^l \sum_{j=1}^l \beta_{ti} \beta_{sj} \mathbf{x}_{ti}^T \mathbf{x}_{sj} + \frac{3}{64} \sum_{i=1}^l \sum_{j=1}^l \beta_{si} \beta_{sj} \mathbf{x}_{si}^T \mathbf{x}_{sj} - \\ & \frac{20}{64} \sum_{i=1}^l \sum_{j=1}^l \beta_{si} \beta_{tj} \mathbf{x}_{si}^T \mathbf{x}_{tj} - \frac{6}{64} \sum_{i=1}^l \sum_{j=1}^l \beta_{si} \beta_{sj} \mathbf{x}_{si}^T \mathbf{x}_{sj} - \frac{6}{64} \sum_{i=1}^l \sum_{j=1}^l \beta_{ti} \beta_{tj} \mathbf{x}_{ti}^T \mathbf{x}_{tj}, \end{aligned}$$

realizando operações aritméticas e simplificação de valores obtemos:

$$\frac{3}{8} \sum_{i=1}^l \sum_{j=1}^l \beta_{si} \beta_{sj} \mathbf{x}_{si}^T \mathbf{x}_{sj} + \frac{1}{4} \sum_{i=1}^l \sum_{j=1}^l \beta_{si} \beta_{tj} \mathbf{x}_{si}^T \mathbf{x}_{tj} + \frac{3}{8} \sum_{i=1}^l \sum_{j=1}^l \beta_{ti} \beta_{tj} \mathbf{x}_{ti}^T \mathbf{x}_{tj}.$$

Seja (II) = $\frac{c}{2} \left(\sum_{i=1}^l (\xi_{si})^2 + \sum_{i=1}^l (\xi_{si}^*)^2 \right) + \frac{c}{2} \left(\sum_{i=1}^l (\xi_{ti})^2 + \sum_{i=1}^l (\xi_{ti}^*)^2 \right) + \sum_{i=1}^l \alpha_{si} (y_{si} - \mathbf{w}_s^T \mathbf{x}_{si} - \epsilon_s - \xi_{si}) +$
 $\sum_{i=1}^l \alpha_{ti} (y_{ti} - \mathbf{w}_t^T \mathbf{x}_{ti} - \epsilon_t - \xi_{ti}) + \sum_{i=1}^l \alpha_{si}^* (\mathbf{w}_s^T \mathbf{x}_{si} - y_{si} - \epsilon_s - \xi_{si}^*) + \sum_{i=1}^l \alpha_{ti}^* (\mathbf{w}_t^T \mathbf{x}_{ti} - y_{ti} - \epsilon_t - \xi_{ti}^*),$

realizando operações aritméticas e substituindo os valores de: $\xi_{si}^*, \xi_{ti}^*, \xi_{si}, \xi_{ti}$ obtemos:

$$\begin{aligned} & \underbrace{\sum_{i=1}^l \frac{(\alpha_{si})^2}{2C}} + \underbrace{\sum_{i=1}^l \frac{(\alpha_{si}^*)^2}{2C}} + \underbrace{\sum_{i=1}^l \frac{(\alpha_{ti})^2}{2C}} + \underbrace{\sum_{i=1}^l \frac{(\alpha_{ti}^*)^2}{2C}} + \sum_{i=1}^l \alpha_{si} y_{si} - \sum_{i=1}^l \alpha_{si} \mathbf{w}_s^T \mathbf{x}_{si} - \sum_{i=1}^l \alpha_{si} \epsilon_s - \\ & \underbrace{\sum_{i=1}^l \alpha_{si} \frac{\alpha_{si}}{C}} + \sum_{i=1}^l \alpha_{ti} y_{ti} - \sum_{i=1}^l \alpha_{ti} \mathbf{w}_t^T \mathbf{x}_{ti} - \sum_{i=1}^l \alpha_{ti} \epsilon_t - \underbrace{\sum_{i=1}^l \alpha_{ti} \frac{\alpha_{ti}}{C}} + \sum_{i=1}^l \alpha_{si}^* \mathbf{w}_s^T \mathbf{x}_{si} - \sum_{i=1}^l \alpha_{si}^* y_{si} - \\ & \sum_{i=1}^l \alpha_{si}^* \epsilon_s - \underbrace{\sum_{i=1}^l \alpha_{si}^* \frac{\alpha_{si}^*}{C}} + \sum_{i=1}^l \alpha_{ti}^* \mathbf{w}_t^T \mathbf{x}_{ti} - \sum_{i=1}^l \alpha_{ti}^* y_{ti} - \sum_{i=1}^l \alpha_{ti}^* \epsilon_t - \underbrace{\sum_{i=1}^l \alpha_{ti}^* \frac{\alpha_{ti}^*}{C}}, \end{aligned}$$

realizando as operações aritméticas indicadas acima, obtemos:

$$\begin{aligned}
& - \sum_{i=1}^l \frac{(\alpha_{si})^2}{2C} - \sum_{i=1}^l \frac{(\alpha_{si}^*)^2}{2C} - \sum_{i=1}^l \frac{(\alpha_{ti})^2}{2C} - \sum_{i=1}^l \frac{(\alpha_{ti}^*)^2}{2C} + \sum_{i=1}^l \alpha_{si} y_{si} - \sum_{i=1}^l \alpha_{si} \mathbf{w}_s^T \mathbf{x}_{si} - \sum_{i=1}^l \alpha_{si} \epsilon_s + \\
& \sum_{i=1}^l \alpha_{ti} y_{ti} - \sum_{i=1}^l \alpha_{ti} \mathbf{w}_t^T \mathbf{x}_{ti} - \sum_{i=1}^l \alpha_{ti} \epsilon_t + \sum_{i=1}^l \alpha_{si}^* \mathbf{w}_s^T \mathbf{x}_{si} - \sum_{i=1}^l \alpha_{si}^* y_{si} - \sum_{i=1}^l \alpha_{si}^* \epsilon_s + \sum_{i=1}^l \alpha_{ti}^* \mathbf{w}_t^T \mathbf{x}_{ti} - \\
& \sum_{i=1}^l \alpha_{ti}^* y_{ti} - \sum_{i=1}^l \alpha_{ti}^* \epsilon_t,
\end{aligned}$$

substituindo os valores de: $(\alpha_{si} - \alpha_{si}^*)$ por β_s e $(\alpha_{ti} - \alpha_{ti}^*)$ por β_t temos:

$$\begin{aligned}
& - \sum_{i=1}^l \frac{(\alpha_{si})^2}{2C} - \sum_{i=1}^l \frac{(\alpha_{si}^*)^2}{2C} - \sum_{i=1}^l \frac{(\alpha_{ti})^2}{2C} - \sum_{i=1}^l \frac{(\alpha_{ti}^*)^2}{2C} + \sum_{i=1}^l \beta_{si} y_{si} - \sum_{i=1}^l \beta_{si} \mathbf{w}_s^T \mathbf{x}_{si} - \sum_{i=1}^l |\beta_{si}| \epsilon_s + \\
& \sum_{i=1}^l \beta_{ti} y_{ti} - \sum_{i=1}^l \beta_{ti} \mathbf{w}_t^T \mathbf{x}_{ti} - \sum_{i=1}^l |\beta_{ti}| \epsilon_t.
\end{aligned}$$

Sendo $\beta_{si} = (\alpha_{si} - \alpha_{si}^*)$ então $(\beta_{si})^2 = (\alpha_{si} - \alpha_{si}^*)(\alpha_{si} - \alpha_{si}^*) = (\alpha_{si})^2 - 2\alpha_{si}\alpha_{si}^* + (\alpha_{si}^*)^2$.

Como $2\alpha_{si}\alpha_{si}^* = 0$ então $(\beta_{si})^2 = (\alpha_{si})^2 + (\alpha_{si}^*)^2$.

Assim,

$$- \sum_{i=1}^l \frac{(\alpha_{si})^2}{2C} - \sum_{i=1}^l \frac{(\alpha_{si}^*)^2}{2C} - \sum_{i=1}^l \frac{(\alpha_{ti})^2}{2C} - \sum_{i=1}^l \frac{(\alpha_{ti}^*)^2}{2C} = - \sum_{i=1}^l \frac{(\beta_{si})^2}{2C} - \sum_{i=1}^l \frac{(\beta_{ti})^2}{2C}.$$

Abrindo $-\sum_{i=1}^l \beta_{si} \mathbf{w}_s^T \mathbf{x}_{si}$ e $\sum_{i=1}^l \beta_{ti} \mathbf{w}_t^T \mathbf{x}_{ti}$ temos:

$$\begin{aligned}
& = - \sum_{i=1}^l \beta_{si} \mathbf{x}_{si} \left(\frac{3}{4} \sum_{j=1}^l \beta_{sj} \mathbf{x}_{sj} + \frac{1}{4} \sum_{j=1}^l \beta_{tj} \mathbf{x}_{tj} \right) - \sum_{i=1}^l \beta_{ti} \mathbf{x}_{ti} \left(\frac{3}{4} \sum_{j=1}^l \beta_{tj} \mathbf{x}_{tj} + \frac{1}{4} \sum_{j=1}^l \beta_{sj} \mathbf{x}_{sj} \right) = \\
& - \frac{3}{4} \sum_{i=1}^l \sum_{j=1}^l \beta_{si} \beta_{sj} \mathbf{x}_{si} \mathbf{x}_{sj} - \frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l \beta_{si} \beta_{tj} \mathbf{x}_{si} \mathbf{x}_{tj} - \frac{3}{4} \sum_{i=1}^l \sum_{j=1}^l \beta_{ti} \beta_{tj} \mathbf{x}_{ti} \mathbf{x}_{tj}.
\end{aligned}$$

Portanto, obtemos em (II):

$$\begin{aligned}
& - \sum_{i=1}^l \frac{(\beta_{si})^2}{2C} - \sum_{i=1}^l \frac{(\beta_{ti})^2}{2C} + \sum_{i=1}^l \beta_{si} y_{si} - \sum_{i=1}^l |\beta_{si}| \epsilon_s + \sum_{i=1}^l \beta_{ti} y_{ti} - \sum_{i=1}^l |\beta_{ti}| \epsilon_t - \frac{3}{4} \sum_{i=1}^l \sum_{j=1}^l \beta_{si} \beta_{sj} \mathbf{x}_{si} \mathbf{x}_{sj} \\
& - \frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l \beta_{si} \beta_{tj} \mathbf{x}_{si} \mathbf{x}_{tj} - \frac{3}{4} \sum_{i=1}^l \sum_{j=1}^l \beta_{ti} \beta_{tj} \mathbf{x}_{ti} \mathbf{x}_{tj}.
\end{aligned}$$

Juntando (I) e (II) teremos o dual do problema de otimização (15):

$$\begin{aligned}
& \frac{3}{8} \sum_{i=1}^l \sum_{j=1}^l \beta_{si} \beta_{sj} \mathbf{x}_{si}^T \mathbf{x}_{sj} + \frac{1}{4} \sum_{i=1}^l \sum_{j=1}^l \beta_{si} \beta_{tj} \mathbf{x}_{si}^T \mathbf{x}_{tj} + \frac{3}{8} \sum_{i=1}^l \sum_{j=1}^l \beta_{ti} \beta_{tj} \mathbf{x}_{ti}^T \mathbf{x}_{tj} - \sum_{i=1}^l \frac{(\beta_{si})^2}{2C} - \sum_{i=1}^l \frac{(\beta_{ti})^2}{2C} \\
& + \sum_{i=1}^l \beta_{si} y_{si} - \sum_{i=1}^l |\beta_{si}| \epsilon_s + \sum_{i=1}^l \beta_{ti} y_{ti} - \sum_{i=1}^l |\beta_{ti}| \epsilon_t - \frac{3}{4} \sum_{i=1}^l \sum_{j=1}^l \beta_{si} \beta_{sj} \mathbf{x}_{si} \mathbf{x}_{sj} - \frac{1}{2} \sum_{i=1}^l \sum_{j=1}^l \beta_{si} \beta_{tj} \mathbf{x}_{si} \mathbf{x}_{tj} \\
& - \frac{3}{4} \sum_{i=1}^l \sum_{j=1}^l \beta_{ti} \beta_{tj} \mathbf{x}_{ti} \mathbf{x}_{tj},
\end{aligned}$$

realizando as operações aritméticas obtemos:

$$\begin{aligned}
& -\frac{3}{8} \sum_{i=1}^l \sum_{j=1}^l \beta_{si} \beta_{sj} \mathbf{x}_{\mathbf{s}i}^T \mathbf{x}_{\mathbf{s}j} - \frac{1}{4} \sum_{i=1}^l \sum_{j=1}^l \beta_{si} \beta_{tj} \mathbf{x}_{\mathbf{s}i}^T \mathbf{x}_{\mathbf{t}j} - \frac{3}{8} \sum_{i=1}^l \sum_{j=1}^l \beta_{ti} \beta_{tj} \mathbf{x}_{\mathbf{t}i}^T \mathbf{x}_{\mathbf{t}j} - \sum_{i=1}^l \frac{(\beta_{si})^2}{2C} - \sum_{i=1}^l \frac{(\beta_{ti})^2}{2C} \\
& + \sum_{i=1}^l \beta_{si} y_{si} - \sum_{i=1}^l |\beta_{si}| \epsilon_s + \sum_{i=1}^l \beta_{ti} y_{ti} - \sum_{i=1}^l |\beta_{ti}| \epsilon_t.
\end{aligned}$$

Apêndice B – Resultados TrAdaBoost

Neste apêndice, apresenta-se os resultados experimentais do método TrAdaBoost, variando o tamanho do conjunto formado. Os valores em destaque ressaltam o melhor resultado para cada experimento com o valor do parâmetro custo C considerado.

B.1 Conjunto de dados *Kin-family*

Experimento	Tamanho	$C=0.1$	$C=10$	$C=100$
50 + 20% m	1689	0,583	0,583	0,583
50 + 40% m	3327	0,581	0,581	0,581
50 + 60% m	4966	0,586	0,586	0,586
50 + 80% m	6604	0,564	0,564	0,564
50 + 100% m	8242	0,567	0,567	0,566

Tabela 22 – Resultados experimentais com *kernel* linear: conjunto *Kin-family* com 50 instâncias de treinamento na tarefa de destino.

Figura 29 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.

Experimento	Tamanho	$C=0.1$	$C=10$	$C=100$
50 + 20% m	1689	0,583	0,583	0,583
50 + 40% m	3327	0,474	0,474	0,485
50 + 60% m	4966	0,412	0,411	0,407
50 + 80% m	6604	0,437	0,437	0,444
50 + 100% m	8242	0,431	0,432	0,418

Tabela 23 – Resultados experimentais com *kernel* polinomial: conjunto *Kin-family* com 50 instâncias de treinamento na tarefa de destino.

Figura 30 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.

Experimento	Tamanho	$C=0.1$	$C=10$	$C=100$
100 + 20% m	1739	0,673	0,673	0,674
100 + 40% m	3377	0,699	0,699	0,699
100 + 60% m	5016	0,732	0,733	0,733
100 + 80% m	6654	0,682	0,682	0,682
100 + 100% m	8292	0,679	0,679	0,679

Tabela 24 – Resultados experimentais com *kernel* linear: conjunto *Kin-family* com 100 instâncias de treinamento na tarefa de destino.

Experimento	Tamanho	$C=0.1$	$C=10$	$C=100$
100 + 20% m	1739	0,653	0,653	0,660
100 + 40% m	3377	0,662	0,663	0,656
100 + 60% m	5016	0,663	0,664	0,661
100 + 80% m	6654	0,714	0,715	0,717
100 + 100% m	8292	0,719	0,718	0,678

Tabela 25 – Resultados experimentais com *kernel* polinomial: conjunto *Kin-family* com 100 instâncias de treinamento na tarefa de destino.

Experimento	Tamanho	$C=0.1$	$C=10$	$C=100$
150 + 20% m	1789	0,614	0,614	0,614
150 + 40% m	3427	0,623	0,623	0,623
150 + 60% m	5066	0,641	0,641	0,641
150 + 80% m	6704	0,590	0,589	0,590
150 + 100% m	8342	0,591	0,591	0,592

Tabela 26 – Resultados experimentais com *kernel* linear: conjunto *Kin-family* com 150 instâncias de treinamento na tarefa de destino.

Figura 31 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.

Figura 32 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.

Figura 33 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.

Experimento	Tamanho	$C=0.1$	$C=10$	$C=100$
150 + 20% m	1789	0,573	0,573	0,579
150 + 40% m	3427	0,544	0,544	0,544
150 + 60% m	5066	0,525	0,525	0,544
150 + 80% m	6704	0,557	0,557	0,530
150 + 100% m	8342	0,550	0,550	0,524

Tabela 27 – Resultados experimentais com *kernel* polinomial: conjunto *Kin-family* com 150 instâncias de treinamento na tarefa de destino.

Experimento	Tamanho	$C=0.1$	$C=10$	$C=100$
200 + 20% m	1839	0,611	0,611	0,611
200 + 40% m	3477	0,622	0,623	0,623
200 + 60% m	5116	0,637	0,637	0,637
200 + 80% m	6754	0,600	0,600	0,600
100% n + 100% m	8392	0,600	0,601	0,600

Tabela 28 – Resultados experimentais com *kernel* linear: conjunto *Kin-family* com 200 instâncias de treinamento na tarefa de destino.

Experimento	Tamanho	$C=0.1$	$C=10$	$C=100$
200 + 20% m	1839	0,577	0,577	0,578
200 + 40% m	3477	0,552	0,553	0,553
200 + 60% m	5116	0,542	0,542	0,538
200 + 80% m	6754	0,599	0,600	0,601
200 + 100% m	8392	0,596	0,597	0,581

Tabela 29 – Resultados experimentais com *kernel* polinomial: conjunto *Kin-family* com 200 instâncias de treinamento na tarefa de destino.

Figura 34 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.

Figura 35 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.

Figura 36 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.

B.2 Conjunto de dados Earthquake

Experimento	Tamanho	$C=0.1$	$C=10$	$C=100$
20 + 20% m	189	0,438	0,379	0,347
20 + 40% m	357	0,430	0,367	0,356
20 + 60% m	525	0,448	0,381	0,376
20 + 80% m	693	0,421	0,463	0,459
20 + 100% m	861	0,420	0,458	0,455

Tabela 30 – Resultados experimentais com *kernel* linear: conjunto *Earthquake* com 20 instâncias de treinamento na tarefa de destino.

Experimento	Tamanho	$C=0.1$	$C=10$	$C=100$
20 + 20% m	189	0,409	1,214	2,227
20 + 40% m	357	0,420	0,859	1,998
20 + 60% m	525	0,516	1,011	2,702
20 + 80% m	693	0,685	0,435	1,257
20 + 100% m	861	0,641	0,403	1,445

Tabela 31 – Resultados experimentais com *kernel* polinomial: conjunto *Earthquake* com 20 instâncias de treinamento na tarefa de destino.

Experimento	Tamanho	$C=0.1$	$C=10$	$C=100$
30 + 20% m	199	0,856	0,897	0,971
30 + 40% m	367	0,839	0,872	0,953
30 + 60% m	535	0,839	0,872	0,955
30 + 80% m	703	0,780	0,809	0,870
30 + 100% m	871	0,790	0,831	0,883

Tabela 32 – Resultados experimentais com *kernel* linear: conjunto *Earthquake* com 30 instâncias de treinamento na tarefa de destino.

Figura 37 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.

Figura 38 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.

Figura 39 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.

Experimento	Tamanho	$C=0.1$	$C=10$	$C=100$
30 + 20% m	199	0,943	1,232	1,600
30 + 40% m	367	0,760	0,782	2,142
30 + 60% m	535	0,669	0,656	1,080
30 + 80% m	703	0,949	1,275	1,903
30 + 100% m	871	0,983	1,311	2,068

Tabela 33 – Resultados experimentais com *kernel* polinomial: conjunto *Earthquake* com 30 instâncias de treinamento na tarefa de destino.

Experimento	Tamanho	$C=0.1$	$C=10$	$C=100$
50 + 20% m	219	0,501	0,513	0,583
50 + 40% m	387	0,506	0,522	0,598
50 + 60% m	555	0,496	0,518	0,609
50 + 80% m	723	0,483	0,489	0,490
50 + 100% m	891	0,489	0,495	0,498

Tabela 34 – Resultados experimentais com *kernel* linear: conjunto *Earthquake* com 50 instâncias de treinamento na tarefa de destino.

Experimento	Tamanho	$C=0.1$	$C=10$	$C=100$
50 + 20% m	219	0,465	0,492	0,642
50 + 40% m	387	0,518	0,474	0,491
50 + 60% m	555	0,488	0,528	0,547
50 + 80% m	723	0,417	0,481	0,488
50 + 100% m	891	0,416	0,492	0,511

Tabela 35 – Resultados experimentais com *kernel* polinomial: conjunto *Earthquake* com 50 instâncias de treinamento na tarefa de destino.

Figura 40 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.

Figura 41 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.

Figura 42 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.

Experimento	Tamanho	$C=0.1$	$C=10$	$C=100$
$100\%n + 20\%m$	239	0,544	0,546	0,550
$70 + 40\%m$	407	0,534	0,533	0,529
$70 + 60\%m$	575	0,524	0,528	0,530
$70 + 80\%m$	743	0,478	0,481	0,485
$70 + 100\%m$	911	0,486	0,492	0,493

Tabela 36 – Resultados experimentais com *kernel* linear: conjunto *Earthquake* com 70 instâncias de treinamento na tarefa de destino.

Experimento	Tamanho	$C=0.1$	$C=10$	$C=100$
$70 + 20\%m$	239	0,420	0,473	0,515
$70 + 40\%m$	407	0,429	0,488	0,562
$70 + 60\%m$	575	0,428	0,480	0,590
$70 + 80\%m$	743	0,510	0,499	0,497
$70 + 100\%m$	911	0,517	0,515	0,520

Tabela 37 – Resultados experimentais com *kernel* polinomial: conjunto *Earthquake* com 70 instâncias de treinamento na tarefa de destino.

B.3 Conjunto de dados *Fly*

Experimento	Tamanho	$C=0.1$	$C=10$	$C=100$
$50 + 20\%m$	2050	0,391	0,198	0,212
$50 + 40\%m$	4050	0,483	0,478	0,522
$50 + 60\%m$	6050	0,562	0,582	0,627
$50 + 80\%m$	8050	0,106	0,104	0,108
$50 + 100\%m$	1050	0,111	0,113	0,107

Tabela 38 – Resultados experimentais com *kernel* linear: conjunto *Fly* com 50 instâncias de treinamento na tarefa de destino.

Figura 43 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.

Figura 44 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.

Figura 45 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.

Experimento	Tamanho	$C=0.1$	$C=10$	$C=100$
50 + 20% m	2050	0,924	6,249	4,707
50 + 40% m	4050	1,116	6,366	4,844
50 + 60% m	6050	1,154	2,297	1,912
50 + 80% m	8050	0,116	0,490	0,540
50 + 100% m	1050	0,131	0,388	0,321

Tabela 39 – Resultados experimentais com *kernel* polinomial: conjunto *Fly* com 50 instâncias treinamento na tarefa de destino.

Experimento	Tamanho	$C=0.1$	$C=10$	$C=100$
200 + 20% m	2200	0,157	0,150	0,148
200 + 40% m	4200	0,290	0,259	0,280
200 + 60% m	6200	0,650	0,631	0,603
200 + 80% m	8200	0,173	0,159	0,213
200 + 100% m	10200	0,167	0,152	0,193

Tabela 40 – Resultados experimentais com *kernel* linear: conjunto *Fly* com 200 instâncias de treinamento na tarefa de destino.

Experimento	Tamanho	$C=0.1$	$C=10$	$C=100$
200 + 20% m	2200	10,025	71,921	40,545
200 + 40% m	4200	1947,358	63,907	465,102
200 + 60% m	6200	2526,520	37,962	22442,574
200 + 80% m	8200	0,148	0,362	0,601
200 + 100% m	10200	0,149	0,429	0,243

Tabela 41 – Resultados experimentais com *kernel* polinomial: conjunto *Fly* com 200 instâncias de treinamento na tarefa de destino.

Figura 46 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.

Figura 47 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.

Figura 48 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.

Experimento	Tamanho	$C=0.1$	$C=10$	$C=100$
400 + 20% m	2400	0,161	0,149	0,151
400 + 40% m	4400	0,213	0,211	0,216
400 + 60% m	6400	0,329	0,327	0,354
400 + 80% m	8400	0,183	0,166	0,187
400 + 100% m	10400	0,178	0,163	0,217

Tabela 42 – Resultados experimentais com *kernel* linear: conjunto *Fly* com 400 instâncias de treinamento na tarefa de destino.

Experimento	Tamanho	$C=0.1$	$C=10$	$C=100$
400 + 20% m	2400	0,951	6,182	10,296
400 + 40% m	4400	1,965	16,798	40,380
400 + 60% m	6400	1,798	5,698	4,518
400 + 80% m	8400	0,268	5,769	8,232
400 + 100% m	10400	0,187	0,221	1,381

Tabela 43 – Resultados experimentais com *kernel* polinomial: conjunto *Fly* com 400 instâncias de treinamento na tarefa de destino.

Experimento	Tamanho	$C=0.1$	$C=10$	$C=100$
800 + 20% m	2800	0,104	0,094	0,103
800 + 40% m	4800	0,147	0,140	0,133
800 + 60% m	6800	0,206	0,201	0,209
800 + 80% m	8800	0,123	0,119	0,157
800 + 100% m	10800	0,128	0,123	0,165

Tabela 44 – Resultados experimentais com *kernel* linear: conjunto *Fly* com 800 instâncias de treinamento na tarefa de destino.

Figura 49 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.

Figura 50 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.

Figura 51 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.

Experimento	Tamanho	$C=0.1$	$C=10$	$C=100$
800 + 20% m	2800	3,880	9,959	18,022
800 + 40% m	4800	27,558	2955,028	7,393
800 + 60% m	6800	0,667	11,693	2,447
800 + 80% m	8800	0,422	13,216	6,275
800 + 100% m	10800	0,161	1,674	0,990

Tabela 45 – Resultados experimentais com *kernel* polinomial: conjunto *Fly* com 800 instâncias de treinamento na tarefa de destino.

B.4 Conjunto de dados ALK-5

Experimento	Tamanho	$C=0.1$	$C=10$	$C=100$
25% n + 20% m	39	2,516	2,969	3,185
25% n + 40% m	62	2,941	3,230	3,336
25% n + 60% m	86	2,846	3,115	3,126
25% n + 80% m	109	2,548	2,961	2,989
25% n + 100% m	132	2,854	3,043	3,042

Tabela 46 – Resultados experimentais com *kernel* linear: conjunto ALK-5 com 25% dos dados de treinamento na tarefa de destino.

Experimento	Tamanho	$C=0.1$	$C=10$	$C=100$
25% n + 20% m	39	3,561	6,222	6,222
25% n + 40% m	62	24,389	77,788	541,388
25% n + 60% m	86	5,750	102,783	169,960
25% n + 80% m	109	3,350	11,059	14,505
25% n + 100% m	132	4,019	21,288	24,687

Tabela 47 – Resultados experimentais com *kernel* polinomial: conjunto ALK-5 com 25% dos dados de treinamento na tarefa de destino.

Figura 52 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.

Figura 53 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.

Figura 54 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.

Experimento	Tamanho	$C=0.1$	$C=10$	$C=100$
$50\%n + 20\%m$	54	1,213	1,306	1,340
$50\%n + 40\%m$	77	1,274	1,382	1,451
$50\%n + 60\%m$	101	1,507	1,620	1,663
$50\%n + 80\%m$	124	1,268	1,332	1,323
$50\%n + 100\%m$	147	1,348	1,392	1,408

Tabela 48 – Resultados experimentais com *kernel* linear: conjunto ALK-5 com 50% dos dados de treinamento na tarefa de destino.

Experimento	Tamanho	$C=0.1$	$C=10$	$C=100$
$50\%n + 20\%m$	54	1,464	1,013	1,006
$50\%n + 40\%m$	77	2,026	11,070	12,899
$50\%n + 60\%m$	101	5,796	5,970	13,717
$50\%n + 80\%m$	124	1,549	2,158	1,833
$50\%n + 100\%m$	147	1,247	1,484	1,690

Tabela 49 – Resultados experimentais com *kernel* polinomial: conjunto ALK-5 com 50% dos dados de treinamento na tarefa de destino.

Experimento	Tamanho	$C=0.1$	$C=10$	$C=100$
$75\%n + 20\%m$	69	1,00	0,916	0,914
$75\%n + 40\%m$	92	1,078	1,094	1,093
$75\%n + 60\%m$	116	1,245	1,211	1,215
$75\%n + 80\%m$	139	1,331	1,314	1,307
$75\%n + 100\%m$	162	1,371	1,332	1,336

Tabela 50 – Resultados experimentais com *kernel* linear: conjunto ALK-5 com 75% dos dados de treinamento na tarefa de destino.

Figura 55 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.

Figura 56 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.

Figura 57 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.

Experimento	Tamanho	$C=0.1$	$C=10$	$C=100$
$75\%n + 20\%m$	69	2,074	6,722	17,494
$75\%n + 40\%m$	92	3,420	11,973	26,488
$75\%n + 60\%m$	116	1,794	2,445	2,532
$75\%n + 80\%m$	139	0,914	2,068	3,221
$75\%n + 100\%m$	162	1,059	1,233	1,254

Tabela 51 – Resultados experimentais com *kernel* polinomial: conjunto ALK-5 com 75% dos dados de treinamento na tarefa de destino.

Experimento	Tamanho	$C=0.1$	$C=10$	$C=100$
$100\%n + 20\%m$	83	1,055	1,029	1,006
$100\%n + 40\%m$	106	1,196	1,180	1,181
$100\%n + 60\%m$	130	1,428	1,465	1,480
$100\%n + 80\%m$	153	1,344	1,320	1,326
$100\%n + 100\%m$	176	1,367	1,364	1,356

Tabela 52 – Resultados experimentais com *kernel* linear: conjunto ALK-5 com 100% dos dados de treinamento na tarefa de destino.

Experimento	Tamanho	$C=0.1$	$C=10$	$C=100$
$100\%n + 20\%m$	83	1,441	5,498	7,089
$100\%n + 40\%m$	106	1,240	3,487	3,139
$100\%n + 60\%m$	130	4,323	6,817	4,522
$100\%n + 80\%m$	153	3,325	3,479	2,689
$100\%n + 100\%m$	176	1,273	1,720	1,752

Tabela 53 – Resultados experimentais com *kernel* polinomial: conjunto ALK-5 com 100% dos dados de treinamento na tarefa de destino.

Figura 58 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.

Figura 59 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.

Figura 60 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.

B.5 Conjunto de dados PPAR- α

Experimento	Tamanho	$C=0.1$	$C=10$	$C=100$
$25\%n + 20\%m$	36	0,571	0,556	0,559
$25\%n + 40\%m$	54	0,550	0,592	0,606
$25\%n + 60\%m$	72	0,517	0,565	0,559
$25\%n + 80\%m$	90	0,615	0,733	0,750
$25\%n + 100\%m$	107	0,523	0,575	0,588

Tabela 54 – Resultados experimentais com *kernel* linear: conjunto PPAR- α com 25% dos dados de treinamento na tarefa de destino.

Experimento	Tamanho	$C=0.1$	$C=10$	$C=100$
$25\%n + 20\%m$	36	0,502	3,915	1,800
$25\%n + 40\%m$	54	2,043	0,604	0,598
$25\%n + 60\%m$	72	0,701	1,462	1,568
$25\%n + 80\%m$	90	2,897	9,049	7,857
$25\%n + 100\%m$	107	2,653	7,074	7,200

Tabela 55 – Resultados experimentais: conjunto alpha - método TrAdaBoost variando o tamanho do conjunto de treinamento da tarefa de origem e com 25% dos dados de treinamento na tarefa de destino.

Experimento	Tamanho	$C=0.1$	$C=10$	$C=100$
$50\%n + 20\%m$	53	0,628	0,639	0,638
$50\%n + 40\%m$	71	0,632	0,654	0,653
$50\%n + 60\%m$	89	0,585	0,599	0,600
$50\%n + 80\%m$	107	0,588	0,589	0,589
$50\%n + 100\%m$	124	0,594	0,590	0,590

Tabela 56 – Resultados experimentais com *kernel* linear: conjunto PPAR- α com 50% dos dados de treinamento na tarefa de destino.

Figura 61 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.

Figura 62 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.

Figura 63 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.

Experimento	Tamanho	$C=0.1$	$C=10$	$C=100$
$50\%n + 20\%m$	53	1,186	0,715	1,161
$50\%n + 40\%m$	71	0,440	0,814	0,749
$50\%n + 60\%m$	89	0,498	0,587	0,789
$50\%n + 80\%m$	107	2,684	4,407	3,927
$50\%n + 100\%m$	124	3,021	5,352	5,316

Tabela 57 – Resultados experimentais com *kernel* polinomial: conjunto PPAR- α com 50% dos dados de treinamento na tarefa de destino.

Experimento	Tamanho	$C=0.1$	$C=10$	$C=100$
$75\%n + 20\%m$	71	0,612	0,673	0,689
$75\%n + 40\%m$	89	0,569	0,646	0,653
$75\%n + 60\%m$	107	0,582	0,603	0,603
$75\%n + 80\%m$	125	0,572	0,604	0,606
$75\%n + 100\%m$	142	0,548	0,561	0,563

Tabela 58 – Resultados experimentais com *kernel* linear: conjunto PPAR- α com 75% dos dados de treinamento na tarefa de destino.

Experimento	Tamanho	$C=0.1$	$C=10$	$C=100$
$75\%n + 20\%m$	71	0,614	0,699	0,747
$75\%n + 40\%m$	89	0,641	0,841	0,975
$75\%n + 60\%m$	107	0,699	0,977	0,964
$75\%n + 80\%m$	125	0,628	0,741	0,708
$75\%n + 100\%m$	142	0,522	0,570	0,600

Tabela 59 – Resultados experimentais com *kernel* polinomial: conjunto PPAR- α com 75% dos dados de treinamento na tarefa de destino.

Figura 64 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.

Figura 65 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.

Figura 66 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.

Experimento	Tamanho	$C=0.1$	$C=10$	$C=100$
$100\%n + 20\%m$	88	0,539	0,539	0,533
$100\%n + 40\%m$	106	0,540	0,549	0,549
$100\%n + 60\%m$	124	0,550	0,560	0,561
$100\%n + 80\%m$	142	0,553	0,555	0,559
$100\%n + 100\%m$	159	0,538	0,551	0,549

Tabela 60 – Resultados experimentais com *kernel* linear: conjunto PPAR- α com 100% dos dados de treinamento na tarefa de destino.

Experimento	Tamanho	$C=0.1$	$C=10$	$C=100$
$100\%n + 20\%m$	88	0,864	2,358	2,529
$100\%n + 40\%m$	106	0,899	2,565	2,239
$100\%n + 60\%m$	124	0,984	2,656	2,640
$100\%n + 80\%m$	142	1,070	2,634	2,557
$100\%n + 100\%m$	159	1,384	2,549	2,417

Tabela 61 – Resultados experimentais com *kernel* polinomial: conjunto PPAR- α com 100% dos dados de treinamento na tarefa de destino.

B.6 Conjunto de dados PPAR- δ

Experimento	Tamanho	$C=0.1$	$C=10$	$C=100$
$25\%n + 20\%m$	36	0,841	0,803	0,820
$25\%n + 40\%m$	54	0,799	0,769	0,799
$25\%n + 60\%m$	72	0,792	0,698	0,713
$25\%n + 80\%m$	90	0,735	0,616	0,615
$25\%n + 100\%m$	107	0,718	0,664	0,667

Tabela 62 – Resultados experimentais com *kernel* linear: conjunto PPAR- δ com 25% dos dados de treinamento na tarefa de destino.

Figura 67 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.

Figura 68 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.

Figura 69 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.

Experimento	Tamanho	$C=0.1$	$C=10$	$C=100$
$25\%n + 20\%m$	36	0,771	1,143	2,916
$25\%n + 40\%m$	54	0,692	0,925	2,128
$25\%n + 60\%m$	72	0,624	1,008	1,857
$25\%n + 80\%m$	90	0,473	0,555	0,539
$25\%n + 100\%m$	107	0,444	0,399	0,371

Tabela 63 – Resultados experimentais com *kernel* polinomial: conjunto PPAR- δ com 25% dos dados de treinamento na tarefa de destino.

Experimento	Tamanho	$C=0.1$	$C=10$	$C=100$
$50\%n + 20\%m$	53	1,013	1,011	1,017
$50\%n + 40\%m$	71	1,071	1,051	1,060
$50\%n + 60\%m$	89	1,021	1,036	1,036
$50\%n + 80\%m$	107	0,992	1,012	1,010
$50\%n + 100\%m$	124	0,959	0,968	0,968

Tabela 64 – Resultados experimentais com *kernel* linear: conjunto PPAR- δ com 50% dos dados de treinamento na tarefa de destino.

Experimento	Tamanho	$C=0.1$	$C=10$	$C=100$
$50\%n + 20\%m$	53	3,720	2,795	3,215
$50\%n + 40\%m$	71	5,350	4,906	5,129
$50\%n + 60\%m$	89	2,864	1,227	1,507
$50\%n + 80\%m$	107	2,321	4,446	4,489
$50\%n + 100\%m$	124	2,703	2,966	3,434

Tabela 65 – Resultados experimentais com *kernel* polinomial: conjunto PPAR- δ com 50% dos dados de treinamento na tarefa de destino.

Figura 70 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.

Figura 71 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.

Figura 72 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.

Experimento	Tamanho	$C=0.1$	$C=10$	$C=100$
$75\%n + 20\%m$	71	1,129	1,143	1,148
$75\%n + 40\%m$	89	1,138	1,163	1,163
$75\%n + 60\%m$	107	1,125	1,168	1,168
$75\%n + 80\%m$	125	1,129	1,155	1,154
$75\%n + 100\%m$	142	1,096	1,099	1,093

Tabela 66 – Resultados experimentais com *kernel* linear: conjunto PPAR- δ com 75% dos dados de treinamento na tarefa de destino.

Experimento	Tamanho	$C=0.1$	$C=10$	$C=100$
$75\%n + 20\%m$	71	1,397	2,333	2,980
$75\%n + 40\%m$	89	1,228	1,584	1,739
$75\%n + 60\%m$	107	2,161	2,304	2,764
$75\%n + 80\%m$	125	3,462	5,099	4,851
$75\%n + 100\%m$	142	2,768	3,003	3,845

Tabela 67 – Resultados experimentais com *kernel* polinomial: conjunto PPAR- δ com 75% dos dados de treinamento na tarefa de destino.

Experimento	Tamanho	$C=0.1$	$C=10$	$C=100$
$100\%n + 20\%m$	88	1,003	1,026	1,020
$100\%n + 40\%m$	106	1,017	1,034	1,034
$100\%n + 60\%m$	124	1,023	1,054	1,050
$100\%n + 80\%m$	142	1,010	1,051	1,052
$100\%n + 100\%m$	159	0,959	0,991	1,001

Tabela 68 – Resultados experimentais com *kernel* linear: conjunto PPAR- δ com 100% dos dados de treinamento na tarefa de destino.

Figura 73 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.

Figura 74 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.

Figura 75 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.

Experimento	Tamanho	$C=0.1$	$C=10$	$C=100$
$100\%n + 20\%m$	88	1,447	2,261	2,992
$100\%n + 40\%m$	106	1,584	2,172	2,496
$100\%n + 60\%m$	124	3,309	3,921	3,976
$100\%n + 80\%m$	142	2,629	3,238	3,682
$100\%n + 100\%m$	159	2,360	2,037	2,011

Tabela 69 – Resultados experimentais com *kernel* polinomial: conjunto PPAR- δ com 100% dos dados de treinamento na tarefa de destino.

B.7 Conjunto de dados DHFR

Experimento	Tamanho	$C=0.1$	$C=10$	$C=100$
$25\%n + 20\%m$	246	0,915	0,930	0,933
$25\%n + 40\%m$	394	0,957	0,988	1,000
$25\%n + 60\%m$	542	0,974	1,099	1,113
$25\%n + 80\%m$	690	0,915	1,121	1,152
$25\%n + 100\%m$	837	0,970	1,195	1,222

Tabela 70 – Resultados experimentais com *kernel* linear: conjunto DHFR com 25% dos dados de treinamento na tarefa de destino.

Experimento	Tamanho	$C=0.1$	$C=10$	$C=100$
$25\%n + 20\%m$	246	0,799	0,699	0,708
$25\%n + 40\%m$	394	0,791	0,721	0,709
$25\%n + 60\%m$	542	0,715	0,701	0,998
$25\%n + 80\%m$	690	0,616	0,628	0,640
$25\%n + 100\%m$	837	0,710	0,688	0,887

Tabela 71 – Resultados experimentais com *kernel* polinomial: conjunto DHFR com 25% dos dados de treinamento na tarefa de destino.

Figura 76 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.

Figura 77 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.

Figura 78 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.

Experimento	Tamanho	$C=0.1$	$C=10$	$C=100$
$50\%n + 20\%m$	344	0,670	0,669	0,669
$50\%n + 40\%m$	492	0,691	0,703	0,707
$50\%n + 60\%m$	640	0,756	0,844	0,860
$50\%n + 80\%m$	788	0,729	0,853	0,860
$50\%n + 100\%m$	935	0,756	0,902	0,917

Tabela 72 – Resultados experimentais com *kernel* linear: conjunto DHFR com 50% dos dados de treinamento na tarefa de destino.

Experimento	Tamanho	$C=0.1$	$C=10$	$C=100$
$50\%n + 20\%m$	344	0,510	0,518	0,552
$50\%n + 40\%m$	492	0,525	0,540	0,530
$50\%n + 60\%m$	640	0,537	0,573	0,581
$50\%n + 80\%m$	788	0,493	0,546	0,616
$50\%n + 100\%m$	935	0,484	0,532	0,553

Tabela 73 – Resultados experimentais com *kernel* polinomial: conjunto DHFR com 50% dos dados de treinamento na tarefa de destino.

Experimento	Tamanho	$C=0.1$	$C=10$	$C=100$
$75\%n + 20\%m$	442	0,654	0,654	0,659
$75\%n + 40\%m$	590	0,685	0,704	0,706
$75\%n + 60\%m$	738	0,767	0,862	0,866
$75\%n + 80\%m$	886	0,740	0,839	0,848
$75\%n + 100\%m$	1033	0,773	0,896	0,902

Tabela 74 – Resultados experimentais com *kernel* linear: conjunto DHFR com 75% dos dados de treinamento na tarefa de destino.

Figura 79 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.

Figura 80 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.

Figura 81 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.

Experimento	Tamanho	$C=0.1$	$C=10$	$C=100$
$75\%n + 20\%m$	442	0,501	0,565	0,578
$75\%n + 40\%m$	590	0,526	0,614	0,634
$75\%n + 60\%m$	738	0,591	0,653	0,591
$75\%n + 80\%m$	886	0,542	0,556	0,619
$75\%n + 100\%m$	1033	0,541	0,581	0,594

Tabela 75 – Resultados experimentais com *kernel* polinomial: conjunto DHFR com 75% dos dados de treinamento na tarefa de destino.

Experimento	Tamanho	$C=0.1$	$C=10$	$C=100$
$100\%n + 20\%m$	539	0,652	0,659	0,666
$100\%n + 40\%m$	687	0,708	0,742	0,744
$100\%n + 60\%m$	835	0,798	0,918	0,929
$100\%n + 80\%m$	983	0,722	0,794	0,797
$100\%n + 100\%m$	1130	0,761	0,845	0,853

Tabela 76 – Resultados experimentais com *kernel* linear: conjunto DHFR com 100% dos dados de treinamento na tarefa de destino.

Experimento	Tamanho	$C=0.1$	$C=10$	$C=100$
$100\%n + 20\%m$	539	0,483	0,479	0,482
$100\%n + 40\%m$	687	0,510	0,549	0,538
$100\%n + 60\%m$	835	0,548	0,587	0,657
$100\%n + 80\%m$	983	0,505	0,536	0,562
$100\%n + 100\%m$	1130	0,508	0,547	0,561

Tabela 77 – Resultados experimentais com *kernel* polinomial: conjunto DHFR com 100% dos dados de treinamento na tarefa de destino.

Figura 82 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.

Figura 83 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.

Figura 84 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.

B.8 Conjunto de dados COX2

Experimento	Tamanho	$C=0.1$	$C=10$	$C=100$
$25\%n + 20\%m$	167	0,809	0,855	0,853
$25\%n + 40\%m$	253	0,865	0,929	0,931
$25\%n + 60\%m$	339	0,880	0,939	0,940
$25\%n + 80\%m$	425	0,767	0,777	0,778
$25\%n + 100\%m$	511	0,799	0,816	0,816

Tabela 78 – Resultados experimentais com *kernel* linear: conjunto COX2 com 25% dos dados de treinamento na tarefa de destino.

Experimento	Tamanho	$C=0.1$	$C=10$	$C=100$
$25\%n + 20\%m$	167	0,805	0,934	0,967
$25\%n + 40\%m$	253	0,869	1,038	1,062
$25\%n + 60\%m$	339	0,719	0,700	0,717
$25\%n + 80\%m$	425	0,690	0,669	0,689
$25\%n + 100\%m$	511	0,641	0,647	0,669

Tabela 79 – Resultados experimentais com *kernel* polinomial: conjunto COX2 com 25% dos dados de treinamento na tarefa de destino.

Experimento	Tamanho	$C=0.1$	$C=10$	$C=100$
$50\%n + 20\%m$	246	1,122	1,166	1,169
$50\%n + 40\%m$	332	1,178	1,221	1,220
$50\%n + 60\%m$	418	1,162	1,193	1,994
$50\%n + 80\%m$	504	1,086	1,097	1,097
$50\%n + 100\%m$	590	1,077	1,083	1,082

Tabela 80 – Resultados experimentais com *kernel* linear: conjunto COX2 com 50% dos dados de treinamento na tarefa de destino.

Figura 85 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.

Figura 86 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.

Figura 87 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.

Experimento	Tamanho	$C=0.1$	$C=10$	$C=100$
$50\%n + 20\%m$	246	0,971	1,051	1,076
$50\%n + 40\%m$	332	1,006	1,067	1,060
$50\%n + 60\%m$	418	0,875	0,903	0,928
$50\%n + 80\%m$	504	0,799	0,796	0,822
$50\%n + 100\%m$	590	0,801	0,804	0,808

Tabela 81 – Resultados experimentais com *kernel* polinomial: conjunto COX2 com 50% dos dados de treinamento na tarefa de destino.

Experimento	Tamanho	$C=0.1$	$C=10$	$C=100$
$75\%n + 20\%m$	325	1,024	1,028	1,027
$75\%n + 40\%m$	454	1,042	1,048	1,049
$75\%n + 60\%m$	497	0,959	0,962	0,962
$75\%n + 80\%m$	583	0,942	0,943	0,943
$75\%n + 100\%m$	669	0,946	0,948	0,947

Tabela 82 – Resultados experimentais com *kernel* linear: conjunto COX2 com 75% dos dados de treinamento na tarefa de destino.

Experimento	Tamanho	$C=0.1$	$C=10$	$C=100$
$75\%n + 20\%m$	325	0,919	0,890	0,898
$75\%n + 40\%m$	454	0,875	0,871	0,938
$75\%n + 60\%m$	497	0,788	0,780	0,784
$75\%n + 80\%m$	583	0,804	0,821	0,852
$75\%n + 100\%m$	669	0,743	0,744	0,764

Tabela 83 – Resultados experimentais com *kernel* polinomial: conjunto COX2 com 75% dos dados de treinamento na tarefa de destino.

Figura 88 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.

Figura 89 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.

Figura 90 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.

Experimento	Tamanho	$C=0.1$	$C=10$	$C=100$
$100\%n + 20\%m$	404	0,993	0,999	0,999
$100\%n + 40\%m$	490	1,028	1,041	1,041
$100\%n + 60\%m$	576	0,962	0,976	0,976
$100\%n + 80\%m$	662	0,956	0,963	0,963
$100\%n + 100\%m$	748	0,949	0,950	0,950

Tabela 84 – Resultados experimentais com *kernel* linear: conjunto COX2 com 100% dos dados de treinamento na tarefa de destino.

Experimento	Tamanho	$C=0.1$	$C=10$	$C=100$
$100\%n + 20\%m$	404	0,839	0,869	0,915
$100\%n + 40\%m$	490	0,813	0,810	0,855
$100\%n + 60\%m$	576	0,742	0,743	0,765
$100\%n + 80\%m$	662	0,751	0,747	0,794
$100\%n + 100\%m$	748	0,746	0,730	0,803

Tabela 85 – Resultados experimentais com *kernel* polinomial: conjunto COX2 com 100% dos dados de treinamento na tarefa de destino.

B.9 Conjunto de dados ER

Experimento	Tamanho	$C=0.1$	$C=10$	$C=100$
$25\%n + 20\%m$	157	0,927	0,884	0,882
$25\%n + 40\%m$	233	0,942	0,951	0,959
$25\%n + 60\%m$	309	1,014	1,060	1,104
$25\%n + 80\%m$	385	1,131	1,218	1,245
$25\%n + 100\%m$	461	1,174	1,195	1,205

Tabela 86 – Resultados experimentais com *kernel* linear: conjunto ER com 25% dos dados de treinamento na tarefa de destino.

Figura 91 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.

Figura 92 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.

Figura 93 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.

Experimento	Tamanho	$C=0.1$	$C=10$	$C=100$
$25\%n + 20\%m$	157	1,362	4,101	4,167
$25\%n + 40\%m$	233	1,363	3,415	7,874
$25\%n + 60\%m$	309	1,895	2,333	4,152
$25\%n + 80\%m$	385	1,250	1,455	2,720
$25\%n + 100\%m$	461	1,468	1,338	1,206

Tabela 87 – Resultados experimentais com *kernel* polinomial: conjunto ER com 25% dos dados de treinamento na tarefa de destino.

Experimento	Tamanho	$C=0.1$	$C=10$	$C=100$
$50\%n + 20\%m$	238	98,903	123,596	134,694
$50\%n + 40\%m$	314	81,910	97,630	96,978
$50\%n + 60\%m$	390	8,213	28,845	31,092
$50\%n + 80\%m$	466	1,094	6,319	7,068
$50\%n + 100\%m$	542	1,005	1,029	1,040

Tabela 88 – Resultados experimentais com *kernel* linear: conjunto ER com 50% dos dados de treinamento na tarefa de destino.

Experimento	Tamanho	$C=0.1$	$C=10$	$C=100$
$50\%n + 20\%m$	238	1,03E+07	4,67E+07	8,64E+07
$50\%n + 40\%m$	314	2,65E+07	2,02E+07	1,75E+07
$50\%n + 60\%m$	390	3,03E+07	3,03E+07	5,56E+07
$50\%n + 80\%m$	466	1,23E+06	2,02E+06	8,67E+03
$50\%n + 100\%m$	542	1,328	255,438	789,487

Tabela 89 – Resultados experimentais com *kernel* polinomial: conjunto ER com 50% dos dados de treinamento na tarefa de destino.

Figura 94 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.

Figura 95 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.

Figura 96 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.

Experimento	Tamanho	$C=0.1$	$C=10$	$C=100$
$75\%n + 20\%m$	319	0,945	0,867	0,866
$75\%n + 40\%m$	395	0,998	0,965	0,965
$75\%n + 60\%m$	471	1,029	0,993	0,995
$75\%n + 80\%m$	547	0,970	0,941	0,942
$75\%n + 100\%m$	623	0,994	0,959	0,965

Tabela 90 – Resultados experimentais com *kernel* linear: conjunto ER com 75% dos dados de treinamento na tarefa de destino.

Experimento	Tamanho	$C=0.1$	$C=10$	$C=100$
$75\%n + 20\%m$	319	1,035	28,890	69,294
$75\%n + 40\%m$	395	1,100	5,755	9,589
$75\%n + 60\%m$	471	1,094	16,020	94,901
$75\%n + 80\%m$	547	1,075	42,377	113,268
$75\%n + 100\%m$	623	1,122	24,164	9,432

Tabela 91 – Resultados experimentais com *kernel* polinomial: conjunto ER com 75% dos dados de treinamento na tarefa de destino.

Experimento	Tamanho	$C=0.1$	$C=10$	$C=100$
$100\%n + 20\%m$	399	20,250	25,869	31,489
$100\%n + 40\%m$	475	44,564	49,067	48,753
$100\%n + 60\%m$	551	11,508	5,775	5,475
$100\%n + 80\%m$	627	0,866	0,860	0,860
$100\%n + 100\%m$	703	0,889	0,870	0,875

Tabela 92 – Resultados experimentais com *kernel* linear: conjunto ER com 100% dos dados de treinamento na tarefa de destino.

Figura 97 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.

Figura 98 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.

Figura 99 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.

Experimento	Tamanho	$C=0.1$	$C=10$	$C=100$
$100\%n + 20\%m$	399	1,67E+05	5,73E+05	4,63E+05
$100\%n + 40\%m$	475	3,87E+06	1,30E+06	2,64E+06
$100\%n + 60\%m$	551	5,62E+05	3,03E+06	3,44E+06
$100\%n + 80\%m$	627	1,208	59,381	116,231
$100\%n + 100\%m$	703	0,955	40,739	32,210

Tabela 93 – Resultados experimentais com *kernel* polinomial: conjunto ER com 100% dos dados de treinamento na tarefa de destino.

Figura 100 – Análise gráfica dos resultados para cada valor de custo C e conjunto formado.