

Theory and Algorithms for Hypothesis Transfer Learning

THÈSE N° 8011 (2018)

PRÉSENTÉE LE 27 FÉVRIER 2018

À LA FACULTÉ DES SCIENCES ET TECHNIQUES DE L'INGÉNIEUR

LABORATOIRE DE L'IDIAP

PROGRAMME DOCTORAL EN GÉNIE ÉLECTRIQUE

ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE

POUR L'OBTENTION DU GRADE DE DOCTEUR ÈS SCIENCES

PAR

Ilja KUZBORSKIJ

acceptée sur proposition du jury:

Prof. J.-Ph. Thiran, président du jury
Prof. H. Bourlard, Prof. B. Caputo, directeurs de thèse
Prof. J. Suykens, rapporteur
Prof. A. Habrard, rapporteur
Dr M. Salzmann, rapporteur



ÉCOLE POLYTECHNIQUE
FÉDÉRALE DE LAUSANNE

Suisse
2018

Acknowledgments

I would like to start by expressing sheer gratitude to my advisor Barbara Caputo. Thank you, Barbara, for many years of support and for your open mind that granted me the freedom of learning and the possibility to work on exciting problems.

I am also deeply indebted to Francesco Orabona who helped me to spark an interest in theoretical research, taught me ways of thinking about algorithms and proofs, for his patience, and for hosting me at TTI in Chicago. I will always admire your honesty, your advice, and I will always check my proofs!

A very special thanks to Nicolò Cesa-Bianchi. Thank you, Nicolò, for being an inspiring teacher, for your patience, and for our amazing collaboration.

I have also spent several wonderful months in Vienna, and I would like to extend my gratitude to Christoph Lampert, who generously hosted me at IST Austria, and also for granting me the freedom in my research, and for always having time for stimulating discussions.

I would also like to thank my thesis director Prof. Hervé Bouchard, as well as, Prof. Johan Suykens, Prof. Amaury Habrard, Dr. Mathieu Salzmann, and Prof. Jean-Philippe Thiran for kindly agreeing to read my thesis and for their useful comments.

During the years of my studies I have been very lucky to meet many great friends and bright colleagues. Life is hardly the same after leaving Idiap about three years ago. Thank you my friends, Arjan, Nikos, Cijo, Marco, Tatiana, and many others, who made years in Martigny fun and memorable.

While living in Rome, I met many friends, whom I will deeply miss. From the bottom of my heart I would like to thank Vane for our one-of-a-kind friendship and for teaching me ways of life. Thank you, Ale, for being true friend and for your inspiring easy-going attitude.

Life in Rome would miss out a lot of joy if not for my labmates and friends. Thank you Fabio, Fra, Vale, Arjan, Nizar, Martina, Paolo, Novi, Antonio, Massimiliano, and other Vandals. It will be hard to find such mixture of fun, enthusiasm, and passion for research as among you, guys.

While staying in Vienna, I was fortunate to meet new colleagues and friends. Thank you, Asya, Alex K, Alex Z, Amelie, Mary, Michael, Georg, and others. I will never forget our lunches, table soccer matches, pool games, stimulating talks, and nightly adventures in Genova...

I also would like to thank my old mates, whom I deeply respect and admire. Thank you Slava, Vitia, Tania, Sania, and Alex. Thank you for your wit and cheer, philosophy and passion for art, your friendly critique and ambition, your passion for travel, and for being amazing pen-pals.

Finally, I am very grateful to my family who made these endeavors possible through their unconditional support and love.

Lausanne, 1 September 2017

Abstract

The design and analysis of machine learning algorithms typically considers the problem of learning on a single task, and the nature of learning in such scenario is well explored. On the other hand, very often tasks faced by machine learning systems arrive sequentially, and therefore it is reasonable to ask whether a better approach can be taken than retraining such systems from scratch given newly available data. Indeed, by drawing analogy from human learning, a novel skill could be acquired more easily whenever the learner shares a relevant past experience. In response to this observation, the machine learning community has drawn its attention towards a form of learning known as *transfer learning* – learning a novel task by leveraging upon auxiliary information extracted from previous tasks. Tangible progress has been made in both theory and practice of transfer learning; however, many questions are still to be addressed.

In this thesis we will focus on an efficient type of transfer learning, known as the *Hypothesis Transfer Learning (HTL)*, where auxiliary information is retained in a form of previously induced hypotheses. This is in contrast to the large body of work where one transfers from the data associated with previously encountered tasks. In particular, we theoretically investigate conditions when HTL guarantees improved generalization on a novel task subject to the relevant auxiliary (source) hypotheses. We investigate HTL theoretically by considering three scenarios – HTL through regularized least squares with biased regularization, through convex empirical risk minimization, and through stochastic optimization, which also touches the theory of non-convex transfer learning problems. In addition, we demonstrate the benefits of HTL empirically, by proposing two algorithms tailored for real-life situations with application to visual learning problems – learning a new class in a multi-class classification setting by transferring from known classes, and an efficient greedy HTL algorithm for learning with large number of source hypotheses.

From theoretical point of view this thesis consistently identifies the key quantitative characteristics of relatedness between novel and previous tasks, and explicates them in generalization bounds. These findings corroborate many previous works in the transfer learning literature and provide a theoretical basis for design and analysis of new HTL algorithms.

Key words: transfer learning, domain adaptation, statistical learning theory, stochastic optimization, visual recognition

Résumé

La conception et l'analyse des algorithmes d'apprentissage machine considèrent généralement le problème de l'apprentissage sur une seule tâche et la nature de l'apprentissage dans un tel scénario est bien explorée. D'autre part, très souvent, les tâches auxquelles sont confrontées par les systèmes d'apprentissage par machine arrivent séquentiellement et, par conséquent, il est raisonnable de demander si une meilleure (approche) méthode peut être (prise) effectuée que de recycler ces systèmes à partir de zéro, à partir de nouvelles données disponibles. En effet, en tirant l'analogie de l'apprentissage humain, une nouvelle habileté pourrait être acquise plus facilement chaque fois que l'apprenant partage une expérience passée pertinente. En réponse à cette observation, la communauté de l'apprentissage par machine a attiré son attention vers une forme d'apprentissage connue sous le nom d'apprentissage par transfert, en apprenant une nouvelle tâche en tirant parti des informations auxiliaires extraites des tâches précédentes. Des progrès tangibles ont été réalisés à la fois dans la théorie et dans la pratique de l'apprentissage par transfert; Cependant, de nombreuses questions doivent encore être traitées.

Dans cette thèse, nous nous concentrerons sur un type efficace d'apprentissage par transfert, connu sous le nom de Hypothesis Transfer Learning (HTL), où l'information auxiliaire est retenue sous forme d'hypothèses précédemment induites. Cela contraste avec le grand nombre de travaux où l'on transfère des données associées aux tâches précédemment rencontrées. En particulier, nous étudions théoriquement les conditions lorsque HTL garantit une généralisation améliorée sur une nouvelle tâche soumise aux hypothèses auxiliaires (sources) pertinentes. Nous étudions HTL théoriquement en considérant trois scénarios - HTL à travers des moindres carrés réguliers avec une régularisation biaisée, grâce à une réduction convexe du risque empirique et à une optimisation stochastique, qui touche également la théorie des problèmes d'apprentissage sans transfert convexe. En outre, nous proposons deux algorithmes adaptés aux situations de la vie réelle avec une application aux problèmes d'apprentissage visuel : apprendre une nouvelle classe dans un classement multi-classe en transférant des classes connues et un algorithme HTL gourmand efficace Pour apprendre avec un grand nombre d'hypothèses sources.

Du point de vue théorique, cette thèse identifie systématiquement les principales caractéristiques quantitatives de la relation entre la tâche nouvelle et la précédente, et les explicite dans les limites de généralisation. Ces résultats corroborent de nombreux travaux antérieurs dans la littérature d'apprentissage par transfert et fournissent une base théorique pour la conception et l'analyse de nouveaux algorithmes HTL.

Mots clefs : transfert d'apprentissage, adaptation de domaine, théorie de l'apprentissage statistique, optimisation stochastique, reconnaissance visuelle

Contents

Acknowledgments	i
Abstract (English)	iii
List of figures	xi
List of tables	xiii
1 Introduction	1
1.1 Contributions and Organization	2
2 Definitions and Background	3
2.1 Basic notions	3
2.2 PAC learning	5
2.3 Learning and algorithmic stability	7
I Theory	11
3 Hypothesis Transfer Learning through Regularized Least Squares	13
3.1 Overview	13
3.2 Hypothesis Transfer Learning Problem	14
3.3 Related Work	15
3.4 Hypothesis Transfer Learning through Regularized Least Squares	16
3.4.1 Biased Regularized Least Squares	17
3.5 Analysis by Hypothesis Stability	18
3.5.1 Implications	19
3.6 Conclusion	20
4 Hypothesis Transfer Learning through Empirical Risk Minimization	22
4.1 Overview	22
4.2 Related Work	24
4.3 Transferring from Auxiliary Hypotheses	25
4.4 Main Results	26
4.4.1 Exponential Generalization Bounds for On-average Stable Algorithms	26
4.4.2 Bounds for Hypothesis Transfer Learning through Regularized Empirical Risk Minimization (ERM)	27
4.4.3 Implications	28

Contents

4.4.4	Comparison to Theories of Domain Adaptation and Transfer Learning	30
4.5	Conclusion	31
5	Hypothesis Transfer Learning through Stochastic Optimization	33
5.1	Overview	33
5.2	Related Work	34
5.3	Stability of Stochastic Gradient Descent	35
5.3.1	Uniform Stability and Generalization	35
5.4	Data-dependent Stability Bounds for SGD	37
5.5	Main Results	37
5.5.1	Convex Losses	38
5.5.2	Non-convex Losses	39
5.5.3	Application to Transfer Learning	41
5.6	Conclusion	42
II	Algorithms	43
6	Greedy Algorithms for Hypothesis Transfer Learning	45
6.1	Overview	45
6.2	Related Work	46
6.3	Transfer Learning through Subset Selection	48
6.4	Greedy Algorithm for k -Source Selection	49
6.5	Experiments	53
6.5.1	Datasets and Features	53
6.5.2	Baselines	54
6.5.3	Results	54
6.5.4	Approximated GreedyTL	56
6.5.5	Selected Source Analysis	57
6.6	Conclusion	58
7	Class-incremental Hypothesis Transfer Learning	61
7.1	Introduction	61
7.2	Related Work	63
7.3	Multiclass Incremental Transfer Learning	64
7.3.1	Multiclass Regularized Least Squares	64
7.3.2	MULTIPLE Algorithm	65
7.3.3	Self-tuning of Transfer Parameters	66
7.4	Experiments	67
7.4.1	Data setup	69
7.4.2	Algorithmic setup	69
7.4.3	Evaluation results	71
7.5	Discussion and Conclusions	73
8	Conclusions and Future Directions	74

A Proofs from Chapter 3	77
A.1 Proof of Theorem 7	77
A.1.1 General Statements	77
A.1.2 Perturbation Bounds	78
A.1.3 Bounding M and $\mathbb{E}_S[\ell(A_{S \setminus i}, z_i)]$	80
A.1.4 Hypothesis Stability γ and Generalization Bound	81
B Proofs from Chapter 4	84
C Proofs from Chapter 5	90
C.1 Preliminaries	91
C.2 Convex Losses	94
C.3 Non-convex Losses	97
D Proofs from Chapter 6	105
E Appendix for Chapter 7	110
E.1 Closed-form LOO prediction in Multiclass RLS	110
Bibliography	122
Curriculum Vitae	123

List of Figures

5.1	Empirical tightness of data-dependent and uniform generalization bounds evaluated by training a convolutional neural network.	41
6.1	Performance on Caltech-256, subsets of Imagenet (1000 classes) and SUN09 (819 classes). Averaged class-balanced accuracies in the leave-one-class-out setting.	55
6.2	Baselines and number of additional noise dimensions sampled from a standard distribution. Averaged class-balanced recognition accuracies in the leave-one-class-out setting.	56
6.3	Comparison of the approximated GreedyTL: GreedyTL-59 to GreedyTL with exhaustive search and most competitive baselines on three largest datasets considered in our experiments.	57
6.4	Semantic transferrability matrix for GreedyTL evaluated on Imagenet (DECAF7 features). Columns correspond to targets and rows to sources. Stronger color intensity means larger source weight. 6.4a corresponds to learning from 2 positive and 10 negative examples, while 6.4b, with 10 positive.	58
6.5	Semantic transferrability matrix for RLS (src+feat) evaluated on Imagenet (DECAF7 features).	59
6.6	GreedyTL evaluated on Imagenet (DECAF7 features): a closer look at some strongly related sources and targets.	59
6.7	Semantic transferrability matrix for the approximated GreedyTL evaluated on Imagenet (DECAF7 features).	60
7.1	Binary (left) versus $K \rightarrow K + 1$ transfer learning (right). In both cases, transfer learning implies that the target class is learned close to where informative sources models are. This is likely to affect negatively performance in the $K \rightarrow K + 1$ case, where one aims for optimal accuracy on the sources and target classes simultaneously.	62
7.2	Experimental results for $K + 1 = 5$, Caltech-256. From left to right, columns report results for the unrelated, mixed and related settings. Top row: no transfer baselines, linear case. Middle row: transfer learning baselines, linear case. Bottom row: transfer and competitive no transfer baselines, average of Radial Basis Function (RBF) kernels over all features. Stars represent statistical significance of MULTiclass Transfer Incremental LEarning (MULTIpLE) over MultiKT-OVA, $p < 0.05$	70
7.3	Results for $K + 1 = 20$, AwA, transfer and competitive no transfer baselines, average of RBF kernels, all features. Left to right: unrelated, mixed and related settings. Stars represent statistical significance of MULTIpLE over MultiKT-OVA.	71

List of Figures

7.4 Results for $K + 1 = 20$, A_{wA} , unrelated: accuracy over the K sources (left) and over the
+1 target (right). 73

List of Tables

6.1 Training time in seconds for transferring to a single target class. Results are averaged over 10 splits.	58
--	----

Acronyms

DA Domain Adaptation

ERM Empirical Risk Minimization

FR Forward Regression

GLS General Learning Setting

HTL Hypothesis Transfer Learning

LOO Leave-One-Out

LSSVM Least-Squares Support Vector Machine

MKL Multiple Kernel Learning

MULTIPLE MULTiclass Transfer Incremental LEarning

OVA One-Versus-All

PAC Probably Approximately Correct

PSD Positive Semi-Definite

RBF Radial Basis Function

RKHS Reproducing kernel Hilbert space

RLS Regularized Least Squares

SGD Stochastic Gradient Descent

SVM Support Vector Machine

TL Transfer Learning

UC Uniform Convergence

R-ERM-HTL Regularized ERM for Transferring from Auxiliary Hypotheses

Notation

$[n]$	the set $\{1, \dots, n\}$ for $n \in \mathbb{N}$
$\mathbf{x}, \mathbf{v}, \mathbf{u}$	column vectors
\mathbf{A}, \mathbf{M}	matrices, e.g. $\mathbf{A} = [\mathbf{a}_1, \dots, \mathbf{a}_d]$
$\ \mathbf{x}\ _p$	$= \left(\sum_{i=1}^d x_i ^p \right)^{\frac{1}{p}}$, Lp -norm of \mathbf{x}
$\ \mathbf{A}\ _2$	$= \sup_{\ \mathbf{u}\ _2=1} \ \mathbf{A}\mathbf{u}\ _2$, spectral norm of \mathbf{A}
\mathcal{X}	Input (instance) space
\mathcal{Y}	Output (label) space
\mathcal{Z}	Example space, $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$
\mathcal{D}	Unknown probability distribution over \mathcal{Z}
m	Training set size
S	$= \{z_i\}_{i=1}^m$, a training set drawn iid from \mathcal{D}^m
\mathcal{H}	Hypothesis space, e.g. $\mathcal{H} = \mathcal{Y}^{\mathcal{X}}$ or $\mathcal{H} \subseteq \mathbb{R}^d$
ℓ	Non-negative loss function $\mathcal{H} \times \mathcal{Z} \mapsto \mathbb{R}_+$
$\ h\ _\infty$	$= \sup_{\mathbf{x} \in \mathcal{X}} h(\mathbf{x})$
$\ \ell\ _\infty$	$= \sup_{\mathbf{w} \in \mathcal{H}, z \in \mathcal{Z}} \ell(\mathbf{w}, z)$
$R_{\mathcal{D}}(h)$	$= \mathbb{E}_{z \sim \mathcal{D}} [\ell(h, z)]$, risk
$\widehat{R}_S(h)$	$= m^{-1} \sum_{i=1}^m \ell(h, z_i)$, empirical risk
$R_{\mathcal{D}}^*(\mathcal{H})$	$= \inf_{h \in \mathcal{H}} R_{\mathcal{D}}(h)$, risk of the best-in-the-class
$T_\alpha(x)$	$= \max\{-\alpha, \min\{\alpha, x\}\}$, α -truncation
$U(\{1, \dots, n\})$	uniform distribution over $1, \dots, n$
$f = \mathcal{O}(g)$	$\exists x_0, \alpha \in \mathbb{R}_+$, such that $\forall x > x_0, f(x) \leq \alpha g(x)$
$f = \widetilde{\mathcal{O}}(g)$	$\exists k \in \mathbb{N}$ such that $f = \mathcal{O}(\log^k(g(x))g(x))$
$\nabla^2 f(\mathbf{x})$	Hessian matrix of a differentiable multi-variate function f
$\text{supp}(\mathbf{x})$	$= \{i \in [d]: x_i \neq 0\}$
$\ \mathbf{x}\ _0$	$= \text{supp}(\mathbf{x}) $

1 Introduction

The field of machine learning has undergone remarkable advancements in recent years, getting better at addressing rich and highly structured problems, in domains such as computer vision [72, 60], speech recognition [119], machine translation [143], and reinforcement learning [98]. From the algorithmic point of view this is largely a consequence of our increasing capacity to train effective models of high complexity, such as in deep learning. Yet, these qualitative gains usually come at the high price of a tremendous amount of annotated data required to obtain a model of high effectiveness.

This raises a question attractive from both theoretical and practical point of view: how to reduce the sample complexity of an algorithm by exploiting some form of a non-trivial prior knowledge? In the machine learning literature this direction is collectively known as *transfer learning*. Of course one could argue that it is always possible to simply add more data while increasing the capacity of the model, so why should we bother? There are few counter-arguments to this criticism. First, some problems might come with a small amount of annotated data, thus impeding the use of data-hungry learning, such as deep learning. Indeed, in many applied areas, such as in visual detection, transfer learning is already used in the form of fine-tuning [51] and extraction of feature representations from intermediate layers of neural networks [64]. Second, transfer learning can facilitate training of models with even higher accuracy than those without, at the matching or lower computational cost. Obviously, in such a scenario one can only expect improvements from additional training data. An example is class-incremental transfer learning, where new classes are incorporated into the recognition system, yet every new class could be learned with fewer examples due to some shared similarities with previously observed ones. Finally, it is also conceivable that sample complexity of some learning problems is so high, that practically (w.r.t. available computational resources) we can achieve an acceptable degree of accuracy only through transfer learning. An example of such problem appears in reinforcement learning, where exploration of the vast state-action spaces is intractable, yet due to prior knowledge a successful agent can still be trained. For example, one of the few crucial components in a famous AlphaGo program [126] was based on a neural network pre-trained on a large dataset of human-played games, a form of transfer learning.

In this thesis we focus on the *Hypothesis Transfer Learning (HTL)*, a type of efficient transfer learning where information about previously encountered tasks is retained in the form of pre-trained models, or *source hypotheses* [76, 77, 80]. This is in contrast to many previous approaches in transfer learning that assume access to the data from another tasks. The critical advantage of HTL is in the computational

scalability, which from the transfer learning point of view is constrained only by the amount of source hypotheses and by their computational complexity.

1.1 Contributions and Organization

A large part of this thesis concerns the development of an intuitive and practical explanation for established transfer learning algorithms. The first part of this thesis lays out the theoretical foundations of Hypothesis Transfer Learning (HTL) in two learning settings – through Empirical Risk Minimization (ERM) and through stochastic optimization. In Chapter 3 we consider a generalization of a basic yet powerful approach to HTL known as the *biased regularization*, related to the Bayesian transfer learning. In particular, we analyze the Regularized Least Squares (RLS) as a hypothesis transfer learning algorithm, and show that its generalization ability critically depends on the quality of the source hypothesis, that is a form of prior knowledge. In this chapter we identify the key quantitative characteristic of relatedness between novel and previous tasks – the expected loss of the source hypothesis on the novel task, and develop generalization bounds explicating this quantity. This supports theoretically many previous works in the transfer learning literature. In Chapter 4 we further generalize arguments of Chapter 3 and show that the same message holds for ERM approach with respect to any strongly-convex and smooth loss function, with high probability. This chapter also goes beyond generalization analysis and shows that the quality of the source knowledge can accelerate the convergence of the solution to the optimal one within the given set of predictors. Next, in Chapter 5 we consider a different approach to HTL, namely by learning through stochastic optimization. We prove novel data-dependent generalization bounds for Stochastic Gradient Descent (SGD), a popular optimization algorithm used to train deep neural networks and in large-scale learning in general. These bounds allow us to study the generalization ability of HTL through SGD on both convex and non-convex smooth problems. Importantly, this analysis extends the arguments and messages of Chapters 3 and 4 to non-convex problems, such as deep learning. On non-convex problems, in addition to previously established measures of source quality, this analysis identifies the expected curvature at the initialization point as another characteristic that governs success of the transfer learning.

On a more practical side, backed up by theoretical results, the second part of the thesis presents HTL algorithms for binary classification with applications in computer vision. First, in Chapter 6 we present algorithms that address transfer learning with a large number of source hypotheses where the goal is to pick a subset that improves performance on the novel task. Concretely, we propose greedy algorithms for picking such a subset, and in particular a randomized variant with computational complexity independent from the number of source hypotheses. We also show the potential of these algorithms theoretically and experimentally. Generalization bounds corroborate these results, demonstrating that under reasonable assumptions on the source hypotheses these algorithms are able to learn effectively with very limited data. We also investigate HTL beyond the binary classification setting. In Chapter 7 we propose a multiclass classification scenario, where a novel class is learned from few examples by transferring from previously observed classes. This is particularly relevant in lifelong learning, where tasks, e.g. visual categories, faced by the system arrive sequentially. Here, the main algorithmic idea is based upon the biased regularization investigated theoretically in Chapter 3. Again, we demonstrate the transfer learning potential of an algorithm on a visual recognition dataset. Finally, we conclude and present future directions in Chapter 8.

2 Definitions and Background

The primary goal of this thesis is to provide theoretical foundations for the Hypothesis Transfer Learning (HTL), a successful framework that enables machine learning algorithms to learn from fewer examples on a novel task by leveraging upon auxiliary hypotheses. To accomplish this goal we use tools from statistical learning theory and compare our results to existing theories of learning in a standard non-transfer setting. This chapter introduces necessary definitions, notions, and tools to comprehend the following material. Next, we introduce the learning setting and standard bounds on the performance of learning algorithms. In this thesis we largely follow a constructive theoretical analysis, that is we analyze concrete algorithms and formulations, which is in contrast to the usual uniform convergence argument [141] prevalent in the statistical learning literature. We finalize this chapter by introducing required concepts necessary for this type of analysis.

2.1 Basic notions

Suppose that we have been tasked to design a visual recognition module for a self-driving car that given an image from the camera could tell whether there is a pedestrian in sight. One could try to manually write down the set of all possible visual features characterizing an object in question and try to recognize an object by detecting them. However, due to high natural visual variation this approach would be brittle to unanticipated conditions and most likely would fail. Alternatively one could try to design an algorithm which given a large collection of images under a variety of conditions, with an object and without one, could *learn* these discriminative characteristics. Thus, the goal of such a *supervised machine learning* algorithm is, given a set of examples, to come up with a hypothesis (a function) able to give correct predictions on yet unseen instances. This is of course only possible by making appropriate assumptions on the environment generating these examples and on the algorithm itself. In this thesis we follow the framework of a statistical learning formalizing such problems, and next we briefly summarize its notions.

In the following we will indicate the space of examples by \mathcal{Z} and its member by $z \in \mathcal{Z}$. For instance, in a supervised setting $\mathcal{Z} = \mathcal{X} \times \mathcal{Y}$, such that \mathcal{X} is the input and \mathcal{Y} is the output space of a learning problem. In our object recognition case, \mathcal{X} would stand for the set of all possible images of a certain size and \mathcal{Y} would describe all possible annotations, e.g. $\mathcal{Y} = \{\text{pedestrian, no pedestrian}\}$. In what follows, without loss of generality we assume that the input space \mathcal{X} is a unit-radius $L2$ ball. In addition

Chapter 2. Definitions and Background

we introduce a hypothesis class \mathcal{H} , a set of all admissible hypotheses that the algorithm is allowed to generate. Thus, formally we define a learning algorithm as a map

$$A : \bigcup_{m=1}^{\infty} \mathcal{Z}^m \mapsto \mathcal{H} \quad (2.1)$$

and for brevity we will use the notation $A_S = A(S)$, where S is a training set. To measure the accuracy of a learning algorithm, we have a non-negative *loss function* $\ell : \mathcal{H} \times \mathcal{Z} \mapsto \mathbb{R}_+$, which measures the cost incurred by predicting with some hypothesis from \mathcal{H} on an example from \mathcal{Z} . We will make use of the following properties of the loss function as necessary.

Definition 1 (*L-Lipschitz ℓ*). A loss function ℓ is *L-Lipschitz* if $\|\nabla \ell(\mathbf{w}, z)\| \leq L, \forall \mathbf{w} \in \mathcal{H}$ and $\forall z \in \mathcal{Z}$. Note that this also implies that

$$|\ell(\mathbf{w}, z) - \ell(\mathbf{v}, z)| \leq L \|\mathbf{w} - \mathbf{v}\|.$$

Definition 2 (*β -smooth ℓ*). A loss function is *β -smooth* if $\forall \mathbf{w}, \mathbf{v} \in \mathcal{H}$ and $\forall z \in \mathcal{Z}$,

$$\|\nabla \ell(\mathbf{w}, z) - \nabla \ell(\mathbf{v}, z)\| \leq \beta \|\mathbf{w} - \mathbf{v}\|,$$

which also implies

$$\ell(\mathbf{w}, z) - \ell(\mathbf{v}, z) \leq \nabla \ell(\mathbf{v}, z)^\top (\mathbf{w} - \mathbf{v}) + \frac{\beta}{2} \|\mathbf{w} - \mathbf{v}\|^2.$$

Definition 3 (*ρ -Lipschitz Hessian of ℓ*). A loss function f has a *ρ -Lipschitz Hessian* if $\forall \mathbf{w}, \mathbf{v} \in \mathcal{H}$ and $\forall z \in \mathcal{Z}$,

$$\|\nabla^2 \ell(\mathbf{w}, z) - \nabla^2 \ell(\mathbf{v}, z)\|_2 \leq \rho \|\mathbf{w} - \mathbf{v}\|.$$

The last condition is occasionally used in analysis of optimization algorithms and holds whenever ℓ has a bounded third derivative [48].

Intuitively, the prediction should only be possible whenever examples used for training and yet unseen ones share some regularities. The framework of statistical learning captures these regularities by assuming that both training and unseen, or testing data are drawn independently from the same unknown distribution \mathcal{D} over the example space \mathcal{Z} . Then, formally we will denote the training set as $S = \{z_i\}_{i=1}^m \stackrel{\text{iid}}{\sim} \mathcal{D}^m$. The distribution \mathcal{D} plays a central role in the statistical learning theory, and in some contexts it is also referred to as the *task*. Ultimately we are interested in the performance of a learning algorithm on the testing data sampled from the same task as the training data. This performance is captured by the expected loss or the *risk* of hypothesis h , with respect to \mathcal{D} ,

$$R_{\mathcal{D}}(h) = \mathbb{E}_{z \sim \mathcal{D}} [\ell(h, z)],$$

and typically we will simply indicate $R(h) = R_{\mathcal{D}}(h)$ whenever \mathcal{D} is clear from the context. Naturally, under realistic conditions we cannot observe the risk, and instead we can compute its empirical

counterpart measured on the training set, or the *empirical risk* defined as

$$\widehat{R}_S(h) = \frac{1}{m} \sum_{i=1}^m \ell(h, z_i).$$

2.2 PAC learning

The risk of an algorithm, $R(A_S)$, is a random variable with randomness arising due to the stochastic origin of the training set (assuming that A is deterministic). Therefore, risk cannot be computed directly from data, but rather can be estimated using probabilistic bounds. One of the major topics of study in the statistical learning theory is stating such bounds on the *generalization error*, defined as the difference between the risk and the empirical risk of a hypothesis generated by an algorithm A given a training set S , that is

$$R_{\mathcal{D}}(A_S) - \widehat{R}_S(A_S). \tag{2.2}$$

On an intuitive level, if we can describe generalization error in terms of quantities controlled by the algorithm and supplied by the user, such as the training set S and the hypothesis class \mathcal{H} , then we can characterize how close the empirical risk will be to the actual performance on unseen data. Thus, whenever generalization error is small or decreases with the size of the training set, we say that the learning algorithm *generalizes*. This is typically sufficient for design of learning algorithms because these bounds point out the ingredients that control generalization. In this thesis we will primarily focus on the generalization bounds.

However, from a theoretical point of view, we sometimes desire to know how optimal the algorithm is, where by optimality we mean the ability of an algorithm to recover the best hypothesis in a given class of functions. To capture this particular notion of optimality¹ we define the risk of the *best-in-the-class* as

$$R_{\mathcal{D}}^*(\mathcal{H}) = \inf_{h \in \mathcal{H}} R_{\mathcal{D}}(h),$$

and the optimality of an algorithm is then represented by an *estimation error*

$$R_{\mathcal{D}}(A_S) - R_{\mathcal{D}}^*(\mathcal{H}). \tag{2.3}$$

An estimation error is one of the central notions in statistical learning theory since it formally characterizes *learnability* in a Probably Approximately Correct (PAC) model of learning proposed by Valiant. Here we present its slight extension due to [59].

Definition 4 (Agnostic PAC learnability with General Loss Functions). *A hypothesis class \mathcal{H} is agnostic PAC learnable w.r.t. example space \mathcal{Z} and a loss function $\ell : \mathcal{H} \times \mathcal{Z} \mapsto \mathbb{R}_+$ if there exists a function $m_{\mathcal{H}} : (0, 1)^2 \mapsto \mathbb{N}$ and a learning algorithm A such that for every $\epsilon, \delta \in (0, 1)$ and for every distribution \mathcal{D} over \mathcal{Z} and every $m \geq m_{\mathcal{H}}(\epsilon, \delta)$, with probability at least $1 - \delta$ over a training set $S \stackrel{iid}{\sim} \mathcal{D}^m$,*

$$R_{\mathcal{D}}(A_S) - R_{\mathcal{D}}^*(\mathcal{H}) \leq \epsilon.$$

¹In this thesis we do not cover *Bayes optimality* and approximation error.

Chapter 2. Definitions and Background

In other words, agnostic PAC learnability formally captures the computational feasibility to address any statistical learning problem by a given class of functions up to certain precision and probability. Another important concept of statistical learning theory, related to PAC learnability, is the uniform convergence.

Definition 5 (Uniform Convergence). *A hypothesis class \mathcal{H} has a uniform convergence property w.r.t. example space \mathcal{X} and a loss function $\ell : \mathcal{H} \times \mathcal{X} \mapsto \mathbb{R}_+$ if there exists a function $m_{\mathcal{H}}^{UC} : (0, 1)^2 \mapsto \mathbb{N}$ such that for every $\epsilon, \delta \in (0, 1)$ and for every distribution \mathcal{D} over \mathcal{X} , and every $m \geq m_{\mathcal{H}}(\epsilon, \delta)$, with probability at least $1 - \delta$ over a training set $S \stackrel{iid}{\sim} \mathcal{D}^m$,*

$$\sup_{h \in \mathcal{H}} |R_{\mathcal{D}}(h) - \widehat{R}_S(h)| \leq \epsilon .$$

Uniform Convergence (UC) goes beyond claims about generalization ability of concrete algorithms and enables analysis of a generalization error for the entire hypothesis class \mathcal{H} (that is for *any* hypothesis in \mathcal{H}). An important property of UC is that it is known to imply PAC learnability, see e.g. [123, Corollary 4.4]. Thus, if one could state a generalization bound for a class \mathcal{H} following a UC argument, and then design an algorithm that outputs hypotheses in restriction to that class, then this would imply generalization bound for the algorithm.

Proving the generalization bound for a single fixed hypothesis is straightforward through the standard concentration argument, e.g. using Chernoff bound or a similar one. If we consider more than one hypothesis, forming a finite class, then we could extend this argument by applying union bound, and now our generalization bound would depend on the cardinality of the class. However, since in UC setting we are interested in generalization w.r.t. all hypotheses in a potentially infinitely uncountable class, this approach needs extension towards a more sophisticated way of capturing the capacity of the class.

One popular way to prove UC bounds is through Rademacher complexity analysis, which can be used to prove bounds for both parametric and non-parametric hypothesis classes, e.g. when the class is a subset of a Reproducing kernel Hilbert space (RKHS). This makes its applicability more general than classical combinatorial class capacity measures such as VC-dimension.

Definition 6 (Rademacher complexity). *Let \mathcal{H} be a class of functions mapping from \mathcal{X} to \mathcal{Y} and $S \stackrel{iid}{\sim} \mathcal{D}^m$. Then Rademacher complexity is defined as*

$$\mathfrak{R}_m(\mathcal{H}) = \mathbb{E}_{S, \boldsymbol{\sigma}} \left[\sup_{h \in \mathcal{H}} \frac{1}{m} \sum_{i=1}^m \sigma_i h(\mathbf{x}_i) \right], \quad (2.4)$$

where $\boldsymbol{\sigma} = [\sigma_1, \dots, \sigma_m]^\top$, with $\sigma_i \sim U(\{-1, +1\})$.

Then one can show [70, 6] the following basic probabilistic UC bound on the generalization error.

Theorem 1. *Assume that the loss function is L -Lipschitz and satisfies $\|\ell\|_\infty \leq 1$, and that $\|h\|_\infty < \infty$. Then with probability at least $1 - e^{-2\eta}$ over a training set $S \stackrel{iid}{\sim} \mathcal{D}^m$, for every $h \in \mathcal{H}$ we have*

$$R(h) - \widehat{R}_S(h) \leq 2L\mathfrak{R}_m(\mathcal{H}) + \sqrt{\frac{\eta}{m}} . \quad (2.5)$$

2.3. Learning and algorithmic stability

Then stating a generalization bound boils down to actually analyzing the Rademacher complexity of that class. The following lemma states the bound on the Rademacher complexity when \mathcal{H} is an L_2 ball.

Lemma 1 (Lemma 22 in [6], Theorem 1 in [66]). *Let \mathcal{X} be a unit L_2 ball, and let the hypothesis class be*

$$\mathcal{H}_\tau = \{ \mathbf{x} \mapsto \langle \mathbf{w}, \mathbf{x} \rangle : \|\mathbf{w}\|_2 \leq \tau \}.$$

Then Rademacher complexity obeys

$$\mathfrak{R}_m(\mathcal{H}_\tau) \leq \frac{\tau}{\sqrt{m}}.$$

Theorem 1 in combination with Lemma 1 can be used to state a generalization bound for an algorithm

$$A_S = \operatorname{argmin}_{\|\mathbf{w}\|_2^2 \leq \tau^2} \widehat{R}_S(\mathbf{w}), \quad (2.6)$$

which is a special case of the *regularized ERM*. For example, a particular instance of this algorithm is a well-known Support Vector Machine (SVM).

The bound of Theorem 1 can further be improved to the *optimistic* one, which exhibits fast $\mathcal{O}(1/m)$ rate of convergence rather than typical $\mathcal{O}(1/\sqrt{m})$ subject to some conditions. One example of such bound presented next guarantees faster generalization subject to the vanishing empirical risk. Practically speaking, whenever a learning algorithm is initialized close to a minimizer of an empirical risk or approached it sufficiently close, the learning switches to the fast rate of convergence.

Theorem 2 (Theorem 1 in [129]). *Let the non-negative loss function be β -smooth and let $\|\ell\|_\infty \leq 1$. Then with high probability over a training set $S \stackrel{iid}{\sim} \mathcal{D}^m$, for every $h \in \mathcal{H}$ we have*

$$R(h) - \widehat{R}_S(h) = \tilde{\mathcal{O}} \left(\tau \sqrt{\frac{\beta \widehat{R}_S(h)}{m}} + \frac{1 + \beta\tau}{m} \right). \quad (2.7)$$

2.3 Learning and algorithmic stability

As discussed in the previous section, statistical learning theory usually studies probabilistic bounds on the generalization error that hold for all hypotheses in a given class, that is for any distribution \mathcal{D} with probability at least $1 - \delta$ for $\delta \in (0, 1)$,

$$\sup_{h \in \mathcal{H}} |R(h) - \widehat{R}_S(h)| \leq F(1/\delta, m, \text{size}(\mathcal{H})), \quad (2.8)$$

where F is a polynomial function of $1/\delta$, the number of training examples m , and some notion of “size” of a class, such as VC-dimension or Rademacher complexity. These bounds are independent from the choice of a learning algorithm.

However, very often we design a concrete learning algorithm and only then analyze its generalization ability. It is also possible that the class \mathcal{H} is so large that our algorithm explores only a small subset of it, and therefore UC type of analysis can be too general and would not necessarily lead to good

Chapter 2. Definitions and Background

estimates. Therefore in many situations it would be sufficient to claim that for any distribution \mathcal{D} with probability at least $1 - \delta$,

$$|R(A_S) - \widehat{R}_S(A_S)| \leq F(1/\delta, m, A), \quad (2.9)$$

where F is a function polynomial in $1/\delta$, m , and some property of A . Sometimes it is even sufficient to state a deterministic generalization bound

$$\mathbb{E}_S [R(A_S) - \widehat{R}_S(A_S)] \leq F(m, \mathcal{D}, A). \quad (2.10)$$

In contrast to the PAC learnability, this type of *constructive* analysis is captured by the General Learning Setting (GLS) due to Vapnik, and the following notion of learnability [124].

Definition 7 (Learnability in General Learning Setting). *A hypothesis class \mathcal{H} is learnable w.r.t. example space \mathcal{Z} and a loss function $\ell : \mathcal{H} \times \mathcal{Z} \mapsto \mathbb{R}_+$ if there exists a learning rule A and a monotonically decreasing sequence ϵ_m^{cons} , such that $\epsilon_m^{\text{cons}} \rightarrow 0$ as $m \rightarrow \infty$, and*

$$\forall \mathcal{D}, \quad \mathbb{E}_{S \sim \mathcal{D}^m} [R(A_S) - R_{\mathcal{D}}^*(\mathcal{H})] \leq \epsilon_m^{\text{cons}}. \quad (2.11)$$

Shalev-Shwartz *et al.* [124] argued that GLS includes most of statistical learning problems, however for some of them UC actually does not hold. Instead they identified a different well-known property of learning algorithms known as the *uniform stability* [17] as necessary and sufficient condition for learnability in GLS. Algorithmic stability will be instrumental in constructive analysis of algorithms in this thesis and next we introduce the necessary background.

On an intuitive level, a learning algorithm is said to be *stable* whenever a small perturbation in the training set does not affect its outcome too much. Of course, there is a number of ways to formalize the perturbation and the extent of the change in the outcome, and we will discuss some of them below. The most important consequence of a stable algorithm is that it *generalizes* from the training set to the unseen data sampled from the same distribution. In other words, the generalization error of an algorithm is controlled by the quantity that captures how stable the algorithm is. So, to observe good performance, or a decreasing risk, we must have a stable algorithm *and* decreasing empirical risk (training error), which usually comes by design of the algorithm.

First we consider the following (weak) notion of stability which is known to imply generalization in expectation whenever an algorithm is insensitive to re-sampling of one point in the training set.

Definition 8 (On-average stability). *Let $i \stackrel{\text{iid}}{\sim} U([m])$. Then, a deterministic algorithm A is ϵ_m -on-average stable if it is true that*

$$\mathbb{E}_{S, z, i} [\ell(A_{S^{(i)}}, z_i) - \ell(A_S, z_i)] \leq \epsilon_m. \quad (2.12)$$

where $S \stackrel{\text{iid}}{\sim} \mathcal{D}^m$ and $S^{(i)}$ is its copy with i -th example replaced by $z \stackrel{\text{iid}}{\sim} \mathcal{D}$.

Theorem 3 (Theorem 13.2 in [123]). *Let algorithm A be ϵ_m -on-average stable. Then,*

$$\mathbb{E}_S [R(A_S) - \widehat{R}_S(A_S)] \leq \epsilon_m. \quad (2.13)$$

Proof.

$$\mathbb{E}_S [R(A_S) - \widehat{R}_S(A_S)] = \mathbb{E}_{S, z, i} [\ell(A_S, z) - \ell(A_S, z_i)] \quad (2.14)$$

$$\begin{aligned} &= \mathbb{E}_{S, z, i} [\ell(A_{S^{(i)}}, z_i) - \ell(A_S, z_i)] && \text{(Swap } z \text{ and } z_i \text{ since } z, z_i \stackrel{\text{iid}}{\sim} \mathcal{D}.) \\ &\leq \epsilon_m. \end{aligned} \quad (2.15)$$

□

As an instructive example consider again a regularized ERM problem (note that this is equivalent to (2.6) whenever $\mathcal{H} \subseteq \mathbb{R}^d$, for some mapping between λ and τ)

$$A_S^\lambda = \operatorname{argmin}_{\mathbf{w} \in \mathbb{R}^d} \{ \widehat{R}_S(\mathbf{w}) + \lambda \|\mathbf{w}\|^2 \}. \quad (2.16)$$

Then A_S^λ can be shown to be ϵ_m -on-average stable by appealing to the strong convexity and smoothness of the objective function.

Theorem 4 (Corollary 13.7 in [123]). *Assume that the loss function is β -smooth and non-negative. Then algorithm A_S^λ , assuming that $\lambda \geq \frac{2\beta}{m}$, satisfies*

$$\epsilon_m = \frac{48\beta}{m\lambda} \mathbb{E}_S [\widehat{R}_S(A_S)]. \quad (2.17)$$

This immediately implies a generalization bound in expectation due to Theorem 3. Despite that this bound holds in expectation, other forms of generalization bounds, such as high-probability ones, can be derived from the above [124]. The theorem above can also be used to state a bound on the estimation error and thus prove learnability in GLS.

Corollary 1. *Let ℓ be β -smooth and convex w.r.t. hypothesis class \mathcal{H} and example space \mathcal{Z} with $\|\ell\|_\infty \leq 1$. Then setting $\lambda = \sqrt{\frac{150\beta}{9m}}$, we have that for every distribution \mathcal{D} ,*

$$\mathbb{E}_S [R(A_S)] - \min_{\mathbf{w} \in \mathcal{H}} R(\mathbf{w}) \leq \sqrt{\frac{150\beta}{m}}. \quad (2.18)$$

On-average stability discussed above captures sensitivity of an algorithm with respect to a concrete distribution \mathcal{D} . Therefore we can say that this on-average stability is *data-dependent*. The following notion of stability is much more restrictive because it holds uniformly for the choice of any data and characterizes stability as a property of a learning algorithm.

Definition 9 (Uniform stability). *A deterministic algorithm A is ϵ_m^{uni} -uniformly stable if for all datasets $S, S^{(i)} \in \mathcal{Z}^m$ such that S and $S^{(i)}$ differ in the i -th example, we have*

$$\sup_{z \in \mathcal{Z}, i \in [m]} \{ \ell(A_{S^{(i)}}(z)) - \ell(A_S(z)) \} \leq \epsilon_m^{\text{uni}}. \quad (2.19)$$

Although more restrictive than on-average stability, uniform stability is usually easier to work with, because one can completely rely on a geometrical argument and tools from optimization leaving out

Chapter 2. Definitions and Background

probabilistic details. The following theorem implies that uniform stability implies generalization with high probability.

Theorem 5 (Theorem 12 in [17]). *Assume that the loss function satisfies $\|\ell\|_\infty \leq M$. Then with probability at least $1 - e^{-2\eta}$ over a training set $S \stackrel{iid}{\sim} \mathcal{D}^m$, for algorithm A we have that*

$$R(A_S) - \widehat{R}_S(A_S) \leq 2\epsilon_m^{\text{uni}} + (2m\epsilon_m^{\text{uni}} + M)\sqrt{\frac{\eta}{m}}. \quad (2.20)$$

Naturally $\epsilon_m \leq \epsilon_m^{\text{uni}}$, and by Theorem 4 we get that

$$\epsilon_m^{\text{uni}} \leq \frac{48M\beta}{m\lambda}. \quad (2.21)$$

Theory **Part I**

3 Hypothesis Transfer Learning through Regularized Least Squares

The material of this chapter is based on the publication:

I. Kuzborskij and F. Orabona. Stability and Hypothesis Transfer Learning.
In *Proceedings of the International Conference on Machine Learning (ICML)*, 2013.

The doctoral candidate formalized the problem, proved the results, and wrote most of the publication.

3.1 Overview

The standard assumption made during design of supervised machine learning algorithms is to have models trained and tested on samples drawn from the same probability distribution, or *domains*. However, this assumption is often violated in practical applications.

A more general setting is the one in which the marginal distributions associated with training and testing domains are different but related. This is the problem of Domain Adaptation (DA), where a successful scheme typically utilizes large unlabeled samples from both domains to adapt a source hypothesis to the target domain. Previous work has addressed in detail the theory of DA and proposed algorithms that critically depend on optimal weighting parameters given by the theoretical analysis [8, 9, 93, 27]. However, in practice, the learner needs access to sufficient unlabeled samples from both domains to estimate these parameters. Even if unlabeled data are abundant, the estimation of these parameters can be computationally prohibitive in some scenarios. A hypothetical example is a large number of domains involved or, for instance, when one acquires new domains incrementally. Here, keeping unlabeled data from all the domains and re-estimating parameters is a necessity.

To overcome this practical limitation, a new framework has been analyzed by a number of works [45, 145, 103, 94, 136, 78]. In this framework, that we will call Hypothesis Transfer Learning (HTL), unlike DA, only *source hypotheses* are retained from the source domain, but not the source data. The attractive quality of HTL is the fact that it assumes no explicit access to the source domain, nor any knowledge

about the relatedness of the source and target distributions. Although, this setting has been explored empirically with success, a formal theory of HTL is mostly missing. Hence it is unclear how to recover optimal transfer parameters and what properties of the source hypothesis affect generalization.

In this chapter, we take a step towards a theory of HTL. In particular, we analyze the generalization ability of an HTL algorithm stemming from the Regularized Least Squares (RLS) with biased regularization. We assume access to a given number of source hypotheses and a *small* set of training samples from the target domain. Rather than relying on oracle inequalities for tuning the optimal parameters, we use the Leave-One-Out (LOO) risk. The LOO risk is known to have low bias compared to empirical risk or cross-validation [43], thus making it preferable in a small sample regime.

In this chapter we will show that the generalization error of the considered HTL algorithm decreases with the increasing quality of the source hypothesis over the target domain. We do so by employing the notion of *hypothesis stability* [17], a form of on-average stability, and upper bounding the second-order moment of the difference between the expected risk and the LOO risk. In addition, we propose a hypothetical algorithm that can avoid negative transfer in the case of unrelated domains, while in worst case scenario recovering the generalization guarantees of RLS. Finally, from the stability theory point of view, this chapter also touches upon a question raised by [43]: “*Is there a way to incorporate prior knowledge via stability?*”, thus exposing a connection between stability and the Hypothesis Transfer Learning.

The rest of the chapter is organized as follows. We formally state the HTL problem in Section 3.2 and introduce analyzed algorithms in Section 3.4. The main result comes in Section 3.5, particularly in Theorem 7, with implications discussed in Section 3.5.1. The proof of the main result can be found in Appendix A, while related work on DA and HTL is covered in Section 3.3. Finally we draw some conclusions and discuss future work in Section 3.6.

3.2 Hypothesis Transfer Learning Problem

First we formally describe the transfer learning problem we consider in this chapter. Assume that in addition to the training set S we also receive a *source* hypothesis $h^{\text{src}} \in \mathcal{H}^{\text{src}} \subseteq \mathcal{Y}^{\mathcal{X}}$. Then, the aim of an HTL algorithm is to use the source hypothesis h^{src} to improve the performance compared to the supervised learning algorithm that has access only to S . More formally, we define the HTL algorithm as follows

Definition 10 (HTL algorithm). *An HTL algorithm is a map*

$$A^{\text{htl}} : \left(\bigcup_{m=1}^{\infty} \mathcal{Z}^m \right) \times \mathcal{H}^{\text{src}} \mapsto \mathcal{H}. \quad (3.1)$$

The following definition captures the goal of an HTL algorithm A^{htl} more formally.

Definition 11 (Usefulness and Collaboration). *We say that hypothesis $h^{\text{src}} \in \mathcal{H}^{\text{src}}$ is useful [76] for A^{htl} with respect to distribution \mathcal{D} and training set size m , if*

$$\mathbb{E}_{\mathcal{S}} [R_{\mathcal{D}}(A^{\text{htl}}(S, h^{\text{src}}))] < \mathbb{E}_{\mathcal{S}} [R_{\mathcal{D}}(A^{\text{htl}}(S, \mathbf{0}))]. \quad (3.2)$$

In addition, we will say that $h^{\text{src}} \in \mathcal{H}^{\text{src}}$ and a distribution \mathcal{D} collaborate [11] for A^{htl} , w.r.t. training set size m , if

$$\mathbb{E}_S [R_{\mathcal{D}}(A^{\text{htl}}(S, h^{\text{src}}))] < \min \left\{ R_{\mathcal{D}}(A^{\text{htl}}(\emptyset, h^{\text{src}})), \mathbb{E}_S [R_{\mathcal{D}}(A^{\text{htl}}(S, \mathbf{0}))] \right\} .$$

The first notion (usefulness) is satisfied whenever algorithm A_S^{htl} can achieve lower risk by using the source hypothesis. The second one (collaboration) [11] is satisfied whenever simultaneous access to h^{src} and S improves the performance compared to when they are used separately. The failure to satisfy any of these two conditions is usually called the *negative transfer*. Thus, we are only interested in the observable improvement of the generalization error on the target domain. From now on we will indicate $A_S^{\text{htl}} = A^{\text{htl}}(S, h^{\text{src}})$ as for the most part of this chapter we will consider the use of a single source hypothesis h^{src} .

3.3 Related Work

We start by introducing related work from Domain Adaptation (DA), closely related to the HTL problem. Most DA algorithms assume access to the labeled m^{src} -sized training set sampled from the source domain \mathcal{D}^{src} , a sample of m^{unlab} unlabeled examples from both domains (marginal distributions), and occasionally an m -sized training set sampled from the target domain. Typically it is assumed that $m \ll m^{\text{src}}, m \ll m^{\text{unlab}}$.

A milestone work on theoretical analysis of DA is due to Ben-David *et al.* [8], where they considered unsupervised setting. The main result of [8] is a UC-type bound on the difference between risks on the target and source domains. This bound is controlled by two critical terms: the “divergence” term and the minimal-joint-error, defined with respect to some hypothesis class. The divergence term quantifies how different distributions (domains) are, while the second term quantitatively captures the existence of a hypothesis with low error on both domains. The theoretical nature of the divergence term was later further investigated by [9]: the additive divergence term is inevitable unless an algorithm has access to labeled training data from the target domain. A similar theoretical setting was also studied by [93]. Once again, an additive divergence term appears in the bounds. Although related, the HTL problem is not covered by the DA theory because it depends on the properties of the learning algorithm that generates the source hypothesis. More importantly, the source domain is inaccessible. These facts render the mentioned DA bounds unapplicable for analysis of the Hypothesis Transfer Learning.

HTL has been also investigated from Bayesian perspective. Li and Bilmes [86] proposed a PAC-Bayes-type of analysis and derived bounds capturing the relationship between domains by an additive KL-divergence term. They showed that for logistic regression, the divergence term is upper bounded by $\|h - h^{\text{src}}\|^2$, motivating the biased regularization term in logistic regression. Indeed, Bayesian linear regression with h^{src} -mean Gaussian prior over h leads to exact recovery of $\|h - h^{\text{src}}\|^2$ in optimization problem [13]. Results of [86] hint that generative methods like the one in [45] could also be related to biased regularization.

The majority of algorithmic works on DA focus on recovery of transformations that approximately minimize the divergence term. Many of these works have further investigated the use of more powerful transformations by lifting computations into the RKHS [55, 104, 4, 148], thus paving the way for DA

in non-linear learning problems. The found transformations are then used to map the hypothesis generated on the source domain in hope that it would perform well on the target one. At the same time the transformation is typically found on the basis of large unlabeled samples drawn from marginal distributions. Unfortunately many of these works are known to scale badly in m^{unlab} and m^{src} , therefore evaluations are rarely done beyond the proof-of-concept benchmarks. At the same time HTL is only limited by computational complexity of the source hypothesis.

Motivated from this point of view, a number of empirical attempts have tried to justify HTL. An SVM-like algorithm with regularizer $\|h - h^{\text{src}}\|^2$ was proposed by [145] for video concept detection. Orabona *et al.* [103] suggested a parametrized variant, $\|h - \beta h^{\text{src}}\|^2$ -regularized Least-Squares Support Vector Machine (LSSVM), then extended to weight and combine multiple source hypotheses in [136]. Leveraging on this idea, an HTL multiclass formulation explored a class-incremental transfer setting [78]. While some of these methods demonstrated impressive practical potential, their theoretical nature remains unclear.

3.4 Hypothesis Transfer Learning through Regularized Least Squares

First we introduce additional definitions used only throughout this chapter. Without loss of generality, in the following we will assume that $\mathcal{Y}, \mathcal{Y}' = [-B; B]$, where $B \in \mathbb{R}$ and $\|\mathbf{x}\| \leq 1$, $\mathbf{x} \in \mathcal{X} = \mathbb{R}^d$. In addition to the training set S we also defined the LOO training set as $S^{\setminus i} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_{i-1}, y_{i-1}), \dots, (\mathbf{x}_{i+1}, y_{i+1}), (\mathbf{x}_m, y_m)\}$ and hypothesis $h_{S^{\setminus i}}$ is produced by an algorithm A given training set $S^{\setminus i}$. Then, we also define the *LOO risk* as

$$\widehat{R}^{\text{loo}}(A, S) = \frac{1}{m} \sum_{i=1}^m \ell(A_{S^{\setminus i}}, (\mathbf{x}_i, y_i)).$$

We will consider linear algorithms, extended to non-linear ones through the use of kernels. Hence hypothesis $h(\mathbf{x})$ will be expressed as the inner product of a vector \mathbf{w} , learned from the training data, and the sample \mathbf{x} .

We assume, that only the target training set S and source hypothesis h^{src} are given, so that the source training set is not required. The main objective of this analysis is to identify the effect of h^{src} on the generalization properties of A^{htl} . For this reason, we would like to bound the expected risk of A^{htl} with terms depending on the characteristics of h^{src} . In particular, we expect that a smaller risk $R(h^{\text{src}})$ should improve the generalization of A^{htl} , compared to the case when $h^{\text{src}} \equiv 0$.

As said above, we proceed by specializing A^{htl} to a particular class of algorithms, the RLS with biased regularization. This will allow us to arrive at a generalization bound where all the relevant quantities are computable in a closed form.

3.4.1 Biased Regularized Least Squares

The RLS algorithm consists in solving the following optimization problem

$$\min_{\mathbf{w} \in \mathbb{R}^d} \left\{ \frac{1}{m} \sum_{i=1}^m (\mathbf{w}^\top \mathbf{x}_i - y_i)^2 + \lambda \|\mathbf{w}\|^2 \right\}. \quad (3.3)$$

The interest of RLS lies in its strong theoretical guarantees and in the fact that the solution can be expressed in a closed form [116]. As a useful consequence, its LOO prediction function is expressed in closed form as well, allowing a very efficient model selection [21]. It is also possible to arrive at (3.3) from a Bayesian perspective by putting a $\mathbf{0}$ -mean Gaussian prior over the parameters of a linear regression model [13]. Note that the same formulation can be used for both classification and regression problems [116, 132].

Assuming that the source hypothesis $h^{\text{src}}(\mathbf{x})$ is expressed as $\mathbf{x}^\top \mathbf{w}_0$, and \mathbf{w}_0 belongs to the same space of \mathbf{w} , [103] proposed the use of a biased regularization to solve hypothesis transfer learning problems efficiently. More formally they defined the following algorithm.

Algorithm 1. *The Hypothesis Transfer Learning Algorithm based on Regularized Least Squares generates a hypothesis $A_S^{\text{htl-bias}}(\mathbf{x}) = \mathbf{x}^\top \mathbf{w}_S$, where*

$$\mathbf{w}_S = \operatorname{argmin}_{\mathbf{w} \in \mathbb{R}^d} \left\{ \frac{1}{m} \sum_{i=1}^m (\mathbf{w}^\top \mathbf{x}_i - y_i)^2 + \lambda \|\mathbf{w} - \mathbf{w}_0\|^2 \right\}. \quad (3.4)$$

Analogously, one can see the formulation of Algorithm 1 as a Bayesian linear regression with \mathbf{w}_0 -mean Gaussian prior distribution. The solution of Algorithm 1 can be expressed in closed form, in fact from the first order optimality condition we get

$$\begin{aligned} \mathbf{w}_S &= \operatorname{argmin}_{\mathbf{w} \in \mathbb{R}^d} \left\{ \frac{1}{m} \|\mathbf{X}^\top \mathbf{w} - \mathbf{y}\|^2 + \lambda \|\mathbf{w} - \mathbf{w}_0\|^2 \right\} \\ &\Rightarrow \mathbf{X}(\mathbf{X}^\top \mathbf{w}_S - \mathbf{y}) + m\lambda(\mathbf{w}_S - \mathbf{w}_0) = 0 \\ &\Rightarrow \mathbf{X}(\mathbf{X}^\top \hat{\mathbf{w}}_S + \mathbf{X}^\top \mathbf{w}_0 - \mathbf{y}) + m\lambda \hat{\mathbf{w}}_S = 0 \\ &\Rightarrow (\mathbf{X}\mathbf{X}^\top + m\lambda\mathbf{I}) \hat{\mathbf{w}}_S = \mathbf{X}\mathbf{y} - \mathbf{X}\mathbf{X}^\top \mathbf{w}_0 \\ &\Rightarrow \hat{\mathbf{w}}_S = (\mathbf{X}\mathbf{X}^\top + m\lambda\mathbf{I})^{-1} \mathbf{X}(\mathbf{y} - \mathbf{X}^\top \mathbf{w}_0) \\ &\Rightarrow \hat{\mathbf{w}}_S = \mathbf{X}(\mathbf{X}^\top \mathbf{X} + m\lambda\mathbf{I})^{-1} (\mathbf{y} - \mathbf{X}^\top \mathbf{w}_0) \end{aligned} \quad (3.5)$$

where in (3.5) we used $\hat{\mathbf{w}}_S = \mathbf{w}_S - \mathbf{w}_0$ and in the last step we used the identity $(\mathbf{X}\mathbf{X}^\top + m\lambda\mathbf{I})^{-1} \mathbf{X} = \mathbf{X}(\mathbf{X}^\top \mathbf{X} + m\lambda\mathbf{I})^{-1}$ to express the solution in dual variables. So, the solution to the problem is given by $\mathbf{w}_S = \mathbf{X}(\mathbf{X}^\top \mathbf{X} + m\lambda\mathbf{I})^{-1} (\mathbf{y} - \mathbf{X}^\top \mathbf{w}_0) + \mathbf{w}_0$, due the definition of $\hat{\mathbf{w}}_S$.

Using the fact that the LOO risk of Algorithm 1 can be written in closed form, [103] proposed to weight the source hypothesis \mathbf{w}_0 by a scalar β , optimized in order to minimize the LOO risk.

In the following we show how to generalize this approach to the generic source hypotheses h^{src} and how to obtain a generalization guarantee for it.

3.5 Analysis by Hypothesis Stability

We now propose a more general version of Algorithm 1.

Algorithm 2. *RLS transfer algorithm by altering training set as $\{(\mathbf{x}_i, y_i - h^{\text{src}}(\mathbf{x}_i)) : 1 \leq i \leq m\}$ produces a hypothesis*

$$A_S^{\text{htl}}(\mathbf{x}) = T_C(\mathbf{x}^\top \hat{\mathbf{w}}_S) + h^{\text{src}}(\mathbf{x}),$$

where

$$\hat{\mathbf{w}}_S := \operatorname{argmin}_{\mathbf{w} \in \mathbb{R}^d} \left\{ \frac{1}{m} \sum_{i=1}^m (\mathbf{w}^\top \mathbf{x}_i - y_i + h^{\text{src}}(\mathbf{x}_i))^2 + \lambda \|\mathbf{w}\|^2 \right\},$$

and the truncation function $T_C(\hat{y})$ is defined as $T_C(\hat{y}) = \min\{\max\{\hat{y}, -C\}, C\}$.

If $h^{\text{src}}(\mathbf{x})$ is equal to $\mathbf{x}^\top \mathbf{w}_0$, where \mathbf{w}_0 belongs to the same space as \mathbf{w}_S , and $C = \infty$, Algorithms 1 and 2 are completely equivalent, because they have exactly the same solution. However, Algorithm 2 is more general because it allows h^{src} to belong to a another hypothesis class. Hence it captures the notion of biased regularization, and generalizes it to any type of source hypothesis h^{src} . This algorithm also captures and generalizes many of the ideas present in the previous works on HTL [45, 145, 103, 94, 136]. Still the use of a specific loss, the square loss, will allow us to have an efficient computation as well. Also, this formulation allows us to truncate the prediction within the range $[-C; C]$, which improves the theoretical guarantees and the practical performance. In fact it is easy to see that if $C \geq B + \|h^{\text{src}}\|_\infty$, then $(T_C(\mathbf{x}^\top \hat{\mathbf{w}}_S) + h^{\text{src}}(\mathbf{x}) - y)^2 \leq (\mathbf{x}^\top \hat{\mathbf{w}}_S + h^{\text{src}}(\mathbf{x}) - y)^2$.

Our goal is to upper bound the expected risk of Algorithm 2, keeping in mind the effect of h^{src} . To this end, we propose to employ the stability framework of [17]. Our choice is motivated by the fact that bounds arising from the stability analysis are free from complexity measures. Hence, the generalization bound of interest will be composed mostly from computable quantities, thus making it more practical, e.g. for finding the optimal transfer parameters.

In particular, we can upper bound the moments of the random variable $R(A_S) - \hat{R}^{\text{loo}}(A, S)$ with a quantity that captures the stability of the learning algorithm. The second order moment can then be used to obtain polynomial bounds, through Chebyshev's inequality [17].

There are various definitions of algorithmic stability [17], but the one we will use is the hypothesis stability.

Definition 12 (Hypothesis Stability [17]). *An algorithm A has a hypothesis stability γ with respect to the loss function ℓ if for all $i \in \{1, \dots, m\}$ the following holds*

$$\mathbb{E}_{S, (\mathbf{x}, y)} [|\ell(A_S, (\mathbf{x}, y)) - \ell(A_{S^i}, (\mathbf{x}, y))|] \leq \gamma.$$

We will use a slight variation of the polynomial bound of [17]. The reason is that Theorem 11 in [17] has the term $\frac{M^2}{2}$, that is not affected by $R(h^{\text{src}})$. Instead, we exchange $\frac{M^2}{2}$ for the term $\mathbb{E}_S[\ell(h_{S^i}, z_i)]$.

Theorem 6. For a supervised learning algorithm A with hypothesis stability γ , and M such that $\ell(A_{S^i}, (\mathbf{x}, y)) \leq M$, for any $i \in \{1, \dots, m\}$, we have

$$\mathbb{E}_S[(R(A_S) - \widehat{R}^{\text{loo}}(A, S))^2] \leq \frac{M}{m} \mathbb{E}_S[\ell(A_{S^i}, (\mathbf{x}_i, y_i))] + 3M\gamma. \quad (3.6)$$

Note that our bound in the worst case loses only a constant multiplicative factor with respect to the one of [17]. We use this theorem to prove our main result, Theorem 7.

Theorem 7. Set $\lambda \geq \frac{1}{m}$. If $C \geq B + \|h^{\text{src}}\|_\infty$, then for Algorithm 2 with probability at least $1 - \delta$ over $S \stackrel{\text{iid}}{\sim} \mathcal{D}^m$ we have

$$R(A_S^{\text{htl}}) - \widehat{R}^{\text{loo}}(A^{\text{htl}}, S) = \mathcal{O} \left(C \cdot \frac{\sqrt[4]{R(h^{\text{src}}) T_{C^2} \left(\frac{R(h^{\text{src}})}{\lambda} \right) + R(h^{\text{src}})^2}}{\sqrt{m\delta\lambda^{3/4}}} \right). \quad (3.7)$$

If $C = \infty$, then for Algorithm 2 we have

$$R(A_S^{\text{htl}}) - \widehat{R}^{\text{loo}}(A^{\text{htl}}, S) = \mathcal{O} \left(\frac{\sqrt{R(h^{\text{src}})} (\|h^{\text{src}}\|_\infty + B)}{\sqrt{m\delta\lambda}} \right). \quad (3.8)$$

Proofs of both theorems can be found in Appendix A, while in the next section we discuss the implications of this theorem.

3.5.1 Implications

First consider the case of $h^{\text{src}} \equiv 0$. This case corresponds to learning without any source hypothesis, without transfer learning. If we set $C = \infty$, we have that the generalization error is bounded by $\mathcal{O} \left(\frac{B}{\sqrt{m\lambda}} \right)$, which is exactly the bound that can be obtained using the results in [17] for RLS, [34, p17, footnote 2].

However, if we know the range $[-B; B]$, we can set C accordingly and obtain that the generalization error is bounded by $\mathcal{O} \left(\frac{B}{\sqrt{m\lambda^{3/4}}} \right)$. Thanks to the truncation, the bound is improved compared the polynomial bound with square loss in [17].

We now turn our attention to the case where $h^{\text{src}} \neq 0$. In this case, the key quantity is $R(h^{\text{src}})$, an indirect measure of how the source and target domains are related. This term takes the role of the divergence between source and target distribution [8, 9, 93], however, this is a more intuitive measure which is directly linked to the loss: how the source hypothesis is going to perform on the target domain, the new task? In addition, it is multiplicative to all bound terms, while the mentioned divergence terms are additive, even if the bounds are generally incomparable. Based on its value, we have various regimes of interest. If $\frac{R(h^{\text{src}})}{\lambda} \rightarrow 0$, we have the surprising result that $R(A_S^{\text{htl}}) - \widehat{R}^{\text{loo}}(A^{\text{htl}}, S) \rightarrow 0$. This implies that the risk approaches the LOO risk, with probability 1. In other words, the transfer learning decreases the variance of the LOO in case when the source and target domains are related. This also implies that we can expect the tuning of any parameter of the algorithm (e.g. the type of kernel) through the minimization of the LOO risk, to have optimal performance, even with a small training set.

This is the first theoretical explanation of why the algorithms of [103, 136] showed reliable performance despite a small training set. Note that $R(h^{\text{src}})$ has to be small with respect to λ . In other words, the better the source hypothesis on the target domain, the more stable an HTL algorithm must be. Looking at Algorithm 1, this makes sense, since a very stable algorithm will generate a hypothesis that does not deviate much from the source \mathbf{w}_0 .

So far we have outlined the benefits of $R(h^{\text{src}})$, but it is reasonable to ask what happens when this quantity is high, that is when the two domains are unrelated. From the bound, we see that Algorithm 2 is also robust against a misspecified source hypothesis h^{src} . In fact, due to truncation, the rate is exactly the same as obtained in the non-transfer case. If we supply the algorithm with a “bad” source hypothesis, in the limit it will have the performance of an algorithm that learns just using the training set. Again, this robustness is achieved also thanks to the truncation, which avoids excessive growth of the loss. In other words, Algorithm 2 is resistant to negative transfer. We actually suspect that the truncation is necessary only for the proof, and in fact, [136] already noticed this robust behaviour of Algorithm 1.

We now consider the case when the source hypothesis h^{src} is a weighted combination of n source hypotheses h_i^{src} , that is $h^{\text{src}} = \sum_{i=1}^n \beta_i h_i^{\text{src}}$ for some $\beta_i \in \mathbb{R}$. This weighting strategy is equivalent to the ones used in the works on DA, but with the important difference that now these weights can be efficiently estimated from the target training set. In particular, one interpretation of Theorem 7 yields

$$\min_{\boldsymbol{\beta} \in \mathbb{R}^n} R(A_S^{\text{htl}}) \leq \min_{\boldsymbol{\beta} \in \mathbb{R}^n} \left\{ \widehat{R}^{\text{loo}}(A^{\text{htl}}, S) + \mathcal{O} \left(\frac{\|\boldsymbol{\beta}\|}{\sqrt{m\lambda^{3/4}}} \right) \right\}.$$

Hence the bound suggests an efficient and principled way to find $\boldsymbol{\beta} = [\beta_1, \dots, \beta_n]^\top$. In other words, it is enough to minimize the LOO risk with respect to $\boldsymbol{\beta}$, taking into account the regularization term, thus turning $\boldsymbol{\beta}$ into a parameter of an optimization problem. Note that [103, 136] already realized the empirical need to constrain $\boldsymbol{\beta}$, but here we demonstrate a principled form of the regularization. Note that the $\mathcal{O}(\cdot)$ notation used in the bound above hides the confidence variable δ , which should be tuned. Yet, here we are mainly interested in the correct form of the objective function for finding the transfer parameters, as a way of using theory to guide practice. Moreover, regardless of the specific procedure used to estimate the optimal value of $\boldsymbol{\beta}$, as noted above we expect the algorithm to be robust to negative transfer, at least in the asymptotic limit.

3.6 Conclusion

In this chapter we have formally introduced the HTL problem and analyzed a class of RLS algorithms with biased regularization that can be used to solve this problem. Our main result is a generalization bound in terms of the Leave-One-Out (LOO) risk, obtained through the notion of hypothesis stability. We point out the key quantity $R(h^{\text{src}})$ and expose its theoretical and practical advantages over analogues in the theory of DA. In particular, we showed that if source and target domains are related, hence $R(h^{\text{src}})$ is small, the LOO risk converges faster to the expected risk and the HTL decreases the variance of the LOO. In the case of unrelated domains, we still match the theoretical guarantees of Regularized Least Squares trained solely on the target domain. As a side effect of our analysis, thanks

to the truncation we have improved the polynomial generalization bounds of [17] for RLS¹.

In the next chapter we will focus on the extension of our theory to a more general family of algorithms. In particular, we will prove bounds on generalization and estimation error for the *regularized Empirical Risk Minimization* for smooth convex loss functions. We will also improve our results in another direction by obtaining bounds that hold with high probability.

¹The suboptimality of bounds in [17] for RLS is also discussed by [149].

4 Hypothesis Transfer Learning through Empirical Risk Minimization

The material of this chapter is based on the publication:

I. Kuzborskij and F. Orabona. Fast rates by transferring from auxiliary hypotheses.

In *Machine Learning* 106.2 (2017): 171-195.

The theoretical results presented in this chapter slightly differ from the ones presented in the publication.

The doctoral candidate formalized the problem, proved the results, and wrote most of the publication.

4.1 Overview

In the standard supervised machine learning setting the learner receives a set of labeled examples, known as the training set. However, very often we have additional information at hand that could be beneficial to the learning process. One such example is the use of unlabeled data drawn from the marginal distributions, that gives rise to the semi-supervised learning setting [22]. Another example is when the training data is coming from a related problem, as in multi-task learning [19], domain adaptation [8, 93], and transfer learning [105, 133]. Among others, there is the use of structural information, such as taxonomy, different views on the same data [15], or even a sort of privileged information [140, 125]. In the recent years all these directions have received a considerable empirical and theoretical attention.

In this chapter we focus on a less theoretically studied direction in the use of supplementary information – learning with *auxiliary hypotheses*, that is classifiers or regressors originating from another task. In particular, in addition to the training set we assume that the learner is supplied with a collection of hypotheses and their predictions on the training set itself. The goal of the learner is to figure out which hypotheses are helpful and use them to improve the prediction performance of the trained classifier. We will call these auxiliary hypotheses the *source* hypotheses and we will say that helpful

ones accelerate the learning on the *target* task. We focus on the linear setting, that is, we train a linear¹ classifier and the source hypotheses are used additively in the prediction process, weighted by some weights. In particular, this captures the setting in which the outputs of the source hypotheses are concatenated with the feature vector, a widely used heuristic [12, 85, 137].

The scenario described above is related to the Transfer Learning (TL) and DA ones, or learning effectively from possibly small amounts of data by reusing prior knowledge [134, 105, 133, 8]. However, transferring from hypotheses offers an advantage compared to the TL and DA frameworks, where one requires access to the data of the *source* domain. For example, in DA [8], one employs large unlabeled samples to estimate the relatedness of the source and target domains to perform the adaptation. Even if unlabeled data are abundant, the estimation of adaptation parameters can be computationally prohibitive. This is the case, for example, when a large number of domains is involved or when one acquires new domains incrementally.

A recently proposed setting, closer to the one we consider, is the HTL [76, 11], where the practical limitations of TL and DA are alleviated through indirect access to the *source domain* by means of a *source hypothesis*. Also, in the HTL setting there are no restrictions on how the source hypotheses can be used to boost the performance on the target task.

Albeit empirically the setting considered in this chapter has already been extensively exploited in the past [145, 103, 136, 65, 78], a first theoretical treatment of this setting was given in Chapter 3 and [76], where we analyzed the linear HTL algorithm that solves a regularized least-squares problem with a single fixed, unweighted, source hypothesis. We proved a polynomial generalization bound that depends on the performance of the fixed source hypothesis on the target task.

Contributions of this chapter. We extend the formulation of Chapter 3 and [76], with a general regularized Empirical Risk Minimization (ERM) problem with respect to any non-negative smooth loss function, and any strongly convex regularizer. We prove high-probability generalization bounds that exhibit *fast rate*, i.e. $\mathcal{O}(1/m)$, of convergence whenever a *weighted combination* of multiple source hypotheses, with weights found by the same ERM procedure, performs well on the target task. In addition, we show that, if the combination is perfect, the error on the training set becomes equal to the generalization error with probability 1. Furthermore, we analyze an estimation error of our formulation, and conclude that a good source hypothesis also speeds up the convergence to the performance of the best hypothesis in the entire class. To prove this result we derive a simple and powerful exponential on-average stability bound.

The rest of the chapter is organized as follows. In the next section we make a brief review of the previous work. Next, we formally state our formulation in Section 4.3 and present one main results right after, in Section 4.4. In Section 4.4.3 we discuss the implications and compare them to the body of literature in learning with fast rates and transfer learning. Finally, in Appendix B, we present the proofs of our results. Section 4.5 concludes the chapter.

¹Non-linear classifiers can be easily produced with the use of kernels.

4.2 Related Work

Chapter 3 showed that the generalization ability of the regularized least-squares HTL algorithm improves if the supplied *source* hypothesis performs well on the target task. More specifically, we proposed a key criterion, *the risk of the source hypothesis on the target domain*, that captures the relatedness of the source and target domains. Later, [11] showed a similar bound, but with a different quantity capturing the relatedness between source and target. Instead of considering a general source hypothesis, they have confined their analysis to the linear hypothesis class. This allowed them to show that the target hypothesis generalizes better when it is close to the good source hypothesis. From this perspective it is easy to interpret the source hypothesis as an initialization point in the hypothesis class. Naturally, given a starting position that is close to the best in the class, one generalizes well.

Prior to these works there were few studies trying to understand learning with auxiliary hypotheses subject to different conditions. [86] have analyzed a Bayesian approach to HTL. Employing a PAC-Bayes analysis they showed that given a prior on the hypothesis class, the generalization ability of logistic regression improves if the prior is informative on the target task. [92] analyzed a setting of *multiple source hypotheses* combination. There, in addition to the source hypotheses, the learner receives unlabeled samples drawn from the source distributions, that are used to weigh and combine these source hypotheses. They have studied the possibility of learning in such a scenario, however, they did not address the generalization properties of any particular algorithm.

Unlike these works, we focus on the generalization ability of a large family of HTL algorithms, that generate the target predictor given a set of multiple source hypotheses. In particular, we analyze Regularized Empirical Risk Minimization with the choice of any non-negative smooth loss and any strongly convex regularizer. Thus our analysis covers a wide range of algorithms, explaining their empirical success. One category of those, prevalent in computer vision [69, 145, 136, 2, 78, 137], employs the principle of biased regularization [121]. For example, instead of penalizing large weights by introducing the term $\|\mathbf{w}\|^2$ into the objective function, one enforces them to be close to some “prior” model, that is $\|\mathbf{w} - \mathbf{w}^{\text{prior}}\|^2$. This principle also found its applications in other fields, such as NLP [32, 33], and electromyography classification [103, 138]. Many empirical works have also investigated the use of the source hypotheses in a “black box” sense, sometimes not even posing the problem as transfer learning [39, 85, 65, 12], and recently in conjunction with deep neural networks [101].

In the literature there are several other machine learning directions conceptually similar to the one we consider in this chapter. Arguably, the most well known one is the Domain Adaptation (DA) problem. The standard machine learning assumption is that the training and the testing sets are sampled from the same probability distribution. In such case, we expect that a hypothesis generated by the learner from that training set will lead to sensible predictions on the testing set. The difficulty arises when training and testing distributions differ, that is we have a training set sampled from the *source domain* and a testing set from the *target domain*. Clearly, the hypothesis generated from the source domain can perform arbitrarily badly on the target one. A paradigm of DA, addressing this issue has received a lot of attention in recent years [8, 93]. Although this framework is different from the one we study in this chapter, we identify similarities and compare our findings with the theory of learning from different domains in Section 4.4.4.

4.3 Transferring from Auxiliary Hypotheses

In the following we will capture and generalize many transfer learning formulations that employ a collection of given *source hypotheses* $\{h_i^{\text{src}} : \mathcal{X} \mapsto \mathcal{Y}\}_{i=1}^n$ within the framework of Regularized Empirical Risk Minimization (ERM). These problems typically involve a criterion for source hypothesis selection and combination with the goal to increase performance on the *target task* [145, 137, 79]. Indeed, some source hypotheses might come from tasks similar to the target task and the goal of an algorithm is to select only relevant ones. In this chapter we will consider source combination

$$h_{\boldsymbol{\beta}}^{\text{src}}(\mathbf{x}) = \sum_{i=1}^n \beta_i h_i^{\text{src}}(\mathbf{x}),$$

and target hypothesis of a form

$$h_{\mathbf{w}, \boldsymbol{\beta}}(\mathbf{x}) = \langle \mathbf{w}, \mathbf{x} \rangle + h_{\boldsymbol{\beta}}^{\text{src}}(\mathbf{x}), \quad (4.1)$$

with the relevance of the sources characterized by the parameter $\boldsymbol{\beta} \in \mathbb{R}^n$. We will focus on the Regularized ERM formulations with the choice of any non-negative smooth loss function and any strongly-convex regularizer. This puts our problem into the class of the ones that can be solved efficiently, yet endowed with interesting properties.

Regularized ERM for Transferring from Auxiliary Hypotheses (R-ERM-HTL). *Let $\phi : \mathcal{Y} \times \mathcal{Y} \mapsto \mathbb{R}_+$ be a L -Lipschitz, convex, and H -smooth loss function and let $\Omega : \mathcal{H} \mapsto \mathbb{R}_+$ be a 2-strongly convex function w.r.t. a norm $\|\cdot\|$, such that $\Omega(\mathbf{0}) = 0$. Given the target training set $S = \{(\mathbf{x}_i, y_i)\}_{i=1}^m$, $\lambda \geq 0$, source hypotheses $\{h_i^{\text{src}}\}_{i=1}^n$, the algorithm generates the target hypothesis*

$$A_S^{\text{htl}}(\mathbf{x}) = \langle \widehat{\mathbf{w}}_S, \mathbf{x} \rangle + h_{\widehat{\boldsymbol{\beta}}_S}^{\text{src}}(\mathbf{x}),$$

such that

$$(\widehat{\mathbf{w}}_S, \widehat{\boldsymbol{\beta}}_S) = \arg \min_{(\mathbf{w}, \boldsymbol{\beta}) \in \mathbb{R}^{2d}} \left\{ \frac{1}{m} \sum_{i=1}^m \phi(h_{\mathbf{w}, \boldsymbol{\beta}}(\mathbf{x}_i), y_i) + \lambda \Omega(\mathbf{w}) + \lambda \Omega(\boldsymbol{\beta}) \right\}. \quad (4.2)$$

In the following we will pay special attention to a quantity that captures the performance of the source hypothesis combination $h_{\widehat{\boldsymbol{\beta}}_S}^{\text{src}}(\mathbf{x})$ on the target domain

$$R^{\text{src}} = \mathbb{E}_S \left[R(h_{\widehat{\boldsymbol{\beta}}_S}^{\text{src}}) \right].$$

Our analysis will focus on the generalization properties of A_S^{htl} . In particular, our main goal will be to understand the impact of the source hypothesis combination on the performance of the target hypothesis. In our analysis we will discuss various regimes of interest, for example considering the perfect and arbitrarily bad source hypothesis. Our discussion will touch scenarios where the auxiliary hypotheses accelerate the learning and the conditions when we can provably expect perfect generalization. Finally, we will consider the consistency of the formulation and pinpoint conditions when we achieve faster convergence to the performance of the best-in-the-class.

One special example covered by our analysis, commonly applied in transfer learning, is the *biased*

regularization [121]. Consider the following least-squares based formulation.

Least-Squares with Biased Regularization. Given the target training set S , source hypotheses $\{\mathbf{w}_i^{\text{src}}\}_{i=1}^n \subset \mathcal{H}$, parameters $\boldsymbol{\beta} \in \mathbb{R}^n$ and $\lambda \geq 0$, the algorithm generates the target hypothesis $A_S^{\text{htl-bias}}(\mathbf{x}) = \langle \hat{\mathbf{w}}_S, \mathbf{x} \rangle$, where

$$\hat{\mathbf{w}}_S = \operatorname{argmin}_{\mathbf{w} \in \mathcal{H}} \left\{ \frac{1}{m} \sum_{i=1}^m (\langle \mathbf{w}, \mathbf{x}_i \rangle - y_i)^2 + \lambda \left\| \mathbf{w} - \sum_{j=1}^n \beta_j \mathbf{w}_j^{\text{src}} \right\|_2^2 \right\}. \quad (4.3)$$

This problem has a simple intuitive interpretation: minimize the training error on the target training set while keeping the solution close to the linear combination of the source hypotheses. One can naturally arrive at (4.3) from a probabilistic perspective: The solution $\hat{\mathbf{w}}$ is a maximum a posteriori estimate when the conditional distribution is Gaussian and the prior is a $\mathbf{W}^{\text{src}} \boldsymbol{\beta}$ -mean, $\frac{1}{\lambda} \mathbf{I}$ -covariance Gaussian distribution. Even though biased regularization is a simple idea, it found success in a plethora of transfer learning applications, ranging from computer vision [69, 145, 136, 2, 78, 137] to NLP [32], to electromyography classification [103, 138].

Claim 1. *Least-Squares with Biased Regularization is a special case of R-ERM-HTL.*

Proof. Introduce \mathbf{w}' , such that $\mathbf{w}' = \mathbf{w} - \mathbf{W}^{\text{src}} \boldsymbol{\beta}$. Then we have that problem (4.3) is equivalent to

$$\min_{\mathbf{w} \in \mathcal{H}} \left\{ \frac{1}{m} \sum_{i=1}^m (\langle \mathbf{w}' + \mathbf{W}^{\text{src}} \boldsymbol{\beta}, \mathbf{x}_i \rangle - y_i)^2 + \lambda \|\mathbf{w}'\|_2^2 \right\},$$

that in turn is a special version of (4.2) when $h_i^{\text{src}}(\mathbf{x}) = \langle \mathbf{w}_i^{\text{src}}, \mathbf{x} \rangle$, we use the square loss, and $\|\cdot\|_2^2$ as regularizer. \square

Albeit practically appealing, the formulation (4.3) is limited in the fact that the source hypothesis must be a linear predictor living in the same space of the target predictor. Instead, R-ERM-HTL naturally generalizes the biased regularization formulation, allowing to treat the source hypothesis as “black box” predictors.

4.4 Main Results

To prove the bound on the generalization error of R-ERM-HTL, we will first prove a novel general algorithmic stability result.

4.4.1 Exponential Generalization Bounds for On-average Stable Algorithms

In particular, we prove that if an algorithm is stable in a data-dependent sense, it generalizes with high probability. To do this, we consider the following two notions of stability.

Definition 13 (On-Average Stability). *Let $t \stackrel{iid}{\sim} U([m])$.*

1) An algorithm A is ϵ_m -on-average stable if it is true that

$$\sup_{z'} \mathbb{E}_{S, z, i} [\ell(A_{S^{(i)}}(z')) - \ell(A_S(z'))] \leq \epsilon_m .$$

2) An algorithm A is $\epsilon_m^{(2)}$ -second-order-on-average stable if it is true that

$$\sup_{z'} \mathbb{E}_{S, z, i} [(\ell(A_S(z')) - \ell(A_{S^{(i)}}(z')))^2] \leq \epsilon_m^{(2)} .$$

The first notion of stability is a slightly stronger version of ‘‘On-Average-Replace-One-Stability’’ presented in Section 2.3, studied by [123, Definition 13.3] and [124], and is closely related to the ‘‘pointwise hypothesis stability’’ from [17]. The second one is intimately related to the variance of the hypothesis generated by the algorithm and is not common in the stability literature. It is partially inspired by a recent work of Maurer [95] on generalized (second-order) McDiarmid’s inequality. It is also not hard to see that whenever the loss function is Lipschitz or smooth, both notions of stability can be analyzed in a similar way. The following theorem characterizes the generalization error of an algorithm A given that it satisfies simultaneously both notions of stability.

Theorem 8. *Let algorithm A be ϵ_m -on-average-stable and $\epsilon_m^{(2)}$ -second-order-on-average stable. Then for a hypothesis A_S we have with probability at least $1 - e^{-\eta}$ that*

$$R(A_S) - \widehat{R}_S(A_S) \leq \epsilon_m + \frac{1.5M\eta}{m \log \left(1 + \frac{2M\eta}{m\sqrt{2\epsilon_m^{(2)}}} \right)} \leq \epsilon_m + \sqrt{4\eta\epsilon_m^{(2)}} + \frac{1.5M\eta}{m} . \quad (4.4)$$

The first inequality is useful whenever an algorithm is completely stable, that is $\epsilon_m = 0$ and $\epsilon_m^{(2)} = 0$. In such case the bound backs up an intuition that generalization error equals to zero. The idea of the proof is to relate the second-order-on-average stability to the variance in Bennett’s and Bernstein’s inequalities through Steele’s inequality.

4.4.2 Bounds for Hypothesis Transfer Learning through Regularized ERM

In this section, we present the main results of this chapter: generalization and estimation error bounds for R-ERM-HTL. In the next section we discuss in detail the implications of these results, while we defer the proofs to Appendix.

The first bound demonstrates the utility of the perfect combination of source hypotheses, while the second lets us observe the dependency on the arbitrary combination. In particular, the first bound explicitates the intuition that given the perfect source hypothesis learning is not required. In other words, when $R^{\text{src}} = 0$ we have that the empirical risk becomes equal to the risk with probability one.

Theorem 9. *Assume that $\|h_i^{\text{src}}\|_\infty \leq 1, \forall i \in [n]$, that $\|\ell\|_\infty \leq M$, and finally that $H \leq \frac{m\lambda}{2}$. Then for*

R -ERM-HTL with probability at least $1 - e^{-\eta}$, $\forall \eta > 0$,

$$R(A_S^{\text{htl}}) - \widehat{R}_S(A_S^{\text{htl}}) \leq \frac{4L\sqrt{HR^{\text{src}}}}{m\lambda} + \frac{1.5M\eta}{m \log\left(1 + \frac{M\eta}{2L\sqrt{2H}} \cdot \frac{\lambda}{\sqrt{R^{\text{src}}}}\right)} \quad (4.5)$$

$$\leq \frac{4L(1 + \sqrt{4\eta})\sqrt{HR^{\text{src}}}}{m\lambda} + \frac{1.5M\eta}{m}. \quad (4.6)$$

Now we focus on the consistency of the HTL. Specifically, we show an upper bound on the excess risk of the Regularized ERM, which depends on R^{src} , that is the risk of the combined source hypothesis h_{β}^{src} on the target domain. We observe that for a small R^{src} , the excess risk shrinks at a fast rate of $\mathcal{O}(1/m)$. In other words, a good prior knowledge guarantees not only good generalization, but also the fast recovery of the performance of the best hypothesis in the class.

This bound is similar in spirit to the results of localized complexities, as in works of [5, 130], however we focus on the linear HTL scenario rather than a generic learning setting. Later, in Section 4.4.3, we compare our bounds to these works and show that our analysis achieves superior results.

Theorem 10. Assume that $\|\ell\|_{\infty} \leq M$, and finally that $H \leq \frac{m\lambda}{2}$, and let

$$\mathbf{w}^* = \underset{\mathbf{w} \in \mathcal{H}}{\text{arginf}} R(\mathbf{w}).$$

Then setting

$$\lambda = \sqrt{\frac{4L\sqrt{H}(1 + \sqrt{4\eta})}{m} \cdot \frac{\sqrt{R^{\text{src}}}}{\Omega(\mathbf{w}^*)}}.$$

in R -ERM-HTL, we have with probability at least $1 - e^{-\eta}$, $\forall \eta > 0$, that

$$R(A_S^{\text{htl}}) - R^*(\mathcal{H}) \leq 4\sqrt{\frac{L\sqrt{H}(1 + \sqrt{4\eta}) \cdot \Omega(\mathbf{w}^*)\sqrt{R^{\text{src}}}}{m}} + \sqrt{\frac{2R^{\text{src}}\eta}{m}} + \frac{3M\eta}{m}. \quad (4.7)$$

4.4.3 Implications

We start by discussing the effect on the generalization ability of the source hypothesis combination. Intuitively, a good source hypothesis combination should facilitate transfer learning, while a reasonable algorithm must not fail if we provide it with bad one. That said, a natural question to ask here is, what makes a good or bad source hypothesis? As in previous works in transfer learning and domain adaptation, we capture this notion via a quantity that has two-fold interpretation: (1) the performance of the source hypothesis combination on the target domain; (2) relatedness of the source and target domains. In the theorems presented in the previous sections we denoted it by R^{src} , that is the risk of the source hypothesis combination on the target domain. In this section we will consider various regimes of interest with respect to R^{src} .

When the source is a bad fit. First consider the case when the source hypothesis combination h_{β}^{src} is useless for the purpose of transfer learning, for example, $h_{\beta}^{\text{src}}(\mathbf{x}) = 0$ for all \mathbf{x} . This corresponds to learning with no auxiliary information. Then we can assume that $R^{\text{src}} \leq M$, and from Theorem 9 we

obtain $R(A_S^{\text{htl}}) - \widehat{R}(A_S^{\text{htl}}) \leq \mathcal{O}(1/(m\lambda))$. This rate matches the one in the analysis of [123, Corollary 13.7] (their bound in expectation can be extended to the high probability one, e.g. by using the technique of [124]).

When the source is a good fit. Here we would like to consider the behavior of the algorithm in the finite-sample and asymptotic scenarios. We first look at the regime when λ^* is minimizing the bound on $R(A_S^{\text{htl}})$ in Theorem 9. Suppose that such λ^* obeys $\lambda^* = \mathcal{O}(\sqrt{R^{\text{src}}})$. In this case, the fast rate *independent* from λ will dominate the bound, and we obtain the convergence rate of $\mathcal{O}(1/m)$. In other words, we can expect a much faster convergence when λ^* provided by the oracle inequality obeys $\lambda^* \ll 1$, and R^{src} , the quality of the combined source hypotheses, is of matching order. Now consider the asymptotic behavior of the algorithm, particularly when m goes to infinity. In such case, the algorithm exhibits a rate of $\mathcal{O}(\sqrt{R^{\text{src}}}/(m\lambda) + 1/m)$, so R^{src} controls the constant factor of the rate. Hence, the quantity R^{src} governs the transient regime for small λ^* and the asymptotic behavior of the algorithm, predicting a faster convergence in both regimes.

When source is a perfect fit. It is conceivable that the source hypothesis exploited is the perfect one, that is $R^{\text{src}} = 0$. In other words, the source hypothesis combination is a perfect predictor for the target domain. Theorem 9 implies that $R(A_S^{\text{htl}}) = \widehat{R}(A_S^{\text{htl}})$ with probability one. We note that for many practically used smooth losses, such as the square loss, this setting is only realistic if the source and target domains match and the problem is noise-free. However, we can observe $R^{\text{src}} = 0$, for example, when the squared hinge loss, $\ell(z, y) = \max\{0, 1 - zy\}^2$, is used and all target domain examples are classified correctly by the source hypothesis combination, case that is not unthinkable for related domains.

Fast rates. There is a number of works in the literature on Uniform Convergence (UC) bounds investigating rates of convergence faster than $1/\sqrt{m}$ subject to different conditions. The PAC literature approached such bounds through relative VC bounds [141], local Rademacher complexity [5], and Rademacher bounds for smooth loss classes [130].

In this chapter we follow a constructive analysis, which is in the spirit of the optimization literature and derive bounds of order $\mathcal{O}(1/(m\lambda) + 1/m)$, that is, free from $1/\sqrt{m}$ terms. Bounds of similar order, albeit in expectation, appear in [123], and also with high probability, albeit w.r.t. the uniform stability [88]. Theorem 8 is in the spirit of these works, albeit with a few differences. The r.h.s. of the first inequality in Theorem 8 vanishes when the algorithm is perfectly stable. Though intuitively trivial, this allows to prove a considerable result in the theory of transfer learning as it quantifies the intuition that no learning is necessary if the source hypothesis has perfect performance on the target task. A bound similar to (4.6) can be achieved through the results of [123], however, in expectation rather than with high probability. The results of [88] cannot be employed to achieve a similar bound because of the stronger, distribution-free notion of stability.

Fast rates for ERM with a smooth loss have been thoroughly analyzed by [130] through the UC argument. Yet, the analysis of our HTL algorithm within their framework would yield a bound that is inferior to ours in two respects. The first concerns the scenario when the combined source hypothesis is perfect, that is $R^{\text{src}} = 0$. The generalization bound of [130] does not offer a way to show that the empirical risk converges to the risk with probability one – instead one can only hope to get a fast rate of convergence. The second problem is in the fact that such bound would depend on the empirical

performance of the combined source hypothesis. As we have noted before, the quantity R^{src} is essential because it captures the degree of relatedness between two domains. In their bounds, one cannot obtain this relationship through the Rademacher complexity term. The reason for this is the strong notion of Rademacher complexity that is employed by that framework, involving a supremum over the sample instead of an expectation.

4.4.4 Comparison to Theories of Domain Adaptation and Transfer Learning

The setting in DA is different from the one we study, however, we will briefly discuss the theoretical relationship between the two. Typically in DA, one trains a hypothesis from an weighted source training set, striving to achieve good performance on the target domain. The key question here is how to alter, or to *adapt*, the source training set. To answer this question, the DA literature introduces the notion of domain relatedness, which quantifies the dissimilarities between the marginal distributions of corresponding domains. Practically, in some cases the domain relatedness can be estimated through a large set of unlabeled samples drawn from both source and target domains. Theories of DA [8, 93, 10, 92, 26] have proposed a number of such domain relatedness criteria. Perhaps the most well known are the $d_{\mathcal{H}\Delta\mathcal{H}}$ -divergence [8] and its more general counterpart, the Discrepancy Distance [93]. Typically, this divergence is explicitated in the generalization bound along with other terms controlling the generalization on the target domain. Let $R_{\mathcal{D}^{\text{trg}}}(h)$ and $R_{\mathcal{D}^{\text{src}}}(h)$ denote the risks of the hypothesis h , measured w.r.t. the target and source distributions. Then a well-known result of [8] suggests that for all $h \in \mathcal{H}$

$$R_{\mathcal{D}^{\text{trg}}}(h) \leq R_{\mathcal{D}^{\text{src}}}(h) + \frac{1}{2} d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D}^{\text{src}}, \mathcal{D}^{\text{trg}}) + \varepsilon_{\mathcal{H}}^*, \quad (4.8)$$

where $\varepsilon_{\mathcal{H}}^* = \min_{h \in \mathcal{H}} \{R_{\mathcal{D}^{\text{trg}}}(h) + R_{\mathcal{D}^{\text{src}}}(h)\}$. This result implies that adaptation is possible given that $d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D}^{\text{src}}, \mathcal{D}^{\text{trg}})$ and ε^* are small. One can try to reduce those by controlling the complexity of the class \mathcal{H} and by minimizing the divergence $d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D}^{\text{src}}, \mathcal{D}^{\text{trg}})$. In practice, the latter can be manipulated through an empirical counterpart on the basis of unlabeled samples. Increasing the complexity of \mathcal{H} indeed reduces ε^* , but inflates $d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D}^{\text{src}}, \mathcal{D}^{\text{trg}})$. On the other hand, minimizing $d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D}^{\text{src}}, \mathcal{D}^{\text{trg}})$ alone puts us under the risk of increasing ε^* , since the empirical divergence is reduced without taking the labeling into account.

Clearly, this bound cannot be directly compared to our result, Theorem 9. However, we note the term R^{src} appearing in our results, which plays a role very similar to $d_{\mathcal{H}\Delta\mathcal{H}}$ in (4.8). Observe that by using (4.8) we can write

$$R^{\text{src}} = R_{\mathcal{D}^{\text{trg}}}(h_{\hat{\beta}_s}^{\text{src}}) \leq R_{\mathcal{D}^{\text{src}}}(h_{\hat{\beta}_s}^{\text{src}}) + d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D}^{\text{src}}, \mathcal{D}^{\text{trg}}) + \varepsilon_{\mathcal{H}}^*.$$

Plugging this into the generalization bound (4.6) we have

$$R(A_S) - \hat{R}_S(A_S) = \mathcal{O} \left(\frac{\sqrt{R_{\mathcal{D}^{\text{src}}}(h_{\hat{\beta}_s}^{\text{src}}) + d_{\mathcal{H}\Delta\mathcal{H}}(\mathcal{D}^{\text{src}}, \mathcal{D}^{\text{trg}}) + \varepsilon_{\mathcal{H}}^*}}{m\lambda} + \frac{1}{m} \right). \quad (4.9)$$

Albeit this inequality shows the generalization ability of the transfer learning algorithm, comparing to (4.8), we observe that DA and our result agree on the fact that the divergence between the domains

has to be small to generalize well. In fact, in the formulation we consider, the divergence is controlled in two ways: implicitly, by the choice of source hypotheses and through the size of class \mathcal{H} , that is by choosing λ . Second, in DA we expect that a hypothesis performs well on the target only if it performs well on the source. In our results, this requirement is relaxed. As a side note, we observe that (4.9) captures an intuitive notion that a good source hypothesis has to perform well on its own domain. Finally, in the theory of DA $\varepsilon_{\mathcal{H}}^*$ is assumed to be small. Indeed, if $\varepsilon_{\mathcal{H}}^*$ is large, there is no hypothesis that is able to perform well on both domains simultaneously, and therefore adaptation is hopeless. In our case, the algorithm can still generalize even with large $\varepsilon_{\mathcal{H}}^*$, however this is due to the supervised nature of HTL.

We now turn our attention to the previous theoretical works studying HTL-related settings. Few papers have addressed the theory of transfer learning, where the only information passed from the source domain is the classifier or regressor. [92] have addressed the problem of multiple source hypotheses combination, however, in a different setting. Specifically, in addition to the source hypotheses, the learner receives the unlabeled samples drawn from the source distributions, that are used to weigh and combine these source hypotheses. The authors have presented a general theory of such a scenario and did not study the generalization properties of any particular algorithm. The first analysis of the generalization ability of HTL in the similar context we consider here was done in Chapter 3 and in [76]. The work focused on the L_2 -regularized least squares and the generalization bound involving the leave-one-out risk instead of the empirical one. The following result, obtained through an algorithmic stability argument [17], holds with probability at least $1 - \delta$

$$R(h^{\text{trg}}) \leq \widehat{R}^{\text{loo}}(h^{\text{trg}}) + \mathcal{O}\left(\frac{\sqrt[4]{R^{\text{src}}}}{\sqrt{m\delta\lambda^{0.75}}}\right), \quad (4.10)$$

where R^{src} is the risk of a single fixed source hypothesis and h^{trg} is the solution of a Regularized Least Square problem. We first observe that the shape of the bound is similar to the one obtained in this chapter, although with a number of differences. First, contrary to our presented bounds, their bound assumes the use of a fixed source hypothesis, that is not even weighted by any coefficient. In practice, this is a very strong assumption, as one can receive an arbitrarily bad source and have no way to exclude it. Second, the bound (4.10) seems to have a vanishing behavior whenever the risk of the source R^{src} is equal to zero. This comes at the cost of the use of a weaker concentration inequality. In Theorem 9 we manage to obtain the same behavior with high probability. Finally, we get a better dependency on R^{src} .

4.5 Conclusion

In this chapter we have formally captured and theoretically analyzed a general family of learning algorithms transferring information from multiple supplied source hypotheses. In particular, our formulation stems from the regularized Empirical Risk Minimization principle with the choice of any non-negative smooth loss function and any strongly convex regularizer. Theoretically we have analyzed the generalization ability and excess risk of this family of HTL algorithms. Our analysis showed that a good source hypothesis combination facilitates faster generalization, specifically in $\mathcal{O}(1/m)$ instead of the usual $\mathcal{O}(1/\sqrt{m\lambda})$ of UC argument or $\mathcal{O}(1/(m\lambda))$ of algorithmic stability one. Furthermore, given a perfect source hypothesis combination, our analysis is consistent with the intuition that learning is

not required.

Our conclusions suggest the key importance of a source hypothesis selection procedure. Indeed, when an algorithm is provided with enormous pool of source hypotheses, how to select relevant ones on the basis of only few labeled examples? This might sound similar to the feature selection problem under the condition that $n \gg m$, however, earlier empirical studies by [137] with hundreds of sources did not find much corroboration for this hypothesis when applying $L1$ regularization. In Chapter 6 we will present a *greedy* algorithm that learns well from few examples given hundreds of source hypotheses.

Despite its generality, the analysis presented in this section is limited to smooth, convex, and regularized learning problems. In the next chapter we will approach HTL from a different angle – from a point of view of stochastic optimization, which will allow us to prove generalization bounds also for *non-convex* objective functions, such as in deep learning.

5 Hypothesis Transfer Learning through Stochastic Optimization

The material of this chapter is based on the publication:

I. Kuzborskij and C. H. Lampert. Data-Dependent Stability of Stochastic Gradient Descent. (Under submission) In *arXiv preprint* arXiv:1703.01678, 2017

The doctoral candidate formalized the problem, proved the results, and wrote most of the publication.

5.1 Overview

Stochastic Gradient Descent (SGD) has become one of the workhorses of modern machine learning. In particular, it is the optimization method of choice for training highly complex and non-convex models, such as neural networks. When it was observed that these models generalize better (suffer less from overfitting) than classical machine learning theory suggests, a large theoretical interest emerged to explain this phenomenon. Given that SGD at best finds a local minimum of the non-convex objective function, it has been argued that all such minima might be equally good. However, at the same time, a large body of empirical work and tricks of trade, such as *early stopping*, suggests that in practice one might not even reach a minimum, yet nevertheless observes excellent performance.

In this chapter we follow an alternative route that aims to *directly* analyze the generalization ability of SGD by studying how sensitive it is to small perturbations in the training set. This is known as the *algorithmic stability* approach [17] and was used recently [57] to establish generalization bounds for both convex and non-convex learning settings. To do so they employed a rather restrictive notion of stability that does not depend on the data, but captures only intrinsic characteristics of the learning algorithm and global properties of the objective function. Consequently, their analysis results in worst-case guarantees that in some cases tend to be too pessimistic. As recently pointed out in [147], *deep learning* might indeed be such a case, as this notion of stability is insufficient to give deeper theoretical insights, and a less restrictive one is desirable.

As our main contribution in this chapter we establish that a data-dependent notion of algorithmic stability, very similar to the *On-Average Stability* [124], holds for SGD when applied to convex as well as non-convex learning problems. As a consequence we obtain new generalization bounds that depend on the data-generating distribution and the initialization point of an algorithm. For convex loss functions, the bound on the generalization error is multiplicative in the risk at the initialization point. For non-convex loss functions, besides the risk, it is also critically controlled by the expected second-order information about the objective function at the initialization point. We further corroborate our findings empirically and show that, indeed, the data-dependent generalization bound is tighter than the worst-case counterpart on non-convex objective functions. Finally, the nature of the data-dependent bounds allows us to state *optimistic* bounds that switch to the faster rate of convergence subject to the vanishing empirical risk.

In particular, our findings justify the intuition that SGD is more stable in less curved areas of the objective function and link it to the generalization ability. This also backs up numerous empirical findings in the deep learning literature that solutions with low generalization error occur in less curved regions. At the same time, in pessimistic scenarios, our bounds are no worse than those of [57].

Finally, we exemplify an application of our bounds, and propose a simple yet principled *hypothesis transfer learning* scheme for the convex and non-convex case, which is guaranteed to transfer from the best source of information. In addition, this approach can also be used to select a good initialization given a number of random starting positions. This is a theoretically sound alternative to the purely random commonly used in non-convex learning.

The rest of the chapter is organized as follows. We revisit the connection between stability and generalization of SGD in Section 5.3 and introduce a data-dependent notion of stability in Section 5.4. We state the main results in Section 5.5, in particular, Theorem 13 for the convex case, and Theorem 15 for the non-convex one. Next we demonstrate empirically that the bound shown in Theorem 15 is tighter than the worst-case one in Section 5.5.2. Finally, we suggest application of these bounds by showcasing principled transfer learning approaches in Section 5.5.3, and we conclude in Section 5.6.

5.2 Related Work

Algorithmic stability has been a topic of interest in learning theory for a long time, however, the modern approach on the relationship between stability and generalization goes back to the milestone work of [17]. They analyzed several notions of stability, which fall into two categories: distribution-free and distribution-dependent ones. The first category is usually called *uniform* stability and focuses on the intrinsic stability properties of an algorithm without regard to the data-generating distribution. Uniform stability was used to analyze many algorithms, including regularized ERM [17], randomized aggregation schemes [42], and recently SGD by [57, 89], and [112]. Despite the fact that uniform stability has been shown to be sufficient to guarantee learnability, it can be too pessimistic, resulting in worst-case rates.

In this chapter we are interested in the data-dependent behavior of SGD, thus the emphasis will fall on the distribution-dependent notion of stability, known as *on-average* stability, explored thoroughly in [124]. The attractive quality of this less restrictive stability type is that the resulting bounds are

5.3. Stability of Stochastic Gradient Descent

controlled by how stable the algorithm is under the data-generating distribution. For instance, in [17] and [36], the on-average stability is related to the variance of an estimator. In [123, Sec. 13], the authors show risk bounds that depend on the expected empirical risk of a solution to the regularized ERM. In turn, one can exploit this fact to state improved *optimistic* risk bounds, for instance, ones that exhibit *fast-rate* regimes [71, 52], or even to design enhanced algorithms that minimize these bounds in a data-driven way, e.g. by exploiting side information as in transfer [76, 11] and metric learning [110]. Here, we mainly focus on the latter direction in the context of SGD: how stable is SGD under the data-generating distribution given an initialization point? We also touch the former direction by taking advantage of our data-driven analysis and show optimistic bounds as a corollary.

We will study the on-average stability of SGD for both convex and non-convex loss functions. In the convex setting, we will relate stability to the risk at the initialization point, while previous data-driven stability arguments usually consider minimizers of convex ERM rather than a stochastic approximation [123, 71]. Beside convex problems, our work also covers the generalization ability of SGD on non-convex problems. Here, we borrow techniques of [57] and extend them to the distribution-dependent setting. That said, while the bounds of [57] are stated in terms of worst-case quantities, ours reveal new connections to the data-dependent second-order information. These new insights also partially justify empirical observations in deep learning about the link between the curvature and the generalization error [61, 68, 23]. At the same time, our work is an alternative to the theoretical studies of neural network objective functions [25, 67], as we focus on the direct connection between the generalization and the curvature.

In this light, our work is also related to non-convex optimization by SGD. The literature on this subject typically studies rates of convergence to the stationary points [50, 1, 114], and ways to avoid saddles [48, 84]. However, unlike these works, and similarly to [57], we are interested in the generalization ability of SGD, and thanks to the stability approach, involvement of stationary points in our analysis is not necessary.

Finally, we propose an example application of our findings in TL. For instance, by controlling the stability bound in a data-driven way, one can choose an initialization that leads to improved generalization. This is related to TL where one transfers from pre-trained models [77, 137, 109, 11], especially popular in deep learning due to its data-demanding nature [46]. The theoretical literature on this topic is mostly focused on the ERM setting and PAC-bounds, while our analysis of SGD yields guarantees as a corollary.

5.3 Stability of Stochastic Gradient Descent

First, we briefly revisit the link between stability and generalization focusing on stability of stochastic learning algorithms.

5.3.1 Uniform Stability and Generalization

On an intuitive level, a learning algorithm is said to be *stable* whenever a small perturbation in the training set does not affect its outcome too much. Of course, there is a number of ways to formalize the perturbation and the extent of the change in the outcome, and we will discuss some of them

below. The most important consequence of a stable algorithm is that it *generalizes* from the training set to the unseen data sampled from the same distribution. In other words, the difference between the risk $R(A_S)$ and the empirical risk $\widehat{R}_S(A_S)$ of the algorithm's output is controlled by the quantity that captures how stable the algorithm is. So, to observe good performance, or a decreasing true risk, we must have a stable algorithm *and* decreasing empirical risk (training error), which usually comes by design of the algorithm. In this chapter we focus on the stability of the Stochastic Gradient Descent (SGD) algorithm, and thus, as a consequence, we study its generalization ability.

Recently, [57] used a stability argument to prove generalization bounds for learning with SGD. Specifically, the authors extended the notion of the *uniform stability* originally proposed by [17], to accommodate randomized algorithms.

Definition 14 (Uniform stability). *A randomized algorithm A is ϵ -uniformly stable if for all datasets $S, S^{(i)} \in \mathcal{Z}^m$ such that S and $S^{(i)}$ differ in the i -th example, we have*

$$\sup_{z \in \mathcal{Z}, i \in [m]} \left\{ \mathbb{E}_A [\ell(A_S, z) - \ell(A_{S^{(i)}}, z)] \right\} \leq \epsilon.$$

Since SGD is a randomized algorithm, we have to cope with two sources of randomness: the data-generating process and the randomization of the algorithm A itself, hence we have statements in expectation. The following theorem of [57] shows that the uniform stability implies generalization in expectation.

Theorem 11. *Let A be ϵ -uniformly stable. Then,*

$$\left| \mathbb{E}_{S, A} [\widehat{R}_S(A_S) - R(A_S)] \right| \leq \epsilon.$$

Thus it suffices to characterize the uniform stability of an algorithm to state a generalization bound. In particular, [57] showed generalization bounds for SGD under different assumptions on the loss function ℓ . Despite that these results hold in expectation, other forms of generalization bounds, such as high-probability ones, can be derived from the above [124].

Apart from SGD, uniform stability has been used before to prove generalization bounds for many learning algorithms [17]. However, these bounds typically suggest worst-case generalization rates, and rather reflect intrinsic stability properties of an algorithm. In other words, uniform stability is oblivious to the data-generating process and any other side information, which might reveal scenarios where generalization occurs at a faster rate. In turn, these insights could motivate the design of improved learning algorithms. In the following we address some limitations of the analysis through uniform stability by using a less restrictive notion of stability. We extend the setting of [57] by proving data-dependent stability bounds for convex and non-convex loss functions. In addition, we also take into account the initialization point of an algorithm as a form of supplementary information, and we dedicate special attention to its interplay with the data-generating distribution. Finally, we discuss situations where one can explicitly control the stability of SGD in a data-dependent way.

5.4 Data-dependent Stability Bounds for SGD

In this section we describe a notion of data-dependent algorithmic stability, that allows us to state generalization bounds which depend not only on the properties of the learning algorithm, but also on the additional parameters of the algorithm. We indicate such additional parameters by θ , and therefore we denote stability as a function $\epsilon(\theta)$. In particular, in the following we will be interested in scenarios where θ describes the data-generating distribution and the initialization point of SGD.

Definition 15 (On-Average stability). *A randomized algorithm A is $\epsilon(\theta)$ -on-average stable if it is true that*

$$\sup_{i \in [m]} \left\{ \mathbb{E}_{A, S, z} \mathbb{E} [\ell(A_S, z) - \ell(A_{S^{(i)}}, z)] \right\} \leq \epsilon(\theta),$$

where $S \stackrel{iid}{\sim} \mathcal{D}^m$ and $S^{(i)}$ is its copy with i -th example replaced by $z \stackrel{iid}{\sim} \mathcal{D}$.

Our definition of on-average stability resembles the notion introduced by [124]. The difference lies in the fact that we take supremum over index of replaced example. A similar notion was also used by [17] and later by [42] for analysis of a randomized aggregation scheme, however their definition involves absolute difference of losses. The dependence on θ also bears similarity to the recent work of [89], however, there, it is used in the context of uniform stability. The following theorem shows that an on-average - stable random algorithm is guaranteed to generalize in expectation.

Theorem 12. *Let an algorithm A be $\epsilon(\theta)$ -on-average stable. Then,*

$$\mathbb{E}_{S, A} [R(A_S) - \widehat{R}_S(A_S)] \leq \epsilon(\theta).$$

Proof (sketch). For any $S = \{z_i\}_{i=1}^m \stackrel{iid}{\sim} \mathcal{D}^m$, let $S^{(i)}$ be its copy with i -th example replaced by $z \stackrel{iid}{\sim} \mathcal{D}$. We relate expected empirical risk and expected risk by

$$\mathbb{E}_{S, A} [R(A_S)] = \mathbb{E}_{S, A} [\widehat{R}_S(A_S)] + \delta, \text{ where } \delta = \frac{1}{m} \sum_{i=1}^m \mathbb{E}_{S, z, A} [\ell(A_S, z) - \ell(A_{S^{(i)}}, z)].$$

We further get that

$$\delta \leq \sup_{i \in [m]} \left\{ \mathbb{E}_{S, z, A} [\ell(A_S, z) - \ell(A_{S^{(i)}}, z)] \right\} \leq \epsilon(\theta).$$

The theorem follows as by definition, the r.h.s. is bounded by $\epsilon(\theta)$. \square

5.5 Main Results

Before presenting our main results in this section, we discuss algorithmic details and assumptions. In the following we assume that the hypothesis space (parameter space) $\mathcal{H} \subseteq \mathbb{R}^d$. We will study the following variant of SGD: given a training set $S = \{z_i\}_{i=1}^m \stackrel{iid}{\sim} \mathcal{D}^m$, step sizes $\{\alpha_t\}_{t=1}^T$, random indices

$I = \{j_t\}_{t=1}^T$, and an initialization point \mathbf{w}_1 , perform updates

$$\mathbf{w}_{t+1} = \mathbf{w}_t - \alpha_t \nabla \ell(\mathbf{w}_t, z_{j_t})$$

for $T \leq m$ steps. We assume that the indices in I are sampled from the uniform distribution over $[m]$ *without* replacement, and that this is the only source of randomness for SGD. In practice this corresponds to permuting the training set before making a pass through it, as it is commonly done in practical applications. All presented theorems assume that the loss function used by SGD is non-negative, Lipschitz, and β -smooth. Examples of such commonly used loss functions are the logistic/softmax losses and neural networks with sigmoid activations. Convexity of loss functions or Lipschitzness of Hessians will only be required for some results, and we will denote it explicitly when necessary. Proofs for all the statements in this section are given in Appendix C.

5.5.1 Convex Losses

First, we present a new and data-dependent stability result for convex losses.

Theorem 13. *Assume that ℓ is convex, and that the SGD step sizes satisfy $\alpha_t \leq \frac{2}{\beta}$, $\forall t \in [T]$. Then SGD is $\epsilon(\mathcal{D}, \mathbf{w}_1)$ -on-average stable with*

$$\epsilon(\mathcal{D}, \mathbf{w}_1) \leq \frac{2L\sqrt{2\beta R(\mathbf{w}_1)}}{m} \sum_{t=1}^T \alpha_t.$$

Under the same assumptions, [57] showed a uniform stability bound $\epsilon \leq \frac{2L^2}{m} \sum_{t=1}^T \alpha_t$. Our bound differs since it involves a multiplicative risk at the initialization point, that is $\sqrt{R(\mathbf{w}_1)}$, in place of a Lipschitz constant. Thus, our bound corroborates the intuition that whenever we start at a good location of the objective function, the algorithm is more stable and thus generalizes better. In the extreme case of $R(\mathbf{w}_1) = 0$, the theorem confirms that SGD, in expectation, does not need to make any updates and is therefore perfectly stable. Note that a result of this type cannot be obtained through the more restrictive uniform stability, precisely because such bounds on the stability must hold even for a worst-case choice of data distribution and initialization. In contrast, the notion of stability we employ depends on the data-generating distribution, which allowed us to introduce dependency on the risk.

Furthermore, consider that we start at arbitrary location \mathbf{w}_1 : assuming that the loss function is bounded for a concrete \mathcal{H} and \mathcal{Z} , the rate of our bound up to a constant is no worse than that of [57]. Finally, one can always tighten this result by taking the minimum of two stability bounds.

A data-dependent argument, very similar to the one used in the proof of Theorem 13 can be also applied to prove the following *optimistic* bound for learning on convex problems with SGD.

Theorem 14. *Assume that ℓ is convex, and that the SGD step sizes satisfy $\alpha_t = \frac{c}{t} \leq \frac{2}{\beta}$, $\forall t \in [T]$. Then the output of SGD obeys*

$$\mathbb{E}_{S,A} [R(A_S) - \widehat{R}_S(A_S)] \leq \frac{4\sqrt[4]{\beta R(\mathbf{w}_1)}\sqrt{cT}}{m} \sqrt{\mathbb{E}_{S,A} [\widehat{R}_S(A_S)]} + \frac{16\sqrt{\beta R(\mathbf{w}_1)}cT}{m^2}. \quad (5.1)$$

The bound of Theorem 14 is usually called optimistic because for a vanishing expected empirical risk, it manifests the *fast* decay of the generalization error. In particular, in our case, the fast rate is $\mathcal{O}(\sqrt{R(\mathbf{w}_1)}T/m^2)$. For the common choice of $m = \mathcal{O}(T)$, this expression reduces to the more familiar looking $\mathcal{O}(\sqrt{R(\mathbf{w}_1)}/m)$. Optimistic bounds for convex learning were extensively studied in recent years in PAC and stochastic optimization settings. PAC literature approached such bounds through relative VC bounds [141], local Rademacher complexity [5], and Rademacher bounds for smooth loss classes [130]. The stochastic optimization literature usually studied optimistic bounds constructively, e.g. for stochastic mirror descent [130] when learning with smooth losses, and stochastic online Newton step [91] for exp-concave loss functions. Here we focus on the comparison to [130], since their results assume only smoothness of the loss function, while others impose stronger assumptions. In particular, we consider Corollary 3 of [130], showing the bound on the estimation error $\mathbb{E}_S[R(A_S)] - R^*(\mathcal{H})$ for stochastic optimization. In other words, their bound characterizes the estimation error, and therefore it is not directly comparable to ours. However, their proof technique also allows to obtain the bound on the generalization error of a shape similar to the consistency one (similarly as in [130, Theorem 1]). The main difference of our bound (5.1) from [130] is a novel multiplicative dependency on the risk at the initialization point $R(\mathbf{w}_1)$, and thus our bound suggests improvement over the previous one in warm-start scenarios, especially where the initialization point is close to the optimal one.

5.5.2 Non-convex Losses

Now we state a new stability result for non-convex losses.

Theorem 15. *Assume that $\ell(\cdot, z) \in [0, 1]$ and has a ρ -Lipschitz Hessian, and that step sizes of a form $\alpha_t = \frac{c}{t}$ satisfy $c \leq \min\left\{\frac{1}{\beta}, \frac{1}{4(2\beta \ln(T))^2}\right\}$. Then SGD is $\epsilon(\mathcal{D}, \mathbf{w}_1)$ -on-average stable with*

$$\epsilon(\mathcal{D}, \mathbf{w}_1) \leq \frac{1 + \frac{1}{c\gamma}}{m} (2cL^2)^{\frac{1}{1+c\gamma}} \left(\mathbb{E}_{S,A} [R(A_S)] \cdot T \right)^{\frac{c\gamma}{1+c\gamma}}, \quad \text{where} \quad (5.2)$$

$$\gamma := \min\left\{\beta, \mathbb{E}_z [\|\nabla^2 \ell(\mathbf{w}_1, z)\|_2] + c\rho(1 + \ln(T))\sqrt{2\beta R(\mathbf{w}_1)}\right\}. \quad (5.3)$$

In particular, γ characterizes how the curvature at the initialization point affects stability, and hence the generalization error of SGD. Since γ heavily affects the rate of convergence in (5.2), and in most situations a smaller γ yields higher stability, we now look at a few cases of its behavior. Consider a regime such that γ is of the order $\tilde{\Theta}(\mathbb{E}[\|\nabla^2 \ell(\mathbf{w}_1, z)\|_2] + \sqrt{R(\mathbf{w}_1)})$, or in other words, that stability is controlled by the curvature and the risk of the initialization point \mathbf{w}_1 . This suggests that starting from a point in a less curved region with low risk should yield higher stability, and therefore as predicted by our theory, allow for faster generalization. In addition, we observe that the considered stability regime offers a principled way to pre-screen a good initialization point in practice, by choosing the one that minimizes the spectral norm of the Hessian and the risk.

Next, we focus on a more specific case. Suppose that we choose a step size $\alpha_t = \frac{c}{t}$ such that $\gamma \leq \tilde{\Theta}(\mathbb{E}[\|\nabla^2 \ell(\mathbf{w}_1, z)\|_2])$, yet not too small, so that the empirical risk can still be decreased. Then, stability is dominated by the curvature around \mathbf{w}_1 . Indeed, lower generalization errors on non-convex problems, such as training deep neural networks, have been observed empirically when SGD is actively guided [61, 54, 23] or converges to solutions with low curvature [68]. However, to the best of our knowledge,

Theorem 15 is the first to establish a theoretical link between the curvature of the loss function and the generalization ability of SGD in a data-dependent sense.

Theorem 15 allows us to show the following statement that further reinforces the effect of the initialization point on the generalization error.

Corollary 2. *Under the conditions of Theorem 15 we have that SGD is $\epsilon(\mathcal{D}, \mathbf{w}_1)$ -on-average stable with*

$$\epsilon(\mathcal{D}, \mathbf{w}_1) \leq \mathcal{O} \left(\frac{1 + \frac{1}{c\gamma}}{m} (R(\mathbf{w}_1) \cdot T)^{\frac{c\gamma}{1+c\gamma}} \right). \quad (5.4)$$

We take a moment to discuss the role of the risk term in $(R(\mathbf{w}_1) \cdot T)^{\frac{c\gamma}{1+c\gamma}}$. Observe that $\epsilon(\mathcal{D}, \mathbf{w}_1) \rightarrow 0$ as $R(\mathbf{w}_1) \rightarrow 0$, in other words, the generalization error approaches zero as the risk of the initialization point vanishes. This is an intuitive behavior, however, uniform stability does not capture this due to its distribution-free nature. Finally, we note that [57, Theorem 3.8] showed a bound similar to (5.2), however, in place of γ their bound has a Lipschitz constant of the gradient. The crucial difference lies in the term γ which is now not merely a Lipschitz constant, but rather depends on the data-generating distribution and initialization point of SGD. We compare to their bound by considering the worst case scenario, namely, that SGD is initialized in a point with high curvature, or altogether, that the objective function is highly curved everywhere. Then, at least our bound is no worse than the one of [57], since $\gamma \leq \beta$.

Theorem 15 also allows us to prove an optimistic generalization bound for learning with SGD on non-convex objectives.

Corollary 3. *Under the conditions of Theorem 15 we have that the output of SGD obeys*

$$\mathbb{E}_{S,A} [R(A_S) - \widehat{R}_S(A_S)] \leq \mathcal{O} \left(\frac{1 + \frac{1}{c\gamma}}{m} \cdot \max \left\{ \left(\mathbb{E}_{S,A} [\widehat{R}_S(A_S)] \cdot T \right)^{\frac{c\gamma}{1+c\gamma}}, \left(\frac{T}{m} \right)^{c\gamma} \right\} \right).$$

An important consequence of Corollary 3 is that for a vanishing expected empirical risk, in particular for $\mathbb{E}_{S,A} [\widehat{R}_S(A_S)] = \mathcal{O} \left(\frac{T^{c\gamma}}{m^{1+c\gamma}} \right)$, the generalization error behaves as $\mathcal{O} \left(\frac{T^{c\gamma}}{m^{1+c\gamma}} \right)$. Considering the full pass, that is $m = \mathcal{O}(T)$, we have an optimistic generalization error of order $\mathcal{O}(1/m)$ instead of $\mathcal{O}(m^{-\frac{1}{1+c\gamma}})$. We note that PAC bounds with a similar optimistic message (although not directly comparable), but without curvature information can also be obtained through empirical Bernstein bounds as in [96]. However, a PAC bound does not suggest a way to minimize non-convex empirical risk in general, where, on the other hand, SGD is known to work reasonably well.

Tightness of Non-convex Bounds

Next we empirically assess the tightness of our non-convex generalization bounds on real data. In the following experiment we train a neural network with three convolutional layers interlaced with max-pooling, followed by a fully connected layer with 16 units, on the MNIST dataset. This totals in a model with 18K parameters. Figure 5.1 compares our data-dependent bound (5.2) to the distribution-free one of [57, Theorem 3.8]. As a reference we also include an empirical estimate of the generalization error

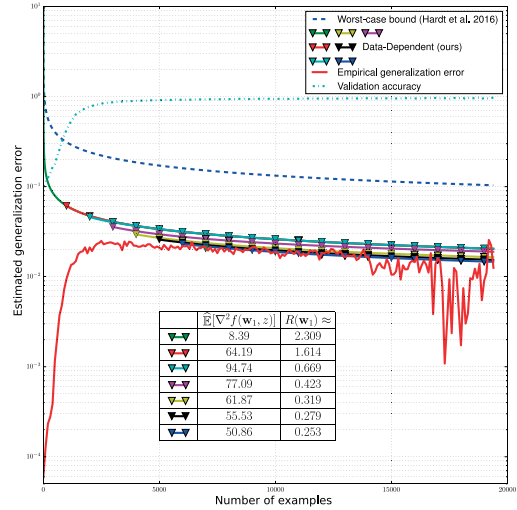
taken as an absolute difference of the validation and training average losses. Since our bound also depends on the initialization point, we plot (5.2) for multiple “warm-starts”, with SGD initialized from a pre-trained position. We consider 7 such warm-starts at every 200 steps, and report data-dependent quantities used to compute (5.2) just beneath the graph. Our first observation is that, clearly, the data-dependent bound gives tighter estimate, by roughly one order of magnitude. Second, simulating start from a pre-trained position suggests even tighter estimates: we suspect that this is due to decreasing validation error which is used as an empirical estimate for $R(\mathbf{w}_1)$ which effects heavily bound (5.2).

We compute an empirical estimate of the expected Hessian spectral norm by the power iteration method using an efficient Hessian-vector multiplication method [108]. Since the bounds depend on constants L , β , and ρ , we estimate them heuristically by tracking maximal values of the gradient and Hessian norms throughout optimization. We compute bounds with estimates $\hat{L} = 78.72$, $\hat{\beta} = 1692.28$, $\hat{\rho} = 3823.73$, and $c = 10^{-3}$. Note that actual constants can only be larger than estimated ones, and thus, discrepancy between the worst-case and the data-dependent bound can be even larger.

5.5.3 Application to Transfer Learning

One example application of the data-dependent bounds presented before lies in *Transfer Learning (TL)*, where we are interested in achieving faster generalization on a *target* task by exploiting side information that originates from different but related *source* tasks. The literature on TL explored many ways to do so, and here we will focus on the one that is most compatible with our bounds. More formally, suppose that the *target* task at hand is characterized by a joint probability distribution \mathcal{D} , and as before we have a training set $S \sim \mathcal{D}^m$. Some TL approaches also assume access to the data sampled from the distributions associated with the *source* tasks. Here we follow a conservative approach – instead of the source data, we receive a set of *source* hypotheses $\{\mathbf{w}_k^{\text{src}}\}_{k=1}^K \subset \mathcal{H}$, trained on the source tasks. The goal of a learner is to come up with a target hypothesis, which in the optimistic scenario generalizes better by relying on source hypotheses. In the TL literature this is known as HTL [77], that is, we transfer from the source hypotheses which act as a proxy to the source tasks and the risk $R(\mathbf{w}_k^{\text{src}})$ quantifies how much the source and target tasks are related. In the following we will consider SGD for HTL, where the source hypotheses act as initialization points. First, consider learning with convex losses: Theorem 13 depends on $R(\mathbf{w}_1)$, thus it immediately quantifies the relatedness of the source and target tasks. So it is enough to pick the point that minimizes the stability bound to transfer from the most related source. Then, bounding $R(\mathbf{w}_k^{\text{src}})$ by $\hat{R}_S(\mathbf{w}_k^{\text{src}})$ through Hoeffding bound along with union bound gives with high probability

Figure 5.1 – Empirical tightness of data-dependent and uniform generalization bounds evaluated by training a convolutional neural network.



that

$$\min_{k \in [K]} \epsilon(\mathcal{D}, \mathbf{w}_k^{\text{src}}) \leq \min_{k \in [K]} \mathcal{O} \left(\widehat{R}_S(\mathbf{w}_k^{\text{src}}) + \sqrt{\frac{\log(K)}{m}} \right).$$

Hence, the most related source is the one that simply minimizes empirical risk. Similar conclusions drawn in the HTL literature, albeit in the context of ERM. Matters are slightly more complicated in the non-convex case. We take a similar approach, however, now we minimize stability bound (5.4), and for the sake of simplicity assume that we make a full pass over the data, so $T = m$. Minimizing the following empirical upper bound selects the best source.

Proposition 1. *Let $\widehat{\gamma}_k^\pm = \frac{1}{m} \sum_{i=1}^m \|\nabla^2 \ell(\mathbf{w}_k^{\text{src}}, z_i)\|_2 + \lambda \sqrt{\widehat{R}_S(\mathbf{w}_k^{\text{src}}) \pm \mathcal{O}(\sqrt{\log(K)/m})}$, where $\lambda = c\rho(1 + \ln(T))\sqrt{2\beta}$. Then with high probability we have that*

$$\min_{k \in [K]} \epsilon(\mathcal{D}, \mathbf{w}_k^{\text{src}}) \leq \min_{k \in [K]} \mathcal{O} \left(\left(1 + \frac{1}{c\widehat{\gamma}_k^-} \right) \widehat{R}_S(\mathbf{w}_k^{\text{src}})^{\frac{c\widehat{\gamma}_k^+}{1+c\widehat{\gamma}_k^+}} \cdot \frac{\sqrt{\log(K)}}{m^{\frac{1}{1+c\widehat{\gamma}_k^+}}} \right).$$

We also note that there is no restriction on the origin of the source hypotheses $\mathbf{w}_k^{\text{src}}$. In general, these can even be random guesses, in which case we would be pre-screening a good starting position. Finally, $\widehat{\gamma}_k$ involves estimation of the spectral norm of the Hessian, which is computationally cheaper to evaluate compared to the complete Hessian matrix [108]. This is particularly relevant for deep learning, where computation of the Hessian matrix can be prohibitively expensive.

5.6 Conclusion

In this chapter we proved data-dependent stability bounds for SGD and revisited its generalization ability. We presented novel bounds for convex and non-convex smooth loss functions, partially controlled by data-dependent quantities, while previous stability bounds for SGD were derived through the worst-case analysis. In particular, for non-convex learning, we demonstrated theoretically that generalization of SGD is heavily affected by the expected curvature around the initialization point. We demonstrated empirically that our bound is indeed tighter compared to the uniform one. In addition, our data-dependent analysis also allowed us to show optimistic bounds on the generalization error of SGD, which exhibit fast rates subject to the vanishing empirical risk of the algorithm's output. Finally, we exploited this fact, presenting a simple and data-driven hypothesis transfer learning approach which directly minimizes the bound.

Algorithms **Part II**

6 Greedy Algorithms for Hypothesis Transfer Learning

The material of this chapter is based on the publication:

I. Kuzborskij, F. Orabona, and B. Caputo. Scalable Greedy Algorithms for Transfer Learning. In *Computer Vision and Image Understanding* 156 (2017): 174-185.

The doctoral candidate formalized the problem, designed the algorithms, evaluated the algorithms, proved the theoretical results, and wrote most of the publication.

6.1 Overview

Over the last few years, the visual recognition research landscape has been heavily dominated by Convolutional Neural Networks, thanks to their ability to leverage effectively massive amounts of training data [38]. This trend dramatically confirms the widely accepted truth that any learning algorithm performs better when trained on a lot of data. This is even more true when facing noisy or “hard” problems such as large-scale recognition [35]. However, when tackling large scale recognition problems, gathering substantial training data for all classes considered might be challenging, if not almost impossible. The occurrence of real-world objects follows a long tail distribution, with few objects occurring very often, and many with few instances. Hence, for the vast majority of visual categories known to human beings, it is extremely challenging to collect training data of the order of $10^4 - 10^5$ instances. The “long tail” distribution problem was noted and studied by Salakhutdinov *et al.* [120], who proposed to address it by leveraging on the prior knowledge available to the learner. Indeed, learning systems are often not trained from scratch: usually they can be build on previous knowledge acquired over time on related tasks [105]. The scenario of learning from few examples by *transferring* from what is already known to the learner is collectively known as Transfer Learning. The target domain usually indicates the task at hand and the source domain the prior knowledge of the learner.

Most of the transfer learning algorithms proposed in the recent years focus on the object detection task (binary transfer learning), assuming access to the training data coming from both source and target domains [105]. While featuring good practical performance [53], they often demonstrate poor

scalability w.r.t. the number of sources. An alternative direction, known as a HTL [76, 11], consists in transferring from the *source hypotheses*, that is classifiers trained from them. This framework is practically very attractive [2, 137, 78], as it treats source hypotheses as black boxes without any regard of their inner workings.

The goal of this chapter is to develop an HTL algorithm able to deal effectively and efficiently with a large number of sources, where our working definition of large is at least 10^3 . Note that this order of magnitude is also the current frontier in visual classification [35]. To this end, we cast Hypothesis Transfer Learning as a problem of *efficient selection* and *combination* of source hypotheses from a large pool. We pose it as a subset selection problem building on results from the literature [30, 150]. We present¹ a greedy algorithm, GreedyTL, which attains state of the art performance even with a very limited amount of data from the target domain. Moreover, we also present a randomized approximate variant of GreedyTL, called GreedyTL-59, that has a complexity *independent* from the number of sources, with no loss in performance. Our key contribution is an L_2 -regularized variant of the Forward Regression algorithm [58]. Since our algorithm can be viewed as a feature selection algorithm as well as an hypothesis transfer learning approach, we extensively evaluate it against popular feature selection and transfer learning baselines. We empirically demonstrate that GreedyTL dominates all the baselines in most small-sample transfer learning scenarios, thus proving the critical role of regularization in our formulation. Experiments over three datasets show the power of our approach: we obtain state of the art results in tasks with up to 1000 classes, totalling 1.2 million examples, with only 11 to 20 training examples from the target domain. We back our experimental results by proving generalization bounds showing that, under reasonable assumptions on the source hypotheses, our algorithm is able to learn effectively with very limited data.

The rest of the chapter is organised as follows: after a review of the relevant literature in the field (section 6.2), we cast the transfer learning problem in the subset selection framework (section 6.3). We then define our GreedyTL, in section 6.4, deriving its formulation, analysing its computational complexity and its theoretical properties. Section 6.5 describes our experimental evaluation and discuss the related findings. We conclude with an overall discussion and presenting possible future research avenues.

6.2 Related Work

The problem of how to exploit prior knowledge when attempting to solve a new task with limited, if any, annotated samples is vastly researched. Previous work spans from transfer learning [105] to domain adaptation [118, 8], and dataset bias [139]. Here we focus on the first. In the literature there are several transfer learning settings [8, 118, 53]. The oldest and most popular is the one assuming access to the data originating from both the source and the target domains [8, 53, 118, 40, 122, 135, 73]. There, one typically assumes that plenty of source data are available, but access to the target data is limited: for instance, we can have many unlabeled examples and only few labeled ones [106]. Here we focus on the Hypothesis Transfer Learning framework (HTL, [76, 11]). It requires to have access only to *source hypotheses*, that is classifiers or regressors trained on the source domains. No assumptions are made on how these source hypotheses are trained, or about their inner workings: they are treated

¹We build upon preliminary results presented in [79].

as “black boxes”, in spirit similar to classifier-generated visual descriptors such as Classemes [12] or Object-Bank [85]. Several works proposed HTL for visual learning [2, 137, 101], some exploiting more explicitly the connection with classeme-like approaches [65, 107], demonstrating an intriguing potential. Although offering scalability, HTL-based approaches proposed so far have been tested on problems with less than a few hundred of sources [137], already showing some difficulties in selecting informative sources.

Recently, the growing need to deal with large data collections [35, 24] has started to change the focus and challenges of research in transfer learning. Scalability with respect to the amount of data and the ability to identify and separate informative sources from those carrying noise for the task at hand have become critical issues. Some attempts have been made in this direction. For example, [87, 142] used taxonomies to leverage learning from few examples on the SUN09 dataset. In [87], the authors attacked the transfer learning problem on the SUN09 dataset by using additional data from another dataset. Zero-shot approaches were investigated by [117] on a subset of the Imagenet dataset. Large-scale visual detection has been explored by [142]. However, all these approaches assume access to all source training data. A slightly different approach to transfer learning that aimed to circumvent this limitation is the reuse of a large convolutional neural network pre-trained on a large visual recognition dataset. The simplest approach is to use the outputs of intermediate layers of such a network, such as DeCAF [38] or Caffe [64]. A more sophisticated way of reuse is fine-tuning, a kind of warm-start, that has been successfully exploited in visual detection [51] and domain adaptation [47, 90].

In many of these works the use of richer sources of information has been supported by an increase in the information available in the target domain as well. From an intuitive point of view, this corresponds to having more data points than dimensions. Of course, this makes the learning and selection process easier, but in many applications it is not a reasonable hypothesis. Also, none of the proposed algorithms has a theoretical backing.

While not explicitly mentioned before, the problem outlined above can also be viewed as a learning scenario where the number of features is by far larger than the number of training examples. Indeed, learning with classeme-like features [12, 85] when only few training examples are available can be seen as a Hypothesis Transfer Learning problem. Clearly, a pure empirical risk minimization would fail due to severe overfitting. In machine learning and statistics this is known as a feature selection problem, and is usually addressed by constraining or penalizing the solution with sparsity-inducing norms. One important sparsity constraint is a non-convex L_0 pseudo-norm constraint $\|\mathbf{w}\|_0 \leq k$, that simply corresponds to choosing up to k non-zero components of a vector \mathbf{w} . One usually resorts to the *subset selection* methods, and greedy algorithms for obtaining solutions under this constraint [30, 31, 150, 151]. However, in some problems introducing an L_0 constraint might be computationally difficult. There, a computationally easier alternative is a convex relaxation of L_0 , the L_1 regularization. Empirical error minimization with L_1 penalty with various loss functions (for square loss, this is known as Lasso) has many favorable properties and is well studied theoretically [18]. Yet, the L_1 penalty is known to suffer from several limitations, one of which is poor empirical performance when there are many correlated features. Perhaps the most famous way to resolve this issue is an *elastic net* regularization which is a weighted mixture of L_1 and squared L_2 penalties [58]. Since our work partially falls into the category of feature selection, we have extensively evaluated the aforementioned baselines in our task. As it will be shown below, none of them achieves competitive performances compared to our approach.

6.3 Transfer Learning through Subset Selection

Additional definitions. We introduce in this section additional definitions we will use in the rest of the chapter. For $\mathbf{x} \in \mathbb{R}^d$, the *support* of \mathbf{x} is $\text{supp}(\mathbf{x}) = \{i \in \{1, \dots, d\} : x_i \neq 0\}$. Then, $\|\mathbf{x}\|_0 = |\text{supp}(\mathbf{x})|$. To measure the accuracy of a learning algorithm, we have a non-negative *loss* function $\ell(h(\mathbf{x}), y)$, which measures the cost incurred in predicting $h(\mathbf{x})$ instead of y . In particular, we will focus on the square loss, $\ell(h(\mathbf{x}), y) = (h(\mathbf{x}) - y)^2$, for its appealing computational properties.

Source Selection. Assume that we are given a finite source hypothesis set $\{h_i^{\text{src}}\}_{i=1}^n$ and the training set $\{(\mathbf{x}_i, y_i)\}_{i=1}^m$. As in previous works [94, 137, 65], we consider the target hypothesis to be of the form

$$h_{\mathbf{w}, \boldsymbol{\beta}}^{\text{trg}}(\mathbf{x}) = \mathbf{w}^\top \mathbf{x} + \sum_{i=1}^n \beta_i h_i^{\text{src}}(\mathbf{x}), \quad (6.1)$$

where \mathbf{w} and $\boldsymbol{\beta}$ are found by the learning procedure. The essential parameter here is $\boldsymbol{\beta}$, that is the one controlling the influence of each source hypothesis. Previous works in transfer learning have focused on finding $\boldsymbol{\beta}$ such that it minimizes the error on the training set, subject to some condition on $\boldsymbol{\beta}$. In particular, [137] proposed to minimize the leave-one-out error w.r.t. $\boldsymbol{\beta}$, subject to $\|\boldsymbol{\beta}\|_2 \leq \tau$, which is known to improve generalization for the right choice of τ [76]. A slightly different approach is to use $\|\boldsymbol{\beta}\|_1 \leq \tau$ regularization for this purpose [137], that induces solutions with most of the coefficients equal to 0, thus assuming that the optimal $\boldsymbol{\beta}$ is sparse.

In this chapter we embrace a weaker assumption, namely, there exist up to k sources that collectively improve the generalization on the target domain. Thus, we pose the problem of the Source Selection as a minimization of the regularized empirical risk on the target training set, while constraining the number of selected source hypotheses.

k -Source Selection. Given the training set $\{([\mathbf{x}_i^\top, h_1^{\text{src}}(\mathbf{x}_i), \dots, h_n^{\text{src}}(\mathbf{x}_i)]^\top, y_i)\}_{i=1}^m$ we have the optimal target hypothesis $h_{\mathbf{w}^*, \boldsymbol{\beta}^*}^{\text{trg}}$ by solving,

$$\begin{aligned} (\mathbf{w}^*, \boldsymbol{\beta}^*) = \underset{\mathbf{w}, \boldsymbol{\beta}}{\text{argmin}} \left\{ \widehat{R}(h_{\mathbf{w}, \boldsymbol{\beta}}^{\text{trg}}) + \lambda \|\mathbf{w}\|_2^2 + \lambda \|\boldsymbol{\beta}\|_2^2 \right\}, \\ \text{s.t. } \|\mathbf{w}\|_0 + \|\boldsymbol{\beta}\|_0 \leq k. \end{aligned} \quad (6.2)$$

Notably, the problem (6.2) is a special case of the *Subset Selection* problem [30]: choose a subset of size k from the n observation variables, which collectively give the best prediction on the variable of interest. However, the Subset Selection problem is **NP-hard** [30]. In practice we can resort to algorithms generating approximate solutions, for many of which we have approximation guarantees. Hence, due to the extensive practical and theoretical results, we will treat the k -Source Selection as a Subset Selection problem, building atop of existing guarantees.

We note that our formulation, (6.2), differs from the classical subset selection for the fact that it is L_2 -regularized. This technical modification makes an essential practical and theoretical difference and it is the crucial part of our algorithm. First, L_2 regularization is known to improve the generalization ability of empirical risk minimization. Second, we show that regularization also improves the quality

of the approximate solution in situations when the sources, or features, are correlated. At the same time, the experimental evaluation corroborates our theoretical findings: Our formulation substantially outperforms standard subset selection, feature selection algorithms, and transfer learning baselines.

6.4 Greedy Algorithm for k -Source Selection

In this section we state the algorithm proposed in this chapter, GreedyTL². In the following we will denote by $U = \{1, \dots, n + d\}$ the index set of all available source hypotheses and features, and by S , the index set of selected ones.

GreedyTL. Let $\mathbf{X} \in \mathbb{R}^{m \times d}$ and $\mathbf{y} \in \{+1, -1\}^m$ be the zero-mean unit-variance training set, $\{h_i^{src}\}_{i=1}^n$, source hypothesis set, and k and λ , regularization parameters. Then, denote $\mathbf{C} = \mathbf{Z}^\top \mathbf{Z}$ and $\mathbf{b} = \mathbf{Z}^\top \mathbf{y}$, where $\mathbf{Z} = \begin{bmatrix} \mathbf{X} & h_1^{src}(\mathbf{x}_1) & \dots & h_n^{src}(\mathbf{x}_1) \\ \dots & \dots & \dots & \dots \\ h_1^{src}(\mathbf{x}_m) & \dots & \dots & h_n^{src}(\mathbf{x}_m) \end{bmatrix}$, and select set S of size k as follows: (I) Initialize $S \leftarrow \emptyset$ and $U \leftarrow \{1, \dots, n + d\}$. (II) Keep populating S with $i \in U$, that maximize $\mathbf{b}_S^\top ((\mathbf{C} + \lambda \mathbf{I})_S^{-1})^\top \mathbf{b}_S$, as long as $|S| \leq k$ and U is non-empty.

In this basic formulation, the algorithm requires to invert a $(d + n)$ -by- $(d + n)$ matrix at each iteration of a greedy search. Clearly, this naive approach gets prohibitive with the growth of the number of source hypotheses, feature dimensions, and desired subset size, since its computational complexity would be in $\mathcal{O}(k(d + n)^4)$. However, we note that in transfer learning one typically assumes that the training set is much smaller than the number of sources and feature dimension. For this reason we apply rank-one updates w.r.t. the dual solution of regularized subset selection, so that the size of the inverted matrix does not change. A similar approach for feature selection with LSSVM was proposed by [100]. The computational complexity then improves to $\mathcal{O}(k(d + n)m^2)$. We present the pseudocode of such a variant of our algorithm, **GreedyTL with Rank-One Updates** in Algorithm 1. The computational complexity of the operations is shown at the end of each line.

Derivation of the Algorithm. We derive GreedyTL by extending the well known Forward Regression (FR) algorithm [30], which gives an approximation to the subset selection problem, one problem of interest. FR is known to find a good approximation as far as features are uncorrelated [30]. In the following, we build upon FR by introducing a Tikhonov (L_2) regularization into the formulation. The purpose of regularization is twofold: first, it improves the generalization ability of the empirical risk minimization, and second, it makes the algorithm more robust to the feature correlations, thus opting to find a better approximate solution.

First, we briefly formalize the subset selection problem. In a subset selection problem one tries to achieve a good prediction accuracy on the *predictor* random variable Y , given a linear combination of a subset of the *observation* random variables $\{X_i\}_{i=1}^n$. The least squares subset selection then reads as

$$\min_{|S|=k, \mathbf{w} \in \mathbb{R}^k} \mathbb{E} \left[\left(Y - \sum_{i \in S} w_i X_i \right)^2 \right].$$

Now denote the covariance matrix of zero-mean unit-variance observation random variables by \mathbf{C}

²Source code is available at <https://iljaku.github.io>

Chapter 6. Greedy Algorithms for Hypothesis Transfer Learning

Algorithm 1 GreedyTL with Rank-One Updates

Input: $\mathbf{Z} \in \mathbb{R}^{m \times (d+n)}$ – m examples formed from features and source predictions,

- 1: $\mathbf{y} \in \{-1, +1\}^m$ – labels,
- 2: $k \in \{1, \dots, d+n\}, \lambda \in \mathbb{R}_+$ – hyperparameters.

Output: \mathbf{w} – target predictor.

- 3: $U \leftarrow \{1, \dots, d+n\}$ ▷ All candidates
- 4: $S \leftarrow \emptyset$ ▷ Selected sources and features
- 5: $\mathbf{K} \leftarrow [\mathbf{0}, \dots, \mathbf{0}] \in \mathbb{R}^{m \times m}$
- 6: $\mathbf{G} \leftarrow \lambda^{-1} \mathbf{I} \in \mathbb{R}^{m \times m}$
- 7: **while** $U \neq \emptyset$ **and** $|S| \leq k$ **do**
- 8:

$$i^* \leftarrow \operatorname{argmax}_{i \in U} \left\{ \mathbf{y}^\top (\mathbf{K} + \mathbf{z}_i \mathbf{z}_i^\top) \mathbf{G}' \mathbf{y} \mid \mathbf{G}' \leftarrow \mathbf{G} - \frac{\mathbf{G} \mathbf{z}_i \mathbf{z}_i^\top \mathbf{G}}{1 + \mathbf{z}_i^\top \mathbf{G} \mathbf{z}_i} \right\}$$

▷ $\mathcal{O}((d+n)(m^2+m))$

9:

10: Computing \mathbf{G}' : ▷ $\mathcal{O}(m^2+m)$

11: Computing score of i : ▷ $\mathcal{O}(m^2+m)$

12: $S \leftarrow S \cup \{i^*\}$

13: $U \leftarrow U \setminus \{i^*\}$

14: $\mathbf{K} \leftarrow \mathbf{K} + \mathbf{z}_{i^*} \mathbf{z}_{i^*}^\top$ ▷ $\mathcal{O}(m^2)$

15: $\mathbf{G} \leftarrow \mathbf{G} - \frac{\mathbf{G} \mathbf{z}_{i^*} \mathbf{z}_{i^*}^\top \mathbf{G}}{1 + \mathbf{z}_{i^*}^\top \mathbf{G} \mathbf{z}_{i^*}}$ ▷ $\mathcal{O}(m^2+m)$

16:

17: **end while** ▷ $\mathcal{O}(k(d+n)m^2)$

18: $\mathbf{w} \leftarrow \mathbf{0} \in \mathbb{R}^{d+n}$

19: $w_i \leftarrow \mathbf{z}_i^\top \mathbf{G} \mathbf{y}, \forall i \in S$

6.4. Greedy Algorithm for k -Source Selection

(a correlation matrix), and the correlations between Y and $\{X_i\}_{i=1}^m$ as \mathbf{b} . Note that the zero-mean unit-variance assumption will be necessary to prove the theoretical guarantees of our algorithm. By virtue of the analytic solution to least-squares and using the introduced notation, we can also state the equivalent *Subset Selection problem*: $\max_{|S|=k} \mathbf{b}_S^\top (\mathbf{C}_S^{-1})^\top \mathbf{b}_S$. However, our goal is to obtain the solution to (6.2), or an *L2-regularized* subset selection. Similarly to the unregularized subset selection, it is easy to see that (6.2) is equivalent to $\max_{|S|=k} \mathbf{b}_S^\top ((\mathbf{C}_S + \lambda \mathbf{I})^{-1})^\top \mathbf{b}_S$. As said above, the Subset Selection problem is **NP-hard**, however, there are several ways to approximate it in practice [31]. We choose FR for this task for its simplicity, appealing computational properties and provably good approximation guarantees. Now, to apply FR to our problem, all we have to do is to provide it with normalized matrix $(\mathbf{C} + \lambda \mathbf{I})^{-1}$ instead of \mathbf{C}^{-1} .

Approximated Randomized Greedy Algorithm. As mentioned above, the complexity of GreedyTL is linear in $d + n$, the number of features and the size of the source hypothesis set. In particular, the search in U for the index to add to S is responsible for the dependency on $d + n$. Here we show how to approximate this search with a randomized strategy. We will use the following theorem.

Theorem 16 ([127](Theorem 6.33)). *Denote by $M := \{x_1, \dots, x_m\} \subset \mathbb{R}$ a set of cardinality m , and by $\tilde{M} \subset M$ a random subset of size \tilde{m} . Then the probability that $\max \tilde{M}$ is greater or equal than n elements of M is at least $1 - (\frac{n}{m})^{\tilde{m}}$.*

The surprising consequence is that, in order to approximate the maximum over a set, we can use a random subset of size $\mathcal{O}(1)$. In particular, if we want to obtain results in the $\frac{n}{m}$ percentile range with $1 - \eta$ confidence, we use³ $\tilde{m} = \frac{\log(\eta)}{\log \frac{n}{m}}$. Practically, if we desire values that are better than 95% of all other estimates with $1 - 0.05$ probability, then 59 samples are sufficient. This rule is commonly called the 59-trick and it has been widely used to speed-up a wide range of algorithms with negligible loss of accuracy, e.g. [37, 128]. Indeed, as we will show in Section 6.5.4, we virtually don't lose any accuracy using this strategy.

With the 59-trick, the search in U becomes a search for the maximum over a random set of size 59. So, the overall complexity is reduced to $\mathcal{O}(km^2)$, that is *independent* from all the quantities that are expected to be big.

Theoretical Guarantees. We now focus on the analysis of the generalization properties of GreedyTL for solving k -Source Selection problem (6.2). Throughout this paragraph we will consider a truncated target predictor $h_{\mathbf{w}, \boldsymbol{\beta}}^{\text{trg}}(\mathbf{x}) := \Upsilon(\mathbf{w}^\top \mathbf{x} + \sum_{i=1}^n \beta_i h_i^{\text{src}}(\mathbf{x}))$, with $\Upsilon(a) := \min\{\max\{a, -1\}, 1\}$. We will also use big-O notation $\tilde{\mathcal{O}}$ to indicate the suppression of a logarithmic factor, in other words, $f(x) \in \tilde{\mathcal{O}}(g(x))$ is a short notation for $\exists n : f(x) \in \mathcal{O}(g(x) \log^n g(n))$. First we state the bound on the risk of an approximate solution returned by GreedyTL.⁴

Theorem 17. *Let GreedyTL generate the solution $(\hat{\mathbf{w}}, \hat{\boldsymbol{\beta}})$, given the training set (\mathbf{X}, \mathbf{y}) , source hypotheses $\{h_i^{\text{src}}\}_{i=1}^n$ with $\tau_\infty^{\text{src}} := \max_i \{\|h_i^{\text{src}}\|_\infty^2\}$, hyperparameters λ and k . Then with high probability,*

$$R\left(h_{\hat{\mathbf{w}}, \hat{\boldsymbol{\beta}}}^{\text{trg}}\right) - \hat{R}\left(h_{\hat{\mathbf{w}}, \hat{\boldsymbol{\beta}}}^{\text{trg}}\right) \leq \tilde{\mathcal{O}}\left(\frac{1 + k\tau_\infty^{\text{src}}}{\lambda m} + \sqrt{\hat{R}^{\text{src}} \frac{1 + k\tau_\infty^{\text{src}}}{\lambda m}}\right),$$

³Note that the formula for \tilde{m} in [127] contains an error, the correct one is the one we report.

⁴Proofs for theorems can be found in the appendix.

where

$$\widehat{R}^{\text{src}} := \frac{1}{m} \sum_{i=1}^m \ell \left(y_{i, \text{T}} \left(\sum_{j \in \text{supp}(\widehat{\beta})} \widehat{\beta}_j h_j^{\text{src}}(\mathbf{x}_i) \right) \right).$$

This results in a generalization bound which tells us how close the performance of the algorithm on the test set will be to the one on the training set. The key quantity here is \widehat{R}^{src} , which captures the quality of the sources selected by the algorithm. To understand its impact, assume that $\lambda = \mathcal{O}(1)$. The bound has two terms, a fast one of the order of $\tilde{\mathcal{O}}(k/m)$ and a slow one of the order $\tilde{\mathcal{O}}\left(\sqrt{\widehat{R}^{\text{src}} k/m}\right)$. When m goes to infinity and $\widehat{R}^{\text{src}} \neq 0$ the slow term will dominate the convergence rate, giving us a rate of the order of $\tilde{\mathcal{O}}\left(\sqrt{\widehat{R}^{\text{src}} k/m}\right)$. If $\widehat{R}^{\text{src}} = 0$ the slow term completely disappears, giving us a so called fast rate of convergence of $\tilde{\mathcal{O}}(k/m)$. On the other hand, for any finite m of the order of $\tilde{\mathcal{O}}(k/\widehat{R}^{\text{src}})$, we still have a rate of the order of $\tilde{\mathcal{O}}(k/m)$. Hence, the quantity \widehat{R}^{src} will govern the finite sample and asymptotic behavior of the algorithm, predicting a faster convergence in both regimes when it is small. In other words, when the source and target tasks are similar, TL facilitates a faster convergence of the empirical risk to the risk. A similar behavior was already observed in [76, 11].

However, one might ask what happens when the selected sources are providing bad predictions. Since $\widehat{R}^{\text{src}} \leq 1$, due to truncation, the empirical risk converges to the risk at the standard rate $\tilde{\mathcal{O}}(\sqrt{k/m})$, the same one we would have without any transferring from the source classifiers.

We now present another result that upper bounds the difference between the risk of solution of the algorithm and the empirical risk of the optimal solution to the k -Source Selection problem.

Theorem 18. *In addition to conditions of Theorem 17, let $(\mathbf{w}^*, \boldsymbol{\beta}^*)$ be the optimal solution to (6.2). Given a sample correlation matrix $\widehat{\mathbf{C}}$, assume that $\widehat{C}_{i,j \neq i} \leq \gamma < \frac{1+\lambda}{6k}$, and $\epsilon := \frac{16(k+1)^2 \gamma}{1+\lambda}$. Then with high probability,*

$$R\left(h_{\widehat{\mathbf{w}}, \widehat{\boldsymbol{\beta}}}^{\text{trg}}\right) - \widehat{R}\left(h_{\mathbf{w}^*, \boldsymbol{\beta}^*}^{\text{trg}}\right) \leq (1+\epsilon) \widehat{R}_\lambda^{\text{src}} + \tilde{\mathcal{O}}\left(\frac{1+k\tau_\infty^{\text{src}}}{\lambda m} + \sqrt{\widehat{R}_\lambda^{\text{src}} \frac{1+k\tau_\infty^{\text{src}}}{\lambda m}}\right),$$

where $\widehat{R}_\lambda^{\text{src}} := \min_{|S| \leq k} \left\{ \frac{\lambda}{|S|} + \frac{1}{|S|} \sum_{i \in S} \widehat{R}(h_i^{\text{src}}) \right\}$.

To analyze the implications of Theorem 18, let us consider a few interesting cases. Similarly as done before, the quantity $\widehat{R}_\lambda^{\text{src}}$ captures how well the source hypotheses are aligned with the target task and governs the asymptotic and finite sample regime. In fact, assume for any finite m that there is at least one source hypothesis with small empirical risk, in particular, in $\tilde{\mathcal{O}}(\sqrt{k/m})$, and set $\lambda = \tilde{\mathcal{O}}(\sqrt{k/m})$. Then we have that $R(h_{\widehat{\mathbf{w}}, \widehat{\boldsymbol{\beta}}}^{\text{trg}}) - \widehat{R}(h_{\mathbf{w}^*, \boldsymbol{\beta}^*}^{\text{trg}}) = \tilde{\mathcal{O}}(\sqrt{k/m})$, that is we get the generalization bound as if we are able to solve the original NP-hard problem in (6.2). In other words, if there are useful source hypotheses, we expect our algorithm to perform similarly to the one that identifies the optimal subset. This might seem surprising, but it is important to note that we do not actually care about identifying the correct subset of source hypotheses. We only care about how well the returned solution is able to generalize. On the other hand, if not even one source hypothesis has low risk, selecting the best subset of k sources becomes meaningless. In this scenario, we expect the selection of any subset to perform in the same way. Thus the approximation guarantee does not matter anymore.

We now state the approximation guarantees of GreedyTL used to prove Theorem 18. In the following Corollary we show how far the optimal solution to the regularized subset selection is from the approximate one found by GreedyTL.

Corollary 4. *Let $\lambda \in \mathbb{R}^+$ and $k \leq n$. Denote $\text{OPT} := \min_{\|\mathbf{w}\|_0=k} \{\widehat{R}(\mathbf{w}) + \lambda \|\mathbf{w}\|_2^2\}$. Assume that $\widehat{\mathbf{C}}$ and $\widehat{\mathbf{b}}$ are normalized, and $\widehat{C}_{i,j \neq i} \leq \gamma < \frac{1+\lambda}{6k}$. Then, the FR algorithm generates an approximate solution $\widehat{\mathbf{w}}$ to the regularized subset selection problem that satisfies $\widehat{R}(\widehat{\mathbf{w}}) + \lambda \|\widehat{\mathbf{w}}\|_2^2 \leq \left(1 + \frac{16(k+1)^2\gamma}{1+\lambda}\right) \text{OPT} - \frac{16(k+1)^2\gamma\lambda}{(1+\lambda)^2}$.*

Apart from being instrumental in the proof of Theorem 18, this statement also points to the secondary role of the regularization parameter λ : unlike in FR, we can control the quality of the approximate solution even if the features are correlated.

6.5 Experiments

In this section we present experiments comparing GreedyTL to several transfer learning and feature selection algorithms. As done previously, we considered the object detection task and, for all datasets, we left out one class considering it as the target class, while the remaining classes were treated as sources [137]. We repeated this procedure for every class and for every dataset at hand, and averaged the performance scores. In the following, we refer to this procedure as *leave-one-class-out*. We performed the evaluation for every class, reporting averaged class-balanced recognition scores.

We used subsets of Caltech-256 [56], Imagenet [35], SUN09 [24], SUN-397 [144]. The largest setting considered involves 1000 classes, totaling in 1.2M examples, where the number of training examples of the target domain varies from 11 to 20. Our experiments aimed at verifying three claims:

1. L_2 -regularization is important when using greedy feature selection as a transfer learning scheme.
2. In a small-sample regime GreedyTL is more robust than alternative feature selection approaches, such as L_1 -regularization.
3. The approximated randomized greedy algorithm improves the computational complexity of GreedyTL with no significant loss in performance.

6.5.1 Datasets and Features

We used the whole Caltech-256, a public subset of Imagenet containing 10^3 classes, all the classes of SUN09 that have more than 1 example, which amounts to 819 classes, and the whole SUN-397 dataset containing 397 place categories. For Caltech-256 and Imagenet, we used as features the publicly-available 1000-dimensional SIFT-BOW descriptors, while for SUN09 we extracted 3400-dimensional PHOG descriptors. In addition, for Imagenet and SUN-397, we also ran experiments using convolutional features extracted from DeCAF neural network [38].

We composed a negative class by merging 100 held-out classes (*surrogate* negative class). We did so for each dataset, and we further split it into the *source* negative and the *target* negative class as 90% + 10% respectively, for training sources and the target. The source classifiers were trained for each class in the dataset, combining all the positive examples of that class and the source negatives. On average,

each source classifier was trained using 10^4 examples for Caltech-256, 10^5 for Imagenet and 10^3 for the SUN09 dataset. The training sets for the target task were composed by $\{2, 5, 10\}$ positive examples, and 10 negative ones. Following [137], the testing set contained 50 positive and 50 negative examples for Caltech-256, Imagenet, and SUN-397. For the skewed SUN09 dataset we took one positive and 10 negative training examples, with the rest left for testing. We drew each target training and testing set randomly 10 times, averaging the results over them.

6.5.2 Baselines

We chose a linear SVM to train the source classifiers [44]. This allows us to compare fairly with relevant baselines (like Lasso) and is in line with recent trends in large scale visual recognition and transfer learning [38]. The models were selected by 5-fold cross-validation having regularization parameter $C \in \{10^{-4}, 10^{-3}, \dots, 10^4\}$. In addition to trained source classifiers, for Caltech-256, we also evaluated transfer from Clasemes [12] and Object Bank [85], which are very similar in spirit to source classifiers. At the same time, for Imagenet, we evaluated transfer from the outputs of the final layers of the DeCAF convolutional neural network [38].

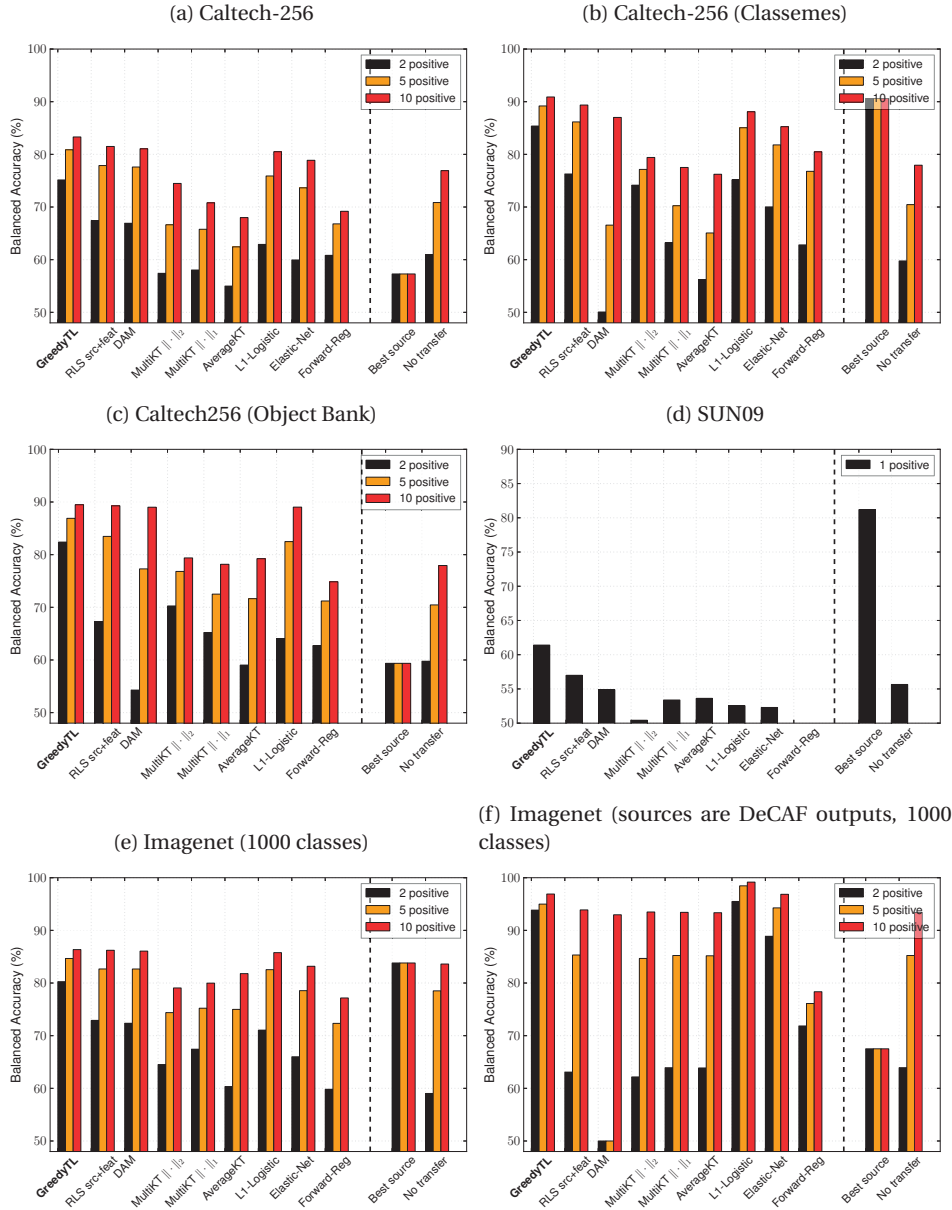
We divided the baselines into two groups - the linear transfer learning baselines that do not require access to the source data, and the feature selection baselines. We included the second group of baselines due to GreedyTL's resemblance to a feature selection algorithm. We focus on the linear baselines, since we are essentially interested in the feature selection in high-dimensional spaces from few examples. In that scope, most feature selection algorithms, such as Lasso, are linear. In particular, amongst TL baselines we chose: *No transfer*: RLS algorithm trained solely on the target data; *Best source*: indicates the performance of the best source classifier selected by its score on the testing set. This is a pseudo-indicator of what an HTL can achieve; *AverageKT*: obtained by averaging the predictions of all the source classifiers; *RLS src+feat*: RLS trained on the concatenation of feature descriptors and source classifier predictions; *MultiKT* $\|\cdot\|_2$: HTL algorithm by [137] selecting β in (6.1) by minimizing the leave-one-out error subject to $\|\beta\|_2 \leq \tau$; *MultiKT* $\|\cdot\|_1$: similar to previous, but applying the constraint $\|\beta\|_1 \leq \tau$; *DAM*: An HTL algorithm by [39], that can handle selection from multiple source hypotheses. It was shown to perform better than the well known and similar ASVM [145] algorithm. For the feature selection baselines we selected well-established algorithms involving sparsity assumption: *L1-Logistic*: Logistic regression with L1 penalty [58]; *Elastic-Net*: Logistic regression with mixture of L1 and L2 penalties [58]; *Forward-Reg*: Forward regression – a classical greedy feature selection algorithm. When comparing our algorithms to the baselines on large datasets, we also consider a Domain Adaptive Dictionary Learning baseline [113]. This baseline represents the family of dictionary learning methods for domain adaptation and transfer learning. In particular, it learns a dictionary on the source domain and adapts it to the target one. However, in our setup the only access to the source data is through the source hypotheses. Therefore, the only way to construct source features is by using the source hypotheses on the target data points.

6.5.3 Results

Figure 6.1 shows the leave-one-class-out performance. In addition, Figures 6.1b, 6.1c, 6.1f show the performance when transferring from off-the-shelf clasemes, object-bank feature descriptors,

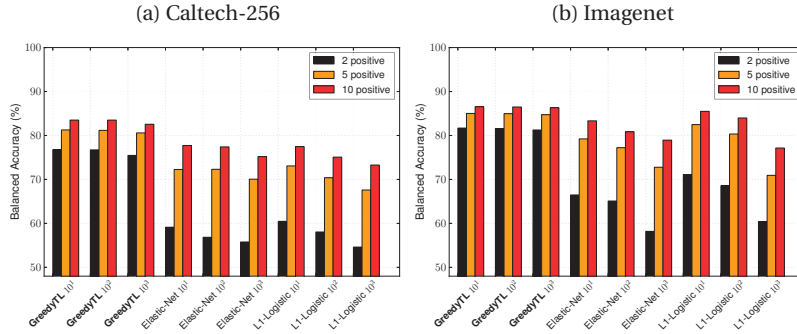
6.5. Experiments

Figure 6.1 – Performance on Caltech-256, subsets of Imagenet (1000 classes) and SUN09 (819 classes). Averaged class-balanced accuracies in the leave-one-class-out setting.



and DeCAF neural network activations. Whenever any baseline algorithm has hyperparameters to tune, we chose the ones that minimize the leave-one-out error on the training set. In particular, we selected the regularization parameter $\lambda \in \{10^{-4}, 10^{-3}, \dots, 10^4\}$. MultiKT and DAM have an additional hyperparameter that we call τ with $\tau \in \{10^{-3}, \dots, 10^3\}$. Kernelized algorithms were supplied with a linear kernel. Model selection for GreedyTL involves two hyperparameters, that is k and λ . Instead of fixing k , we let GreedyTL select features as long as the regularized error between two consecutive steps is larger than δ . In particular, we set $\delta = 10^{-4}$, as in preliminary experiments we have not observed any gain in performance past that point. The λ is fixed to 1. Even better performance could be obtained

Figure 6.2 – Baselines and number of additional noise dimensions sampled from a standard distribution. Averaged class-balanced recognition accuracies in the leave-one-class-out setting.



tuning it.

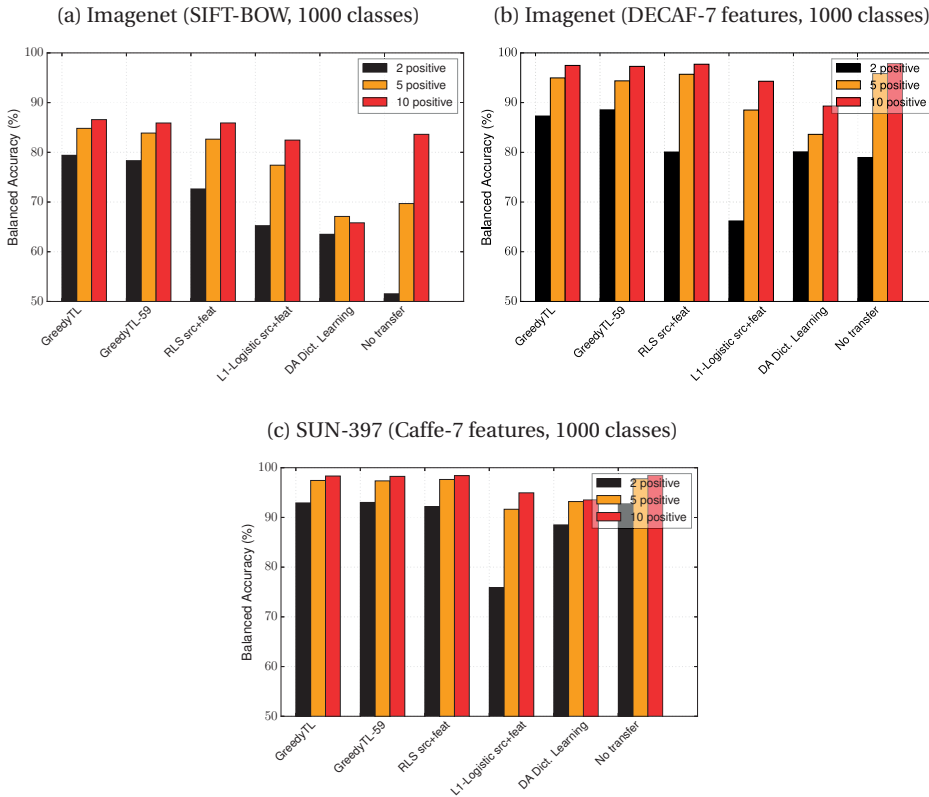
We see that GreedyTL dominates TL and feature selection baselines throughout the benchmark, rarely appearing on-par, especially in the small-sample regime. In addition, on two datasets out of three, it manages to identify the source classifier subset that performs comparably or better than the Best source, that is the single best classifier selected by its performance on the testing set. The significantly stronger performance achieved by GreedyTL w.r.t. FR, on all databases and in all settings, confirms the importance of the regularization in our formulation.

Notably, GreedyTL outperforms RLS src+feat, which is equivalent to GreedyTL selecting all the sources and features. This observation points to the fact that GreedyTL successfully manages to discard irrelevant feature dimensions and sources. To investigate this important point further, we artificially add 10, 100 and 1000 dimensions of pure noise sampled from a standard distribution. Figure 6.2 compares feature selection methods to GreedyTL in robustness to noise. Clearly, in the small-sample setting, GreedyTL is tolerant to large amount of noise, while $L1$ and $L1/L2$ regularization suffer a considerable loss in performance. We also draw attention to the failure of $L1$ -based feature selection methods and MultiKT with $L1$ regularization to match the performance of GreedyTL.

6.5.4 Approximated GreedyTL

As was discussed in Section 6.3, the computational complexity of GreedyTL is linear in the number of source hypotheses and feature dimensions. In this section we assess empirical performance of the approximated GreedyTL, which is *independent* from the number of source hypotheses, implemented through the approximated greedy algorithm described at the end of Section 6.3. In the following we refer to this version of an algorithm as GreedyTL-59. Instead of considering all the transfer learning and feature selection baselines, we restrict the performance comparison to the strongest competitors. To show the power of highly scalable approximated GreedyTL, we focus on the largest datasets in the number of source hypotheses and feature dimensions: Imagenet and SUN-397. In case of Imagenet, we consider standard SIFT-BOW features as in previous section and also DeCAF-7 convolutional features extracted from the seventh layer of the DeCAF neural network [38]. For SUN-397, we use convolutional features of the Caffe network trained on the Places-205 dataset [152], which was shown to perform particularly well in the scene recognition tasks. Figure 6.3 summarizes the new results. Surprisingly, approximated GreedyTL performs on par with the version with exhaustive search over the candidate,

Figure 6.3 – Comparison of the approximated GreedyTL: GreedyTL-59 to GreedyTL with exhaustive search and most competitive baselines on three largest datasets considered in our experiments.



maintaining dominant performance in the small-sample regime on the Imagenet dataset. Yet, training timings are dramatically improved as can be seen from Table 6.1. In the case of SUN-397 dataset, however, GreedyTL performs on par with the top competitors.

6.5.5 Selected Source Analysis

In this section we take a look at the source hypotheses selected by GreedyTL. In particular, we make a qualitative assessment with the goal to see if semantically related sources and targets are correlated, visualizing selected sources and the magnitude of their weights. We do so by grouping sources and targets semantically according to the WordNet [97] distance, and plotting them as matrices with columns corresponding to targets, rows to sources, and entries to the weights of the sources. Figure 6.4 shows such matrices for GreedyTL when evaluated on Imagenet with DECAF7 features and averaged over all splits, for 2 positive and 10 positive examples accordingly. First we note that for certain supercategories there are clearly distinctive patterns, indicating cross-transfer within the same supercategory. We compare those matrices to the ones originating from the strongest RLS (src+feat) baseline, Figure 6.5. We notice a clear difference, as semantic patterns of GreedyTL are more distinctive in a small-sample setting (2+10), while the ones of RLS (src+feat) appear hazier. We argue that this is a consequence of greedy selection procedure implemented by GreedyTL, where sources are selected incrementally, thus many coefficients correspond to zeros. Due to the formulation of RLS (src+feat),

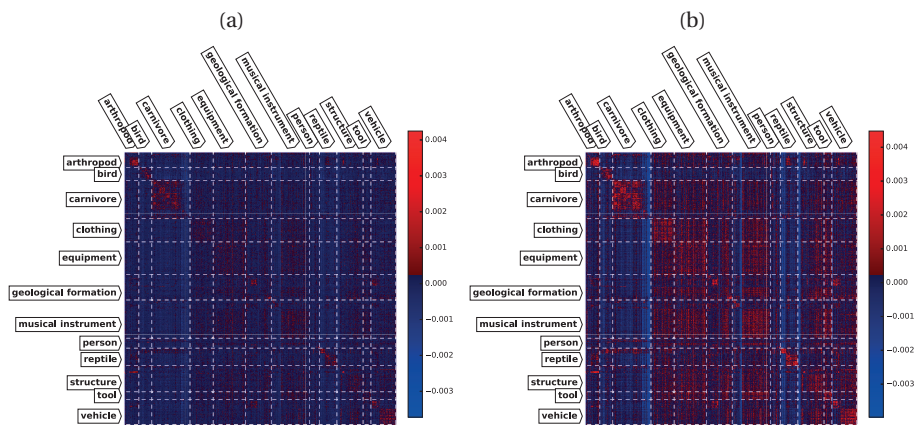
Chapter 6. Greedy Algorithms for Hypothesis Transfer Learning

Table 6.1 – Training time in seconds for transferring to a single target class. Results are averaged over 10 splits.

		GreedyTL		
Training examples pos.+neg.		2 + 10	5 + 10	10 + 10
Imagenet (SIFT-BOW)	1899 source+dim	1.541 ± 0.242	3.083 ± 0.486	5.291 ± 0.870
Imagenet (DECAF7)	4995 source+dim	3.481 ± 0.356	7.492 ± 0.655	13.408 ± 1.165
SUN-397 (Caffe-7)	4492 source+dim	3.245 ± 0.495	6.764 ± 1.051	11.282 ± 1.630

		GreedyTL-59		
Training examples pos.+neg.		2 + 10	5 + 10	10 + 10
Imagenet (SIFT-BOW)	1899 source+dim	0.043 ± 0.005	0.088 ± 0.011	0.149 ± 0.021
Imagenet (DECAF7)	4995 source+dim	0.055 ± 0.006	0.114 ± 0.013	0.198 ± 0.020
SUN-397 (Caffe-7)	4492 source+dim	0.060 ± 0.021	0.120 ± 0.038	0.198 ± 0.055

Figure 6.4 – Semantic transferrability matrix for GreedyTL evaluated on Imagenet (DECAF7 features). Columns correspond to targets and rows to sources. Stronger color intensity means larger source weight. 6.4a corresponds to learning from 2 positive and 10 negative examples, while 6.4b, with 10 positive.



however, even if a source is less relevant, its coefficient most likely will not be exactly equal to zero.

It is also instructive to compare exact GreedyTL to the approximated one. Figure 6.7 pictures semantic matrices for the approximated version. We note that approximated version appears to be slightly more conservative in a small-sample case (2+10), but overall, semantic patterns seem to match, thus emphasizing the quality of the solution provided by the approximated version and empirically corroborating the theoretical motivation behind the randomized selection.

Finally, we take a closer look at some patterns of Figure 6.4a, that is in the case of learning from only 2 positive examples. This new analysis is shown in Figure 6.6. We notice that even at the smaller scale, there are emergent semantic patterns.

6.6 Conclusion

In this chapter we studied the transfer learning problem involving hundreds of sources. The kind of transfer learning scenario we consider assumes no access to the source data directly, but through the use of the source hypotheses induced by them. In particular, we focused on the efficient source

Figure 6.5 – Semantic transferrability matrix for RLS (src+feat) evaluated on Imagenet (DECAF7 features).

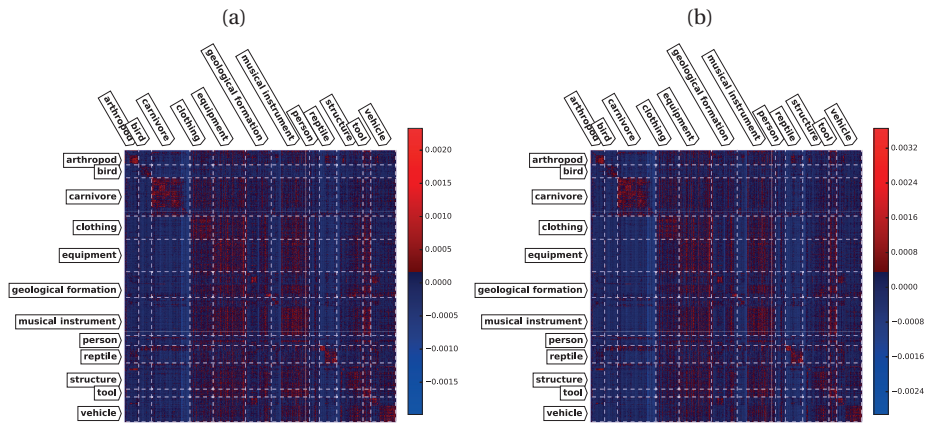
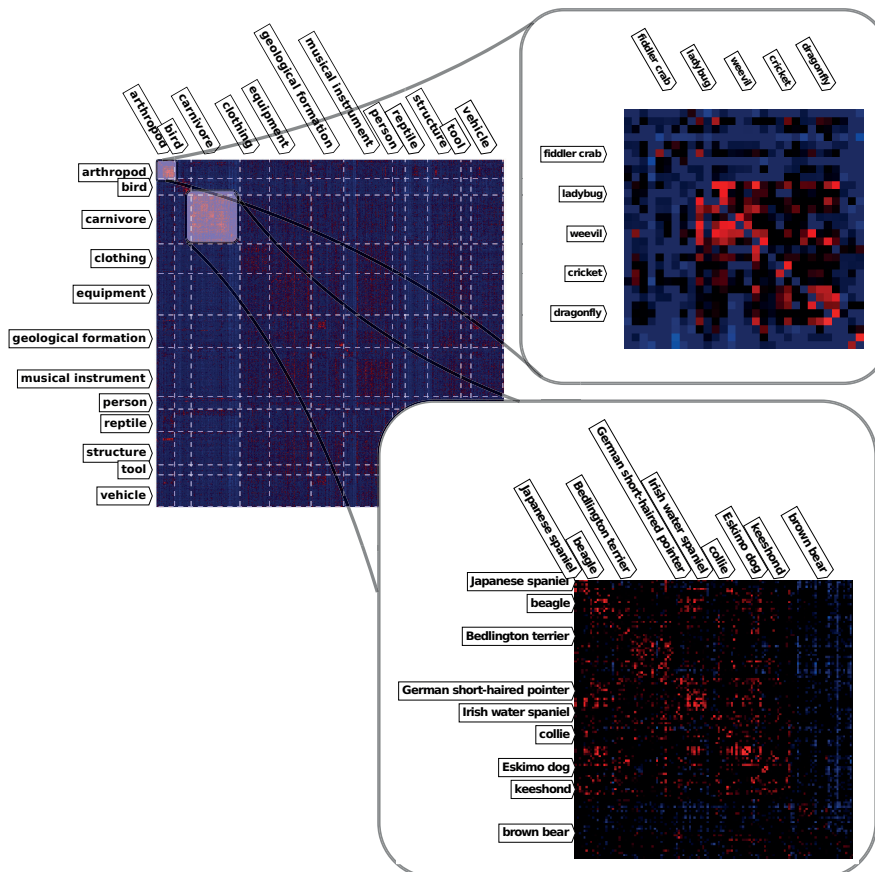


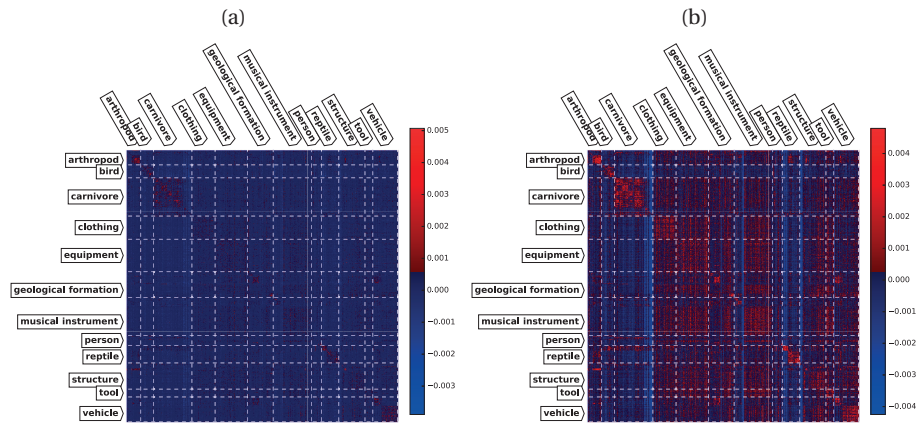
Figure 6.6 – GreedyTL evaluated on Imagenet (DECAF7 features): a closer look at some strongly related sources and targets.



hypothesis selection and combination, improving the performance on the target task. We proposed a greedy algorithm, GreedyTL, capable of selecting relevant sources and feature dimensions at the same

Chapter 6. Greedy Algorithms for Hypothesis Transfer Learning

Figure 6.7 – Semantic transferrability matrix for the approximated GreedyTL evaluated on Imagenet (DECAF7 features).



time. We verified these claims by obtaining the best results among the competing feature selection and TL algorithms, on the Imagenet, SUN09 and Caltech-256 datasets. At the same time, comparison against the non-regularized version of the algorithm clearly show the power of our intuition. We support our empirical findings by showing theoretically that under reasonable assumptions on the sources, the algorithm can learn effectively from few target examples.

In the next chapter we go beyond the binary classification setting and consider an HTL multiclass learning scenario, where classes are introduced to the learner incrementally.

7 Class-incremental Hypothesis Transfer Learning

The material of this chapter is partially based on the publication:

I. Kuzborskij, F. Orabona, and B. Caputo. From N to $N + 1$: Multiclass Transfer Incremental Learning.

In *Computer Vision and Pattern Recognition (CVPR)*, 2013.

The doctoral candidate formalized the problem, designed the algorithms, evaluated the algorithms, and partially wrote the publication.

7.1 Introduction

Vision-based applications that appear in assisted ambient living, home robotics, and intelligent car driver assistants all share the need to distinguish between several object categories. They also share the need to update their knowledge over time, by learning new category models whenever faced with unknown objects. Consider for instance the case of a service robot, designed for cleaning up kitchens in public hospitals. Its manufacturers will have equipped it with visual models of objects expected to be found in a kitchen, but inevitably the robot will encounter something not anticipated at design time – perhaps an object out of context, such as a personal item forgotten by a patient on her food tray, or a new type of food processor that entered the market after the robot. To learn such new object, the robot will generally have to rely on little data and explanation from its human supervisor. Also, it will have to preserve its current range of competences while adding the new object to its set of known visual models. This challenge, which holds for any intelligent system equipped with a camera, can be summarized as follows: suppose you have a system that knows K objects (source). Now you need to extend its object knowledge to the $K + 1$ -th (target), using only few new annotated samples, without having the possibility to re-train everything from scratch. Can you add effectively the new target $K + 1$ -th class model to the known K source models by leveraging over them, while at the same time preserving their classification abilities?

The problem of how to learn a new object category from few annotated samples by exploiting prior

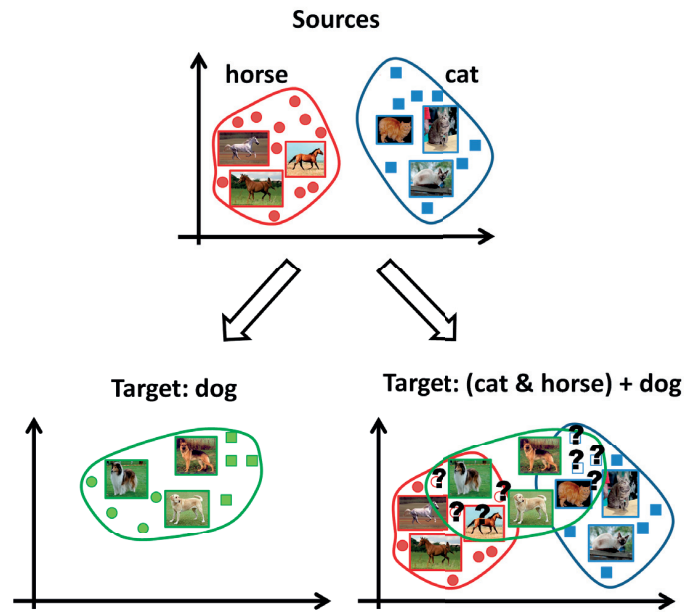


Figure 7.1 – Binary (left) versus $K \rightarrow K + 1$ transfer learning (right). In both cases, transfer learning implies that the target class is learned close to where informative sources models are. This is likely to affect negatively performance in the $K \rightarrow K + 1$ case, where one aims for optimal accuracy on the sources and target classes simultaneously.

knowledge has been extensively studied [146, 65, 41]. The majority of previous work focused on object category detection (i.e. binary classification) rather than the multiclass case [2, 145, 136]. It is natural to ask if such previous methods would work well in the scenario depicted, by just extending them to the multiclass. We argue that to solve the $K \rightarrow K + 1$ transfer learning problem one needs to address a deeper algorithmic challenge.

In addition, learning from scratch and preserving training sets from all the source tasks might be infeasible due to the large number of tasks or when acquiring tasks incrementally, especially for large datasets [94]. In the object categorization case this might come as training source classifiers from large scale visual datasets, in abundance of data.

Consider the following example: a transfer learning task of learning a dog detector, given that the system has already learned other kind of animal detectors. This is achieved, in one form or another, by constraining the dog model to be somehow “similar” to the horse and cat detectors learned before [65, 136]. Success in this setting is defined as optimizing the accuracy of the dog detector, with a minimal number of annotated training samples (Figure 7.1, left).

But if we consider the multiclass case, the different tasks now “overlap”. Hence we are faced with two opposite needs: on one side, we want to learn to recognize dogs from few samples, and for that we need to impose that the dog model is close to the horse and cat models learned before. On the other side, we want to optimize the overall system performance, which means that we need to avoid mispredictions between classes at hand (Figure 7.1, right). These two seemingly contradictory requirements are true for many $K \rightarrow K + 1$ transfer learning scenarios: how to reconcile them in a principled manner is the contribution of this paper.

We build on the algorithm of Tommasi *et al.* [136], a transfer learning method based on the multiclass extension of RLS [132]. Thanks to the linear nature of RLS, we cast transfer learning as a constraint for the classifier of the $K + 1$ target class to be close to a subset of the K source classifiers. At the same time, we impose a stability to the system, biasing the formulation towards solutions close to the hyperplanes of the K source classes. In practice, given K source models, we require that these models would not change much when going from K to $K + 1$.

As in [136], we learn how much to transfer from each of the source classifiers, by minimizing the LOO error, which is an unbiased estimator of the generalization error for a classifier [20]. We call our algorithm MULTIPLE.

Experiments on various subsets of the Caltech-256 [56] and AwA datasets [81] show that our algorithm outperforms the One-Versus-All (OVA) extension of [136], as well as other baselines [65, 146, 2]. Moreover, its performance often is comparable to what would be obtained by re-training the whole $K + 1$ classifier from all data, without the need to store the source training data.

The paper is organized as follows: after a review of previous work (Section 7.2), we describe our algorithm in Section 7.3. Experiments are reported in Section 7.4, followed by conclusions in Section 7.5.

7.2 Related Work

Prior work in transfer learning addresses mostly the binary classification problem (object detection). Some approaches transfer information through samples belonging to both source and target domains during the training process, as in [83] for reinforcement learning. Feature space approaches consider transferring or sharing feature space representations between source and target domains. Typically, in this setting source and target domain samples are available to the learner. In that context, Blitzer *et al.* [14] proposed a heuristic for finding corresponding features, that appear frequently in both domains. Daumé [32] showed a simple and effective way to replicate feature spaces for performing adaptation for the case of natural language processing. Yao and Doretto [146] proposed an AdaBoost-based method using multiple source domains for the object detection task.

Another research line favors model-transfer (or parameter-transfer) methods, where the only knowledge available to the learner is “condensed” within a model trained on the source domain. Thus, samples from the source domain are not preserved. Model-transfer is theoretically sound as was shown by Kuzborskij and Orabona [76], since relatedness of the source and target tasks enables quick convergence of the empirical error estimate to the true error. Within this context, Yang *et al.* [145] proposed a kernelizable SVM-like classifier with a biased regularization term. There, instead of the standard ℓ_2 regularization, the goal of the algorithm is to keep the target domain classifier “close” to the one trained on the source domain. Tommasi *et al.* [136] proposed a multi-source transfer model with a similar regularizer, where each source classifier was weighed by learned coefficients. The method obtained strong results on the visual object detection task, using only a small amount of samples from the target domain. Aytar and Zisserman [2] proposed a similar model, with a linear formulation for the problem of object localization. Both methods rely on weighed source classifiers, which is crucial when attempting to avoid negative transfer. Several Multiple Kernel Learning (MKL) methods were proposed for solving transfer learning problems. Jie *et al.* [65] suggested to use MKL kernel weights as

source classifier weights, proposing one of the few truly multiclass transfer learning models. An MKL approach was also proposed by Duan *et al.* [41]. There, kernel weights affect both the source classifiers and the representation of the target domain.

7.3 Multiclass Incremental Transfer Learning

In the following we propose a multiclass classification algorithm able to quickly learn and incorporate new classes in an incremental fashion. More specifically, suppose that we are given a predictor trained to distinguish between the K categories, and we receive a small amount of examples belonging to a new $K + 1$ -th category. To this end, our task is to generate a new $K + 1$ class predictor which maintains performance on the K source classes and yet is able to generalize quickly on a new target $K + 1$ -th class. We address the issue of quick generalization by *transferring* from the K source classes and at the same time, we constrain excessive updates of the source models to maintain their performance.

We stress that we assume no access to the examples used to train the source classifiers. Such access would be a limiting factor, because we would be impeded by the need to re-train a classifier from scratch every time we learn a new class. This would pose a challenge to the application of class-incremental scheme in life-long learning scenarios.

Now we proceed with the description of the algorithm.

7.3.1 Multiclass Regularized Least Squares

In the following we will address our problem as a multi-class problem through reduction to a number of binary ones [115]. This is well-known as a One-Versus-All (OVA) approach to a multiclass classification. Suppose we are given a training set $S = \{(\mathbf{x}_i, \text{label}_i)\}_{i=1}^m$, such that $\mathbf{x}_i \in \mathcal{X}$ and $\text{label}_i \in [K]$. Then, to make predictions we will use a combined multiclass predictor

$$f(\mathbf{x}) = \arg \max_{k \in [K]} \{ \hat{\mathbf{w}}_k^\top \mathbf{x} \},$$

supplied with a set of binary ones, $\{\hat{\mathbf{w}}_k\}_{k=1}^K$, where each k -th classifier is generated from a training set with binarized labels, $S_k = \{(\mathbf{x}_i, y_{k,i} = 2\mathbb{1}\{\text{label}_i = k\} - 1)\}_{i=1}^m$.

We assume that we have access to a small amount of labeled examples for all classes, including the source ones. In this setting we still build upon the OVA formulation, and to train binary classifiers we will use a well-known RLS algorithm, very closely related to the LSSVM, which can be used both for regression and classification [116, 132]. More formally, given a training set S_k , an RLS generates a linear model $[\hat{\mathbf{w}}_k, \hat{b}_k]$ by solving

$$\min_{\mathbf{w} \in \mathbb{R}^d, b \in \mathbb{R}} \left\{ \frac{1}{m} \sum_{i=1}^m (\mathbf{w}_k^\top \mathbf{x}_i + b_k - y_{k,i})^2 + \lambda \|\mathbf{w}_k\|^2 \right\}. \quad (7.1)$$

Thus, every resulting binary classifier is characterized by a hyperplane $\hat{\mathbf{w}}_k$ and therefore is a linear classifier. We note that it is straightforward to convert this approach into a non-linear one through the use of kernels, and for clarity we describe the algorithm in a linear notation. In the next section we

7.3. Multiclass Incremental Transfer Learning

build our transfer learning algorithm upon multiclass OVA RLS.

7.3.2 MULTIPLE Algorithm

In the rest of this paper, we will assume that the source classifier is an OVA linear classifier, and we will use matrix notation to denote its binary components, that is

$$\mathbf{W}^{\text{src}} = [\mathbf{w}_1^{\text{src}}, \dots, \mathbf{w}_K^{\text{src}}].$$

Note that we assume no access to the examples used to train the source classifiers.

The aim of the approach presented here is to find a new set of hyperplanes $\mathbf{W} = [\mathbf{w}_1, \dots, \mathbf{w}_K, \mathbf{w}_{K+1}]$, such that the generalization ability of an OVA multiclass predictor

1. improves on the *target* $K + 1$ -th category by transferring from the source models \mathbf{W}^{src} , and,
2. does not deteriorate, or even improves on the K *source* categories.

We cast both objectives above into the regularized ERM framework by extending a multiclass OVA formulation. Our extension builds upon the HTL literature [77, 76], Chapters 3 and 4. In particular we exploit a variant of HTL known as *biased regularization*. Unlike usual L_2 -regularized ERM formulation, such as SVM, when using biased regularization, we state an objective function as a sum of empirical risk and regularization term $\lambda \|\mathbf{w} - \mathbf{w}_0\|^2$ instead of $\lambda \|\mathbf{w}\|^2$. Thus, thanks to the model linearity, we incorporate a metric between classifiers into the objective, which is used to find classifiers with similar performance by enforcing the distance between them to be small. We propose to achieve both aims above through the use of biased regularization.

We cover the first objective by introducing the regularizer $\lambda \|\mathbf{w}_{K+1} - \mathbf{W}^{\text{src}} \boldsymbol{\beta}\|^2$ into (7.1). This term enforces the target model \mathbf{w}_{K+1} to be close to a linear combination of the source models. At the same time the use of the misspecified source model, that is the negative transfer, is prevented by weighing each source model using the vector $\boldsymbol{\beta} \in \mathbb{R}^K$. This type of regularization has been used in HTL for visual learning [78, 137, 138], and its theoretical merits are understood [77].

The second objective is again tackled by biased regularization. Simply introducing a target category into a classifier may affect the performance of the source models and it is therefore necessary to update the source models. However, to prevent excessive updates, which may drive updated classifiers too far from the source ones, we enforce the new hyperplanes \mathbf{W} to remain close to the source hyperplanes \mathbf{W}^{src} using the term $\lambda \|\mathbf{w}_k - \mathbf{w}_k^{\text{src}}\|_2^2$ in (7.1). We consider a biased regularization approach to transfer learning. Specifically, we combine RLS formulation (7.1) with biased regularization. To this end we formulate the objective for the target $K + 1$ -th class model $\hat{\mathbf{w}}_{K+1}$ as

$$\min_{\substack{\mathbf{w}_{K+1} \in \mathbb{R}^d, \\ b \in \mathbb{R}}} \left\{ \frac{1}{m} \sum_{i=1}^m (\mathbf{w}_{K+1}^\top \mathbf{x}_i + b_{K+1} - y_{k,i})^2 + \lambda \|\mathbf{w}_{K+1} - \mathbf{W}^{\text{src}} \boldsymbol{\beta}\|^2 \right\}.$$

and for every k -th updated source model $\hat{\mathbf{w}}_k$ as

$$\min_{\mathbf{w}_k \in \mathbb{R}^d, b_k \in \mathbb{R}} \left\{ \frac{1}{m} \sum_{i=1}^m (\mathbf{w}_k^\top \mathbf{x}_i + b_k - y_{k,i})^2 + \lambda \|\mathbf{w}_k - \mathbf{w}_k^{\text{src}}\|^2 \right\}$$

Thus, as in the OVA formulation we predict with

$$f^{\text{trg}}(\mathbf{x}) = \operatorname{argmax}_{k \in [K+1]} \{ \hat{\mathbf{w}}_k^\top \mathbf{x} \}.$$

The solutions to the minimization problems above are given by

$$\begin{aligned} \hat{\mathbf{w}}_k &= \mathbf{w}_k^{\text{src}} + \mathbf{X}(\mathbf{a}_k - \mathbf{a}_k^{\text{src}}), \quad \forall k \in [K] \\ \hat{b}_k &= b_k - b_k^{\text{src}}, \\ \hat{\mathbf{w}}_{K+1} &= \mathbf{W}^{\text{src}} \boldsymbol{\beta} + \mathbf{X}(\mathbf{a}_{K+1} - \mathbf{A}^{\text{src}} \boldsymbol{\beta}), \\ \hat{b}_{K+1} &= b_{K+1} - \mathbf{b}_k^{\text{src}} \boldsymbol{\beta}. \end{aligned}$$

where

$$\begin{bmatrix} \mathbf{A} \\ \mathbf{b}^\top \end{bmatrix} = \mathbf{M} \begin{bmatrix} \mathbf{Y} \\ \mathbf{0} \end{bmatrix}, \quad (7.2)$$

$$\begin{bmatrix} \mathbf{A}^{\text{src}} \\ \mathbf{b}^{\text{src}\top} \end{bmatrix} = \mathbf{M} \begin{bmatrix} \mathbf{X}^\top \mathbf{W}^{\text{src}} \\ \mathbf{0} \end{bmatrix}, \quad (7.3)$$

$$\mathbf{M} = \begin{bmatrix} \mathbf{X}^\top \mathbf{X} + \lambda \mathbf{I} & \mathbf{1} \\ \mathbf{1}^\top & 0 \end{bmatrix}^{-1}. \quad (7.4)$$

The solution of the transfer learning problem is completely defined once we set the parameters $\boldsymbol{\beta}$. In the next section we describe how to automatically tune these parameters.

7.3.3 Self-tuning of Transfer Parameters

Our goal is to tune the transfer coefficients $\boldsymbol{\beta}$ to improve the performance of the linear model for the new $K + 1$ -class by exploiting only relevant source models while preventing negative transfer. We optimize the coefficients $\boldsymbol{\beta}$ automatically using an objective based on the Leave-One-Out (LOO) error, which is an almost unbiased estimator of the generalization error of a classifier [20]. An advantage of RLS, used as a basis to our approach, over other methods is that it allows the LOO error to be computed efficiently in analytic form. Specifically, we cast the optimization of $\boldsymbol{\beta}$ as the minimization of a convex upper bound of the LOO error. The LOO predictions for the entire training set with respect to hyperplane $\hat{\mathbf{w}}_k$ is given by (derivation is available in the appendix).

$$\begin{aligned} \mathbf{y}_k^{\text{loo}} &= \mathbf{y}_k - (\mathbf{M} \circ \mathbf{I})^{-1}(\mathbf{a}_k - \mathbf{a}_k^{\text{src}}) \quad \forall k \in [K], \\ \mathbf{y}_{K+1}^{\text{loo}}(\boldsymbol{\beta}) &= \mathbf{y}_{K+1} - (\mathbf{M} \circ \mathbf{I})^{-1}(\mathbf{a}_{K+1} - \mathbf{A}^{\text{src}} \boldsymbol{\beta}). \end{aligned} \quad (7.5)$$

We stress that (7.5) is a linear function of $\boldsymbol{\beta}$. We now need a convex multiclass loss to measure the LOO errors. A fairly standard choice would be a convex multiclass loss as in [28], which keeps samples of different classes at the unit marginal distance. Slightly abusing notation in (7.5), such multiclass loss

function would look like

$$\ell_i^{\text{mc}}(\boldsymbol{\beta}) = \max_{r \neq y_i} \left[1 + y_{r,i}^{\text{loo}}(\boldsymbol{\beta}) - y_{y_i,i}^{\text{loo}}(\boldsymbol{\beta}) \right]_+ . \quad (7.6)$$

However, from (7.5) observe that changing $\boldsymbol{\beta}$ will only change the score of the target $K + 1$ -th class. Thus, when using this loss, almost all examples are neglected during optimization with respect to $\boldsymbol{\beta}$. We address this issue by proposing a modified version of (7.6),

$$\ell_i^{\text{mc-mod}}(\boldsymbol{\beta}) = \begin{cases} \left[1 + y_{K+1,i}^{\text{loo}}(\boldsymbol{\beta}) - y_{y_i,i}^{\text{loo}}(\boldsymbol{\beta}) \right]_+ & : \text{label}_i \neq K + 1 \\ \max_{r \neq y_i} \left[1 + y_{r,i}^{\text{loo}}(\boldsymbol{\beta}) - y_{y_i,i}^{\text{loo}}(\boldsymbol{\beta}) \right]_+ & : \text{label}_i = K + 1 \end{cases}$$

The rationale behind this loss is to enforce a margin of 1 between the target $K + 1$ -th class and the correct one, even when the $K + 1$ -th class does not have the highest score. This has the advantage of forcing the use of all examples during the tuning of $\boldsymbol{\beta}$. Given the analytic form of LOO predictions (7.5) and the multiclass loss function above, we can obtain $\boldsymbol{\beta}$ by solving the convex regularized problem

$$\min_{\boldsymbol{\beta} \in \Omega} \left\{ \frac{1}{m} \sum_{i=1}^m \ell_i^{\text{mc-mod}}(\boldsymbol{\beta}) \right\} , \quad (7.7)$$

$$\text{with } \Omega = \{ \boldsymbol{x} \mid \|\boldsymbol{x}\|_2 \leq 1 \wedge \boldsymbol{x} \geq \mathbf{0} \} .$$

Constraining $\boldsymbol{\beta}$ within a unit L_2 ball is a form of regularization imposed on $\boldsymbol{\beta}$, which prevents overfitting as was shown in theoretical works on HTL [77, 76]. This optimization procedure can be implemented elegantly using projected subgradient descent, which is not affected by the fact that the objective function in (7.7) is not differentiable everywhere. The pseudocode of the optimization algorithm is summarized in Algorithm 2.

Finally we make a few comments on the computational complexity of the entire approach. The computational complexity for obtaining \mathbf{A} , \mathbf{A}^{src} , and \mathbf{M} is in $\mathcal{O}(m^3 + m^2(K + 1))$, which comes from matrix operations (7.3)-(7.4). Algorithm 2 is in $\mathcal{O}(mK(T + 1))$, where we assume that most terms in (7.5) are precomputed. Each iteration of the algorithm is efficient since it depends linearly on both the training set size and number of classes.¹

7.4 Experiments

We present here a series of experiments designed to investigate the behavior of our algorithm when (a) the source classes and the target class are related/unrelated, and when (b) the overall number of classes increases. All experiments were conducted on subsets of two different public datasets, and the results were benchmarked against several baselines. In the rest of the section we first describe our experimental setup (section 7.4.1), then we describe the chosen baselines (section 7.4.2). Section 7.4.3 reports our findings.

¹The source code of MULTIPLE is available online at <https://iljaku.github.io>

Algorithm 2 Projected subgradient descent to find β

Input: $M, Y, A, A^{\text{src}}, T$

Output: β

```

1:  $\mathbf{y}_k^{\text{loo}} \leftarrow \mathbf{y}_k - (\mathbf{M} \circ \mathbf{I})^{-1}(\mathbf{a}_k - \mathbf{a}_k^{\text{src}}) \quad \forall k \in [K]$ 
2:  $\beta_1 \leftarrow \mathbf{0}$ 
3: for  $t \in [T]$  do ▷ Iterations of subgradient descent.
4:    $\mathbf{y}_{K+1}^{\text{loo}} = \mathbf{y}_{K+1} - (\mathbf{M} \circ \mathbf{I})^{-1}(\mathbf{a}_{K+1} - A^{\text{src}} \beta_t)$ 
5:    $\Delta \leftarrow \mathbf{0}$ 
6:   for  $i \in [m]$  do ▷ Passing through the training set.
7:     if  $\text{label}(y_i) \neq K + 1$  then
8:       if  $1 + y_{K+1,i}^{\text{loo}} - y_{y_i,i}^{\text{loo}} > 0$  then
9:          $\Delta \leftarrow \Delta + \text{diag}(\mathbf{M})_i^{-1} \mathbf{a}_i^{\text{src}}$ 
10:      end if
11:     else if  $\max_{r \neq y_i} (1 + y_{r,i}^{\text{loo}} - y_{y_i,i}^{\text{loo}}) > 0$  then
12:        $\Delta \leftarrow \Delta - \text{diag}(\mathbf{M})_i^{-1} \mathbf{a}_i^{\text{src}}$ 
13:     end if
14:   end for
15:    $\beta \leftarrow \beta_t - \frac{\Delta}{M\sqrt{t}}$ 
16:    $\beta \leftarrow [\beta]_+$ 
17:   if  $\|\beta\|_2 > 1$  then
18:      $\beta = \frac{\beta}{\|\beta\|_2}$ 
19:   end if
20:    $\beta_{t+1} \leftarrow \beta$ 
21: end for

```

7.4.1 Data setup

We run all experiments on subsets of the Caltech-256 database [56] and of the Animal with Attributes (AwA) database [81]. From the Caltech-256 database, we selected a total of 14 classes and for the AwA dataset, 42 classes. We did not carry out any image pre-selection or pre-processing. Moreover, for both databases we used pre-computed features available online². Specifically, for the Caltech-256 experiments we used the following features: oriented and unoriented PHOG shape descriptors, SIFT appearance descriptors, region covariance and local binary patterns totalling in 14 descriptor types [49]. For the AwA experiments the chosen features were SIFT, rgSIFT, SURE, PHOG, RGB color histograms and local self-similarity histograms [81].

For each class considered, we randomly selected 80 image samples. These were then split in three disjoint sets: 30 samples for the source classifier, 20 samples for training and 30 samples for test. The samples of the source classifier were used for training the K models \mathbf{W}^{src} .

The performance of each method (see Section 7.4.2) was evaluated using progressively $\{5, 10, 15, 20\}$ training samples for each of the $K + 1$ classes. The experiments were repeated 10 times, using different randomly sampled training and test sets, which we refer to as data splits. Furthermore, to get a reliable estimate of the performance of transfer with respect to different classes, we used a leave-one-class-out approach, considering in turn each class as the $K + 1$ target class, and the other K as source classifiers. We report results averaged over all data splits and leave-one-class-out evaluations.

7.4.2 Algorithmic setup

We compared MULTIPLE against two categories of baselines. The first, that we call no transfer baselines, consists of a group of algorithms addressing the $K \rightarrow K + 1$ problem without leveraging source models; the second, that we call transfer baselines, consists of a group of methods attempting to solve the $K \rightarrow K + 1$ problem by leveraging source models. The no transfer baselines are the following:

No transfer corresponds to RLS trained only on the new training data.

Batch corresponds to a RLS trained using all available samples, i.e. assuming to have access to all the data used to build the source models plus the new training data. The performance of this method might be seen as an indicator of the best performance achievable on the problem, thus as an important reference for assessing the results obtained by transfer learning methods.

Source is the RLS K -class source classifier. In this case, classification on the sample belonging to $K + 1$ -th class is assigned 0 accuracy.

Source+1 corresponds to a binary RLS trained to discriminate between the target class vs the source classes given the training data. It is evaluated on the $K + 1$ problem by combining it with Source in a OVA setting. It is arguably the simplest possible approach to address the $K \rightarrow K + 1$ problem.

Source+1 (hinge) is the scheme analogous to Source+1, but utilizing the hinge loss $\ell(x, z) = |1 - xz|_+$, thus corresponding to a classical SVM formulation.

As transfer baselines, we chose the following methods:

MKTL We compared against MKTL [65], which is one of the few existing discriminative transfer learning algorithm in multiclass formulation.

²Caltech-256: http://www.vision.caltech.edu/Image_Datasets/Caltech256/
AwA: <http://attributes.kyb.tuebingen.mpg.de/>

Chapter 7. Class-incremental Hypothesis Transfer Learning

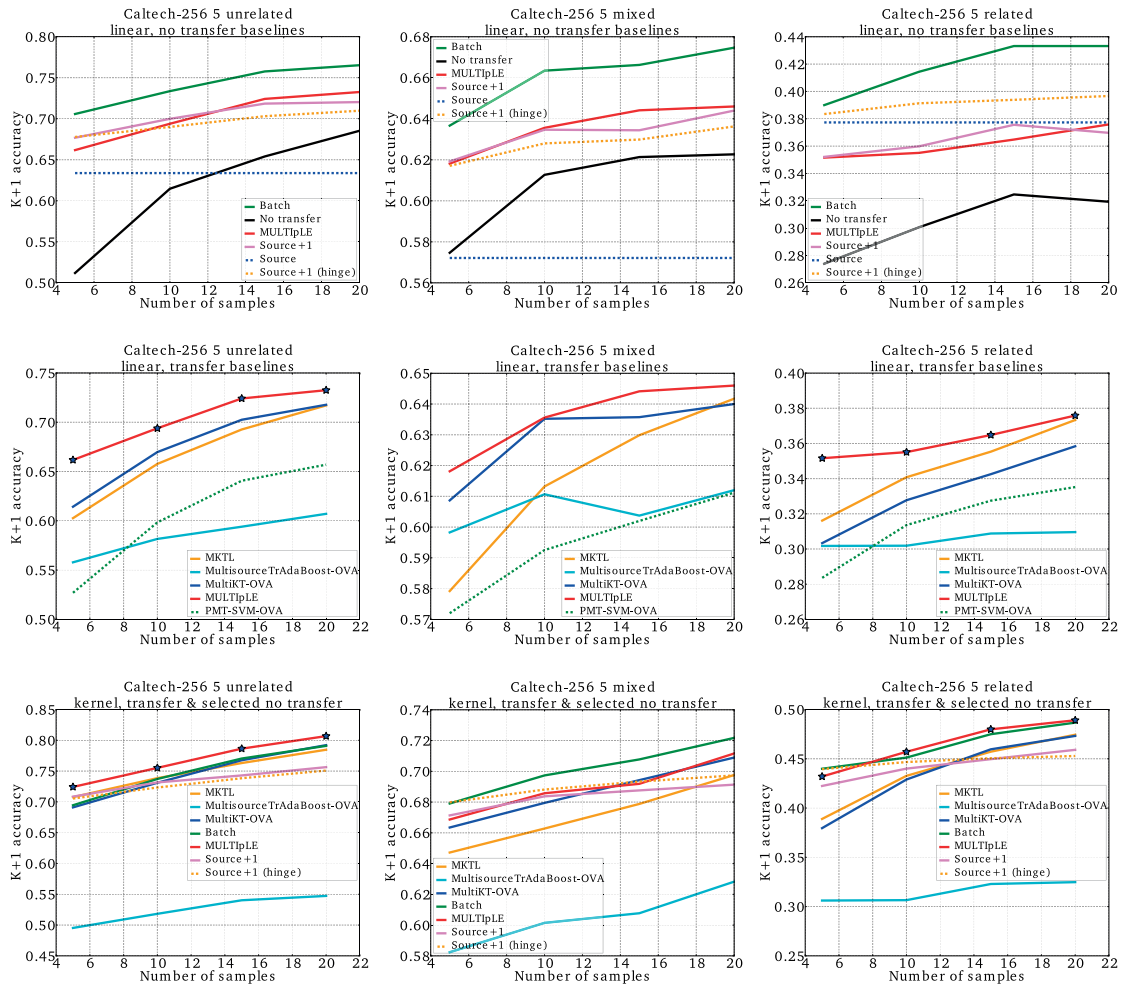


Figure 7.2 – Experimental results for $K + 1 = 5$, Caltech-256. From left to right, columns report results for the unrelated, mixed and related settings. Top row: no transfer baselines, linear case. Middle row: transfer learning baselines, linear case. Bottom row: transfer and competitive no transfer baselines, average of RBF kernels over all features. Stars represent statistical significance of MULTipLE over MultiKT-OVA, $p < 0.05$.

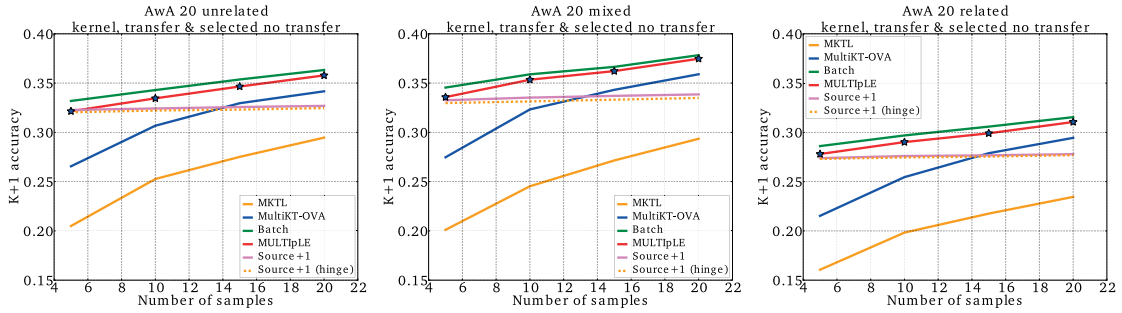


Figure 7.3 – Results for $K + 1 = 20$, AWA, transfer and competitive no transfer baselines, average of RBF kernels, all features. Left to right: unrelated, mixed and related settings. Stars represent statistical significance of MULTIPLE over MultiKT-OVA.

MultiKT-OVA We implemented an OVA multiclass extension of the binary transfer learning method by Tommasi *et al.* [136] as follows: as in the standard OVA formulation, we train MultiKT instance to discriminate between one of $K + 1$ classes and the rest K . At the same time we use Source as the source classifier. Thus, eventually we obtain $K + 1$ MultiKT instances.

PMT-SVM-OVA We also implemented an OVA multiclass extension of the binary transfer learning method by Aytar and Zisserman [2], as done for MultiKT-OVA.

MultisourceTrAdaBoost-OVA As a final transfer learning baseline, we implemented an OVA extension of MultisourceTrAdaBoost [146], where each source corresponds to a subset of samples designated for the source classifier, while belonging to a specific class. We follow the authors by using a linear SVM as weak learner.

Apart for PMT-SVM-OVA and MultisourceTrAdaBoost-OVA, which cannot be kernelized, we used all the features available for each dataset via kernel averaging [49], computing the average of RBF kernels over all available features from the dataset at hand and RBF hyperparameters $\gamma \in \{2^{-5}, 2^{-6}, \dots, 2^8\}$. The trade-off hyperparameter $C \in \{10^{-5}, 10^{-6}, \dots, 10^8\}$ was tuned by 5-fold cross-validation for the no transfer baselines. In case of model-transfer algorithms, the source model’s C value was reused.

Since MultisourceTrAdaBoost-OVA is a non-kernel baseline, to test its performance over multiple features we concatenated them. This approach proved computationally unfeasible for PMT-SVM-OVA (we used the implementation made available by the authors). Thus, to compare fairly with it, we also did run experiments using, for all methods, a linear kernel and a single feature (SIFT for the Caltech-256 and PHOG for the AWA). We refer to this setting as linear.

7.4.3 Evaluation results

Mimicking the setting proposed in Tommasi *et al.* [136], we performed experiments on different groups of related, unrelated and mixed categories for both databases.

For the Caltech-256 database, the related classes were chosen from the “quadruped animals” subset; the unrelated classes were chosen randomly from the whole dataset, and the mixed classes were taken from the “quadruped animals” and the “ground transportation” subsets, sampled in equal proportions. For the AWA database, the related classes were chosen from the “quadruped animals” subset; the unrelated classes were randomly chosen from the whole dataset, and the mixed classes were sampled

in equal proportions from the subsets “quadruped animals” and “aquatic animals”. This setting allows us to evaluate how MULTIpLE, and the chosen baselines, are able to exploit the source knowledge in different situations, while considering the overall accuracy. To assess the performance of all methods as the overall number of classes grows, we repeated all experiments increasing progressively their number, with $K + 1 = 5, 10, 20$ respectively. Because of space constraint and redundancy, only a subset of all experiments is reported here³.

Figure 7.2 shows the results obtained for $K + 1 = 5$. The left column shows the results for the unrelated setting; the center column shows the results for the mixed setting, and the right column shows the results for the related setting. The first row compares the results obtained by MULTIpLE with those of the no transfer baselines (Section 7.4.2), using a single feature and a linear kernel. We see that the performance of MULTIpLE is always better than no transfer, and in two cases out of three is better or on par with Source and Source+1 (hinge) (unrelated and mixed), while it is always similar to Source+1. This is not the case anymore when using multiple features through kernel averaging (Figure 7.2, bottom row): when using the kernelized version of all algorithms, our approach always performs equal or better than most baselines, apart for Batch and in rare cases, Source+1 (hinge). Compared to Batch, in two cases out of three (unrelated, related) MULTIpLE performs on par with it. This is a remarkable result, as the Batch method constitutes an important reference for the behavior of transfer learning algorithms in this setting (Section 7.4.2).

Figure 7.2, middle row, reports results obtained for MULTIpLE and all transfer learning baselines, as defined in Section 7.4.2, for one feature and the linear kernel. We see that our algorithm obtains a better performance compared to all the others, especially in the small sample regime. As our method builds on the MultiKT algorithm, we tested the statistical significance of our performance with respect to it, using the Wilcoxon signed-rank test ($p < 0.05$). In two cases out of three (related, unrelated), MULTIpLE is significantly better than its competitor. This is the case also when using all features via kernel averaging. We mark these cases with a star on the plots (Figure 7.2, middle and bottom row). With respect to the transfer baselines, the related setting seems to be the one more favorable to our approach. With respect to the no transfer baselines, MULTIpLE seems to perform better in the unrelated case.

The performance of PMT-SVM-OVA and Multisource-TrAdaBoost-OVA is disappointing, compared with that achieved by the other two transfer learning baselines, i.e. MultiKT and MKTL. This is true for all settings (related, unrelated and mixed). Particularly, the performance of MultisourceTrAdaBoost-OVA does not seem to benefit from using multiple features (Figure 7.2, middle and bottom row). On the basis of these results, we did not consider these two baseline algorithms in the rest of our experiments.

Figure 7.3 shows results for $K + 1 = 20$ classes on the AwA dataset, for the unrelated (left), mixed (center) and related (right) settings, all features (averaged RBF kernels). For sake of readability, we report here only the baselines which were competitive with, or better than, MULTIpLE in the $K + 1 = 5$ case, in at least one setting. We see that here our algorithm consistently outperforms all transfer learning baselines, especially with a small training set, while obtaining a performance remarkably similar to Batch, in terms of accuracy and behavior. The Wilcoxon signed-rank test ($p < 0.05$) indicates that, in all these experiments MULTIpLE is again significantly better than MultiKT-OVA. These results

³All experimental results and the source code are available at <https://iljaku.github.io>.

suggest that, as the number of sources grows, our method gets closer to the Batch performance while using only a considerably smaller amount of data – the ultimate goal of any effective transfer learning method. Results obtained on the whole AWA dataset support this claim³.

7.5 Discussion and Conclusions

All results confirm our claim that the mere extension to multiclass of existing binary transfer learning algorithms is not sufficient to address the $K \rightarrow K + 1$ problem. This is well illustrated by the gap in performance between MULTIpLE and MultiKT, which is consistent across datasets, settings and the number of classes. The main difference between the two algorithms is the term we added into the objective, that allows to learn the new class, while preserving the performance on the old classes. The results we have shown demonstrate the importance of such a term in the behavior of the algorithm. One might argue that the worse performance of the transfer learning baselines depends on how we implemented the OVA extension for such binary methods. Still, the results obtained by MKTL, the only transfer learning baseline with a multiclass formulation, clearly indicate that the ability to handle multiple sources by itself is not the solution.

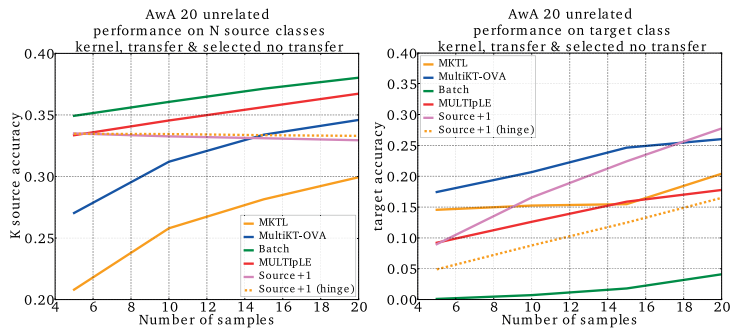


Figure 7.4 – Results for $K + 1 = 20$, AWA, unrelated: accuracy over the K sources (left) and over the $+1$ target (right).

To gain a better understanding on how MULTIpLE balances the need to preserve performance over the sources, and the learning of the target class, we show the accuracy plots for the AWA experiments, $K + 1 = 20$, unrelated, for the K sources and for the $+1$ target separately (Figure 7.4). MULTIpLE and Batch present similar behaviors, as they both preserve the accuracy over the K sources. Both methods do not aggressively leverage over sources for learning the target class, as done by MultiKT-OVA and MKTL (to a lesser extent), although MULTIpLE seems to be able to do so better than Batch. Thus, our choice of optimizing the overall accuracy has resulted in a method able to reproduce the behavior and the performance achievable if all training data would be accessible. Note that training with all the data might not be possible, nor desirable, in all applications. As opposed to this, the OVA extensions of existing binary transfer learning algorithms are more biased towards a strong exploitation of source knowledge when learning the target class, at the expenses of the overall performance. How to combine these two aspects, namely how to design principled methods able to obtain an overall accuracy comparable to that of the Batch method while at the same time boosting the learning of the target class, remains the open challenge of the $K \rightarrow K + 1$ transfer learning problem.

8 Conclusions and Future Directions

This thesis explores an efficient approach to transfer learning, known as the Hypothesis Transfer Learning (HTL), where prior knowledge is retained in a form of *hypotheses*, or models, inherited from previous tasks. The key feature of HTL is its computational advantage with respect to the alternative transfer learning paradigms: transfer is efficient because we do not assume access to the data of previous tasks nor any knowledge about their data-generating mechanisms. This approach to transfer learning has shown its credibility in a wide range of applications, however theoretical justification was largely lacking. The primary goal of this thesis was to contribute to the theoretical foundations of the HTL through analysis of well-known and effective learning algorithms. In the first part of the thesis we outlined such a theory in the context of the convex empirical risk minimization and stochastic optimization with both convex and non-convex loss functions, with implications to HTL. This theoretical analysis quantified the effectiveness of transfer learning by linking the generalization error of the studied algorithms to *the risk of the source hypothesis on the target domain*, a quantity that can be estimated empirically from a training set. In case of stochastic optimization on non-convex objectives, the theory also identified an additional criterion that controls the quality of transfer learning, linked to the expected curvature of the objective function at the initialization point of the optimization procedure. This analysis clearly motivates the design and analysis of novel HTL algorithms. In the second part of the thesis we proposed and evaluated HTL algorithms for two scenarios: efficient binary classification with a very large number of source hypotheses, and multiclass classification where a new class is incorporated incrementally by simultaneous transfer from previous classifiers. The theory and practice of HTL outlined in this thesis have multiple possible directions of future research, discussed in the following.

HTL in Deep Learning. Recent tangible advancements in machine learning are in many ways due to the timely development of algorithms able to find effective representations directly from data and circumventing the need for manual feature engineering, collectively known as deep learning. Deep learning is known for its data-demanding nature and further research in transfer learning coupled with deep learning is a very prominent direction. Indeed, many of the breakthroughs involving deep learning already use rudimentary forms of transfer learning, such as fine-tuning and forms of pre-training. A theoretical understanding of these is generally lacking, and therefore a solid ground for design and analysis of novel deep transfer learning algorithms is missing. The first part of this thesis has illustrated a way of understanding theoretical properties of transfer learning largely *through* the analysis of the

generalization error. Therefore, first it is imperative to contribute to understanding the generalization ability of deep learning algorithms. Empirical evidence suggests that much of the success in training deep learning models comes from the power of stochastic optimization algorithms, such as SGD, and tricks of trade made during model design. That said, particularly interesting directions lie in addressing generalization of deep learning by studying optimization algorithms and architectural choices jointly.

One such direction lies in the *data-dependent* theoretical analysis. Such theory is relevant, not only because it might lead to tight bounds explaining the behavior of deep neural networks, but also because it can suggest how to improve deep learning algorithms by exploiting data-dependent quantities arising during analysis. For instance, Chapter 5, [75] has already identified such a data-dependent quantity, the expected leading eigenvalue of the Hessian matrix, as critical to the generalization of SGD. The data-dependent generalization bounds can reveal the importance of many other interesting characteristics, such as the variance of the gradient or data-dependent impact of architectural recipes. For example, the deep learning community took a long path to discover simple and powerful tricks such as residual layers [60] and batch normalization [63]: can data-dependent theory suggest a systematic way to discover similar recipes? In contrast to the Uniform Convergence arguments of the classical statistical learning theory, a more prominent direction for this type of analysis seems to lie in constructive analysis, such as algorithmic stability and techniques used in the optimization literature [57, 147].

Addressing this problem should allow us to design better deep learning algorithms, for example for transfer learning, as the novel theory could reveal conditions where one could generalize faster. Despite the large body of work on experimental deep transfer learning, gained insights might become valuable when one faces transfer learning problems of optimal choice among a large number of pre-trained source models, individual layers, and in lifelong learning scenarios, where the tasks are acquired sequentially.

HTL in Nonparametric Learning Algorithms. Besides deep learning, HTL has a prominent application in classical problems such as learning in rich nonparametric classes of functions. A few well-known examples of such methods are the nearest-neighbor search, tree-based methods, and kernel methods. Many scalable variants of these algorithms are also actively used in practice. An attractive property of non-parametric algorithms is that some of them are known to achieve Bayes optimality, that is recover the best possible hypothesis within a very large class of functions, e.g. among all smooth functions. Unfortunately, to learn in such a setting, one has to pay a non-parametric price which manifests through the curse of dimensionality in non-asymptotic rates of convergence.

Many works, especially in the context of nearest-neighbor classification and regression, have tried to address this issue through dimensionality reduction and metric learning [7, 74]. Here an interesting opportunity lies in transfer of metrics in both global and local metric learning scenarios with the goal to achieve provable reduction of the curse of dimensionality by HTL.

HTL in Reinforcement Learning. The results presented in this thesis also have the potential to facilitate the design of reinforcement learning algorithms with improved sample complexity. Recently the field of reinforcement learning witnessed impressive achievements due to the introduction of more powerful value functions and policy approximators, again modeled by means of deep learning. Despite

Chapter 8. Conclusions and Future Directions

the long history of work on transfer in reinforcement learning [82, 3], in many cases these agents are trained from scratch on every new task or use a rudimentary form of transfer of approximators, typically adapted from the deep learning literature.

The transfer of approximators clearly bears elements of the hypothesis transfer learning discussed above, with the theory and the algorithms highly applicable. The direction to pursue here is the sample complexity or regret analysis of the popular reinforcement algorithms, such as Q-learning, which would quantitatively involve the notion of the generalization ability of approximators. This might lead to the synthesis between the theory of hypothesis transfer learning and transfer in reinforcement learning. In turn, the goal is to design and analyze reinforcement learning algorithms that are able to transfer by taking advantage of this theory.

A Proofs from Chapter 3

Sketch proof of Theorem 6. We trace the occurrence of $\frac{M^2}{2m}$ to the proof of Lemma 9 [17]. At the beginning of the proof they suggest the following inequality

$$\mathbb{E}_S[(R(A_S) - \hat{R}^{\text{loo}}(A, S))^2] \leq \frac{1}{m} \mathbb{E}_S[\ell(A_{S \setminus i}, z_i)(M - \ell(A_{S \setminus i}, z_j))] \quad (\text{A.1})$$

$$+ \mathbb{E}_{S, z, z'}[\ell(A_{S \setminus i}, z)\ell(A_{S \setminus i}, z') - \ell(A_{S \setminus i}, z)\ell(A_{S \setminus i}, z_i)] \quad (\text{A.2})$$

$$+ \mathbb{E}_{S, z, z'}[\ell(A_{S \setminus i}, z_i)\ell(A_{S \setminus i}, z_j) - \ell(A_{S \setminus i}, z)\ell(A_{S \setminus i}, z_i)]. \quad (\text{A.3})$$

Here we are only interested in the first term, since it is the origin of the term $\frac{M^2}{2m}$. Using the fact that $\ell(A_{S \setminus i}, z_j) \geq 0$, we have

$$\begin{aligned} \frac{1}{m} \mathbb{E}_S[\ell(A_{S \setminus i}, z_i)(M - \ell(A_{S \setminus i}, z_j))] &= \frac{1}{m} \mathbb{E}_S[\ell(A_{S \setminus i}, z_i)(M - \ell(A_{S \setminus i}, z_j))] \\ &\leq \frac{M}{m} \mathbb{E}_S[\ell(A_{S \setminus i}, z_i)]. \quad \square \end{aligned} \quad (\text{A.4})$$

A.1 Proof of Theorem 7

In this section we will mostly use the notation $\mathbf{w}_S \equiv A_S$ and $\mathbf{w}_{S \setminus i} \equiv A_{S \setminus i}$ as a reminder that we are working in a vector space. To prove Theorem 7 we need to upper bound the quantities M , γ , and $\mathbb{E}_S[\ell(A_{S \setminus i}, z_i)]$ of Theorem 6. To do so, we proceed by stating and proving additional lemmas. In particular we first start by proving general statements in subsection A.1.1, used throughout the proof. Next we prove two perturbation bounds which are instrumental in the proof of the stability bound γ . Next we handle terms M and $\mathbb{E}_S[\ell(A_{S \setminus i}, z_i)]$, and corresponding bounds are shown in Lemma 8, while γ is bounded in Theorem 19.

A.1.1 General Statements

Lemma 2. For all $\mathbf{X} \in \mathbb{R}^{m \times d}$, $m, \lambda \geq 0$, we have that the matrix

$$\mathbf{I} - \mathbf{X}(\mathbf{X}^\top \mathbf{X} + m\lambda \mathbf{I})^{-1} \mathbf{X}^\top$$

Appendix A. Proofs from Chapter 3

is PSD and its maximum eigenvalue is less than 1.

Proof.

$$\begin{aligned}
 \mathbf{I} - \mathbf{X}(\mathbf{X}^\top \mathbf{X} + m\lambda \mathbf{I})^{-1} \mathbf{X}^\top &= \mathbf{I} - (\mathbf{X}\mathbf{X}^\top + m\lambda \mathbf{I})^{-1} \mathbf{X}\mathbf{X}^\top \\
 &= \mathbf{I} - \mathbf{U}(\mathbf{\Lambda} + m\lambda \mathbf{I})^{-1} \mathbf{U}^\top \mathbf{U} \mathbf{\Lambda} \mathbf{U}^\top \\
 &= \mathbf{U}(\mathbf{I} - (\mathbf{\Lambda} + m\lambda \mathbf{I})^{-1} \mathbf{\Lambda}) \mathbf{U}^\top,
 \end{aligned} \tag{A.5}$$

where we used the identity $(\mathbf{X}\mathbf{X}^\top + m\lambda \mathbf{I})^{-1} \mathbf{X} = \mathbf{X}(\mathbf{X}^\top \mathbf{X} + m\lambda \mathbf{I})^{-1}$ to obtain (A.5). Subsequently we used the eigendecomposition $\mathbf{X}\mathbf{X}^\top = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top$. \square

Lemma 3. $\forall a, b, \hat{y} \in \mathbb{R}$,

$$|(a - \hat{y})^2 - (b - \hat{y})^2| \leq (a - b)^2 + 2|(b - \hat{y})(a - b)|.$$

Proof.

$$|(a - \hat{y})^2 - (b - \hat{y})^2| = |a^2 - b^2 - 2\hat{y}(a - b)| \tag{A.6}$$

$$= |(a - b)^2 - 2b^2 + 2ab - 2\hat{y}(a - b)| \tag{A.7}$$

$$= |(b - b)^2 + 2(b - \hat{y})(a - b)| \tag{A.8}$$

$$\leq (a - b)^2 + 2|(b - \hat{y})(a - b)|. \tag{A.9}$$

\square

Lemma 4. Let $\alpha \geq 1$, and $C \geq |y|$, then

$$\begin{aligned}
 (T_C(\Delta) - y)^2 &\leq (T_C(y + \alpha(\Delta - y)) - y)^2 \\
 &\leq \alpha^2 (T_C(\Delta) - y)^2.
 \end{aligned}$$

Proof. We only prove the upper bound, noting that the proof of the lower bound is similar. The proof follows from an analysis of all the possible cases. The lemma trivially holds when $|y + \alpha(\Delta - y)| \leq C$. For $\Delta > C$, the bound holds because $y + \alpha(\Delta - y) > C$; the same reasoning applies for $\Delta < -C$. The last case is when $\frac{C-y}{\alpha} + y < \Delta < C$. We have $(T_C(y + \alpha(\Delta - y)) - y)^2 = (C - y)^2$. Note that $C \geq y$ implies that $\frac{C-y}{\alpha} + y > y$, so $\Delta > y$ and this implies the stated bound. The case is analogous $-C < \Delta < -\frac{C+y}{\alpha} + y$: we have that $T_C(y + \alpha(\Delta - y)) = -C$ and $-\frac{C+y}{\alpha} + y \leq y$ because $C + y \geq 0$, hence $\Delta < y$. \square

A.1.2 Perturbation Bounds

The first perturbation bound is given in Lemma 5 capturing the closed-form formula for calculating the change in truncated predictions of RLS when a new sample point is added. This result is related to the well-known closed-form formula for LOO risk for RLS, e.g. see [21]. The second one shown in Lemma 6 bounds the absolute difference between the margins of the LOO estimate and an intact one.

Lemma 5. Let \mathbf{w}_S be the hypothesis produced by the RLS algorithm given training set S . For any i -th example $(\mathbf{x}_i, y_i) \in S$, we have that the hypothesis $\mathbf{w}_{S \setminus i}$ produced by the same RLS algorithm on a training

set $S^{\setminus i}$ satisfies

$$(T_C(\mathbf{x}_i^\top \mathbf{w}_S) - y_i)^2 \leq (T_C(\mathbf{x}_i^\top \mathbf{w}_{S^{\setminus i}}) - y_i)^2 \quad (\text{A.10})$$

$$\leq \left(1 + \frac{1}{m\lambda}\right)^2 (T_C(\mathbf{x}_i^\top \mathbf{w}_S) - y_i)^2. \quad (\text{A.11})$$

Proof. The $\mathbf{w}_{S^{\setminus i}}$ is given by

$$\mathbf{w}_{S^{\setminus i}} = \operatorname{argmin}_{\mathbf{w} \in \mathbb{R}^d} \left\{ \frac{1}{m} \|\mathbf{X}^\top \mathbf{w} - \mathbf{y}\|_2^2 + \lambda \|\mathbf{w}\|^2 \right\},$$

where \mathbf{X} is a matrix $d \times (m-1)$ and \mathbf{y} an $m-1$ dimensional vector, respectively the matrix of the training examples and vector of the training labels without the i -th example. Let $\mathbf{M} := \mathbf{X}^\top \mathbf{X} + m\lambda \mathbf{I}$, then

$$\mathbf{x}_i^\top \mathbf{w}_{S^{\setminus i}} = \mathbf{x}_i^\top \mathbf{X} \mathbf{M}^{-1} \mathbf{y}.$$

It is straightforward to see that $\mathbf{x}_i^\top \mathbf{w}_S$ is equal to

$$\begin{bmatrix} \mathbf{x}_i^\top \mathbf{X} & \|\mathbf{x}_i\|^2 \end{bmatrix} \begin{bmatrix} \mathbf{M} & \mathbf{X}^\top \mathbf{x}_i \\ \mathbf{x}_i^\top \mathbf{X} & \|\mathbf{x}_i\|^2 + m\lambda \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{y} \\ y_i \end{bmatrix}. \quad (\text{A.12})$$

Expanding the middle term and using the block-wise matrix inversion property [111] we get

$$\begin{bmatrix} \mathbf{M} & \mathbf{X}^\top \mathbf{x}_i \\ \mathbf{x}_i^\top \mathbf{X} & \|\mathbf{x}_i\|^2 + m\lambda \end{bmatrix}^{-1} = \begin{bmatrix} \mathbf{M}^{-1} & \mathbf{0} \\ \mathbf{0}^\top & 0 \end{bmatrix} \quad (\text{A.13})$$

$$+ \frac{1}{a} \begin{bmatrix} \mathbf{M}^{-1} \mathbf{X}^\top \mathbf{x}_i \\ -1 \end{bmatrix} \begin{bmatrix} \mathbf{x}_i^\top \mathbf{X} \mathbf{M}^{-1} & -1 \end{bmatrix}, \quad (\text{A.14})$$

where $a := \|\mathbf{x}_i\|^2 + m\lambda - \mathbf{x}_i^\top \mathbf{X} \mathbf{M}^{-1} \mathbf{X}^\top \mathbf{x}_i$. Plugging this result into (A.12) yields

$$\mathbf{x}_i^\top \mathbf{w}_S = \mathbf{x}_i^\top \mathbf{w}_{S^{\setminus i}} - \frac{a - m\lambda}{a} (\mathbf{x}_i^\top \mathbf{w}_{S^{\setminus i}} - y_i).$$

So we have

$$(T_C(\mathbf{x}_i^\top \mathbf{w}_{S^{\setminus i}}) - y_i)^2 = \left(T_C \left(\frac{a}{m\lambda} (\mathbf{x}_i^\top \mathbf{w}_S - y_i) + y_i \right) - y_i \right)^2. \quad (\text{A.15})$$

Observing that $0 \leq \mathbf{x}_i^\top \mathbf{X} \mathbf{M}^{-1} \mathbf{X}^\top \mathbf{x}_i \leq \|\mathbf{x}_i\|^2$, we have that $1 \leq \frac{a}{m\lambda} \leq 1 + \frac{\|\mathbf{x}_i\|^2}{m\lambda}$, hence we use the upper bound in Lemma 4 to derive the stated upper bound. Analogously, the lower bound follows from (A.15) and the lower bound in Lemma 4. \square

Lemma 6. *Let \mathbf{w}_S be the hypothesis produced by the RLS algorithm given training set S . For any sample $(\mathbf{x}, y) \stackrel{iid}{\sim} \mathcal{D}$ and $(\mathbf{x}_i, y_i) \in S$, such that $\|\mathbf{x}\|, \|\mathbf{x}_i\| \leq 1$, we have that the hypothesis $\mathbf{w}_{S^{\setminus i}}$ produced by the same RLS algorithm on a training set $S^{\setminus i} \forall i \in \{1, \dots, m\}$, satisfies*

$$|\mathbf{x}^\top \mathbf{w}_S - \mathbf{x}^\top \mathbf{w}_{S^{\setminus i}}| \leq \frac{1}{m\lambda} |\mathbf{x}_i^\top \mathbf{w}_{S^{\setminus i}} - y_i|.$$

Appendix A. Proofs from Chapter 3

Proof. Define $\mathbf{X} = [\mathbf{x}_1, \dots, \mathbf{x}_{i-1}, \mathbf{x}_{i+1}, \dots, \mathbf{x}_m]$, $\mathbf{M} = \mathbf{X}^\top \mathbf{X} + m\lambda \mathbf{I}$. It is straightforward to see that $\mathbf{x}^\top \mathbf{w}_S$ is equal to

$$\begin{bmatrix} \mathbf{x}^\top \mathbf{X} & \mathbf{x}^\top \mathbf{x}_i \end{bmatrix} \begin{bmatrix} \mathbf{M} & \mathbf{X}^\top \mathbf{x}_i \\ \mathbf{x}_i^\top \mathbf{X} & \|\mathbf{x}_i\|^2 + m\lambda \end{bmatrix}^{-1} \begin{bmatrix} \mathbf{y} \\ y_i \end{bmatrix}. \quad (\text{A.16})$$

Expanding the middle term and using the block-wise matrix inversion property [111] we get

$$\begin{bmatrix} \mathbf{M} & \mathbf{X}^\top \mathbf{x}_i \\ \mathbf{x}_i^\top \mathbf{X} & \|\mathbf{x}_i\|^2 + m\lambda \end{bmatrix}^{-1} = \begin{bmatrix} \mathbf{M}^{-1} & \mathbf{0} \\ \mathbf{0}^\top & 0 \end{bmatrix} + \frac{1}{a} \begin{bmatrix} \mathbf{M}^{-1} \mathbf{X}^\top \mathbf{x}_i \\ -1 \end{bmatrix} \begin{bmatrix} \mathbf{x}_i^\top \mathbf{X} \mathbf{M}^{-1} & -1 \end{bmatrix}, \quad (\text{A.17})$$

where $a := \|\mathbf{x}_i\|^2 + m\lambda - \mathbf{x}_i^\top \mathbf{X} \mathbf{M}^{-1} \mathbf{X}^\top \mathbf{x}_i$. Plugging this result into (A.16) yields

$$\mathbf{x}^\top \mathbf{w}_S = \mathbf{x}^\top \mathbf{w}_{S^i} + \frac{\mathbf{x}^\top (\mathbf{I} - \mathbf{X} \mathbf{M}^{-1} \mathbf{X}^\top) \mathbf{x}_i}{a} (y_i - \mathbf{x}_i^\top \mathbf{w}_{S^i}). \quad (\text{A.18})$$

Using the result of Lemma 2, we have that $m\lambda \leq a$ and in addition by the Cauchy-Schwarz inequality we have that $\mathbf{x}^\top (\mathbf{I} - \mathbf{X} \mathbf{M}^{-1} \mathbf{X}^\top) \mathbf{x}_i \leq 1$, since $\|\mathbf{x}\|, \|\mathbf{x}_i\| \leq 1$. \square

A.1.3 Bounding M and $\mathbb{E}_S[\ell(A_{S^i}, z_i)]$

Next we bound $\mathbb{E}_S[\ell(\mathbf{w}_{S^i}, (\mathbf{x}_i, y_i))]$ and M in Lemma 8, but first we also need to prove the following helpful lemma which bounds the norm of the hypothesis.

Lemma 7. *The following bounds hold for the hypothesis $\hat{\mathbf{w}}_S$ produced by Algorithm 2*

$$\mathbb{E}_S[\|\hat{\mathbf{w}}_S\|^2] \leq \frac{1}{\lambda} R(h^{\text{src}}),$$

and

$$\|\hat{\mathbf{w}}_S\|^2 \leq \frac{1}{\lambda} (B + \|h^{\text{src}}\|_\infty)^2.$$

Proof. We define

$$Q(\mathbf{w}) := \frac{1}{m} \sum_{i=1}^m (\mathbf{x}_i^\top \mathbf{w} - y_i + h^{\text{src}}(\mathbf{x}_i))^2 + \lambda \|\mathbf{w}\|^2.$$

Using the definition of $\hat{\mathbf{w}}_S$ in Algorithm 2, we have that

$$Q(\hat{\mathbf{w}}_S) \leq Q(\mathbf{0}) = \hat{R}(h^{\text{src}}). \quad (\text{A.19})$$

Hence we get $\|\hat{\mathbf{w}}_S\|^2 \leq \frac{\hat{R}(h^{\text{src}})}{\lambda}$. Now

$$\mathbb{E}_S \|\hat{\mathbf{w}}_S\|^2 \leq \frac{1}{\lambda} \mathbb{E}_S \hat{R}(h^{\text{src}}) = \frac{1}{\lambda} R(h^{\text{src}}).$$

For the second upper bound, from (A.19) it also follows

$$\|\widehat{\mathbf{w}}_S\|^2 \leq \frac{1}{m\lambda} \sum_{i=1}^m (h^{\text{src}}(\mathbf{x}_i) - y_i)^2 \leq \frac{1}{\lambda} (B + \|h^{\text{src}}\|_\infty)^2. \quad \square$$

Lemma 8. Assume $(\mathbf{x}, y) \stackrel{iid}{\sim} \mathcal{D}$. For Algorithm 2 the following bounds hold $\forall i \in \{1, \dots, m\}$

$$\sup_{\mathbf{x}, y} (T_C(\mathbf{x}^\top \widehat{\mathbf{w}}_{S^i}) - y + h^{\text{src}}(\mathbf{x}))^2 \leq \left(1 + \frac{1}{m\lambda}\right)^2 \left(T_C\left(\frac{B + \|h^{\text{src}}\|_\infty}{\sqrt{\lambda}}\right) + B + \|h^{\text{src}}\|_\infty\right)^2, \quad (\text{A.20})$$

and

$$\mathbb{E}_S [(T_C(\mathbf{x}^\top \widehat{\mathbf{w}}_S) - y + h^{\text{src}}(\mathbf{x}_i))^2] \leq 2 \left(T_{C^2}\left(\frac{R(h^{\text{src}})}{\lambda}\right) + R(h^{\text{src}})\right). \quad (\text{A.21})$$

Proof. We use Lemma 5, and the Cauchy-Schwarz inequality to derive

$$\sup_{\mathbf{x}, y} (T_C(\mathbf{x}^\top \widehat{\mathbf{w}}_{S^i}) - y + h^{\text{src}}(\mathbf{x}))^2 \leq \left(1 + \frac{1}{m\lambda}\right)^2 \sup_{\mathbf{x}, y} (T_C(\mathbf{x}^\top \widehat{\mathbf{w}}_S) - y + h^{\text{src}}(\mathbf{x}))^2 \quad (\text{A.22})$$

$$\leq \left(1 + \frac{1}{m\lambda}\right)^2 \sup_{\mathbf{x}, y} (|T_C(\mathbf{x}^\top \widehat{\mathbf{w}}_S)| + B + \|h^{\text{src}}\|_\infty)^2. \quad (\text{A.23})$$

The term $|T_C(\mathbf{x}^\top \widehat{\mathbf{w}}_S)|$ can be simultaneously upper bounded using C and, using the Cauchy-Schwarz inequality, by $\|\widehat{\mathbf{w}}_S\|$. Hence using the second result of Lemma 7 we obtain the first result.

For the second upper bound, using the elementary inequality $(a + b)^2 \leq 2(a^2 + b^2)$, in an analogous way we have

$$\begin{aligned} \mathbb{E}_S [(T_C(\mathbf{x}^\top \widehat{\mathbf{w}}_S) - y + h^{\text{src}}(\mathbf{x}))^2] &\leq 2 \mathbb{E}_S [(T_C(\mathbf{x}^\top \widehat{\mathbf{w}}_S))^2 + (h^{\text{src}}(\mathbf{x}) - y)^2] \\ &= 2 \left(\mathbb{E}_S [(T_C(\mathbf{x}^\top \widehat{\mathbf{w}}_S))^2] + R(h^{\text{src}})\right). \end{aligned}$$

Again, the first term in the left hand side of the last inequality can be simultaneously upper bounded using C^2 and, using the Cauchy-Schwarz inequality, by $\|\widehat{\mathbf{w}}\|^2$. Hence the first result of Lemma 7 concludes the proof. \square

A.1.4 Hypothesis Stability γ and Generalization Bound

Now, we are ready to upper-bound the hypothesis stability for Algorithm 2.

Theorem 19. The hypothesis stability of Algorithm 2 is upper bounded as

$$\gamma \leq T_{4C^2} \left(\frac{2R(h^{\text{src}})}{m^2\lambda^2} \left(1 + \frac{1}{\lambda}\right)\right) + 2T_{2C} \left(\frac{\sqrt{2R(h^{\text{src}})}}{m\lambda} \sqrt{1 + \frac{1}{\lambda}}\right) \sqrt{2T_{C^2} \left(\frac{R(h^{\text{src}})}{\lambda}\right) + 2R(h^{\text{src}})}. \quad (\text{A.24})$$

Proof. From Lemma 3 with $a = T_C(\Delta + \epsilon)$, $b = T_C(\Delta)$, $\widehat{y} = y - h^{\text{src}}(\mathbf{x})$ and also using the fact that

Appendix A. Proofs from Chapter 3

$|T_C(\Delta + \epsilon) - T_C(\Delta)| \leq \min(|\epsilon|, 2C)$, we have

$$\begin{aligned} & |(T_C(\Delta + \epsilon) - y + h^{\text{src}}(\mathbf{x}))^2 - (T_C(\Delta) - y + h^{\text{src}}(\mathbf{x}))^2| \\ & \leq \min(\epsilon^2, 4C^2) + 2 \min(|\epsilon|, 2C) |T_C(\Delta) - y + h^{\text{src}}(\mathbf{x})|. \end{aligned}$$

Set $\Delta := \mathbf{x}^\top \mathbf{w}_{S^i}$, and $\Delta + \epsilon := \mathbf{x}^\top \mathbf{w}_S$. Taking the expectation $\mathbb{E}[\cdot] = \mathbb{E}_{S,(\mathbf{x},y)}[\cdot]$, and using Jensen's and Cauchy-Schwarz's inequalities, we have

$$\mathbb{E} \left[|(T_C(\Delta + \epsilon) - y + h^{\text{src}}(\mathbf{x}))^2 - (T_C(\Delta) - y + h^{\text{src}}(\mathbf{x}))^2| \right] \quad (\text{A.25})$$

$$\leq \min(\mathbb{E}[\epsilon^2], 4C^2) + 2 \min(\sqrt{\mathbb{E}[\epsilon^2]}, 2C) \sqrt{\mathbb{E}[(T_C(\Delta) - y + h^{\text{src}}(\mathbf{x}))^2]} \quad (\text{A.26})$$

$$\leq \min(\mathbb{E}[\epsilon^2], 4C^2) + 2 \min(\sqrt{\mathbb{E}[\epsilon^2]}, 2C) \sqrt{\mathbb{E}[2T_C^2(\|\widehat{\mathbf{w}}_S\|^2) + 2(h^{\text{src}}(\mathbf{x}) - y)^2]} \quad (\text{A.27})$$

$$\leq \min(\mathbb{E}[\epsilon^2], 4C^2) + 2 \min(\sqrt{\mathbb{E}[\epsilon^2]}, 2C) \sqrt{2T_C^2 \left(\frac{R(h^{\text{src}})}{\lambda} \right) + 2R(h^{\text{src}})}. \quad (\text{A.28})$$

In (A.27) we apply the Cauchy-Schwarz inequality and the elementary inequality $(a + b)^2 \leq 2a^2 + 2b^2$, while (A.28) comes from the first result of Lemma 7.

We now use Lemma 6 to have that

$$\begin{aligned} \mathbb{E}[\epsilon^2] & \leq \frac{1}{m^2 \lambda^2} \mathbb{E}[(\mathbf{x}^\top \widehat{\mathbf{w}}_{S^i} - y_i + h^{\text{src}}(\mathbf{x}))^2] \\ & \leq \frac{2}{m^2 \lambda^2} \mathbb{E}[\|\widehat{\mathbf{w}}_{S^i}\|^2 + (y - h^{\text{src}}(\mathbf{x}))^2] \\ & \leq \frac{2R(h^{\text{src}})}{m^2 \lambda^2} \left(\frac{m-1}{m} \frac{1}{\lambda} + 1 \right) \\ & \leq \frac{2R(h^{\text{src}})}{m^2 \lambda^2} \left(\frac{1}{\lambda} + 1 \right). \end{aligned}$$

Putting all together we have

$$\begin{aligned} & \mathbb{E}_{S,(\mathbf{x},y)} \left[|(T_C(\Delta) - y + h^{\text{src}}(\mathbf{x}))^2 - (T_C(\Delta + \epsilon) - y + h^{\text{src}}(\mathbf{x}))^2| \right] \\ & \leq T_{4C^2} \left(\frac{2R(h^{\text{src}})}{m^2 \lambda^2} \left(1 + \frac{1}{\lambda} \right) \right) + 2T_{2C} \left(\frac{\sqrt{2R(h^{\text{src}})}}{m\lambda} \sqrt{1 + \frac{1}{\lambda}} \right) \sqrt{2T_C^2 \left(\frac{R(h^{\text{src}})}{\lambda} \right) + 2R(h^{\text{src}})}. \end{aligned}$$

□

With the results above we now prove Theorem 7.

Proof of Theorem 7. We apply Theorem 6. To apply this theorem, we need to upper-bound quantities $M, \mathbb{E}_S[\ell(\mathbf{w}_{S^i}, (\mathbf{x}_i, y_i))]$, and γ . Using the upper bound in Lemma 5 and the second result in Lemma 8, we have

$$\mathbb{E}_S[\ell(A_{S^i}, (\mathbf{x}_i, y_i))] \leq 2 \left(1 + \frac{1}{m\lambda} \right)^2 \left(T_C^2 \left(\frac{R(h^{\text{src}})}{\lambda} \right) + R(h^{\text{src}}) \right), \quad (\text{A.29})$$

we use bound on γ given by Theorem 19. By assumption on the loss function

$$\|\ell\|_\infty \leq M.$$

So we have

$$\sup_{\mathbf{x}, y} (T_C(\mathbf{x}^\top \mathbf{w}_{S^i}) - \hat{y} + h^{\text{src}}(\mathbf{x}))^2 \leq \left(T_C \left(\frac{B + \|h^{\text{src}}\|_\infty}{\sqrt{\lambda}} \right) + B + \|h^{\text{src}}\|_\infty \right)^2.$$

We have this result, because the term $T_C(\mathbf{x}^\top \mathbf{w}_{S^i})$ can be simultaneously upper-bounded by C and, using the Cauchy-Schwarz inequality, $\|\mathbf{w}_{S^i}\|$. Consequently, $\|\mathbf{w}_{S^i}\|$ is bounded using the second result of Lemma 7. Putting it all together and applying Theorem 6, we have that

$$\mathbb{E}_S [(R(A^{\text{htl}}) - \hat{R}^{\text{loo}}(A^{\text{htl}}))^2] = \mathcal{O} \left(\frac{C^2 \sqrt{R(h^{\text{src}}) T_C^2 \left(\frac{R(h^{\text{src}})}{\lambda} \right) + R(h^{\text{src}})^2}}{m\lambda^{1.5}} \right), \quad (\text{A.30})$$

where $C \geq B + \|h^{\text{src}}\|_\infty$. Applying Chebyshev's inequality we get the statement. The dominant rates in $\mathcal{O}(\cdot)$ notation, in both truncated and untruncated cases, come from the bound on the component $M\gamma$ in Theorem 6. \square

B Proofs from Chapter 4

Our main result is Theorem 8, and before proving it we introduce a few instrumental theorems.

Theorem 20 (Steele's inequality [131]). *Let $F : \mathcal{X}^m \rightarrow \mathbb{R}$ be any measurable function. Then,*

$$\mathbb{E}_S \left[\left(F(S) - \mathbb{E}_{S'} [F(S')] \right)^2 \right] \leq \frac{1}{2} \sum_{k=1}^m \mathbb{E}_{S,z} \left[\left(F(S) - F(S^{(k)}) \right)^2 \right].$$

Theorem 21 (Bennett's and Bernstein's inequalities [16]). *Let X_1, \dots, X_m be independent random variables with finite variance such that $\mathbb{P}(|X_i - \mathbb{E}[X_i]| < M) = 1$ for some $M > 0$ and all $i = 1, \dots, m$. Let*

$$Z = \sum_{i=1}^m X_i \quad \text{and} \quad v = \sum_{i=1}^m \mathbb{V}[X_i].$$

Then for any $t > 0$

$$\begin{aligned} \mathbb{P}(Z - \mathbb{E}[Z] \geq t) &\leq \exp \left(-\frac{v}{M^2} h \left(\frac{Mt}{v} \right) \right) \\ &\leq \exp \left(-\frac{t^2/2}{v + Mt/3} \right), \end{aligned}$$

where $h(u) = (1 + u) \log(1 + u) - u$ for all $u > 0$.

Theorem 22 (Theorem 13.2 in [124]). *Let $i \stackrel{iid}{\sim} U(\{1, \dots, m\})$. Then for any learning algorithm A ,*

$$\mathbb{E}_S [R(A_S) - \widehat{R}_S(A_S)] = \mathbb{E}_{S, z, i} [\ell(A_{S^{(i)}}, z_i) - \ell(A_S, z_i)].$$

Lemma 9. *Let $a, b > 0$ such that $b = (1 + a) \log(1 + a) - a$. Then $a \leq \frac{3b}{2 \log(\sqrt{b+1})}$.*

Proof. It is easy to verify that the inverse function $f^{-1}(b)$ of $f(a) = (1 + a) \log(1 + a) - a$ is

$$f^{-1}(b) = \exp \left[W \left(\frac{b-1}{e} \right) + 1 \right] - 1,$$

where the function $W : \mathbb{R}_+ \rightarrow \mathbb{R}$ is the Lambert function that satisfies

$$x = W(x) \exp(W(x)).$$

Hence, to obtain an upper bound to a , we need an upper bound to the Lambert function. We use Theorem 2.3 in [62], that says that

$$W(x) \leq \log \frac{x+C}{1+\log(C)}, \quad \forall x > -\frac{1}{e}, C > \frac{1}{e}.$$

Setting $C = \frac{\sqrt{b+1}}{e}$, we obtain

$$a = f^{-1}(b) \leq e \frac{\frac{b-1}{e} + \frac{\sqrt{b+1}}{e}}{1 + \log\left(\frac{\sqrt{b+1}}{e}\right)} - 1 = \frac{b + \sqrt{b}}{\log(\sqrt{b+1})} - 1 \leq \frac{3b}{2\log(\sqrt{b+1})},$$

where in the last inequality we used the fact that $x + \sqrt{x} - \log(\sqrt{x+1}) \leq \frac{3}{2}x, \forall x \geq 0$, as it can be easily verified by comparing the derivatives of both terms. \square

Proof of Theorem 8. The idea of the proof is to relate on-average stability to the variance in Bennett's and Bernstein's inequalities. We will ultimately apply these inequalities with random variables

$$X_{S,i} = R(A_S) - \ell(A_S, z_i).$$

Note that $X_{S,i}$ is random only in S (since $z_i \in S$ and A is deterministic). To apply these concentration bounds we have to upper bound two terms:

- 1) Expectation term $\sum_{i=1}^m \mathbb{E}_S[X_{S,i}]$,
- 2) Variance term $v = \sum_{i=1}^m \mathbb{V}_S[X_{S,i}]$.

1) Handling the expectation term We start by looking at its expectation, that is

$$\mathbb{E}_S[X_{S,i}] = \mathbb{E}_S[R(A_S) - \ell(A_S, z_i)]. \tag{B.1}$$

By Theorem 22 we have that

$$\sum_{i=1}^m \mathbb{E}_S[X_{S,i}] = m \mathbb{E}_S[R(A_S) - \widehat{R}_S(A_S)] \tag{B.2}$$

$$= m \mathbb{E}_{S,z,i}[\ell(A_{S^{(i)}}, z_i) - \ell(A_S, z_i)] \tag{B.3}$$

$$\leq m \sup_{z'} \mathbb{E}_{S,z,i}[\ell(A_{S^{(i)}}, z') - \ell(A_S, z')] \tag{B.4}$$

$$\leq m \epsilon_m. \tag{B.5}$$

Appendix B. Proofs from Chapter 4

2) **Handling the variance term** Consider the variance of $X_{S,i}$,

$$\mathbb{V}_S[X_{S,i}] = \mathbb{E}_S \left[\left(R(A_S) - \ell(A_S, z_i) - \mathbb{E}_{S'} [R(A_{S'}) - \ell(A_{S'}, z'_i)] \right)^2 \right]. \quad (\text{B.6})$$

Now we further bound the variance by using Steele's inequality, Theorem 20, with $F(S) = R(A_S) - \ell(A_S, z_i)$,

$$\mathbb{V}_S[X_{S,i}] \leq \frac{1}{2} \sum_{k=1}^m \mathbb{E}_{S,z} \left[(R(A_S) - R(A_{S^{(k)}}) + \ell(A_{S^{(k)}}) - \ell(A_S, z_i))^2 \right] \quad (\text{B.7})$$

$$\leq \sum_{k=1}^m \mathbb{E}_{S,z} \left[(R(A_S) - R(A_{S^{(k)}}))^2 \right] + \sum_{k=1}^m \mathbb{E}_{S,z} \left[(\ell(A_{S^{(k)}}) - \ell(A_S, z_i))^2 \right]. \quad (\text{B.8})$$

We first handle the term involving $R(\cdot)$,

$$\sum_{k=1}^m \mathbb{E}_{S,z} \left[(R(A_S) - R(A_{S^{(k)}}))^2 \right] = m \mathbb{E}_{S,z,k} \left[\left(\mathbb{E}_{z'} [\ell(A_S, z') - \ell(A_{S^{(k)}}) - \ell(A_{S^{(k)}}) - \ell(A_S, z')] \right)^2 \right] \quad (\text{B.9})$$

$$\leq m \mathbb{E}_{S,z,z',k} \left[(\ell(A_S, z') - \ell(A_{S^{(k)}}) - \ell(A_{S^{(k)}}) - \ell(A_S, z'))^2 \right] \quad (\text{By Jensen's inequality})$$

$$\leq m \sup_{z'} \mathbb{E}_{S,z,k} \left[(\ell(A_S, z') - \ell(A_{S^{(k)}}) - \ell(A_{S^{(k)}}) - \ell(A_S, z'))^2 \right] \quad (\text{B.10})$$

$$\leq m \epsilon_m^{(2)}. \quad (\text{B.11})$$

Now for the term involving $\ell(\cdot, \cdot)$ we have that

$$\sum_{k=1}^m \mathbb{E}_{S,z} \left[(\ell(A_{S^{(k)}}) - \ell(A_S, z_i))^2 \right] = m \mathbb{E}_{S,z,k} \left[(\ell(A_{S^{(k)}}) - \ell(A_S, z_i))^2 \right] \quad (\text{B.12})$$

$$\leq m \sup_{z'} \mathbb{E}_{S,z,k} \left[(\ell(A_{S^{(k)}}) - \ell(A_S, z'))^2 \right] \quad (\text{B.13})$$

$$\leq m \epsilon_m^{(2)}. \quad (\text{B.14})$$

Thus we get that

$$\mathbb{V}_S[X_{S,i}] \leq 2m \epsilon_m^{(2)},$$

and

$$v = \sum_{i=1}^m \mathbb{V}_S[X_{S,i}] \leq 2m^2 \epsilon_m^{(2)}. \quad (\text{B.15})$$

Bennett's bound. Using the first inequality of Theorem 21 and bounding $\sum_{i=1}^m \mathbb{E}_S[X_{S,i}]$ using (B.2)-(B.5) we have that

$$\mathbb{P} \left(mR(A_S) - m\widehat{R}_S(A_S) - m\epsilon_m \geq t \right) \leq \exp \left(-\frac{v}{M^2} h \left(\frac{Mt}{v} \right) \right). \quad (\text{B.16})$$

Letting

$$\delta = \exp\left(-\frac{\nu}{M^2}h\left(\frac{Mt}{\nu}\right)\right) \quad (\text{B.17})$$

$$\Rightarrow \frac{M^2 \log(1/\delta)}{\nu} = h\left(\frac{Mt}{\nu}\right), \quad (\text{B.18})$$

where $h(x) = (1+x)\log(1+x) - x$. Then

$$h\left(\frac{Mt}{\nu}\right) = \left(1 + \frac{Mt}{\nu}\right) \log\left(1 + \frac{Mt}{\nu}\right) - \frac{Mt}{\nu}$$

and applying Lemma 9 we get that

$$\begin{aligned} t &\leq \frac{3\nu h\left(\frac{Mt}{\nu}\right)}{2M \log\left(1 + \sqrt{h\left(\frac{Mt}{\nu}\right)}\right)} \\ &\leq \frac{3M \log(1/\delta)}{2 \log\left(1 + M \sqrt{\frac{\log(1/\delta)}{\nu}}\right)}. \end{aligned} \quad (\text{Plugging (B.18)})$$

Thus from (B.16) and (B.15) we conclude that with probability at least $1 - \delta$

$$mR(A_S) - m\widehat{R}_S(A_S) \leq m\epsilon_m + \frac{3M \log(1/\delta)}{2 \log\left(1 + \frac{M}{m} \sqrt{\frac{\log(1/\delta)}{2\epsilon_m^{(2)}}}\right)}.$$

Dividing through by m proves the first inequality.

Bernstein's bound. Now we consider Bernstein's bound. Similarly as in the Bennett case we have that

$$\mathbb{P}(mR(A_S) - m\widehat{R}_S(A_S) - m\epsilon_m \geq t) \leq \exp\left(-\frac{t^2/2}{\nu + Mt/3}\right). \quad (\text{B.19})$$

Letting

$$\delta = \exp\left(-\frac{t^2/2}{\nu + Mt/3}\right) \quad (\text{B.20})$$

$$\Rightarrow t \leq \frac{2}{3}M \ln(1/\delta) + \sqrt{2 \ln(1/\delta) \nu} \quad (\text{B.21})$$

$$t \leq \frac{2}{3}M \ln(1/\delta) + \sqrt{4 \ln(1/\delta) m^2 \epsilon_m^{(2)}}. \quad (\text{B.22})$$

Thus Bernstein's inequality gives us that with probability at least $1 - \delta$,

$$mR(A_S) - m\widehat{R}_S(A_S) \leq m\epsilon_m + \frac{2}{3}M \ln(1/\delta) + m \sqrt{4 \ln(1/\delta) \epsilon_m^{(2)}}, \quad (\text{B.23})$$

and dividing both sides by m proves the second inequality.

□

Proof of Theorem 9. Denote $\hat{\mathbf{u}}_S = [\hat{\mathbf{w}}_S^\top, \hat{\boldsymbol{\beta}}_S^\top]^\top$, observe that $A_S^{\text{htl}}(\mathbf{x}) = \langle \hat{\mathbf{u}}_S, \mathbf{x} \rangle$, and introduce a loss function ℓ , such that

$$\ell(\hat{\mathbf{u}}_S, (\mathbf{x}, y)) = \phi(A_S^{\text{htl}}(\mathbf{x}), y). \quad (\text{B.24})$$

Assuming that the loss function is L -Lipschitz we have that

$$\sup_{z'} \mathbb{E}_{S,z,i} [\ell(\hat{\mathbf{u}}_{S^{(i)}}, z') - \ell(\hat{\mathbf{u}}_S, z')] \leq L \mathbb{E}_{S,z,i} [\|\hat{\mathbf{u}}_S - \hat{\mathbf{u}}_{S^{(i)}}\|], \quad (\text{B.25})$$

and similarly

$$\sup_{z'} \mathbb{E}_{S,z,i} [(\ell(\hat{\mathbf{u}}_S, z') - \ell(\hat{\mathbf{u}}_{S^{(i)}}, z'))^2] \leq L^2 \mathbb{E}_{S,z,i} [\|\hat{\mathbf{u}}_S - \hat{\mathbf{u}}_{S^{(i)}}\|^2], \quad (\text{B.26})$$

[123, end of page 143] showed that the minimizer $\hat{\mathbf{u}}_S$ of ERM with a 2λ -strongly-convex and H -smooth loss function, assuming that $H \leq \frac{m\lambda}{2}$, satisfies

$$\|\hat{\mathbf{u}}_S - \hat{\mathbf{u}}_{S^{(i)}}\| \leq \frac{\sqrt{8H}}{m\lambda} \left(\sqrt{\ell(\hat{\mathbf{u}}_S, z_i)} + \sqrt{\ell(\hat{\mathbf{u}}_{S^{(i)}}, z)} \right).$$

We take the square of both sides, apply the inequality $(a+b)^2 \leq a^2 + b^2$ for $a, b \geq 0$, and take the expectation w.r.t. S, z , and i to get that

$$\mathbb{E}_{S,z,i} [\|\hat{\mathbf{u}}_S - \hat{\mathbf{u}}_{S^{(i)}}\|^2] \leq \frac{8H}{m^2\lambda^2} \left(\mathbb{E}_{S,i} [\ell(\hat{\mathbf{u}}_S, z_i)] + \mathbb{E}_{S,z,i} [\ell(\hat{\mathbf{u}}_{S^{(i)}}, z)] \right) \quad (\text{B.27})$$

$$= \frac{16H\mathbb{E}_S [\hat{R}_S(\hat{\mathbf{u}}_S)]}{m^2\lambda^2}, \quad (\text{B.28})$$

where the last identity comes by observing that $\mathbb{E}_{S,i} [\ell(\hat{\mathbf{u}}_S, z_i)] = \mathbb{E}_{S,z,i} [\ell(\hat{\mathbf{u}}_{S^{(i)}}, z)] = \mathbb{E}_S [\hat{R}_S(\hat{\mathbf{u}}_S)]$ recalling that $z_i, z, S \stackrel{\text{iid}}{\sim} \mathcal{D}^{m+2}$. Since A_S^{htl} is a minimizer of a regularized empirical risk we also have that

$$\hat{R}_S(A_S^{\text{htl}}) \leq \hat{R}_S(A_S^{\text{htl}}) + \lambda\Omega(\hat{\mathbf{w}}_S) + \lambda\Omega(\hat{\boldsymbol{\beta}}_S) \quad (\text{B.29})$$

$$\leq \hat{R}_S(h_{\hat{\boldsymbol{\beta}}_S}^{\text{src}}) + \lambda\Omega(\hat{\boldsymbol{\beta}}_S), \quad (\text{B.30})$$

which implies that

$$\hat{R}_S(A_S^{\text{htl}}) \leq \hat{R}_S(h_{\hat{\boldsymbol{\beta}}_S}^{\text{src}}).$$

Thus, considering the above and plugging (B.28) into (B.26) we get that

$$\epsilon_m^{(2)} = \frac{16L^2 H \mathbb{E}_S [\hat{R}_S(h_{\hat{\boldsymbol{\beta}}_S}^{\text{src}})]}{m^2\lambda^2} = \frac{16L^2 H R^{\text{src}}}{m^2\lambda^2},$$

and similarly

$$\epsilon_m = \frac{4L\sqrt{HR^{\text{src}}}}{m\lambda}.$$

Plugging the stability results into Theorem 8 gives the statement. \square

Proof of Theorem 10. By definition of A_S^{htl} in R-ERM-HTL we have that

$$\widehat{R}_S(A_S^{\text{htl}}) \leq \widehat{R}_S(A_S^{\text{htl}}) + \lambda\Omega(\widehat{\mathbf{w}}_S) + \lambda\Omega(\widehat{\boldsymbol{\beta}}_S) \quad (\text{B.31})$$

$$\leq \widehat{R}_S(\mathbf{w}^*) + \lambda\Omega(\mathbf{w}^*). \quad (\text{B.32})$$

Plugging the above into (4.6) we get that

$$R(A_S^{\text{htl}}) - \widehat{R}_S(\mathbf{w}^*) \leq \lambda\Omega(\mathbf{w}^*) + \frac{4L(1 + \sqrt{4\eta})\sqrt{HR^{\text{src}}}}{m\lambda} + \frac{1.5M\eta}{m} \quad (\text{B.33})$$

$$\lambda = \sqrt{\frac{4L(1 + \sqrt{4\eta})\sqrt{HR^{\text{src}}}}{m\Omega(\mathbf{w}^*)}}.$$

Plugging this back we get

$$R(A_S^{\text{htl}}) - \widehat{R}_S(\mathbf{w}^*) \leq 4\sqrt{\frac{\Omega(\mathbf{w}^*)L(1 + \sqrt{4\eta})\sqrt{HR^{\text{src}}}}{m}} + \frac{1.5M\eta}{m} \quad (\text{B.34})$$

Finally concentrating $\widehat{R}_S(\mathbf{w}^*)$ around $R(\mathbf{w}^*)$ using Bernstein's inequality, that is

$$\widehat{R}_S(\mathbf{w}^*) \leq R(\mathbf{w}^*) + \sqrt{\frac{2R(\mathbf{w}^*)\eta}{m}} + \frac{1.5M\eta}{m}, \quad (\text{B.35})$$

and using the fact that $R(\mathbf{w}^*) \leq R(h_{\widehat{\boldsymbol{\beta}}_S}^{\text{src}})$ we get that

$$R(A_S^{\text{htl}}) - R(\mathbf{w}^*) \leq 4\sqrt{\frac{\Omega(\mathbf{w}^*)L(1 + \sqrt{4\eta})\sqrt{HR^{\text{src}}}}{m}} + \sqrt{\frac{2R^{\text{src}}\eta}{m}} + \frac{3M\eta}{m}, \quad (\text{B.36})$$

which completes the proof. \square

C Proofs from Chapter 5

In this section we present proofs of all the statements.

Proof of Theorem 12. Indicate by $S = \{z_i\}_{i=1}^m$ and $S' = \{z'_i\}_{i=1}^m$ independent training sets sampled i.i.d. from \mathcal{D} , and let $S^{(i)} = \{z_1, \dots, z_{i-1}, z'_i, z_{i+1}, \dots, z_m\}$, such that $z'_i \stackrel{\text{iid}}{\sim} \mathcal{D}$. We relate expected empirical risk and expected risk by

$$\begin{aligned} \mathbb{E}_{S,A} \mathbb{E}[\widehat{R}_S(A_S)] &= \mathbb{E}_{S,A} \mathbb{E} \left[\frac{1}{m} \sum_{i=1}^m \ell(A_S, z_i) \right] \\ &= \mathbb{E}_{S,S',A} \mathbb{E} \left[\frac{1}{m} \sum_{i=1}^m \ell(A_{S^{(i)}}, z'_i) \right] \\ &= \mathbb{E}_{S,S',A} \mathbb{E} \left[\frac{1}{m} \sum_{i=1}^m \ell(A_S, z'_i) \right] - \delta \\ &= \mathbb{E}_{S,A} \mathbb{E}[R(A_S)] - \delta, \end{aligned}$$

where

$$\begin{aligned} \delta &= \mathbb{E}_{S,S',A} \mathbb{E} \left[\frac{1}{m} \sum_{i=1}^m (\ell(A_S, z'_i) - \ell(A_{S^{(i)}}, z'_i)) \right] \\ &= \frac{1}{m} \sum_{i=1}^m \mathbb{E}_{S,z'_i,A} \mathbb{E}[\ell(A_S, z'_i) - \ell(A_{S^{(i)}}, z'_i)]. \end{aligned}$$

Renaming z'_i as z and taking sup over i we get that

$$\delta \leq \sup_{i \in [m]} \left\{ \mathbb{E}_{S,z,A} \mathbb{E}[\ell(A_S, z) - \ell(A_{S^{(i)}}, z)] \right\}.$$

This completes the proof. □

C.1 Preliminaries

We say that the SGD gradient update rule is an operator $G_t : \mathcal{H} \mapsto \mathcal{H}$, such that

$$G_t(\mathbf{w}) := \mathbf{w} - \alpha_t \nabla \ell(\mathbf{w}, z_{i_t}),$$

and it is also a function of the training set S and a random index set I . Then, $\mathbf{w}_{t+1} = G_t(\mathbf{w}_t)$, throughout $t = 1, \dots, T$. Moreover we will use the notation $\mathbf{w}_{S,t}$ to indicate the output of SGD ran on a training set S , at step t , and define

$$\delta_t(S, z) := \|\mathbf{w}_{S,t} - \mathbf{w}_{S^{(i)},t}\|.$$

Next, we summarize a few instrumental facts about G_t and a few statements about the loss functions used in our proofs.

Definition 16 (Expansiveness). *A gradient update rule is η -expansive if for all \mathbf{w}, \mathbf{v} ,*

$$\|G_t(\mathbf{w}) - G_t(\mathbf{v})\| \leq \eta \|\mathbf{w} - \mathbf{v}\|.$$

The following lemma characterizes expansiveness for the gradient update rule under different assumptions on ℓ .

Lemma 10 (Lemma 3.6 in [57]). *Assume that ℓ is β -smooth. Then, we have that:*

- 1) G_t is $(1 + \alpha_t \beta)$ -expansive,
- 2) If ℓ in addition is convex, then, for any $\alpha_t \leq \frac{2}{\beta}$, the gradient update rule G_t is 1-expansive.

An important consequence of β -smoothness of ℓ is self-boundedness [123], which we will use on many occasions.

Lemma 11 (Self-boundedness). *For a β -smooth non-negative function ℓ we have that*

$$\|\nabla \ell(\mathbf{w}, z)\| \leq \sqrt{2\beta \ell(\mathbf{w}, z)}.$$

Self-boundedness in turn implies the following boundedness of a gradient update rule.

Corollary 5. *Assume that ℓ is β -smooth and non-negative. Then,*

$$\|\mathbf{w} - G_t(\mathbf{w})\| = \alpha_t \|\nabla \ell(\mathbf{w}, z_{j_t})\| \leq \alpha_t \min \left\{ \sqrt{2\beta \ell(\mathbf{w}, z_{j_t})}, L \right\}.$$

Proof. By Lemma 11

$$\|\alpha_t \nabla \ell(\mathbf{w}, z_{j_t})\| \leq \alpha_t \sqrt{2\beta \ell(\mathbf{w}, z_{j_t})},$$

and also by Lipschitzness of ℓ , $\|\alpha_t \nabla \ell(\mathbf{w}, z_{j_t})\| \leq \alpha_t L$. □

Appendix C. Proofs from Chapter 5

Next we introduce a bound that relates the risk of the output at step t to the risk of the initialization point \mathbf{w}_1 through the variance of the gradient. Given an appropriate choice of step size, this bound will be crucial at stating stability bounds that depend on the risk at \mathbf{w}_1 . The proof idea is similar to the one of [50]. In particular, it does not require convexity of the loss function.

Lemma 12. *Assume that the loss function ℓ is β -smooth. Then, for $\mathbf{w}_{S,t}$ we have that*

$$\mathbb{E}_S[\ell(\mathbf{w}_{S,t}, z_{j_t}) - \ell(\mathbf{w}_1, z_{j_t})] \leq \sum_{k=1}^{t-1} \alpha_k \left(\frac{\alpha_k \beta}{2} - 1 \right) \mathbb{E}_S[\|\nabla \ell(\mathbf{w}_{S,k}, z_{j_k})\|^2].$$

Proof. For brevity let $\mathbf{w}_k = \mathbf{w}_{S,k}$. Since ℓ is β -smoothness we have

$$\ell(\mathbf{w}_{k+1}, z_{j_t}) - \ell(\mathbf{w}_k, z_{j_t}) \leq \nabla \ell(\mathbf{w}_k, z_{j_t})^\top (\mathbf{w}_{k+1} - \mathbf{w}_k) + \frac{\beta}{2} \|\mathbf{w}_{k+1} - \mathbf{w}_k\|^2.$$

Considering SGD update $\mathbf{w}_{k+1} = \mathbf{w}_k - \alpha_k \nabla \ell(\mathbf{w}_k, z_{j_k})$, where $z_{j_k} \stackrel{\text{iid}}{\sim} \mathcal{D}$, and summing both sides from 1 to $t-1$ we get

$$\ell(\mathbf{w}_t, z_{j_t}) - \ell(\mathbf{w}_1, z_{j_t}) \leq - \sum_{k=1}^{t-1} \alpha_k \nabla \ell(\mathbf{w}_k, z_{j_t})^\top \nabla \ell(\mathbf{w}_k, z_{j_k}) + \frac{\beta}{2} \sum_{k=1}^{t-1} \alpha_k^2 \|\nabla \ell(\mathbf{w}_k, z_{j_k})\|^2.$$

Taking expectation w.r.t. S and z on both sides, using the fact that $\mathbf{w}_{S,k}$ does not depend on z_{j_k} nor on z_{j_t} , and that $z_{j_k}, z_{j_t} \stackrel{\text{iid}}{\sim} \mathcal{D}$, we have that

$$\begin{aligned} & \mathbb{E}_{S,z}[\ell(\mathbf{w}_{S,t}, z_{j_t}) - \ell(\mathbf{w}_1, z_{j_t})] \\ & \leq - \sum_{k=1}^{t-1} \alpha_k \mathbb{E}_{S,z}[\nabla \ell(\mathbf{w}_{S,k}, z_{j_t})^\top \nabla \ell(\mathbf{w}_{S,k}, z_{j_k})] + \frac{\beta}{2} \sum_{k=1}^{t-1} \alpha_k^2 \mathbb{E}_{S,z}[\|\nabla \ell(\mathbf{w}_{S,k}, z_{j_k})\|^2] \\ & = \alpha_k \left(\frac{\alpha_k \beta}{2} - 1 \right) \sum_{k=1}^{t-1} \alpha_k^2 \mathbb{E}_{S,z}[\|\nabla \ell(\mathbf{w}_{S,k}, z_{j_k})\|^2]. \end{aligned}$$

□

The following lemma is a consequence of Lemma 12 and self-boundedness.

Lemma 13. *Assume that the loss function ℓ is β -smooth and non-negative, and that step sizes obey $\alpha_t \leq \frac{2}{\beta}$. Then $\forall t \in [T]$ we have that*

$$\mathbb{E}_{S,z}[\|\nabla \ell(\mathbf{w}_{S,t}, z_{j_t})\|] \leq \sqrt{2\beta R(\mathbf{w}_1)}.$$

Proof. By Lemma 11, $\|\nabla \ell(\mathbf{w}_{S,t}, z_{j_t})\| \leq \sqrt{2\beta \ell(\mathbf{w}_{S,t}, z_{j_t})}$. Now, we invoke Lemma 12 assuming that the

step size is set such that $\alpha_t \leq \frac{2}{\beta}$ to get that

$$\begin{aligned} \mathbb{E}_{S,z} [\|\nabla \ell(\mathbf{w}_{S,t}, z_{j_t})\|] &\leq \sqrt{2\beta} \mathbb{E}_S \left[\sqrt{\ell(\mathbf{w}_{S,t}, z_{j_t})} \right] \\ &\leq \sqrt{2\beta} \mathbb{E}_S [\ell(\mathbf{w}_{S,t}, z_{j_t})] && \text{(By Jensen's inequality)} \\ &\leq \sqrt{2\beta} \mathbb{E}_S [\ell(\mathbf{w}_1, z_{j_t})] = \sqrt{2\beta R(\mathbf{w}_1)}. && \text{(By Lemma 12.)} \end{aligned}$$

□

The following lemma is similar to Lemma 3.11 of [57], and is instrumental in bounding the stability of SGD. However, we make an adjustment and state it in expectation over the data. Note that it does not require convexity of the loss function.

Lemma 14. *Assume that the loss function $\ell(\cdot, z) \in [0, 1]$ is L -Lipschitz for all z . Then, for every $t_0 \in \{0, 1, 2, \dots, m\}$ we have that*

$$\mathbb{E}_{S,z} \mathbb{E}_A [\ell(\mathbf{w}_{S,T}, z) - \ell(\mathbf{w}_{S^{(i)},T}, z)] \tag{C.1}$$

$$\leq L \mathbb{E}_{S,z} \left[\mathbb{E}_A [\delta_T(S, z) \mid \delta_{t_0}(S, z) = 0] \right] + \mathbb{E}_{S,A} [R(A_S)] \frac{t_0}{m}. \tag{C.2}$$

Proof. We proceed with elementary decomposition, Lipschitzness of ℓ , and using the fact that ℓ is non-negative to have that

$$\ell(\mathbf{w}_{S,T}, z) - \ell(\mathbf{w}_{S^{(i)},T}, z) \tag{C.3}$$

$$\begin{aligned} &= (\ell(\mathbf{w}_{S,T}, z) - \ell(\mathbf{w}_{S^{(i)},T}, z)) \mathbb{1}\{\delta_{t_0}(S, z) = 0\} \\ &+ (\ell(\mathbf{w}_{S,T}, z) - \ell(\mathbf{w}_{S^{(i)},T}, z)) \mathbb{1}\{\delta_{t_0}(S, z) \neq 0\} \\ &\leq L \delta_T(S, z) \mathbb{1}\{\delta_{t_0}(S, z) = 0\} + \ell(\mathbf{w}_{S,T}, z) \mathbb{1}\{\delta_{t_0}(S, z) \neq 0\}. \end{aligned} \tag{C.4}$$

Taking expectation w.r.t. algorithm randomization, we get that

$$\mathbb{E}_A [\ell(\mathbf{w}_{S,T}, z) - \ell(\mathbf{w}_{S^{(i)},T}, z)] \tag{C.5}$$

$$\leq L \mathbb{E}_A [\delta_T(S, z) \mathbb{1}\{\delta_{t_0}(S, z) = 0\}] + \mathbb{E}_A [\ell(\mathbf{w}_{S,T}, z) \mathbb{1}\{\delta_{t_0}(S, z) \neq 0\}]. \tag{C.6}$$

Recall that $i \in [m]$ is the index where S and $S^{(i)}$ differ, and introduce a random variable τ_A taking on the index of the first time step where SGD uses the example z_i or a replacement z . Note also that τ_A does not depend on the data. When $\tau_A > t_0$, then it must be that $\delta_{t_0}(S, z) = 0$, because updates on both S and $S^{(i)}$ are identical until t_0 . A consequence of this is that $\mathbb{1}\{\delta_{t_0}(S, z) \neq 0\} \leq \mathbb{1}\{\tau_A \leq t_0\}$. Thus the rightmost term in (C.6) is bounded as

$$\mathbb{E}_A [\ell(\mathbf{w}_{S,T}, z) \mathbb{1}\{\delta_{t_0}(S, z) \neq 0\}] \leq \mathbb{E}_A [\ell(\mathbf{w}_{S,T}, z) \mathbb{1}\{\tau_A \leq t_0\}].$$

Now, focus on the r.h.s. above. Recall that we assume randomization by sampling from the uniform distribution over $[m]$ without replacement, and denote a realization by $\{j_i\}_{i=1}^m$. Then, we can always express our randomization as a permutation function $\pi_A(S) = \{z_{j_i}\}_{i=1}^m$. In addition, introduce an algo-

Appendix C. Proofs from Chapter 5

rithm GD: $\mathcal{X}^m \mapsto \mathcal{H}$, which is identical to A , except that it passes over the training set S sequentially without randomization. That said, we have that

$$\mathbb{E}_A [\ell(\mathbf{w}_{S,T}, z) \mathbb{1}\{\tau_A \leq t_0\}] = \mathbb{E}_A [\ell(\text{GD}_{\pi_A(S)}, z) \mathbb{1}\{\tau_A \leq t_0\}],$$

and taking expectation over the data,

$$\mathbb{E}_{S,z} \left[\mathbb{E}_A [\ell(\mathbf{w}_{S,T}, z) \mathbb{1}\{\tau_A \leq t_0\}] \right] = \mathbb{E}_A \left[\mathbb{E}_{S,z} [\ell(\text{GD}_{\pi_A(S)}, z) \mathbb{1}\{\tau_A \leq t_0\}] \right].$$

Now observe that for any realization of A , $\mathbb{E}_{S,z} [\ell(\text{GD}_{\pi_A(S)}, z)] = \mathbb{E}_A \mathbb{E}_{S,z} [\ell(A_S, z)]$ because expectation w.r.t. S and z does not change under our randomization¹. Thus, we have that

$$\mathbb{E}_A \left[\mathbb{E}_{S,z} [\ell(\text{GD}_{\pi_A(S)}, z) \mathbb{1}\{\tau_A \leq t_0\}] \right] = \mathbb{E}_{S,A} [R(A_S)] \mathbb{P}(\tau_A \leq t_0).$$

Now assuming that τ_A is uniformly distributed over $[m]$ we have that

$$\mathbb{P}(\tau_A \leq t_0) = \frac{t_0}{m}.$$

Putting this together with (C.3) and (C.4), we finally get that

$$\begin{aligned} & \mathbb{E}_{S,z} \mathbb{E}_A [\ell(\mathbf{w}_{S,T}, z) - \ell(\mathbf{w}_{S^{(i)},T}, z)] \\ & \leq L \mathbb{E}_{S,z} \left[\mathbb{E}_A [\delta_T(S, z) \mathbb{1}\{\delta_{t_0}(S, z) = 0\}] \right] + \mathbb{E}_{S,A} [R(A_S)] \frac{t_0}{m} \\ & \leq L \mathbb{E}_{S,z} \left[\mathbb{E}_A [\delta_T(S, z) \mid \delta_{t_0}(S, z) = 0] \right] + \mathbb{E}_{S,A} [R(A_S)] \frac{t_0}{m}. \end{aligned}$$

This completes the proof. \square

We spend a moment to highlight the role of conditional expectation in (C.2). Observe that we could naively bound (C.1) by the Lipschitzness of ℓ , but Lemma 14 follows a more careful argument. First note that t_0 is a free parameter. The expected distance in (C.2) between SGD outputs $\mathbf{w}_{S,t}$ and $\mathbf{w}_{S^{(i)},t}$ is conditioned on the fact that at step t_0 the outputs of SGD are still the same. This means that the perturbed point is encountered after t_0 . Then, the conditional expectation should be a decreasing function of t_0 : the later the perturbation occurs, the smaller deviation between $\mathbf{w}_{S,t}$ and $\mathbf{w}_{S^{(i)},t}$ we should expect. Later we use this fact to minimize the bound (C.2) over t_0 .

C.2 Convex Losses

In this section we prove on-average stability for loss functions that are non-negative, β -smooth, and convex.

Theorem 23. *Assume that ℓ is convex, and that SGD is ran with step sizes $\{\alpha_t\}_{t=1}^T$. Then, for every*

¹Strictly speaking we could omit $\mathbb{E}_A[\cdot]$ and consider *any* randomization by reshuffling, but we keep expectation for the sake of clarity.

$t_0 \in \{0, 1, 2, \dots, m\}$, SGD is $\epsilon(\mathcal{D}, \mathbf{w}_1)$ -on-average stable with

$$\epsilon(\mathcal{D}, \mathbf{w}_1) \leq \frac{2}{m} \sum_{t=t_0+1}^T \alpha_t \mathbb{E}_{S,z} [\|\nabla \ell(\mathbf{w}_t, z_{j_t})\|] + \mathbb{E}_{S,A} [R(A_S)] \frac{t_0}{m}.$$

Proof. For brevity denote $\Delta_t(S, z) := \mathbb{E}_A [\delta_t(S, z) \mid \delta_{t_0}(S, z) = 0]$. We start by applying Lemma 14:

$$\mathbb{E}_{S,z,A} [\ell(\mathbf{w}_{S,T}, z) - \ell(\mathbf{w}_{S^{(i)},T}, z)] \leq L \mathbb{E}_{S,z} [\Delta_T(S, z)] + \mathbb{E}_{S,A} [R(A_S)] \frac{t_0}{m}. \quad (\text{C.7})$$

Our goal is to bound the first term on the r.h.s. as a decreasing function of t_0 , so that eventually we can minimize the bound w.r.t. t_0 . At this point we focus on the first term, and the proof partially follows the outline of the proof of Theorem 3.7 in [57]. The strategy will be to establish the bound on $\Delta_T(S, z)$ by using a recursive argument. In fact we will state the bound on $\Delta_{t+1}(S, z)$ in terms of $\Delta_t(S, z)$ and then unravel the recursion. Finally, we will take the expectation w.r.t. the data after we obtain the bound by recursion.

To do so, we distinguish two cases: 1) SGD encounters a perturbed point at step t , that is $t = i$, and 2) the current point is the same in S and $S^{(i)}$, so $t \neq i$. For the first case, we will use the data-dependent boundedness of the gradient update rule, Corollary 5, that is

$$\|G_t(\mathbf{w}_{S,t}) - G_t(\mathbf{w}_{S^{(i)},t})\| \leq \delta_t(S, z) + 2\alpha_t \|\nabla \ell(\mathbf{w}_{S,t}, z_{j_t})\|.$$

To handle the second case, we will use the expansiveness of the gradient update rule, Lemma 10, which states that for convex loss functions, the gradient update rule is 1-expansive, so $\delta_{t+1}(S, z) \leq \delta_t(S, z)$. Considering both cases of example selection, and noting that SGD encounters the perturbation with probability $\frac{1}{m}$, we write \mathbb{E}_A for a step t as

$$\begin{aligned} \Delta_{t+1}(S, z) &\leq \left(1 - \frac{1}{m}\right) \Delta_t(S, z) + \frac{1}{m} (\Delta_t(S, z) + 2\alpha_t \|\nabla \ell(\mathbf{w}_{S,t}, z_{j_t})\|) \\ &= \Delta_t(S, z) + \frac{2\alpha_t \|\nabla \ell(\mathbf{w}_{S,t}, z_{j_t})\|}{m}. \end{aligned}$$

Unraveling the recursion from T to t_0 and plugging the above into (C.7) yields

$$\mathbb{E}_{A,S,z} [\delta_T(S, z)] \leq \frac{2}{m} \sum_{t=t_0+1}^T \alpha_t \mathbb{E}_{S,z} [\|\nabla \ell(\mathbf{w}_t, z_{j_t})\|] + \mathbb{E}_{S,A} [R(A_S)] \frac{t_0}{m}.$$

This completes the proof. □

The next corollary is a simple consequence of Theorem 23 and Lemma 13.

Appendix C. Proofs from Chapter 5

Proof of Theorem 13. Consider Theorem 23 and set $t_0 = 0$. Then we have that

$$\begin{aligned} \epsilon(\mathcal{D}, \mathbf{w}_1) &\leq \frac{2}{m} \sum_{t=1}^T \alpha_t \|\nabla \ell(\mathbf{w}_t, z_{j_t})\| \\ &\leq \frac{2\sqrt{2\beta R(\mathbf{w}_1)}}{m} \sum_{t=1}^T \alpha_t, \end{aligned}$$

where the last inequality comes from Lemma 13 assuming that $\alpha_t \leq \frac{2}{\beta}$. \square

Proof of Theorem 14. For brevity denote $r = \mathbb{E}_{S,A}[R(A_S)]$. Consider Theorem 23 and assume that the step size obeys $\alpha_t = \frac{c}{t} \leq \frac{2}{\beta}$. We have

$$\begin{aligned} \epsilon(\mathcal{D}, \mathbf{w}_1) &\leq \frac{2c}{m} \sum_{t=t_0+1}^T \frac{\mathbb{E}_{S,z}[\|\nabla \ell(\mathbf{w}_t, z_{j_t})\|]}{t} + r \frac{t_0}{m} \\ &\leq \frac{2c\sqrt{2\beta R(\mathbf{w}_1)}}{m} \ln\left(\frac{T}{t_0}\right) + r \frac{t_0}{m} \\ &\leq \frac{2c\sqrt{2\beta R(\mathbf{w}_1)}}{m} \frac{T}{t_0} + r \frac{t_0}{m}, \end{aligned} \tag{C.8}$$

where in (C.8) we used Lemma 13 to bound the expectation of the norm and bounded the sum of the step sizes by the logarithm. Now, setting $t_0 = \sqrt{\frac{2\sqrt{2\beta R(\mathbf{w}_1)}cT}{r}}$ minimizes the bound above, and plugging it back we get that

$$\epsilon(\mathcal{D}, \mathbf{w}_1) \leq \frac{2\sqrt{2\sqrt{2\beta R(\mathbf{w}_1)}crT}}{m}.$$

By Theorem 12 we then have that

$$r - \mathbb{E}_{S,A}[\widehat{R}_S(A_S)] \leq \frac{4\sqrt[4]{\beta R(\mathbf{w}_1)} \cdot \sqrt{crT}}{m} \tag{C.9}$$

Now using a simple fact that for any non-negative A, B, C ,

$$A \leq B + C\sqrt{A} \Rightarrow A \leq B + C^2 + \sqrt{BC},$$

we get from (C.9) that

$$r - \mathbb{E}_{S,A}[\widehat{R}_S(A_S)] \leq \frac{4\sqrt[4]{\beta R(\mathbf{w}_1)}\sqrt{crT}}{m} \sqrt{\mathbb{E}_{S,A}[\widehat{R}_S(A_S)]} + \frac{16\sqrt{\beta R(\mathbf{w}_1)}cT}{m^2}.$$

This completes the proof. \square

C.3 Non-convex Losses

Our proof of a stability bound for non-convex loss functions, Theorem 15, follows the general outline of [57, Theorem 3.8]. Namely, the outputs of SGD run on a training set S and its perturbed version $S^{(i)}$ will not differ too much, because by the time a perturbation is encountered, the step size has already decayed enough. So, on one hand, stabilization is enforced by diminishing the step size, and on the other hand, by how much updates expand the distance between the gradients after the perturbation. Since [57] work with uniform stability, they capture the expansiveness of post-perturbation update by the Lipschitzness of the gradient. In combination with a recursive argument, their bound has exponential dependency on the Lipschitz constant of the gradient. We argue that the Lipschitz continuity of the gradient can be too pessimistic in general. Instead, we rely on a local data-driven argument: considering that we initialize SGD at point \mathbf{w}_1 , how much the updates expand the gradient under the distribution of interest? The following crucial lemma characterizes such behavior in terms of the curvature at \mathbf{w}_1 .

Lemma 15. *Assume that the loss function $\ell(\cdot, z)$ is β -smooth and that its Hessian is ρ -Lipschitz. Then,*

$$\|G_t(\mathbf{w}_{S,t}) - G_t(\mathbf{w}_{S^{(i)},t})\| \leq (1 + \alpha_t \xi_t(S, z)) \delta_t(S, z)$$

where

$$\xi_t(S, z) := \|\nabla^2 \ell(\mathbf{w}_1, z_t)\|_2 + \rho \sqrt{\frac{\beta}{2}} \sum_{k=1}^T \alpha_k \left(\sqrt{\ell(\mathbf{w}_{S,k}, z_{j_k})} + \sqrt{\ell(\mathbf{w}_{S^{(i)},k}, z'_{j_k})} \right).$$

Furthermore, for any $t \in [T]$,

$$\mathbb{E}_{S,z} [\xi_t(S, z)] \leq \mathbb{E}_z [\|\nabla^2 \ell(\mathbf{w}_1, z)\|_2] + c\rho(1 + \ln(T))\sqrt{2\beta R(\mathbf{w}_1)}.$$

Proof. Recall that the randomness of the algorithm is realized through sampling without replacement from the uniform distribution over $[m]$. Apart from that we will not be concerned with the randomness of the algorithm, and given the set of random variables $\{j_i\}_{i=1}^m$, for brevity we will use indexing notation z_1, z_2, \dots, z_m to indicate $z_{j_1}, z_{j_2}, \dots, z_{j_m}$. Next, let $S^{(i)} = \{z'_i\}_{i=1}^m$, and introduce a shorthand notation $f_k(\mathbf{w}) = \ell(\mathbf{w}, z_k)$ and $f_{k'}(\mathbf{w}) = \ell(\mathbf{w}, z'_k)$. We start by applying the triangle inequality to get

$$\|G_t(\mathbf{w}_{S,t}) - G_t(\mathbf{w}_{S^{(i)},t})\| \leq \|\mathbf{w}_{S,t} - \mathbf{w}_{S^{(i)},t}\| + \alpha_t \|\nabla f_t(\mathbf{w}_{S,t}) - \nabla f_t(\mathbf{w}_{S^{(i)},t})\|.$$

In the following we will focus on the second term of the r.h.s. above. Given SGD outputs $\mathbf{w}_{S,t}$ and $\mathbf{w}_{S^{(i)},t}$ with $t > i$, our goal here is to establish how much the gradients grow apart with every new update. This behavior can be characterized assuming that the gradient is Lipschitz continuous, however, we conduct a local analysis. Specifically, we observe how much the updates expand the gradients, given that we start at some point \mathbf{w}_1 under the data-generating distribution. So, instead of the Lipschitz constant, expansiveness rather depends on the curvature around \mathbf{w}_1 . On the other hand, we are dealing with outputs at an arbitrary time step t , and therefore we first have to relate them to the initialization point \mathbf{w}_1 . We do so by using the gradient update rule and telescopic sums, and conclude that this relationship is controlled by the sum of the gradient norms along the update path. We further establish that this sum is controlled by the risk of \mathbf{w}_1 , through the self-bounding property of the loss

function and Lemma 12. Thus, the proof consists of two parts: 1) Decomposition into curvature and gradients along the update path, and 2) bounding those gradients.

1) Decomposition. Introduce $\boldsymbol{\delta}_t := \mathbf{w}_{S^{(i)},t} - \mathbf{w}_{S,t}$. By Taylor's theorem we get that

$$\nabla f_t(\mathbf{w}_{S,t}) - \nabla f_t(\mathbf{w}_{S^{(i)},t}) = \int_0^1 \left(\nabla^2 f_t(\mathbf{w}_{S,t} + \tau \boldsymbol{\delta}_t) - \nabla^2 f_t(\mathbf{w}_1) \right) d\tau \boldsymbol{\delta}_t + \nabla^2 f_t(\mathbf{w}_1) \boldsymbol{\delta}_t.$$

Taking the norm on both sides, applying the triangle inequality, Cauchy-Schwartz inequality, and assuming that Hessians are ρ -Lipschitz we obtain

$$\|\nabla f_t(\mathbf{w}_{S,t}) - \nabla f_t(\mathbf{w}_{S^{(i)},t})\| \tag{C.10}$$

$$\leq \int_0^1 \|\nabla^2 f_t(\mathbf{w}_{S,t} + \tau \boldsymbol{\delta}_t) - \nabla^2 f_t(\mathbf{w}_1)\| d\tau \|\boldsymbol{\delta}_t\| + \|\nabla^2 f_t(\mathbf{w}_1)\| \|\boldsymbol{\delta}_t\|$$

$$\leq \rho \int_0^1 \|\mathbf{w}_{S,t} - \mathbf{w}_1 + \tau \boldsymbol{\delta}_t\| d\tau \|\boldsymbol{\delta}_t\| + \|\nabla^2 f_t(\mathbf{w}_1)\| \|\boldsymbol{\delta}_t\|. \tag{C.11}$$

2) Bounding gradients. Using telescoping sums and the SGD update rule we get that

$$\begin{aligned} & \mathbf{w}_{S,t} - \mathbf{w}_1 + \tau \boldsymbol{\delta}_t \\ &= \mathbf{w}_{S,t} - \mathbf{w}_1 + \tau (\mathbf{w}_{S^{(i)},t} - \mathbf{w}_1 + \mathbf{w}_1 - \mathbf{w}_{S,t}) \\ &= \sum_{k=1}^{t-1} (\mathbf{w}_{S,k+1} - \mathbf{w}_{S,k}) + \tau \sum_{k=1}^{t-1} (\mathbf{w}_{S^{(i)},k+1} - \mathbf{w}_{S^{(i)},k}) - \tau \sum_{k=1}^{t-1} (\mathbf{w}_{S,k+1} - \mathbf{w}_{S,k}) \\ &= (\tau - 1) \sum_{k=1}^{t-1} \alpha_k \nabla f_k(\mathbf{w}_{S,k}) - \tau \sum_{k=1}^{t-1} \alpha_k \nabla f_{k'}(\mathbf{w}_{S^{(i)},k}). \end{aligned}$$

Plugging above into the integral of (C.11) we have

$$\begin{aligned} & \int_0^1 \left\| \sum_{k=1}^{t-1} \alpha_k ((\tau - 1) \nabla f_k(\mathbf{w}_{S,k}) - \tau \nabla f_{k'}(\mathbf{w}_{S^{(i)},k}) \right\| d\tau \\ & \leq \frac{1}{2} \left\| \sum_{k=1}^{t-1} \alpha_k \nabla f_k(\mathbf{w}_{S,k}) \right\| + \frac{1}{2} \left\| \sum_{k=1}^{t-1} \alpha_k \nabla f_{k'}(\mathbf{w}_{S^{(i)},k}) \right\| \\ & \leq \sqrt{\frac{\beta}{2}} \sum_{k=1}^{t-1} \alpha_k \left(\sqrt{f_k(\mathbf{w}_{S,k})} + \sqrt{f_{k'}(\mathbf{w}_{S^{(i)},k})} \right), \end{aligned}$$

where the last inequality comes from the self-bounding property of β -smooth functions, Lemma 11. Plugging this result back into (C.11) completes the proof of the first statement.

Bounding $\mathbb{E}_{S,z}[\xi_t(S,z)]$. Now we briefly focus on the expectation of $\xi_t(S,z)$, and relate it to the risk of \mathbf{w}_1 and expectation Hessian. By definition of $\xi_t(S,z)$

$$\mathbb{E}_{S,z}[\xi_t(S,z)] \leq \rho \sqrt{\frac{\beta}{2}} \sum_{k=1}^T \alpha_k \left(\mathbb{E}_{S,z} \left[\sqrt{\ell(\mathbf{w}_{S,k}, z_{j_k})} \right] + \mathbb{E}_{S,z} \left[\sqrt{\ell(\mathbf{w}_{S^{(i)},k}, z'_{j_k})} \right] \right) + \mathbb{E}_{S,z} \left[\|\nabla^2 \ell(\mathbf{w}_1, z_t)\|_2 \right].$$

By Jensen's inequality and applying Lemma 12 assuming that $\alpha_t \leq \frac{2}{\beta}$ we have,

$$\mathbb{E} \left[\sqrt{\ell(\mathbf{w}_{S,k}, z_{j_k})} \right] \leq \sqrt{\mathbb{E}[\ell(\mathbf{w}_{S,k}, z_{j_k})]} \leq \sqrt{R(\mathbf{w}_1)}.$$

We arrive at the same bound for the perturbed term by renaming z'_{j_k} into z_{j_k} , using the fact that $\mathbf{w}_{S^{(i)},k}$ does not depend on z'_{j_k} under the randomization of SGD. Finally putting things together,

$$\mathbb{E}_{S,z}[\xi_t(S,z)] \leq \rho \sqrt{2\beta R(\mathbf{w}_1)} \sum_{k=1}^T \alpha_k + \mathbb{E}_z \left[\|\nabla^2 \ell(\mathbf{w}_1, z)\| \right],$$

and upper bounding $\sum_{k=1}^T \alpha_k \leq c(1 + \ln(T))$ proves the second statement. \square

Next, we need the following statement to prove our stability bound.

Proposition 2 (Bernstein-type inequality). *Let Z be a zero-mean real-valued r.v., such that $|Z| \leq b$ and $\mathbb{E}[Z^2] \leq \sigma^2$. Then for all $|c| \leq \frac{1}{2b}$, we have that $\mathbb{E}[e^{cZ}] \leq e^{c^2\sigma^2}$.*

Proof. The stated inequality is a consequence of a Bernstein-type inequality for moment generating functions, Theorem 2.10 in [16]. Observe that a zero-centered r.v. Z bounded by b satisfies Bernstein's condition, that is

$$|\mathbb{E}[(Z - \mathbb{E}[Z])^q]| \leq \frac{q!}{2} \sigma^2 b^{q-2} \quad \text{for all integers } q \geq 3.$$

This in turn satisfies the condition for Bernstein-type inequality stating that

$$\mathbb{E}[\exp(c(Z - \mathbb{E}[Z]))] \leq \exp\left(\frac{c^2\sigma^2/2}{1 - b|c|}\right).$$

Choosing $|c| \leq \frac{1}{2b}$ verifies the statement. \square

Now we are ready to prove Theorem 15, which bounds the $\epsilon(\mathcal{D}, \mathbf{w}_1)$ -on-average stability of SGD.

Proof of Theorem 15. For brevity denote $r := \mathbb{E}_{S,A}[R(A_S)]$ and

$$\Delta_t(S,z) := \mathbb{E}_A[\delta_t(S,z) \mid \delta_{t_0}(S,z) = 0].$$

By Lemma 14, for all $t_0 \in [m]$,

$$\mathbb{E}_{S,z} \mathbb{E}_A[\ell(\mathbf{w}_{S,T}, z) - \ell(\mathbf{w}_{S^{(i)},T}, z)] \leq L \mathbb{E}_{S,z}[\Delta_T(S,z)] + r \frac{t_0}{m}. \quad (\text{C.12})$$

Appendix C. Proofs from Chapter 5

Most of the proof is dedicated to bounding the first term in (C.12). We deal with this similarly as in [57]. Specifically, we state the bound on $\Delta_T(S, z)$ by using a recursion. In our case, however, we also have an expectation w.r.t. the data, and to avoid complications with dependencies, we first unroll the recursion for the random quantities, and only then take the expectation. At this point the proof crucially relies on the product of exponentials arising from the recursion, and all the relevant random quantities end up inside of them. We alleviate this by Proposition 2. Finally, we conclude by minimizing (C.12) w.r.t. t_0 . Thus we have three steps: 1) recursion, 2) bounding $\mathbb{E}[\exp(\dots)]$, and 3) tuning of t_0 .

1) Recursion. We begin by stating the bound on $\Delta_T(S, z)$ by recursion. Thus we will first state the bound on $\Delta_{t+1}(S, z)$ in terms of $\Delta_t(S, z)$, and other relevant quantities and then unravel the recursion. As in the convex case, we distinguish two cases: 1) SGD encounters the perturbed point at step t , that is $t = i$, and 2) the current point is the same in S and $S^{(i)}$, so $t \neq i$. For the first case, we will use the worst-case boundedness of G_t , Corollary 5, that is, $\|G_t(\mathbf{w}_{S,t}) - G_t(\mathbf{w}_{S^{(i)},t})\| \leq \delta_t(S, z) + 2\alpha_t L$. To handle the second case we will use Lemma 15, namely,

$$\|G_t(\mathbf{w}_{S,t}) - G_t(\mathbf{w}_{S^{(i)},t})\| \leq (1 + \alpha_t \xi_t(S, z)) \delta_t(S, z).$$

In addition, as a safety measure we will also take into account that the gradient update rule is at most $(1 + \alpha_t \beta)$ -expansive by Lemma 10. So we will work with the function $\psi_t(S, z) := \min\{\xi_t(S, z), \beta\}$ instead of $\xi_t(S, z)$ and decompose the expectation w.r.t. A for a step t . Noting that SGD encounters the perturbed example with probability $\frac{1}{m}$,

$$\begin{aligned} \Delta_{t+1}(S, z) &\leq \left(1 - \frac{1}{m}\right) (1 + \alpha_t \psi_t(S, z)) \Delta_t(S, z) + \frac{1}{m} (2\alpha_t L + \Delta_t(S, z)) \\ &= \left(1 + \left(1 - \frac{1}{m}\right) \alpha_t \psi_t(S, z)\right) \Delta_t(S, z) + \frac{2\alpha_t L}{m} \\ &\leq \exp(\alpha_t \psi_t(S, z)) \Delta_t(S, z) + \frac{2\alpha_t L}{m}, \end{aligned} \quad (\text{C.13})$$

where the last inequality follows from $1 + x \leq \exp(x)$. This inequality is not overly loose for $x \in [0, 1]$, and, in our case it becomes instrumental in handling the recursion.

Now, observe that the relation $x_{t+1} \leq a_t x_t + b_t$ with $x_{t_0} = 0$ unwinds from T to t_0 as $x_T \leq \sum_{t=t_0+1}^T b_t \prod_{k=t+1}^T a_k$. Consequently, having $\Delta_{t_0}(S, z) = 0$, we unwind (C.13) to get

$$\begin{aligned} \Delta_T(S, z) &\leq \sum_{t=t_0+1}^T \left(\prod_{k=t+1}^T \exp\left(\frac{c\psi_k(S, z)}{k}\right) \right) \frac{2cL}{mt} \\ &= \sum_{t=t_0+1}^T \exp\left(c \sum_{k=t+1}^T \frac{\psi_k(S, z)}{k}\right) \frac{2cL}{mt}. \end{aligned} \quad (\text{C.14})$$

2) Bounding $\mathbb{E}[\exp(\dots)]$. We take the expectation w.r.t. S and z on both sides and focus on the expectation of the exponential in (C.14). First, introduce $\mu_k := \mathbb{E}_{S,z}[\psi_k(S, z)]$, and proceed as

$$\mathbb{E}_{S,z} \left[\exp\left(c \sum_{k=t+1}^T \frac{\psi_k(S, z)}{k}\right) \right] = \mathbb{E}_{S,z} \left[\exp\left(c \sum_{k=t+1}^T \frac{\psi_k(S, z) - \mu_k}{k}\right) \right] \exp\left(c \sum_{k=t+1}^T \frac{\mu_k}{k}\right). \quad (\text{C.15})$$

Observe that the zero-mean version of $\psi_k(S, z)$ is bounded as

$$\sum_{k=t+1}^T \frac{|\psi_k(S, z) - \mu_k|}{k} \leq 2\beta \ln(T),$$

and assume the setting of c as $c \leq \frac{1}{2(2\beta \ln(T))^2}$. By Proposition 2, we have

$$\begin{aligned} & \mathbb{E} \left[\exp \left(c \sum_{k=t+1}^T \frac{\psi_k(S, z) - \mu_k}{k} \right) \right] \\ & \leq \exp \left(c^2 \mathbb{E} \left[\left(\sum_{k=t+1}^T \frac{\psi_k(S, z) - \mu_k}{k} \right)^2 \right] \right) \\ & = \exp \left(\frac{c}{2} \mathbb{E} \left[\left(\frac{1}{2\beta \ln(T)} \sum_{k=t+1}^T \frac{\psi_k(S, z) - \mu_k}{k} \right)^2 \right] \right) \\ & \leq \exp \left(\frac{c}{2} \mathbb{E} \left[\left| \sum_{k=t+1}^T \frac{\psi_k(S, z) - \mu_k}{k} \right| \right] \right) \\ & \leq \exp \left(\frac{c}{2} \sum_{k=t+1}^T \frac{\mathbb{E}[|\psi_k(S, z) - \mu_k|]}{k} \right) \\ & \leq \exp \left(c \sum_{k=t+1}^T \frac{\mu_k}{k} \right). \end{aligned}$$

Getting back to (C.15) we conclude that

$$\mathbb{E}_{S, z} \left[\exp \left(c \sum_{k=t+1}^T \frac{\psi_k(S, z)}{k} \right) \right] \leq \exp \left(c \sum_{k=t+1}^T \frac{2\mu_k}{k} \right). \quad (\text{C.16})$$

Next, we give an upper-bound on μ_k , that is $\mu_k \leq \min\{\beta, \mathbb{E}_{S, z}[\xi_k(S, z)]\}$. Finally, we bound $\mathbb{E}_{S, z}[\xi_k(S, z)]$ using the second result of Lemma 15, which holds for any $k \in [T]$, to get that $\mu_k \leq \gamma$, with γ defined in (5.3).

3) Tuning of t_0 . Now we turn our attention back to (C.14). Considering that we took an expectation w.r.t. the data, we use (C.16) and the fact that $\mu_k \leq \gamma$ to get that

$$\begin{aligned} \mathbb{E}_{S, z} [\Delta_T(S, z)] & \leq \sum_{t=t_0+1}^T \exp \left(2c\gamma \sum_{k=t+1}^T \frac{1}{k} \right) \frac{2cL}{mt} \\ & \leq \sum_{t=t_0+1}^T \exp \left(2c\gamma \ln \left(\frac{T}{t} \right) \right) \frac{2cL}{mt} \\ & = \frac{2cL}{m} (T^{2c\gamma}) \sum_{t=t_0+1}^T t^{-2c\gamma-1} \\ & \leq \frac{1}{2c\gamma} \frac{2cL}{m} \left(\frac{T}{t_0} \right)^{2c\gamma}. \end{aligned}$$

Appendix C. Proofs from Chapter 5

Plug the above into (C.12) to get

$$\mathbb{E}_{S,z} \mathbb{E}_A [\ell(\mathbf{w}_{S,T}, z) - \ell(\mathbf{w}_{S^{(t)},T}, z)] \leq \frac{L^2}{\gamma m} \left(\frac{T}{t_0}\right)^{2c\gamma} + r \frac{t_0}{m}. \quad (\text{C.17})$$

Let $q = 2c\gamma$. Then, setting

$$t_0 = \left(\frac{2cL^2}{r}\right)^{\frac{1}{1+q}} T^{\frac{q}{1+q}}$$

minimizes (C.17). Plugging t_0 back we get that (C.17) equals to

$$\frac{1 + \frac{1}{q}}{m} (2cL^2)^{\frac{1}{1+q}} (rT)^{\frac{q}{1+q}}.$$

This completes the proof. \square

This theorem implies the following result that is further controlled by the initialization point.

Proof of Corollary 2. Consider the statement of Theorem 15. Assuming that the step size $\alpha_t \leq \frac{2}{\beta}$, Lemma 12 implies that $\mathbb{E}_{S,A}[R(A_S)] \leq R(\mathbf{w}_1)$, which completes the proof. \square

Optimistic Rates for Learning with Non-convex Loss Functions

Next we will prove an optimistic bound based on Theorem 15, in other words, the bound that demonstrates fast convergence rate subject to the vanishing empirical risk. First we will need the following technical statement.

Lemma 16. [29, Lemma 7.2] *Let $c_1, c_2, \dots, c_l > 0$ and $s > q_1 > q_2 > \dots > q_{l-1} > 0$. Then the equation*

$$x^s - c_1 x^{q_1} - c_2 x^{q_2} - \dots - c_{l-1} x^{q_{l-1}} - c_l = 0$$

has a unique positive solution x^ . In addition,*

$$x^* \leq \max \left\{ (lc_1)^{\frac{1}{s-q_1}}, (lc_2)^{\frac{1}{s-q_2}}, \dots, (lc_{l-1})^{\frac{1}{s-q_{l-1}}}, (lc_l)^{\frac{1}{s}} \right\}.$$

Next we prove a useful technical lemma similarly as in [102, Lemma 7].

Lemma 17. *Let $a, c > 0$ and $0 < \alpha < 1$. Then the inequality*

$$x - ax^\alpha - c \leq 0$$

implies

$$x \leq \max \left\{ 2^{\frac{\alpha}{1-\alpha}} a^{\frac{1}{1-\alpha}}, (2c)^\alpha a \right\} + c.$$

Proof. Consider a function $h(x) = x - ax^\alpha - c$. Applying Lemma 16 with $s = 1$, $l = 2$, $c_1 = a$, $c_2 = c$,

and $q_1 = a$ we get that $h(x) = 0$ has a unique positive solution x^* and

$$x^* \leq \max \left\{ (2a)^{\frac{1}{1-\alpha}}, 2c \right\}. \quad (\text{C.18})$$

Moreover, the inequality $h(x) \leq 0$ is verified for $x = 0$, and $\lim_{x \rightarrow +\infty} h(x) = +\infty$, so we have that $h(x) \leq 0$ implies $x \leq x^*$. Now, using this fact and the fact that $h(x^*) = 0$, we have that

$$x \leq x^* = a(x^*)^\alpha + c,$$

and upper-bounding x^* by (C.18) we finally have

$$x \leq a \max \left\{ (2a)^{\frac{\alpha}{1-\alpha}}, (2c)^\alpha \right\} + c,$$

which completes the proof. \square

Proof of Corollary 3. Consider Theorem 15 and observe that it verifies the condition of Lemma 17 with $x = \mathbb{E}_{S,A}[R(A_S)]$, $c = \mathbb{E}_{S,A}[\widehat{R}_S(A_S)]$, $\alpha = \frac{c\gamma}{1+c\gamma}$, and

$$a = \frac{1 + \frac{1}{c\gamma}}{m} (2cL^2)^{\frac{1}{1+c\gamma}} T^{\frac{c\gamma}{1+c\gamma}}.$$

Note that $\alpha/(1-\alpha) = c\gamma$ and $1/(1-\alpha) = 1+c\gamma$. Then, we obtain that

$$\begin{aligned} \mathbb{E}_{S,A} [R(A_S) - \widehat{R}_S(A_S)] &\leq \max \left\{ 2^{c\gamma} \left(\frac{1 + \frac{1}{c\gamma}}{m} \right)^{1+c\gamma} (2cL^2)^{c\gamma} T^{c\gamma}, \right. \\ &\quad \left. \left(2 \mathbb{E}_{S,A} [\widehat{R}_S(A_S)] \right)^{\frac{c\gamma}{1+c\gamma}} \left(\frac{1 + \frac{1}{c\gamma}}{m} (2cL^2)^{\frac{1}{1+c\gamma}} T^{\frac{c\gamma}{1+c\gamma}} \right) \right\} \\ &= \max \left\{ \left(2 + \frac{2}{c\gamma} \right)^{1+c\gamma} (cL^2)^{c\gamma} \left(\frac{T^{c\gamma}}{m^{1+c\gamma}} \right), \right. \\ &\quad \left. \frac{1 + \frac{1}{c\gamma}}{m} (2cL^2)^{\frac{1}{1+c\gamma}} \left(2 \mathbb{E}_{S,A} [\widehat{R}_S(A_S)] \cdot T \right)^{\frac{c\gamma}{1+c\gamma}} \right\}. \end{aligned}$$

This completes the proof. \square

Proof of Proposition 1. Consider minimizing (5.4) over a discrete set of source hypotheses $\{\mathbf{w}_k^{\text{src}}\}_{k=1}^K$,

$$\min_{k \in [K]} \epsilon(\mathcal{D}, \mathbf{w}_k^{\text{src}}) \leq \min_{k \in [K]} \mathcal{O} \left(\frac{1 + \frac{1}{c\gamma_k}}{m} (R(\mathbf{w}_k^{\text{src}}) \cdot T)^{\frac{c\gamma_k}{1+c\gamma_k}} \right), \quad (\text{C.19})$$

and recall that

$$\gamma_k = \mathbb{E}_{z \sim \mathcal{D}} [\|\nabla^2 \ell(\mathbf{w}_k^{\text{src}}, z)\|_2] + \lambda \sqrt{R(\mathbf{w}_k^{\text{src}})},$$

Appendix C. Proofs from Chapter 5

such that $\lambda = c\rho(1 + \ln(T))\sqrt{2\beta}$. Let

$$\tilde{\gamma}_k = \frac{1}{m} \sum_{i=1}^m \|\nabla^2 \ell(\mathbf{w}_k^{\text{src}}, z_i)\|_2 + \lambda \sqrt{\widehat{R}_S(\mathbf{w}_k^{\text{src}})}.$$

By Hoeffding's inequality, with high probability, we have that $|\gamma_k - \tilde{\gamma}_k| \leq \mathcal{O}\left(\frac{1}{\sqrt[4]{m}}\right)$. Now we further upper bound (C.19) by upper bounding $R(\mathbf{w}_k^{\text{src}})$ and applying the union bound to get

$$\begin{aligned} \min_{k \in [K]} \epsilon(\mathcal{D}, \mathbf{w}_k^{\text{src}}) &\leq \min_{k \in [K]} \mathcal{O} \left(\left(1 + \frac{1}{c\tilde{\gamma}_k}\right) \left(\widehat{R}_S(\mathbf{w}_k^{\text{src}}) + \sqrt{\frac{\log(K)}{m}} \right)^{\frac{c\tilde{\gamma}_k^+}{1+c\tilde{\gamma}_k^+}} m^{-\frac{1}{1+c\tilde{\gamma}_k^+}} \right) \\ &\leq \min_{k \in [K]} \mathcal{O} \left(\left(1 + \frac{1}{c\tilde{\gamma}_k}\right) \widehat{R}_S(\mathbf{w}_k^{\text{src}})^{\frac{c\tilde{\gamma}_k^+}{1+c\tilde{\gamma}_k^+}} \cdot \frac{\sqrt{\log(K)}}{m^{\frac{1}{1+c\tilde{\gamma}_k^+}}} \right), \end{aligned}$$

which concludes the proof. □

D Proofs from Chapter 6

For brevity, we define $\mathbf{h}^{\text{src}}(\mathbf{x}) := [h_1^{\text{src}}(\mathbf{x}), \dots, h_n^{\text{src}}(\mathbf{x})]^\top$, and we will consider a truncated target predictor

$$h_{\mathbf{w}, \boldsymbol{\beta}}^{\text{trg}}(\mathbf{x}) := \mathsf{T}(\mathbf{w}^\top \mathbf{x} + \boldsymbol{\beta}^\top \mathbf{h}^{\text{src}}(\mathbf{x})),$$

with $\mathsf{T}(a) := \min\{\max\{a, -1\}, 1\}$. That said, we will assume that

$$\widehat{R}(h_{\mathbf{w}, \boldsymbol{\beta}}^{\text{trg}}) \leq \frac{1}{m} \sum_{i=1}^m (\mathbf{w}^\top \mathbf{x}_i + \boldsymbol{\beta}^\top \mathbf{h}^{\text{src}}(\mathbf{x}_i) - y_i)^2,$$

in other words, the empirical risk of a truncated predictor cannot be greater, since all the labels belong to $\{-1, 1\}$.

To prove Theorem 17 we need the following supplementary lemmas.

Lemma 18. *Let GreedyTL generate solution $(\widehat{\mathbf{w}}, \widehat{\boldsymbol{\beta}})$, given the training set (\mathbf{X}, \mathbf{y}) , source hypotheses $\{h_i^{\text{src}}\}_{i=1}^n$, and hyperparameters λ and k . Then we have that,*

$$\lambda \|\widehat{\mathbf{w}}\|^2 + \lambda \|\widehat{\boldsymbol{\beta}}\|^2 + \widehat{R}(h_{\widehat{\mathbf{w}}, \widehat{\boldsymbol{\beta}}}^{\text{trg}}) \leq \min_{|S| \leq k} \left\{ \frac{1}{|S|} \sum_{j \in S} \widehat{R}(h_j^{\text{src}}) + \frac{\lambda}{|S|} \right\},$$

$$\lambda \|\widehat{\mathbf{w}}\|^2 + \lambda \|\widehat{\boldsymbol{\beta}}\|^2 + \widehat{R}(h_{\widehat{\mathbf{w}}, \widehat{\boldsymbol{\beta}}}^{\text{trg}}) \leq \widehat{R}(h_{\mathbf{0}, \widehat{\boldsymbol{\beta}}}^{\text{trg}}).$$

and also,

$$\lambda \|\widehat{\mathbf{w}}\|^2 + \lambda \|\widehat{\boldsymbol{\beta}}\|^2 + \widehat{R}(h_{\widehat{\mathbf{w}}, \widehat{\boldsymbol{\beta}}}^{\text{trg}}) \leq 1.$$

Appendix D. Proofs from Chapter 6

Proof. Define $J(\mathbf{w}, \boldsymbol{\beta}) := \widehat{R}(h_{\mathbf{w}, \boldsymbol{\beta}}^{\text{trg}}) + \lambda \|\mathbf{w}\|^2 + \lambda \|\boldsymbol{\beta}\|^2$. For any $\boldsymbol{\alpha} \in \left\{0, \frac{1}{p}\right\}^n$ such that $\|\boldsymbol{\alpha}\|_0 = p$ we have,

$$\begin{aligned} J(\widehat{\mathbf{w}}, \widehat{\boldsymbol{\beta}}) &\leq J(\mathbf{0}, \boldsymbol{\alpha}) = \frac{1}{m} \sum_{i=1}^m \ell \left(y_i, \frac{1}{p} \sum_{j \in \text{supp}(\boldsymbol{\alpha})} h_j^{\text{src}}(\mathbf{x}_i) \right) + \frac{\lambda}{p} \\ &\leq \frac{1}{p} \sum_{j \in \text{supp}(\boldsymbol{\alpha})} \widehat{R}(h_j^{\text{src}}) + \frac{\lambda}{p}. \end{aligned} \quad (\text{D.1})$$

We have the last inequality due to Jensen's inequality. The fact that (D.3) holds for any $p \in \{1, \dots, k\}$ proves the first statement.

We have the second statement from,

$$\begin{aligned} \widehat{R}(h_{\widehat{\mathbf{w}}, \widehat{\boldsymbol{\beta}}}^{\text{trg}}) + \lambda \|\widehat{\mathbf{w}}\|^2 + \lambda \|\widehat{\boldsymbol{\beta}}\|^2 &\leq \widehat{R}(h_{\mathbf{0}, \widehat{\boldsymbol{\beta}}}^{\text{trg}}) + \lambda \|\widehat{\boldsymbol{\beta}}\|^2 \\ \Rightarrow \widehat{R}(h_{\widehat{\mathbf{w}}, \widehat{\boldsymbol{\beta}}}^{\text{trg}}) &\leq \widehat{R}(h_{\widehat{\mathbf{w}}, \widehat{\boldsymbol{\beta}}}^{\text{trg}}) + \lambda \|\widehat{\mathbf{w}}\|^2 \leq \widehat{R}(h_{\mathbf{0}, \widehat{\boldsymbol{\beta}}}^{\text{trg}}). \end{aligned}$$

The last statement comes from,

$$\lambda \|\widehat{\mathbf{w}}\|^2 + \lambda \|\widehat{\boldsymbol{\beta}}\|^2 \leq J(\mathbf{0}, \mathbf{0}) \leq 1. \quad (\text{D.2})$$

□

Lemma 19. *Let $(\mathbf{w}^*, \boldsymbol{\beta}^*)$ be the optimal solution to (6.3), given the training set (\mathbf{X}, \mathbf{y}) , source hypotheses $\{h_i^{\text{src}}\}_{i=1}^n$, and hyperparameters λ and k . Then, the following holds,*

$$\begin{aligned} &\lambda \|\mathbf{w}^*\|^2 + \lambda \|\boldsymbol{\beta}^*\|^2 + \widehat{R}(h_{\mathbf{w}^*, \boldsymbol{\beta}^*}^{\text{trg}}) \\ &\leq \min_{|S| \leq k} \left\{ \frac{1}{|S|} \sum_{j \in S} \widehat{R}(h_j^{\text{src}}) + \frac{\lambda}{|S|} \right\}. \end{aligned}$$

Proof. Define $J(\mathbf{w}, \boldsymbol{\beta}) := \widehat{R}(h_{\mathbf{w}, \boldsymbol{\beta}}^{\text{trg}}) + \lambda \|\mathbf{w}\|^2 + \lambda \|\boldsymbol{\beta}\|^2$. For any $\boldsymbol{\alpha} \in \left\{0, \frac{1}{p}\right\}^n$ such that $\|\boldsymbol{\alpha}\|_0 = p$ we have,

$$\begin{aligned} J(\mathbf{w}^*, \boldsymbol{\beta}^*) &\leq J(\mathbf{0}, \boldsymbol{\alpha}) = \frac{1}{m} \sum_{i=1}^m \ell \left(y_i, \frac{1}{p} \sum_{j \in \text{supp}(\boldsymbol{\alpha})} h_j^{\text{src}}(\mathbf{x}_i) \right) + \frac{\lambda}{p} \\ &\leq \frac{1}{p} \sum_{j \in \text{supp}(\boldsymbol{\alpha})} \widehat{R}(h_j^{\text{src}}) + \frac{\lambda}{p}. \end{aligned} \quad (\text{D.3})$$

We have the last inequality due to Jensen's inequality. The fact that (D.3) holds for any $p \in \{1, \dots, k\}$ proves the statement.

□

Proof of Theorem 17. To prove the statement we will use the optimistic rate Rademacher complexity bounds of [130]. In particular, we will have to do two things: upper-bound the worst-case Rademacher complexity of the hypothesis class of GreedyTL, and upper-bound the empirical risk of members of that hypothesis class. Before proceeding, we spend a moment to define the loss class of GreedyTL,

assuring that it is consistent with the definition by [130],

$$\mathcal{L} := \left\{ (\mathbf{x}, y) \mapsto \frac{1}{2} (h(\mathbf{x}) - y)^2 : h \in (\mathbb{T} \circ \mathcal{H}), \widehat{R}(h) \leq r \right\}. \quad (\text{D.4})$$

Here, $(\mathbb{T} \circ \mathcal{H})$ is the class of truncated hypotheses, \mathcal{H} is the hypothesis class of GreedyTL and r is the mentioned bound on the empirical risk. We define the hypothesis class as,

$$\mathcal{H} := \left\{ \mathbf{x} \mapsto \mathbf{w}^\top \mathbf{x} + \boldsymbol{\beta}^\top \mathbf{h}^{\text{src}}(\mathbf{x}) : \|\mathbf{w}\|_2^2 + \|\boldsymbol{\beta}\|_2^2 \leq \frac{1}{\lambda} \right\}.$$

In this definition we have used the fact shown in Lemma 18 that is the constraint on $\|\mathbf{w}\|_2^2 + \|\boldsymbol{\beta}\|_2^2$, which translates into a constraint on the hypothesis class. Now we are ready to analyze its complexity.

Recall that the worst case Rademacher complexity is defined as,

$$\mathfrak{R}(\mathcal{F}) := \sup_{\mathbf{x}_1, \dots, \mathbf{x}_m \in \mathcal{X}} \left\{ \mathbb{E} \left[\sup_{f \in \mathcal{F}} \left\{ \frac{1}{m} \sum_{i=1}^m \sigma_i f(\mathbf{x}_i) \right\} \right] \right\},$$

where σ_i is r.v., such that $\mathbb{P}(\sigma_i = 1) = \mathbb{P}(\sigma_i = -1) = \frac{1}{2}$.

Let us focus on the analysis of the empirical Rademacher complexity $\widehat{\mathfrak{R}}(\mathbb{T} \circ \mathcal{H})$, that is the part inside the outer supremum. The truncation $\mathbb{T}(\cdot)$ is 1-Lipschitz, therefore by Talagrand's contraction lemma [99] we have that $\widehat{\mathfrak{R}}(\mathbb{T} \circ \mathcal{H}) \leq \widehat{\mathfrak{R}}(\mathcal{H})$. Hence, now we proceed with an upper-bound on $\widehat{\mathfrak{R}}(\mathcal{H})$. Define

$\boldsymbol{\iota} \in \{0, 1\}^n$ such that $\iota_i := \begin{cases} 1, & i \in \text{supp}(\boldsymbol{\beta}) \\ 0, & \text{otherwise} \end{cases}$. Then we have that

$$\widehat{\mathfrak{R}}(\mathbb{T} \circ \mathcal{H}) \leq \widehat{\mathfrak{R}}(\mathcal{H}) \quad (\text{D.5})$$

$$= \mathbb{E} \left[\sup_{\|\mathbf{w}\|_2^2 + \|\boldsymbol{\beta}\|_2^2 \leq \frac{1}{\lambda}} \frac{1}{m} \sum_{i=1}^m \sigma_i (\mathbf{w}^\top \mathbf{x}_i + \boldsymbol{\beta}^\top \mathbf{h}^{\text{src}}(\mathbf{x}_i)) \right]$$

$$= \frac{1}{m\sqrt{\lambda}} \mathbb{E} \left[\left\| \sum_{i=1}^m \sigma_i \begin{bmatrix} \mathbf{x}_i \\ \boldsymbol{\iota} \circ \mathbf{h}^{\text{src}}(\mathbf{x}_i) \end{bmatrix} \right\| \right] \quad (\text{D.6})$$

$$\leq \sqrt{\frac{1}{m^2 \lambda} \sum_{i=1}^m \|\mathbf{x}_i\|^2 + \|\boldsymbol{\iota} \circ \mathbf{h}^{\text{src}}(\mathbf{x}_i)\|^2} \quad (\text{D.7})$$

$$\leq \sqrt{\frac{1 + k \|\mathbf{h}^{\text{src}}\|_\infty^2}{\lambda m}}. \quad (\text{D.8})$$

To obtain (D.6) we have applied the Cauchy-Schwartz inequality on the inner product of $[\mathbf{w}^\top \ \boldsymbol{\beta}^\top]^\top$ and $[\mathbf{x}_i^\top \ \mathbf{h}^{\text{src}}(\mathbf{x}_i)^\top]^\top$, then upper-bounding norms with constraints given by definition of a class \mathcal{H} . To get (D.7) we have applied Jensen's inequality w.r.t. $\mathbb{E}[\cdot]$, along with the fact that $\mathbb{E}[\sigma_i \sigma_{j \neq i}] = 0$ and $\mathbb{E}[\sigma_i \sigma_i] = 1$. Next, we have bounded the L_2 norms of features and sources, recalling that by assumption, $\|\mathbf{x}_i\|^2 \leq 1$. Finally, taking supremum over (D.8) w.r.t. data, we obtain,

$$\mathfrak{R}(\mathbb{T} \circ \mathcal{H}) \leq \sqrt{\frac{1 + k \|\mathbf{h}^{\text{src}}\|_\infty^2}{\lambda m}}.$$

Appendix D. Proofs from Chapter 6

Next, we upper bound the empirical risk of the members of \mathcal{H} by Lemma 18. By plugging the bound on the $\mathfrak{R}(\mathcal{H})$, and the bound on the empirical risk of (D.4) into Theorem 1 in [130] we have the statement. \square

Next we prove the approximation guarantee of a regularized subset selection, Corollary 4, that is needed for the proof of Theorem 18. First we note that the solution returned by FR enjoys the following guarantees in solving the Subset Selection.

Theorem 24 ([30]). *Assume that \mathbf{C} and \mathbf{b} are normalized, and $C_{i,j \neq i} \leq \gamma < \frac{1}{6k}$ for subset size $k \leq n$. Then, the FR algorithm generates an approximate solution $\hat{\mathbf{w}}$ to the Subset Selection such that, $R(\hat{\mathbf{w}}) \leq (1 + 16(k+1)^2\gamma) \min_{\|\mathbf{w}\|_0=k} R(\mathbf{w})$.*

This theorem is instrumental in stating our corollary.

Proof of Corollary 4. In addition to the sample covariance matrix $\hat{\mathbf{C}}$, define also the correlations $\mathbf{b} := \frac{1}{m} \mathbf{X}^\top \mathbf{y}$. Denote $\hat{\mathbf{C}}' = \frac{\hat{\mathbf{C}} + \lambda \mathbf{I}}{1 + \lambda}$. Now, suppose that $\hat{\mathbf{w}}_S$ is the solution found by the forward regression algorithm, given the input $(\hat{\mathbf{C}}', \hat{\mathbf{b}}, k)$. So, the empirical risk that the algorithm attains is $1 - \hat{\mathbf{b}}_S^\top (\hat{\mathbf{C}}'_S)^{-1} \hat{\mathbf{b}}_S$, as follows from the analytic solution to empirical risk minimization for a given S . In fact, we can upper-bound it right away using Theorem 24. But, recall that our goal is to upper-bound the quantity $\hat{R}(\hat{\mathbf{w}}) + \lambda \|\hat{\mathbf{w}}\|^2 = 1 - \hat{\mathbf{b}}_S^\top (\hat{\mathbf{C}}_S + \lambda \mathbf{I})^{-1} \hat{\mathbf{b}}_S$, that is the regularized empirical risk of the approximation $\hat{\mathbf{w}}_S$ to the regularized subset selection. This quantity is obtained via the unnormalized covariance matrix, therefore we cannot analyze it directly by Theorem 24. For this reason we rewrite it as $\hat{R}(\hat{\mathbf{w}}) + \lambda \|\hat{\mathbf{w}}\|^2 = 1 - \frac{1}{1+\lambda} \hat{\mathbf{b}}_S^\top (\hat{\mathbf{C}}'_S)^{-1} \hat{\mathbf{b}}_S$. From Theorem 24 we then have $(\hat{\mathbf{C}}'_S)_{i,j \neq i} \leq \gamma' \leq \frac{1}{6k}$, denote $\epsilon = 16(k+1)^2\gamma'$, and let S^* be the optimal subset of size k . Now we plug $1 - \hat{\mathbf{b}}_{S^*}^\top (\hat{\mathbf{C}}'_{S^*})^{-1} \hat{\mathbf{b}}_{S^*}$ into Theorem 24, and proceed with algebraic transformations,

$$\begin{aligned} 1 - \hat{\mathbf{b}}_S^\top (\hat{\mathbf{C}}'_S)^{-1} \hat{\mathbf{b}}_S &\leq (1 + \epsilon) (1 - \hat{\mathbf{b}}_{S^*}^\top (\hat{\mathbf{C}}'_{S^*})^{-1} \hat{\mathbf{b}}_{S^*}) \\ &\Rightarrow \frac{1}{1 + \lambda} (1 - \hat{\mathbf{b}}_S^\top (\hat{\mathbf{C}}'_S)^{-1} \hat{\mathbf{b}}_S) \leq \frac{1 + \epsilon}{1 + \lambda} (1 - \hat{\mathbf{b}}_{S^*}^\top (\hat{\mathbf{C}}'_{S^*})^{-1} \hat{\mathbf{b}}_{S^*}) \\ &\Rightarrow 1 - \frac{1}{1 + \lambda} \hat{\mathbf{b}}_S^\top (\hat{\mathbf{C}}'_S)^{-1} \hat{\mathbf{b}}_S \end{aligned} \tag{D.9}$$

$$\begin{aligned} &\leq (1 + \epsilon) \left(\frac{1}{1 + \lambda} - \frac{1}{1 + \lambda} \hat{\mathbf{b}}_{S^*}^\top (\hat{\mathbf{C}}'_{S^*})^{-1} \hat{\mathbf{b}}_{S^*} \right) + \frac{\lambda}{1 + \lambda} \\ &\Rightarrow 1 - \frac{1}{1 + \lambda} \hat{\mathbf{b}}_S^\top (\hat{\mathbf{C}}'_S)^{-1} \hat{\mathbf{b}}_S \end{aligned} \tag{D.10}$$

$$\leq (1 + \epsilon) \left(1 - \frac{1}{1 + \lambda} \hat{\mathbf{b}}_{S^*}^\top (\hat{\mathbf{C}}'_{S^*})^{-1} \hat{\mathbf{b}}_{S^*} \right) - \frac{\epsilon \lambda}{1 + \lambda}.$$

The last step is to relate γ' to γ . The fact $(\hat{\mathbf{C}}'_S)_{i,j \neq i} \leq \gamma' \leq \frac{1}{6k}$ is equivalent to $\frac{(\hat{\mathbf{C}}_S)_{i,j \neq i}}{1 + \lambda} \leq \gamma' \leq \frac{1}{6k}$. Therefore we can set $\gamma = \gamma'(1 + \lambda)$ and obtain $(\hat{\mathbf{C}}_S)_{i,j \neq i} \leq \gamma \leq \frac{1 + \lambda}{6k}$. This concludes the proof. \square

Proof of Theorem 18. The proof follows the composition of Theorem 17, Corollary 4 and Lemma 19. In particular, we upper-bound the empirical risk of Theorem 17 with an approximation given by Corollary 4, ignoring the negative term. Next, we upper-bound $\epsilon(\lambda \|\mathbf{w}^*\|^2 + \lambda \|\boldsymbol{\beta}^*\|^2 + \hat{R}(h_{\mathbf{w}^*, \boldsymbol{\beta}^*}^{\text{trg}})) + \lambda \|\mathbf{w}^*\|^2 + \lambda \|\boldsymbol{\beta}^*\|^2$ by Lemma 19. \square

The following proposition is used to derive the GreedyTL in Section 6.4.

Proposition 3. Define the regularized accuracy as,

$$\widehat{A}^\lambda(\mathbf{w}) := 1 - \left(\frac{1}{m} \|\mathbf{X}^\top \mathbf{w} - \mathbf{y}\|_2^2 + \lambda \|\mathbf{w}\|_2^2 \right).$$

We are given $\mathbf{X} \in \mathbb{R}^{n \times m}$, $\mathbf{y} \in \mathbb{R}^m$, $S \subseteq \{1, \dots, n\}$, and $\lambda \in \mathbb{R}^+$. Furthermore, assume that $\frac{\|\mathbf{y}\|_2^2}{m} = 1$, and let $\widehat{\mathbf{X}}$ be the submatrix of \mathbf{X} , selecting rows indexed by S . Then we have that,

$$\max_{\mathbf{w}, \text{supp}(\mathbf{w})=S} \{\widehat{A}^\lambda(\mathbf{w})\} = \frac{1}{m} \mathbf{y}^\top \widehat{\mathbf{X}}^\top (\widehat{\mathbf{X}} \widehat{\mathbf{X}}^\top + m\lambda \mathbf{I})^{-1} \widehat{\mathbf{X}} \mathbf{y} \quad (\text{D.11})$$

$$= \frac{1}{m} \mathbf{y}^\top (\widehat{\mathbf{X}}^\top \widehat{\mathbf{X}} + m\lambda \mathbf{I})^{-1} \widehat{\mathbf{X}}^\top \widehat{\mathbf{X}} \mathbf{y}. \quad (\text{D.12})$$

Proof. Expanding the $\|\cdot\|^2$ in $\widehat{A}^\lambda(\mathbf{w})$ and using the fact that $\frac{\|\mathbf{y}\|_2^2}{m} = 1$, gives us

$$\widehat{A}^\lambda(\mathbf{w}) = \frac{2}{m} \mathbf{w}^\top \widehat{\mathbf{X}} \mathbf{y} - \frac{1}{m} \mathbf{w}^\top (\widehat{\mathbf{X}} \widehat{\mathbf{X}}^\top + m\lambda \mathbf{I}) \mathbf{w}.$$

Now we have that $\frac{\partial \widehat{A}^\lambda(\mathbf{w})}{\partial \mathbf{w}} = \mathbf{0} \Rightarrow \mathbf{w} = (\widehat{\mathbf{X}} \widehat{\mathbf{X}}^\top + m\lambda \mathbf{I})^{-1} \widehat{\mathbf{X}} \mathbf{y}$. Denote $\mathbf{G} = (\widehat{\mathbf{X}} \widehat{\mathbf{X}}^\top + m\lambda \mathbf{I})^{-1}$ and set optimal solution $\mathbf{w}^* = \mathbf{G} \widehat{\mathbf{X}} \mathbf{y}$. By putting \mathbf{w}^* into the objective we have,

$$\begin{aligned} \widehat{A}^\lambda(\mathbf{w}^*) &= \frac{2}{m} \mathbf{y}^\top \widehat{\mathbf{X}}^\top \mathbf{G}^\top \widehat{\mathbf{X}} \mathbf{y} - \frac{1}{m} \mathbf{y}^\top \widehat{\mathbf{X}}^\top \mathbf{G}^\top \mathbf{G}^{-1} \mathbf{G} \widehat{\mathbf{X}} \mathbf{y} \\ &= \frac{1}{m} \mathbf{y}^\top \widehat{\mathbf{X}}^\top \mathbf{G}^\top \widehat{\mathbf{X}} \mathbf{y}. \end{aligned}$$

This proves the first statement.

Now we turn to the second statement, that is the solution in the dual variables. By using dual variable identity $(\widehat{\mathbf{X}} \widehat{\mathbf{X}}^\top + m\lambda \mathbf{I})^{-1} \widehat{\mathbf{X}} = \widehat{\mathbf{X}} (\widehat{\mathbf{X}}^\top \widehat{\mathbf{X}} + m\lambda \mathbf{I})^{-1}$ [99], we write the solution w.r.t. \mathbf{w} as $\mathbf{w}^* = \widehat{\mathbf{X}} (\widehat{\mathbf{X}}^\top \widehat{\mathbf{X}} + m\lambda \mathbf{I})^{-1} \mathbf{y}$. Denoting $\mathbf{G} = (\widehat{\mathbf{X}}^\top \widehat{\mathbf{X}} + m\lambda \mathbf{I})^{-1}$, setting optimal solution $\mathbf{w}^* = \widehat{\mathbf{X}} \mathbf{G} \mathbf{y}$, and putting \mathbf{w}^* into the objective we have,

$$\begin{aligned} \widehat{A}^\lambda(\mathbf{w}^*) &= \frac{2}{m} \mathbf{y}^\top \mathbf{G}^\top \widehat{\mathbf{X}}^\top \widehat{\mathbf{X}} \mathbf{y} \\ &\quad - \frac{1}{m} \mathbf{y}^\top \mathbf{G}^\top \widehat{\mathbf{X}}^\top (\widehat{\mathbf{X}} \widehat{\mathbf{X}}^\top + m\lambda \mathbf{I}) \widehat{\mathbf{X}} \mathbf{G} \mathbf{y} = \frac{1}{m} \mathbf{y}^\top \mathbf{G} \widehat{\mathbf{X}}^\top \widehat{\mathbf{X}} \mathbf{y}. \end{aligned}$$

The last fact comes from the observation that $\widehat{\mathbf{X}} \mathbf{G} = (\widehat{\mathbf{X}} \widehat{\mathbf{X}}^\top + m\lambda \mathbf{I})^{-1} \widehat{\mathbf{X}}$ by dual variable identity. This concludes the proof of the second statement. \square

E Appendix for Chapter 7

E.1 Closed-form LOO prediction in Multiclass RLS

We follow closely the proof given by Cawley [20] with generalization to the multiclass scenario.

Additional notation:

\mathbf{X} – Sample matrix, where each column is a sample

\mathbf{Y} – Encoded OVA label matrix, where each label code is a column

\mathbf{A} – model parameter matrix, where each model parameters form a column

$\mathbf{A}^{(i)}$ – i -th row of a matrix \mathbf{A}

$\mathbf{A}^{(-i)}$ – all, but i -th row of a matrix \mathbf{A}

\mathbf{b} – transfer parameter vector

In the following we will assume that the solution in terms of \mathbf{A} and \mathbf{b} is given by solving

$$\begin{bmatrix} \mathbf{A} \\ \mathbf{b}^\top \end{bmatrix} = \mathbf{M}^{-1} \begin{bmatrix} \mathbf{Y} - \mathbf{X}^\top \mathbf{W}' \boldsymbol{\beta} \\ \mathbf{0} \end{bmatrix} \quad (\text{E.1})$$

To derive the LOO prediction formula, we need to solve Equation E.1 when one of the elements of \mathbf{X} is missing. For this reason we dissect matrix \mathbf{M} as follows (notice r.h.s.)

$$\begin{bmatrix} \mathbf{X}^\top \mathbf{X} + \frac{1}{C} \mathbf{I} & \mathbf{1} \\ \mathbf{1}^\top & 0 \end{bmatrix} = \mathbf{M} = \begin{bmatrix} m_{11} & \mathbf{m}_1^\top \\ \mathbf{m}_1 & M_1 \end{bmatrix}$$

Consequently, we recover a closed-form solution

$$\begin{bmatrix} \mathbf{A}^{(-1)} \\ \mathbf{b}^{\top(-1)} \end{bmatrix} = \mathbf{M}_1^{-1} (\mathbf{Y}^{(-1)} - [\mathbf{X}^{\top(-1)} \mathbf{W}' \quad \mathbf{X}^{\top(-1)} \mathbf{W}' \boldsymbol{\beta}])$$

E.1. Closed-form LOO prediction in Multiclass RLS

Using parameters $\begin{bmatrix} \mathbf{A}^{(-1)} \\ \mathbf{b}^{\top(-1)} \end{bmatrix}$ we obtain the prediction on the missing sample

$$\begin{aligned} \tilde{\mathbf{Y}}^{(1)} &= \mathbf{m}_1^\top \begin{bmatrix} \mathbf{A}^{(-1)} \\ \mathbf{b}^{\top(-1)} \end{bmatrix} + [\mathbf{X}^{\top(1)} \mathbf{W}' \mathbf{X}^{\top(1)} \mathbf{W}' \boldsymbol{\beta}] \\ &= \mathbf{m}_1 \mathbf{b}^\top \mathbf{M}_1^{-1} (\mathbf{Y}^{(-1)} - [\mathbf{X}^{\top(-1)} \mathbf{W}' \mathbf{X}^{\top(-1)} \mathbf{W}' \boldsymbol{\beta}]) \\ &\quad + [\mathbf{X}^{\top(1)} \mathbf{W}' \mathbf{X}^{\top(1)} \mathbf{W}' \boldsymbol{\beta}] \end{aligned} \tag{E.2}$$

Noting that predictions with respect to all, but the first element are

$$\begin{aligned} & \begin{bmatrix} \mathbf{m}_1 & \mathbf{M}_1 \end{bmatrix} \begin{bmatrix} \mathbf{A} \\ \mathbf{b}^\top \end{bmatrix} \\ &= \mathbf{M}_1^{-1} (\mathbf{Y}^{(-1)} - [\mathbf{X}^{\top(-1)} \mathbf{W}' \mathbf{X}^{\top(-1)} \mathbf{W}' \boldsymbol{\beta}]) \end{aligned}$$

we rewrite Equation E.2 as

$$\begin{aligned} \tilde{\mathbf{Y}}^{(1)} &= \mathbf{m}_1^\top \mathbf{M}_1^{-1} \begin{bmatrix} \mathbf{m}_1 & \mathbf{M}_1 \end{bmatrix} \begin{bmatrix} \mathbf{A} \\ \mathbf{b}^\top \end{bmatrix} \\ &\quad + [\mathbf{X}^{\top(1)} \mathbf{W}' \mathbf{X}^{\top(1)} \mathbf{W}' \boldsymbol{\beta}] \\ &= \mathbf{m}_1^\top \mathbf{M}_1^{-1} \mathbf{m}_1 \mathbf{A}^{(1)} + \mathbf{m}_1^\top \begin{bmatrix} \mathbf{A}^{(-1)} \\ \mathbf{b}^\top \end{bmatrix} \\ &\quad + [\mathbf{X}^{\top(1)} \mathbf{W}' \mathbf{X}^{\top(1)} \mathbf{W}' \boldsymbol{\beta}] \end{aligned} \tag{E.3}$$

Noting that the first equation in System E.1 is

$$\mathbf{Y}^{(1)} - [\mathbf{X}^{\top(1)} \mathbf{W}' \mathbf{X}^{\top(1)} \mathbf{W}' \boldsymbol{\beta}] = m_{11} \mathbf{A}^{(1)} + \mathbf{m}_1^\top \begin{bmatrix} \mathbf{A}^{(-1)} \\ \mathbf{b}^\top \end{bmatrix}$$

we rearrange and plug $\mathbf{m}_1^\top \begin{bmatrix} \mathbf{A}^{(-1)} \\ \mathbf{b}^\top \end{bmatrix}$ into Equation E.3 to arrive at

$$\begin{aligned} \tilde{\mathbf{Y}}^{(1)} &= \mathbf{m}_1^\top \mathbf{M}_1^{-1} \mathbf{m}_1 \mathbf{A}^{(1)} + \mathbf{Y}^{(1)} \\ &\quad - [\mathbf{X}^{\top(1)} \mathbf{W}' \mathbf{X}^{\top(1)} \mathbf{W}' \boldsymbol{\beta}] - m_{11} \mathbf{A}^{(1)} \\ &\quad + [\mathbf{X}^{\top(1)} \mathbf{W}' \mathbf{X}^{\top(1)} \mathbf{W}' \boldsymbol{\beta}] \\ &= \mathbf{Y}^{(1)} + (\mathbf{m}_1^\top \mathbf{M}_1^{-1} \mathbf{m}_1 - m_{11}) \mathbf{A}^{(1)} \end{aligned}$$

Expressing \mathbf{M}^{-1} by the Schur complement lemma, we observe that the inverse of the complement $\mu = m_{11} - \mathbf{m}_1^\top \mathbf{M}_1^{-1} \mathbf{m}_1$ is the first matrix element.

$$\mathbf{M}^{-1} = \begin{bmatrix} \mu^{-1} & -\mu^{-1} \mathbf{m}_1 \mathbf{M}_1^{-1} \\ \mathbf{M}_1^{-1} + \mu^{-1} \mathbf{M}_1^{-1} \mathbf{m}_1^\top \mathbf{m}_1 \mathbf{M}_1^{-1} & -\mu^{-1} \mathbf{M}_1^{-1} \mathbf{m}_1^\top \end{bmatrix}$$

Appendix E. Appendix for Chapter 7

Combining this fact with insensitivity of system to row-wise permutations, for the i -th sample we have:

$$\tilde{\mathbf{Y}}^{(i)} = \mathbf{Y}^{(i)} - \frac{\mathbf{A}^{(i)}}{M_{ii}^{-1}}$$

Bibliography

- [1] Z. Allen-Zhu and E. Hazan. Variance reduction for faster non-convex optimization. In *International Conference on Machine Learning (ICML)*, pages 699–707, 2016.
- [2] Y. Aytar and A. Zisserman. Tabula rasa: Model transfer for object category detection. In *International Conference on Computer Vision (ICCV)*, 2011.
- [3] M. G. Azar, A. Lazaric, and E. Brunskill. Regret bounds for reinforcement learning with policy advice. In *European Conference on Machine Learning (ECML)*, pages 97–112. Springer-Verlag New York, Inc., 2013.
- [4] M. Baktashmotlagh, M. Harandi, B. Lovell, and M. Salzmann. Unsupervised domain adaptation by domain invariant projection. In *International Conference on Computer Vision (ICCV)*, 2013.
- [5] P. L. Bartlett, O. Bousquet, and S. Mendelson. Local rademacher complexities. *Annals of Statistics*, pages 1497–1537, 2005.
- [6] P. L. Bartlett and S. Mendelson. Rademacher and gaussian complexities: risk bounds and structural results. *Journal of Machine Learning Research*, 2003.
- [7] A. Bellet, A. Habrard, and M. Sebban. Metric learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 9(1):1–151, 2015.
- [8] S. Ben-David, J. Blitzer, K. Crammer, A. Kulesza, F. Pereira, and J. Vaughan. A theory of learning from different domains. *Machine Learning*, 79(1):151–175, 2010.
- [9] S. Ben-David, T. Lu, T. Luu, and D. Pál. Impossibility Theorems for Domain Adaptation. *JMLR W&CP*, 9:129–136, 2010.
- [10] S. Ben-David and R. Urner. On the hardness of domain adaptation and the utility of unlabeled target samples. In *Algorithmic Learning Theory (ALT)*, pages 139–153. Springer, 2012.
- [11] S. Ben-David and R. Urner. Domain adaptation as learning with auxiliary information. NIPS workshop on New Directions in Transfer and Multi-Task, 2013.
- [12] A. Bergamo and L. Torresani. Classemes and other classifier-based features for efficient object categorization. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2014.
- [13] C. Bishop. *Pattern Recognition and Machine Learning*. Springer-Verlag New York, 2006.
- [14] J. Blitzer, R. McDonald, and F. Pereira. Domain adaptation with structural correspondence learning. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2006.

Bibliography

- [15] A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *Conference on Learning Theory (COLT)*, pages 92–100. ACM, 1998.
- [16] S. Boucheron, G. Lugosi, and P. Massart. *Concentration inequalities: A nonasymptotic theory of independence*. Oxford University Press, 2013.
- [17] O. Bousquet and A. Elisseeff. Stability and Generalization. *Journal of Machine Learning Research*, 2:499–526, 2002.
- [18] P. Bühlmann and S. Van De Geer. *Statistics for high-dimensional data: methods, theory and applications*. Springer, 2011.
- [19] R. Caruana. *Multitask learning*. Springer, 1998.
- [20] G. Cawley. Leave-one-out cross-validation based model selection criteria for weighted LS-SVMs. In *IEEE International Joint Conference on Neural Networks (IJCNN)*, 2006.
- [21] G. C. Cawley and N. L. C. Talbot. Preventing Over-Fitting during Model Selection via Bayesian Regularisation of the Hyper-Parameters. *Journal of Machine Learning Research*, 8:841–861, 2007.
- [22] O. Chapelle, B. Schölkopf, A. Zien, et al. *Semi-supervised learning*, volume 2. The MIT Press, 2006.
- [23] P. Chaudhari, A. Choromanska, S. Soatto, Y. LeCun, C. Baldassi, C. Borgs, J. Chayes, L. Sagun, and R. Zecchina. Entropy-sgd: Biasing gradient descent into wide valleys. In *International Conference on Learning Representations (ICLR)*, 2017.
- [24] M. J. Choi, J. J. Lim, A. Torralba, and A. S. Willsky. Exploiting hierarchical context on a large database of object categories. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.
- [25] A. Choromanska, M. Henaff, M. Mathieu, G. B. Arous, and Y. LeCun. The loss surfaces of multilayer networks. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2015.
- [26] C. Cortes and M. Mohri. Domain adaptation and sample bias correction theory and algorithm for regression. *Theoretical Computer Science*, 519:103–126, 2014.
- [27] C. Cortes, M. Mohri, M. Riley, and A. Rostamizadeh. Sample Selection Bias Correction Theory. In *Algorithmic Learning Theory (ALT)*, pages 38–53. Springer, 2008.
- [28] K. Crammer and Y. Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of Machine Learning Research*, 2:265–292, 2002.
- [29] F. Cucker and D. X. Zhou. *Learning theory: an approximation theory viewpoint*, volume 24. Cambridge University Press, 2007.
- [30] A. Das and D. Kempe. Algorithms for subset selection in linear regression. In *STOC*, 2008.
- [31] A. Das and D. Kempe. Submodular meets spectral: Greedy algorithms for subset selection, sparse approximation and dictionary selection. In *International Conference on Machine Learning (ICML)*, 2011.

- [32] H. Daumé III. Frustratingly easy domain adaptation. In *ACL*, 2007.
- [33] H. Daumé III, A. Kumar, and A. Saha. Frustratingly easy semi-supervised domain adaptation. In *Proceedings of the 2010 Workshop on Domain Adaptation for Natural Language Processing*, pages 53–59. Association for Computational Linguistics, 2010.
- [34] E. De Vito, A. Caponnetto, and L. Rosasco. Model Selection for Regularized Least-Squares Algorithm in Learning Theory. *Found. Comput. Math.*, 5(1):59–85, Feb. 2005.
- [35] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.
- [36] L. Devroye and T. Wagner. Distribution-free performance bounds for potential function rules. *IEEE Transactions on Information Theory*, 25(5):601–604, 1979.
- [37] C. Domingo and O. Watanabe. MadaBoost: A modification of AdaBoost. In *Conference on Learning Theory (COLT)*, pages 180–189, 2000.
- [38] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *International Conference on Machine Learning (ICML)*, 2014.
- [39] L. Duan, I. W. Tsang, D. Xu, and T.-S. Chua. Domain adaptation from multiple sources via auxiliary classifiers. In *International Conference on Machine Learning (ICML)*, 2009.
- [40] L. Duan, D. Xu, and S.-F. Chang. Exploiting web images for event recognition in consumer videos: A multiple source domain adaptation approach. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.
- [41] L. Duan, D. Xu, I. Tsang, and J. Luo. Visual event recognition in videos by learning from web data. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(9):1667–1680, 2012.
- [42] A. Elisseeff, T. Evgeniou, and M. Pontil. Stability of randomized learning algorithms. *Journal of Machine Learning Research*, 6(Jan):55–79, 2005.
- [43] A. Elisseeff and M. Pontil. Leave-one-out Error and Stability of Learning Algorithms with Applications. In *Advances in Learning Theory: Methods, Models and Applications*, pages 111–125. VIOS Press, 2003.
- [44] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. Liblinear: A library for large linear classification. *Journal of Machine Learning Research*, 2008.
- [45] L. Fei-Fei, R. Fergus, and P. Perona. One-shot learning of object categories. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(4):594–611, 2006.
- [46] T. Galanti, L. Wolf, and T. Hazan. A theoretical framework for deep transfer learning. *Information and Inference*, page iaw008, 2016.
- [47] Y. Ganin and V. S. Lempitsky. Unsupervised domain adaptation by backpropagation. In *International Conference on Machine Learning (ICML)*, pages 1180–1189, 2015.

Bibliography

- [48] R. Ge, F. Huang, C. Jin, and Y. Yuan. Escaping from saddle points-online stochastic gradient for tensor decomposition. In *Conference on Learning Theory (COLT)*, pages 797–842, 2015.
- [49] P. Gehler and S. Nowozin. On feature combination for multiclass object classification. In *International Conference on Computer Vision (ICCV)*, 2009.
- [50] S. Ghadimi and G. Lan. Stochastic first-and zeroth-order methods for nonconvex stochastic programming. *SIAM Journal on Optimization*, 23(4):2341–2368, 2013.
- [51] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Region-based convolutional networks for accurate object detection and segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PP(99):1–1, 2015.
- [52] A. Gonen and S. Shalev-Shwartz. Fast rates for empirical risk minimization of strict saddle problems. *arXiv preprint arXiv:1701.04271*, 2017.
- [53] B. Gong, Y. Shi, F. Sha, and K. Grauman. Geodesic flow kernel for unsupervised domain adaptation. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.
- [54] I. Goodfellow, Y. Bengio, and A. Courville. *Deep learning*. The MIT Press, 2016.
- [55] A. Gretton, A. Smola, J. Huang, M. Schmittfull, K. Borgwardt, and B. Schölkopf. Covariate shift by kernel mean matching. *Dataset shift in machine learning*, 3(4):5, 2009.
- [56] G. Griffin, A. Holub, and P. Perona. Caltech-256 object category dataset. Technical report, Caltech, 2007.
- [57] M. Hardt, B. Recht, and Y. Singer. Train faster, generalize better: Stability of stochastic gradient descent. In *International Conference on Machine Learning (ICML)*, 2016.
- [58] T. Hastie, R. Tibshirani, and J. Friedman. *The Elements Of Statistical Learning*. Springer, 2009.
- [59] D. Haussler. Decision theoretic generalizations of the pac model for neural net and other learning applications. *Information and computation*, 100(1):78–150, 1992.
- [60] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.
- [61] S. Hochreiter and J. Schmidhuber. Flat minima. *Neural Computation*, 9(1):1–42, 1997.
- [62] A. Hoorfar and M. Hassani. Inequalities on the lambert w function and hyperpower function. *J. Inequal. Pure and Appl. Math*, 9(2), 2008.
- [63] S. Ioffe and C. Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International Conference on Machine Learning (ICML)*, pages 448–456, 2015.
- [64] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *ACM International Conference on Multimedia*, pages 675–678. ACM, 2014.

-
- [65] L. Jie, T. Tommasi, and B. Caputo. Multiclass transfer learning from unconstrained priors. In *International Conference on Computer Vision (ICCV)*, 2011.
- [66] S. M. Kakade, K. Sridharan, and A. Tewari. On the complexity of linear prediction: Risk bounds, margin bounds, and regularization. In *Conference on Neural Information Processing Systems (NIPS)*, pages 793–800, 2008.
- [67] K. Kawaguchi. Deep learning without poor local minima. In *Conference on Neural Information Processing Systems (NIPS)*, pages 586–594, 2016.
- [68] N. S. Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, and P. T. P. Tang. On large-batch training for deep learning: Generalization gap and sharp minima. In *International Conference on Learning Representations (ICLR)*, 2017.
- [69] W. Kienzle and K. Chellapilla. Personalized handwriting recognition via biased regularization. In *International Conference on Machine Learning (ICML)*, pages 457–464, 2006.
- [70] V. Koltchinskii and D. Panchenko. Rademacher processes and bounding the risk of function learning. In *High Dimensional Probability II*, pages 443–457. Springer, 2000.
- [71] T. Koren and K. Levy. Fast rates for exp-concave empirical risk minimization. In *Conference on Neural Information Processing Systems (NIPS)*, pages 1477–1485, 2015.
- [72] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *Conference on Neural Information Processing Systems (NIPS)*, pages 1097–1105, 2012.
- [73] I. Kuzborskij, F. M. Carlucci, and B. Caputo. When Naive Bayes Nearest Neighbours Meet Convolutional Neural Networks. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.
- [74] I. Kuzborskij and N. Cesa-Bianchi. Nonparametric Online Regression while Learning the Metric. *arXiv preprint arXiv:1705.07853*, 2017.
- [75] I. Kuzborskij and C. H. Lampert. Data-Dependent Stability of Stochastic Gradient Descent. *arXiv preprint arXiv:1703.01678*, 2017.
- [76] I. Kuzborskij and F. Orabona. Stability and hypothesis transfer learning. In *International Conference on Machine Learning (ICML)*, pages 942–950, 2013.
- [77] I. Kuzborskij and F. Orabona. Fast Rates by Transferring from Auxiliary Hypotheses. *Machine Learning*, pages 1–25, 2016.
- [78] I. Kuzborskij, F. Orabona, and B. Caputo. From N to N+1: Multiclass Transfer Incremental Learning. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3358–3365. IEEE, 2013.
- [79] I. Kuzborskij, F. Orabona, and B. Caputo. Transfer learning through greedy subset selection. In *Image Analysis and Processing - ICIAP 2015 - 18th International Conference, Proceedings, Part I*, pages 3–14, 2015.

Bibliography

- [80] I. Kuzborskij, F. Orabona, and B. Caputo. Scalable greedy algorithms for transfer learning. *Computer Vision and Image Understanding*, 156:174–185, 2017.
- [81] C. H. Lampert, H. Nickisch, and S. Harmeling. Learning to detect unseen object classes by between-class attribute transfer. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.
- [82] A. Lazaric. Transfer in reinforcement learning: a framework and a survey. *Reinforcement Learning*, 12:143–173, 2012.
- [83] A. Lazaric, M. Restelli, and A. Bonarini. Transfer of samples in batch reinforcement learning. In *International Conference on Machine Learning (ICML)*, 2008.
- [84] J. D. Lee, M. Simchowitz, M. I. Jordan, and B. Recht. Gradient descent only converges to minimizers. In *Conference on Learning Theory (COLT)*, pages 1246–1257, 2016.
- [85] L. Li, H. Su, E. Xing, and L. Fei-Fei. Object bank: A high-level image representation for scene classification & semantic feature sparsification. In *Conference on Neural Information Processing Systems (NIPS)*, 2010.
- [86] X. Li and J. Bilmes. A bayesian divergence prior for classifier adaptation. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, pages 275–282, 2007.
- [87] J. J. Lim, A. Torralba, and R. Salakhutdinov. Transfer learning by borrowing examples for multi-class object detection. In *Conference on Neural Information Processing Systems (NIPS)*, 2011.
- [88] T. Liu, G. Lugosi, G. Neu, and D. Tao. Algorithmic stability and hypothesis complexity. In *International Conference on Machine Learning (ICML)*, 2017.
- [89] B. London. Generalization bounds for randomized learning with application to stochastic gradient descent. In *NIPS Workshop on Optimizing the Optimizers*, 2016.
- [90] M. Long, Y. Cao, J. Wang, and M. I. Jordan. Learning transferable features with deep adaptation networks. In *International Conference on Machine Learning (ICML)*, pages 97–105, 2015.
- [91] M. Mahdavi, L. Zhang, and R. Jin. Lower and Upper Bounds on the Generalization of Stochastic Exponentially Concave Optimization. In *Conference on Learning Theory (COLT)*, pages 1305–1320, 2015.
- [92] Y. Mansour, M. Mohri, and A. Rostamizadeh. Domain adaptation with multiple sources. In *Conference on Neural Information Processing Systems (NIPS)*, pages 1041–1048, 2008.
- [93] Y. Mansour, M. Mohri, and A. Rostamizadeh. Domain adaptation: Learning bounds and algorithms. In *Conference on Learning Theory (COLT)*, 2009.
- [94] Y. Mansour, M. Mohri, and A. Rostamizadeh. Domain Adaptation with Multiple Sources. In *Conference on Neural Information Processing Systems (NIPS)*, 2009.
- [95] A. Maurer. A second-order look at stability and generalization. In *Conference on Learning Theory (COLT)*, 2017.

- [96] A. Maurer and M. Pontil. Empirical bernstein bounds and sample variance penalization. *arXiv preprint arXiv:0907.3740*, 2009.
- [97] G. A. Miller. Wordnet: a lexical database for english. *Communications of the ACM*, 38(11):39–41, 1995.
- [98] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski, et al. Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533, 2015.
- [99] M. Mohri, A. Rostamizadeh, and A. Talwalkar. *Foundations of machine learning*. The MIT Press, 2012.
- [100] F. Ojeda, J. A. Suykens, and B. De Moor. Low rank updated ls-svm classifiers for fast variable selection. *Neural Networks*, 21(2):437–449, 2008.
- [101] M. Oquab, L. Bottou, I. Laptev, and J. Sivic. Learning and transferring mid-level image representations using convolutional neural networks. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.
- [102] F. Orabona. Simultaneous model selection and optimization through parameter-free stochastic learning. In *Conference on Neural Information Processing Systems (NIPS)*, pages 1116–1124, 2014.
- [103] F. Orabona, C. Castellini, B. Caputo, A. Fiorilla, and G. Sandini. Model Adaptation with Least-Squares SVM for Adaptive Hand Prosthetics. In *Robotics and Automation, IEEE International Conference on*, pages 2897–2903. IEEE, 2009.
- [104] S. J. Pan, I. W. Tsang, J. T. Kwok, and Q. Yang. Domain adaptation via transfer component analysis. *Neural Networks, IEEE Transactions on*, 22(2):199–210, 2011.
- [105] S. J. Pan and Q. Yang. A survey on transfer learning. *Knowledge and Data Engineering, IEEE Transactions on*, 22(10):1345–1359, 2010.
- [106] V. M. Patel, R. Gopalan, R. Li, and R. Chellappa. Visual domain adaptation: a survey of recent advances. *IEEE Signal Processing Magazine*, 2014.
- [107] N. Patricia and B. Caputo. Learning to learn, from transfer learning to domain adaptation: A unifying perspective. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1442–1449. IEEE, 2014.
- [108] B. A. Pearlmutter. Fast exact multiplication by the hessian. *Neural Computation*, 6(1):147–160, 1994.
- [109] A. Pentina and C. H. Lampert. A pac-bayesian bound for lifelong learning. In *International Conference on Machine Learning (ICML)*, 2014.
- [110] M. Perrot and A. Habrard. A theoretical analysis of metric hypothesis transfer learning. In *International Conference on Machine Learning (ICML)*, pages 1708–1717, 2015.
- [111] K. Petersen and M. Pedersen. The matrix cookbook. *Technical University of Denmark*, 2008.

Bibliography


- [112] T. Poggio, S. Voinea, and L. Rosasco. Online learning, stability, and stochastic gradient descent. *arXiv preprint arXiv:1105.4701*, 2011.
- [113] Q. Qiu, V. M. Patel, P. Turaga, and R. Chellappa. Domain adaptive dictionary learning. In *European Conference on Computer Vision (ECCV)*, pages 631–645. Springer, 2012.
- [114] S. J. Reddi, A. Hefny, S. Sra, B. Póczos, and A. Smola. Stochastic variance reduction for nonconvex optimization. In *International Conference on Machine Learning (ICML)*, pages 314–323, 2016.
- [115] R. Rifkin and A. Klautau. In defense of one-vs-all classification. *Journal of Machine Learning Research*, 5:101–141, 2004.
- [116] R. Rifkin, G. Yeo, and T. Poggio. Regularized Least-Squares Classification. In J. A. K. Suykens, G. Horvath, S. Basu, C. Micchelli, and J. Vandewalle, editors, *Advances in Learning Theory: Methods, Models and Applications*, pages 131–154. VIOS Press, 2003.
- [117] M. Rohrbach, M. Stark, and B. Schiele. Evaluating knowledge transfer and zero-shot learning in a large-scale setting. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.
- [118] K. Saenko, B. Kulis, M. Fritz, and T. Darrell. Adapting visual category models to new domains. In *European Conference on Computer Vision (ECCV)*, 2010.
- [119] H. Sak, A. Senior, K. Rao, and F. Beaufays. Fast and accurate recurrent neural network acoustic models for speech recognition. In *Conference of the International Speech Communication Association*, 2015.
- [120] R. Salakhutdinov, A. Torralba, and J. Tenenbaum. Learning to share visual appearance for multiclass object detection. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.
- [121] B. Schölkopf, R. Herbrich, and A. J. Smola. A generalized representer theorem. In *Conference on Learning Theory (COLT)*, pages 416–426. Springer, 2001.
- [122] C.-W. Seah, I. W.-H. Tsang, and Y.-S. Ong. Healing sample selection bias by source classifier selection. In *IEEE International Conference on Data Mining (ICDM)*, pages 577–586. IEEE, 2011.
- [123] S. Shalev-Shwartz and S. Ben-David. *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014.
- [124] S. Shalev-Shwartz, O. Shamir, N. Srebro, and K. Sridharan. Learnability, stability and uniform convergence. *Journal of Machine Learning Research*, 11(Oct):2635–2670, 2010.
- [125] V. Sharmanska, N. Quadrianto, and C. H. Lampert. Learning to rank using privileged information. In *International Conference on Computer Vision (ICCV)*, pages 825–832. IEEE, 2013.
- [126] D. Silver, A. Huang, C. J. Maddison, A. Guez, L. Sifre, G. van den Driessche, J. Schrittwieser, I. Antonoglou, V. Panneershelvam, M. Lanctot, S. Dieleman, D. Grewe, J. Nham, N. Kalchbrenner, I. Sutskever, T. Lillicrap, M. Leach, K. Kavukcuoglu, T. Graepel, and D. Hassabis. Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484–489, Jan 2016.
- [127] A. Smola and B. Schölkopf. *Learning with Kernels*. MIT press, Cambridge, MA, USA, 2002.

- [128] A. J. Smola and B. Schölkopf. Sparse greedy matrix approximation for machine learning. In *International Conference on Machine Learning, ICML '00*, pages 911–918, San Francisco, CA, USA, 2000. Morgan Kaufmann Publishers Inc.
- [129] N. Srebro, K. Sridharan, and A. Tewari. Smoothness, low noise and fast rates. In *Conference on Neural Information Processing Systems (NIPS)*, 2010.
- [130] N. Srebro, K. Sridharan, and A. Tewari. Smoothness, low noise and fast rates. In *Conference on Neural Information Processing Systems (NIPS)*, pages 2199–2207, 2010.
- [131] J. M. Steele. An efron-stein inequality for nonsymmetric statistics. *The Annals of Statistics*, pages 753–758, 1986.
- [132] J. A. K. Suykens, T. Van Gestel, and J. De Brabanter. *Least squares support vector machines*. World Scientific, 2002.
- [133] M. Taylor and P. Stone. Transfer learning for reinforcement learning domains: A survey. *Journal of Machine Learning Research*, 10:1633–1685, 2009.
- [134] S. Thrun. Lifelong learning algorithms. In *Learning to learn*, pages 181–209. Springer, 1998.
- [135] T. Tommasi and B. Caputo. Frustratingly easy nbnn domain adaptation. In *International Conference on Computer Vision (ICCV)*, 2013.
- [136] T. Tommasi, F. Orabona, and B. Caputo. Safety in numbers: Learning categories from few examples with multi model knowledge transfer. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3081–3088, 2010.
- [137] T. Tommasi, F. Orabona, and B. Caputo. Learning categories from few examples with multi model knowledge transfer. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(5):928–941, 2014.
- [138] T. Tommasi, F. Orabona, C. Castellini, and B. Caputo. Improving control of dexterous hand prostheses using adaptive learning. *IEEE Transactions on Robotics*, 29(1):207–219, 2013.
- [139] A. Torralba and A. A. Efros. Unbiased look at dataset bias. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.
- [140] V. Vapnik and A. Vashist. A new learning paradigm: Learning using privileged information. *Neural Networks*, 22(5):544–557, 2009.
- [141] V. N. Vapnik and A. Y. Chervonenkis. On the Uniform Convergence of Relative Frequencies of Events to Their Probabilities. *Theory of Probability & Its Applications*, 16(2):264–280, 1971.
- [142] A. Vezhnevets and V. Ferrari. Associative embeddings for large-scale knowledge transfer with self-assessment. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.
- [143] Y. Wu, M. Schuster, Z. Chen, Q. V. Le, M. Norouzi, W. Macherey, M. Krikun, Y. Cao, Q. Gao, K. Macherey, J. Klingner, A. Shah, M. Johnson, X. Liu, L. Kaiser, S. Gouws, Y. Kato, T. Kudo, H. Kazawa, K. Stevens, G. Kurian, N. Patil, W. Wang, C. Young, J. Smith, J. Riesa, A. Rudnick, O. Vinyals, G. Corrado, M. Hughes, and J. Dean. Google’s neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144, 2016.

Bibliography

- [144] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.
- [145] J. Yang, R. Yan, and A. Hauptmann. Cross-Domain Video Concept Detection Using Adaptive SVMs. In *Proceedings of the 15th international conference on Multimedia*, pages 188–197. ACM, 2007.
- [146] Y. Yao and G. Doretto. Boosting for transfer learning with multiple sources. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.
- [147] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals. Understanding deep learning requires rethinking generalization. In *International Conference on Learning Representations (ICLR)*, 2017.
- [148] K. Zhang, V. Zheng, Q. Wang, J. Kwok, Q. Yang, and I. Marsic. Covariate shift in hilbert space: A solution via surrogate kernels. In *International Conference on Machine Learning (ICML)*, 2013.
- [149] T. Zhang. Leave-one-out Bounds for Kernel Methods. *Neural Computation*, 15(6):1397–1437, 2003.
- [150] T. Zhang. Adaptive forward-backward greedy algorithm for sparse learning with linear models. In *Conference on Neural Information Processing Systems (NIPS)*, 2008.
- [151] T. Zhang. On the consistency of feature selection using greedy least squares regression. *Journal of Machine Learning Research*, 2009.
- [152] B. Zhou, A. Lapedriza, J. Xiao, A. Torralba, and A. Oliva. Learning deep features for scene recognition using places database. In *Conference on Neural Information Processing Systems (NIPS)*, pages 487–495, 2014.

Ilja Kuzborskij

ilja.kuzborskij@gmail.com
<http://idiap.ch/~ikuzbor>
[Google Scholar](#) 
Citizenship: Lithuania (EU)

My current research interest is in data-dependent analysis of learning algorithms. In particular I am interested in analysis of deep learning problems, stochastic optimization, online learning, nonparametric prediction, and efficient forms of transfer learning.

- EDUCATION**
- École polytechnique fédérale de Lausanne (EPFL)** 2012 - Sept 2017 (planned)
PhD student, Electrical Engineering
Thesis: *Hypothesis Transfer Learning*
Advisors: Prof. Barbara Caputo and Prof. Francesco Orabona
- University of Edinburgh** 2010 - 2011
MSc Artificial Intelligence
Thesis: *Large-Scale Pattern Mining of Computer Program Source Code*
Advisor: Prof. Charles Sutton
- EXPERIENCE**
- IST Austria (Scientific Visitor)** Oct 2016 - Apr 2017
Data-Dependent Generalization Bounds for Non-convex Problems
Advisor: Prof. Christoph Lampert
Analyzed learning ability of SGD algorithm widely used in deep learning in a data-dependent setting. Derived theory which motivates new transfer learning algorithms, and wrote efficient software to compute top Hessian eigenvalue of deep neural nets.
- EPFL and Idiap Research Institute (PhD student)** Sep 2012 - present
Hypothesis Transfer Learning
Advisor: Prof. Barbara Caputo and Prof. Francesco Orabona
Designed and analyzed efficient transfer learning algorithms that can learn much faster by leveraging on auxiliary pre-trained models. Developed new theory that corroborates success of many previous works in the area, and designed novel algorithms that are able to learn efficiently by reusing thousands of auxiliary models. Designed and analyzed online and stochastic locally-linear learning algorithms.
- University of Rome La Sapienza (Research Assistant)** Oct 2014 - present
Hypothesis Transfer Learning and Locally-Linear Learning
Advisor: Prof. Barbara Caputo
In collaboration with Prof. Nicolò Cesa-Bianchi designed and analyzed online learning algorithm that learns in rich (nonparametric) environments, and simultaneously reduces curse of dimensionality. Designed and analyzed randomized greedy transfer learning algorithms that can learn faster by leveraging on auxiliary models.

Toyota Technological Institute as Chicago (Intern)

Summer 2013

Hypothesis Transfer Learning

Supervisor: Prof. Francesco Orabona

Analyzed the family of transfer learning algorithms that learn by reusing auxiliary pre-trained models induced from previous tasks. Identified key quantitative characteristics of relatedness between new and previous tasks, and developed theory explicating this quantity, supporting theoretically many previous works in the literature.

Idiap Research Institute (Intern)

Jan 2012 - Sep 2012

Electromyography Classification with Large Number of Grasps

Supervisor: Prof. Barbara Caputo

Conducted evaluation on feasibility of recognition of 52 hand grasps from surface electromyography to investigate potential application in robotic hand prosthetics.

CERN (Intern)

Summer 2009

Supervisor: Dr. Vincenzo Innocente

Developed a domain-specific information retrieval and natural language-based system for semi-automatic software error resolution used in the CMS experiment.

PUBLICATIONS**Technical Reports**

I. Kuzborskij and C. H. Lampert. [Data-Dependent Stability of Stochastic Gradient Descent](#). *arXiv preprint arXiv:1703.01678*, 2017.

I. Kuzborskij and N. Cesa-Bianchi. [Nonparametric Online Regression while Learning the Metric](#). *arXiv preprint arXiv:1705.07853*, 2017.

Journal Papers

I. Kuzborskij and F. Orabona. [Fast Rates by Transferring from Auxiliary Hypotheses](#). *Machine Learning, Springer*, 2017.

I. Kuzborskij, F. Orabona, and B. Caputo. [Scalable Greedy Algorithms for Transfer Learning](#). *Computer Vision and Image Understanding, Elsevier*, 2016.

M. Atzori, A. Gijsberts, **I. Kuzborskij**, S. Heynen, A. Mittaz Hager, O. Deriaz, C. Castellini, H. Müller, and B. Caputo. [Characterization of a Benchmark Database for Myoelectric Movement Classification](#). *IEEE Transactions on Neural Systems and Rehabilitation Engineering (TNSRE)*, 2014.

Peer Reviewed Conferences

I. Kuzborskij, F. M. Carlucci, and B. Caputo. [When Naïve Bayes Nearest Neighbors Meet Convolutional Neural Networks](#). In *Computer Vision and Pattern Recognition (CVPR)*, 2016.

I. Kuzborskij, B. Caputo, and F. Orabona. [Transfer Learning through Greedy Subset Selection](#). In *International Conference on Image Analysis and Processing (ICIAP)*, Oral presentation, **Best Paper Award**, 2015.

I. Kuzborskij and F. Orabona. [Stability and Hypothesis Transfer Learning](#). In *International Conference on Machine Learning (ICML)*, 2013.

I. Kuzborskij, F. Orabona, and B. Caputo. [From N to N+1: Multiclass Transfer Incremental Learning](#). In *Computer Vision and Pattern Recognition (CVPR)*, 2013.

I. Kuzborskij, A. Gijsberts, and B. Caputo. [On the Challenge of Classifying 52 Hand Movements from Surface Electromyography](#). In *Engineering in Medicine and Biology Society (EMBC)*, 2012.

ACTIVITIES

Reviewer

International Conference on Machine Learning (ICML)	2014-2017
Conference on Neural Information Processing Systems (NIPS)	2017
Journal of Machine Learning Research (JMLR)	2017
Conference On Learning Theory (COLT)	2017
International Conference on Artificial Intelligence and Statistics (AISTATS)	2016/17
International Conference on Algorithmic Learning Theory (ALT)	2016
Elsevier Journal on Computer Vision and Image Understanding (CVIU)	2015-2017

Events

Google ML Summit, Zürich	June 2017
Online Learning Summer School, Copenhagen	June 2015

Teaching

Artificial Intelligence and Machine Learning 2015/16

Class of 176 student, TA and substitute for ML part, designed homework assignments, graded reports.

HONORS	Best Paper Award at the International Conference on Image Analysis and Processing	2015
	Postgraduate Tuition Award, Student Awards Agency for Scotland	2010
TECHNICAL SKILLS	Machine Learning: empirical risk minimization, neural networks, SVMs, regularization, kernel methods, locally-linear methods, feature selection.	
	Optimization: (Stochastic) gradient methods, accelerated methods, proximal methods, elements of submodular optimization.	
	Learning theory: algorithmic stability, concentration bounds, uniform deviation bounds, Rademacher complexity, tools from online and nonparametric analysis.	
	Programming: python (numpy, scipy, scikit-learn), C++, elements of Tensorflow, grid computation (SGE / MapReduce).	
REFEREES	Prof. Barbara Caputo University of Rome La Sapienza caputo@dis.uniroma1.it	
	Prof. Francesco Orabona Stony Brook University francesco@orabona.com	
	Prof. Nicolò Cesa-Bianchi Università degli Studi di Milano nicolo.cesa-bianchi@unimi.it	
	Prof. Christoph H. Lampert IST Austria chl@ist.ac.at	

