Algorithmic Stability and Meta-Learning

Andreas Maurer
Adalbertstr. 55
D-80799 München
andreasmaurer@compuserve.com

June 5, 2005

Abstract

A mechnism of transfer learning is analysed, where samples drawn from different learning tasks of an environment are used to improve the learners performance on a new task. We give a general method to prove generalisation error bounds for such meta-algorithms. The method can be applied to the bias learning model of J.Baxter and to derive novel generalisation bounds for meta-algorithms searching spaces of uniformly stable algorithms. We also present an application to regularized least squares regression.

1 Introduction

We formally study the phenomenon of *transfer*, where novel tasks and concepts are learned more quickly and reliably through the application of past experience. Transfer is fundamental to human learning (see [13] for an overview of the psychological literature) and offers a way to partially escape the implications of the *No Free Lunch Theorem* (NFLT).

The NFLT states that no algorithm is superior to another when averaged uniformly across all learning tasks. In a real environment, however, not all learning tasks occur equally likely. They are distributed according to some environmental distribution $\mathcal{E}$, which is far from uniform. By gathering information on this distribution of tasks, a learner can possibly find an algorithm to outperform other algorithms, but, of course, only on average over the distribution $\mathcal{E}$.

This mechanism of *meta-learning* has been analysed by Jonathan Baxter ([2], [3]) and there have been several successful experiments in practical machine-learning contexts (see [5],[14],[15] and section 6). In this paper we extend the results in [3] and offer a general method to control the generalization error of meta-learning. We begin by reviewing some notions of learning theory.

Generalization error bounds. Statistical learning theory deals with *data* and *hypotheses*. A data point z may be an input-output pair $z = (x, y)$ and a

hypothesis c may be some function $x \mapsto c(x)$, but for many theoretical results data and hypotheses can be arbitrary objects z and c, related only through a nonnegative *loss function* $l(c,z)$ which measures how poorly the hypothesis c applies to the data point z. The familiar square loss $l(c,(x,y)) = (c(x)-y)^2$ is an example where $z=(x,y)$ with $y \in \mathbb{R}$ and $c : x \mapsto c(x) \in \mathbb{R}$.

A *learning task* is modelled by a probability distribution D on the set of data points, $D(z)$ being interpreted as the probability that the data point z will be encountered under the conditions of the task D. For a given hypothesis c the *risk*

$$R(c,D) = E_{z\sim D}[l(c,z)] \tag{1}$$

measures how poorly the hypothesis c is expected to perform on D.

A *learning algorithm* A takes a *sample* $S=(z_1,...,z_m)$ of data, drawn iid from the distribution D defining the learning task, and computes a hypothesis $A(S)$. The returned hypothesis should work well on the same learning task D, so we want the risk $R(A(S),D)$ to be small. The quantity

$$E_{S\sim D^m}[R(A(S),D)] \tag{2}$$

would be a natural measure for the performance of a given algorithm A with respect to a given learning task D.

Unfortunately the distribution D itself is generally unknown, so that we cannot compute or bound (2) directly. We do, however, know the sample S which was drawn from D, and we may give a performance guarantee for A conditioned on S, but for arbitrary D. Such a *generalization error bound* is typically given by specifying a two argument function $B(\delta,S)$, where $\delta > 0$ is a confidence parameter, and the requirement that

$$\forall D, D^m\{S : R(A(S),D) \leq B(\delta,S)\} \geq 1-\delta. \tag{3}$$

The bound above states that with high probability $(1-\delta)$ in S the learning-result $A(S)$ will have risk bounded by B. Section 3 will give examples of generalization error bounds.

Meta-Learning. This paper describes a mechanism by which a sequence $\mathbf{S}=(S_1,...,S_n)$ of samples, drawn from different learning tasks $D_1,...,D_n$, can be used to improve and predict the performance of a learner on an *unknown future task*. We will give bounds analogous to (3) and also present a practical algorithm.

The crucial idea, due to J.Baxter ([2], [3]), is that the learning tasks D_i originate from an *environment* of tasks, which is a probability distribution $\mathcal{E}$ on the set of learning tasks. The encounter with a new learning task is thus modelled as a random event, a draw $D \sim \mathcal{E}$ of a task D. Subsequent to the draw of D a sample $S=(z_1,...,z_m)$ may be generated by a sequence of m *independent* draws from D. Let $\mathbf{D}_{\mathcal{E}}(S)$ be the overall probability for an m-sample S to arise in this way,

$$\mathbf{D}_{\mathcal{E}}(S) = E_{D\sim\mathcal{E}}[D^m(S)].$$

The accumulation of experience is then modelled by n *independent* draws of samples $S_i \sim \mathbf{D}_{\mathcal{E}}$, resulting in the sample-sequence or *meta-sample* $\mathbf{S}=(S_1,...,S_n)$ (also called 'support sets' by S.Thrun in [15] or (n,m)-samples in [3]). The probability for $\mathbf{S}$ to arise in this manner is $(\mathbf{D}_{\mathcal{E}})^n(\mathbf{S})$ and depends completely on the environment $\mathcal{E}$. We generally use m to denote the size of the ordinary samples and n for the size of the meta samples. We also use bold letters $\mathbf{D}$, $\mathbf{S}$, $\mathbf{l}$, etc to distinguish objects of meta-learning from the corresponding objects of ordinary learning D, S, l, etc.

A learners behaviour is formally described by a learning algorithm A. To say that the meta-sample $\boldsymbol{S}$ is used to determine the behaviour of the learner on future learning tasks can therefore be expressed in the equation

$$A = \mathbf{A}(\mathbf{S})$$

where $\mathbf{A}$ is a function which returns a learning algorithm for every meta-sample $\mathbf{S}$. The object $\mathbf{A}$ will be called a *meta-algorithm*. Since $\mathbf{A}(\mathbf{S})$ is an algorithm we can train it with a sample S to obtain a hypothesis $\mathbf{A}(\mathbf{S})(S)$.

An example of a meta-algorithm is feature-learning where $\mathbf{A}$ selects a feature map to preprocess the input of a fixed algorithm. Another example is given in section 6. In general, any method that adjusts the parameters of an algorithm on the basis of the experience made with other learning tasks can be regarded as a meta-algorithm.

To state generalization error bounds for meta-algorithms, we need to define a statistical measure of the performance of an algorithm A with respect to an environment $\mathcal{E}$, analogous to the risk $R(c,D)$ of a hypothesis c with respect to a task D. The risk (1) measures the expected loss of a hypothesis for future data drawn from the task distribution D, so the analogous quantity for an algorithm should measure the expected loss of the hypothesis returned by the algorithm for future tasks drawn from the environmental distribution $\mathcal{E}$. A corresponding experiment involves the random draw of a task D from $\mathcal{E}$, training the algorithm with a sample S drawn randomly and independently from D, and applying the resulting hypothesis to data randomly drawn from D. Formally

$$\mathbf{R}(A,\mathcal{E}) = E_{D\sim\mathcal{E}}\left[E_{S\sim D^m}\left[R(A(S),D)\right]\right] = E_{D\sim\mathcal{E}}\left[E_{S\sim D^m}\left[E_{z\sim D}\left[l(A(S),z)\right]\right]\right]. \tag{4}$$

The *transfer risk* $\mathbf{R}(A,\mathcal{E})$ measures how well the algorithm A is adapted to the environment $\mathcal{E}$. If $\mathcal{E}$ is non-uniform the NFLT doesn't apply, and we may hope to optimize $\mathbf{R}(A,\mathcal{E})$ in A.

If the environment was known, we could in principle select A so as to minimize (4), but the only available information is the past experience or meta-sample $\mathbf{S}$. The situation is analogous to ordinary learning. Now suppose that $\mathbf{A}$ is a meta algorithm. The idea is to bound $\mathbf{R}(\mathbf{A}(\mathbf{S}),\mathcal{E})$ in terms of $\mathbf{S}$ with high probability in $\mathbf{S}$, as $\mathbf{S}$ is drawn from the environment $\mathcal{E}$ *for every environment* $\mathcal{E}$. Given $\mathbf{S}$ we can then reason that, regardless of $\mathcal{E}$, the bound is

true with high probability. Formally we seek a function B such that, given a confidence parameter δ,

$$\forall \mathcal{E}, (\mathbf{D}_{\mathcal{E}})^n \{\mathbf{S} : \mathbf{R}(\mathbf{A}(\mathbf{S}), \mathcal{E}) \leq B(\delta, \mathbf{S})\} \geq 1 - \delta. \tag{5}$$

The principal contribution of this paper is a general method to prove bounds of this type for different classes of meta-algorithms.

The Method. Given an algorithm A, let $\mathbf{l}(A, S)$ be an *estimator* for the risk of $A(S)$ given the sample $S = (z_1, ..., z_m)$. For example set $\mathbf{l} = l_{emp}$ with the empirical estimator

$$l_{emp}(A, S) = \sum_{i=1}^{m} l(A(S), z_i).$$

We then write, using $E_{S \sim \mathbf{D}_{\mathcal{E}}}[f(S)] = E_{D \sim \mathcal{E}}[E_{S \sim D^m}[f(S)]]$,

$$\begin{aligned}
&\mathbf{R}(\mathbf{A}(\mathbf{S}), \mathcal{E}) \\
&= E_{S \sim \mathbf{D}_{\mathcal{E}}}[\mathbf{l}(\mathbf{A}(\mathbf{S}), S)] + E_{D \sim \mathcal{E}}[E_{S \sim D^m}[R(\mathbf{A}(\mathbf{S})(S), D) - \mathbf{l}(\mathbf{A}(\mathbf{S}), S)]] \\
&\leq E_{S \sim \mathbf{D}_{\mathcal{E}}}[\mathbf{l}(\mathbf{A}(\mathbf{S}), S)] + \sup_{D, \mathbf{S}'} |E_{S \sim D^m}[R(\mathbf{A}(\mathbf{S}')(S), D) - \mathbf{l}(\mathbf{A}(\mathbf{S}'), S)]|.
\end{aligned} \tag{6}$$

To control the first term in the last line it suffices to prove a bound of the type

$$\forall \mathbf{D} \in M_1(Z^m), \mathbf{D}^n \{\mathbf{S} : E_{S \sim \mathbf{D}}[\mathbf{l}(\mathbf{A}(\mathbf{S}), S)] \leq \Pi(\delta, \mathbf{S})\} \geq 1 - \delta, \tag{7}$$

where $\mathbf{D} \in M_1(Z^m)$ refers to any probability distribution on the set Z^m of m-samples. Notice that (7) has exactly the same structure as an ordinary generalization error bound (3) where D has been repaced with $\mathbf{D}$, S with $\mathbf{S}$, A with $\mathbf{A}$, l with $\mathbf{l}$, and B with Π. We therefore propose to use established results of learning theory to obtain the statement (7). Because it controls future values of the estimator, a two-argument function Π satisfying (7) will be called an *estimator prediction bound* for $\mathbf{A}$ with respect to the estimator $\mathbf{l}$.

The simplest case, where a nontrivial estimator prediction bound can be found, occurs when $\mathbf{A}$ searches only a finite set of algorithms, but there are many other possibilities, some are listed in section 3.

Suppose that we have established (7). To obtain (5) it will be sufficient to bound the second term in the last line of (6).

Methods for deriving ordinary generalization error bounds often use an intermediate bound on the estimation error

$$|R(A(S), D) - \mathbf{l}(A, S)|,$$

valid for all distributions with high probability in S, for example by bounding the complexity of a hypothesis space searched by A. Such bounds lead to a

general method to control the second term in (6) and to prove (5). Theorem 5 states a corresponding result, which is applied in section 5.2 to improve on the results in [3].

A second method to bound the estimation error in (6) involves the notion of *algorithmic stability*. This method is less general but more elegant and often gives tighter bounds. Bousquet and Elisseeff [4] have shown how generalization error bounds for learning algorithms can be obtained in an easy, elegant and direct way. Instead of measuring the size of the space which the algorithm searches, they concentrate directly on continuity properties of the algorithm in its dependence on the training sample. A learning algorithm is *uniformly β-stable* if the omission of a single example doesn't change the loss of the returned hypothesis by more than β, for any data point and training sample possible. Many algorithms are stable and stable algorithms have simple bounds on their estimation error. Corresponding theorems can be found in [4]. The requirement of stability has been weakened and the results have been extended by Nyogi and Kutin in [10].

If for some β and all $\mathbf{S}$ the algorithm $\mathbf{A}(\mathbf{S})$ is uniformly β-stable, then the estimation term in (6) can be bounded in a particularly simple way, namely by 2β, as stated in Theorem 6.

Results. Algorithmic stability is also useful at a different level to prove that a meta-algorithm $\mathbf{A}$ has an estimator prediction bound. This can be done by appealing to Theorem 12 in [4] (stated as Theorem 2 in section 3). The following is an immediate consequence of this theorem in combination with our Theorem 6:

Theorem 1 *Suppose the meta-algorithm $\mathbf{A}$ satisfies the following two conditions:*

1. For every meta sample $\mathbf{S}=(S_1,...,S_n)$, let $\mathbf{S}^{\setminus i}$ be the same as $\mathbf{S}$ except that one of the S_i has been deleted. Then for every $\mathbf{S}$,$\mathbf{S}^{\setminus i}$ and every ordinary sample S we have

$$\left|l_{emp}\left(\mathbf{A}\left(\mathbf{S}\right),S\right)-l_{emp}\left(\mathbf{A}\left(\mathbf{S}^{\setminus i}\right),S\right)\right|\leq\beta'.$$

2. For every ordinary sample $S=(z_1,...,z_m)$, let $S^{\setminus i}$ is the same as S except that one of the z_i has been deleted. Then for every meta sample $\mathbf{S}$ and every S and $S^{\setminus i}$ we have

$$\left|l\left(\mathbf{A}\left(\mathbf{S}\right)\left(S\right),z\right)-l\left(\mathbf{A}\left(\mathbf{S}\right)\left(S^{\setminus i}\right),z\right)\right|\leq\beta.$$

Then for every environment $\mathcal{E}$ we have, with probability greater than $1-\delta$ in the meta-sample $\mathbf{S}=(S_1,...,S_n)$ drawn from $(\mathbf{D}_{\mathcal{E}})^n$, the inequality

$$\mathbf{R}\left(\mathbf{A}\left(\mathbf{S}\right),\mathcal{E}\right)\leq\frac{1}{n}\sum_{i=1}^{n}l_{emp}\left(\mathbf{A}\left(\mathbf{S}\right),S_i\right)+2\beta'+\left(4n\beta'+M\right)\sqrt{\frac{\ln\left(1/\delta\right)}{2n}}+2\beta. \quad (8)$$

The left hand side of the last inequality measures the expected performance of the algorithm $\mathbf{A}(\mathbf{S})$ for all, and potentially yet unknown, tasks of the environment $\mathcal{E}$. The right side is composed of an empirical estimate and terms depending on the sample sizes n and m, the stability parameters β' and β and the confidence parameter δ. If $\beta' \approx 1/n^a$ and $\beta \approx 1/m^b$, with $a > 1/2$ and $b > 0$, the bound of the theorem becomes non-trivial.

We apply these results to a practical meta-algorithm for least squares regression. This meta-algorithm is related to the *Chorus of Prototypes* introduced by Edelman in [8], so we call it *CP-Regression*. CP-Regression takes the meta-sample $\mathbf{S}=(S_1, ..., S_n)$ and uses a primitive algorithm A_0 to compute a set of corresponding regression functions $h_1, ..., h_n$. For any new input object x the feature vector of x is then mixed with (or even replaced by) the vector $(h_1(x), ..., h_n(x))$. Finally $\mathbf{A}(\mathbf{S})$ is defined to be regularized least squares regression with this modified input representation. We show that Theorem 1 applies to this meta-algorithm, with $\beta' \approx 1/n$ and $\beta \approx 1/m$ as required.

CP-Regression can be implemented in practice and preliminary experiments seem to indicate that meta-learning gives a practical advantage over ordinary regularized least squares regression.

Outline of the Paper. In section 2 we give a summary of the definitions and notation used in the paper. This section is intended as a reference for the reader. In section 3 we show how to obtain estimator prediction bounds from standard results in learning theory. In section 4 we derive transfer risk bounds for meta-algorithms. In section 5 we attempt a comparison of our bounds to ordinary generalization error bounds and compare our method and results to the approach taken by J.Baxter in [3]. In section 6 we discuss regularized least squares regression, introduce CP-regression, analyse its properties and present some preliminary experimental results.

2 Definitions and Notation

This section is intended as a reference for the notation and definitions used in the paper.

Measurability. Any subset which we explicitely define on a measurable space will be assumed measurable, as will be any function. Thus for example '$F \subseteq \mathbb{R}$' is shorthand for the statement '$F \subseteq \mathbb{R}$ and F is Lebesgue-measurable'. $M_1(X)$ will always denote the space of probability measures on a measurable space X. We supply $M_1(X)$ with any σ-algebra containing the σ-algebra generated by the set of functions

$$\mu \in M_1(X) \mapsto E_{x\sim\mu}[f]$$

for all bounded measurable functions f and all singleton sets $\{\mu\}$ for $\mu \in M_1(X)$. In this way $M_1(X)$ becomes itself a measurable space and it makes sense to talk about $M_1(M_1(X))$.

Learning and Algorithms. Throughout Z will be a measurable space of *data-points* $z \in Z$, C a space of *hypotheses* or *concepts* $c \in C$ and $l : C \times Z \to [0, M]$ a *loss function*. *Samples* are polytuples $S \in \bigcup_{m=1}^{\infty} Z^m$, and *learning algorithms* are symmetric functions

$$A : \bigcup_{m=1}^{\infty} Z^m \to C.$$

Symmetry, which will be essential for our use of stability, means that for any permutation π on $\{1, ..., m\}$ and any $S \in Z^m$ we have $A(\pi(S)) = A(S)$ where $\pi(S)$ refers to the permuted sample

$$\pi(z_1, ..., z_m) = \left(z_{\pi(1)}, ..., z_{\pi(m)}\right).$$

The set of such algorithms depends only on C and Z and will be denoted by $\mathcal{A}(C, Z)$. The hypothesis $A(S)$ is what results when A is trained with S.

Learning Tasks and Risk. A *learning task* is specified by a probability measure $D \in M_1(Z)$. Given such a task D and a hypothesis $c \in C$ and a loss function l we use

$$R(c, D) = E_{z \sim D}[l(c, z)]$$

to denote the *risk* (=expected loss) of the hypothesis c in task D w.r.t. the loss function l.

Generalization Error Bounds. A function $B : (0, 1] \times \bigcup_{m=1}^{\infty} Z^m \to [0, M]$ is a *generalization error bound* for the algorithm $A \in \mathcal{A}(C, Z)$ with respect to the loss function l iff

$$\forall D \in M_1(Z), \forall \delta > 0, D^m \{S : R(A(S), D) \leq B(\delta, S)\} \geq 1 - \delta.$$

Estimators and Algorithmic Stability. The *leave-one-out estimator* l_{loo} and the *empirical estimator* l_{emp} are the functions (the notation is from [4])

$$l_{loo}, l_{emp} : \mathcal{A}(C, Z) \times (Z^m) \to [0, M]$$

defined for $A \in \mathcal{A}(C, Z)$ and $S = (z_1, ..., z_m) \in Z^m$ by

$$l_{loo}(A, S) = \frac{1}{m} \sum_{i=1}^{m} l\left(A\left(S^{\setminus i}\right), z_i\right),$$

where $S^{\setminus i}$ generally denotes the sample S with the i-th element deleted, and

$$l_{emp}(A, S) = \frac{1}{m} \sum_{i=1}^{m} l(A(S), z_i).$$

For $\beta > 0$ an algorithm $A \in \mathcal{A}(C, Z)$ is called *uniformly β-stable w.r.t. the loss function l* if

$$|l(A(S), z) - l\left(A\left(S^{\setminus i}\right), z\right)| < \beta,$$

for every m, for every $S \in Z^m$, $z \in Z$ and $i \in \{1, ..., m\}$.

Environments and Induced Distributions. A meta-learning task is specified by an *environment*

$$\mathcal{E} \in M_1(M_1(Z))$$

which models the drawing of learning tasks $D \sim \mathcal{E}$. The environment $\mathcal{E}$ defines an *induced distribution* $\mathbf{D}_{\mathcal{E}} \in M_1(Z^m)$, by

$$\mathbf{D}_{\mathcal{E}}(F) = E_{D\sim\mathcal{E}}[D^m(F)] \text{ for } F \subseteq Z^m \text{ measurable.} \tag{9}$$

The corresponding expectation for a measurable function f on Z^m is then

$$E_{S\sim\mathbf{D}_{\mathcal{E}}}[f] = E_{D\sim\mathcal{E}}[E_{S\sim D^m}[f(S)]].$$

The induced distribution $\mathbf{D}_{\mathcal{E}}$ models the probability $\mathbf{D}_{\mathcal{E}}(S)$ for an m-sample S to arise when a task D is drawn from the environment $\mathcal{E}$, followed by m independent draws of examples from the same distribution D. $\mathbf{D}_{\mathcal{E}}$ is not a product measure, but a mixture of symmetric product measures, and therefore itself symmetric. Repeated, independent draws from $\mathbf{D}_{\mathcal{E}}$ give rise to *meta-samples* (see below).

Transfer Risk. Given an environment $\mathcal{E} \in M_1(M_1(Z))$, an algorithm $A \in \mathcal{A}(C, Z)$ and a loss function $l : C \times Z \to [0, M]$ the *transfer risk* of A in the environment $\mathcal{E}$ w.r.t. the loss function l is given by

$$\mathbf{R}(A, \mathcal{E}) = E_{D\sim\mathcal{E}}[E_{S\sim D^m}[R(A(S), D)]].$$

It gives the expected risk of the hypothesis $A(S)$ for a task D randomly drawn from the environment and the sample S randomly drawn from this task. It measures how poorly the algorithm A is suited to the environment $\mathcal{E}$.

Meta-Samples and Meta-Algorithms. We use the letter $\mathbf{S}$ to denote a *meta-sample*, $\mathbf{S} = (S_1, ..., S_n) \in (Z^m)^n$. Such can be generated by a sequence of n independent draws from some distribution $\mathbf{D} \in M_1(Z^m)$, typically the distribution $\mathbf{D}_{\mathcal{E}}$ induced by an environment $\mathcal{E}$, that is $\mathbf{S} \sim (\mathbf{D}_{\mathcal{E}})^n$.

$\mathcal{A}(\mathcal{A}(C, Z), Z^m)$ is the set of *meta algorithms*. That is for $\mathbf{A} \in \mathcal{A}(\mathcal{A}(C, Z), Z^m)$ and $\mathbf{S} \in \bigcup_{n=1}^{\infty}(Z^m)^n$ the object $\mathbf{A}(\mathbf{S})$ is the algorithm $A = \mathbf{A}(\mathbf{S}) \in \mathcal{A}(C, Z)$ which results from training $\mathbf{A}$ with the meta-sample $\mathbf{S}$. Given an m-sample S, the object $\mathbf{A}(\mathbf{S})(S)$ is the hypothesis returned by the algorithm $\mathbf{A}(\mathbf{S})$, when trained with an ordinary sample S.

Estimator Prediction Bounds. A function $\Pi : (0,1] \times \bigcup_{n=1}^{\infty} (Z^m)^n \to [0,M]$ is an *estimator prediction bound* for the meta-algorithm $\mathbf{A} \in \mathcal{A}(\mathcal{A}(C,Z), Z^m)$ with respect to the estimator $\mathbf{l} : \mathcal{A}(C,Z) \times (Z^m) \to [0,M]$ iff

$$\forall \mathbf{D} \in M_1(Z^m), \forall \delta > 0, \mathbf{D}^n \{\mathbf{S} : E_{S\sim\mathbf{D}}[\mathbf{l}(\mathbf{A}(\mathbf{S}), S)] \leq \Pi(\delta, \mathbf{S})\} \geq 1 - \delta. \quad (10)$$

An estimator prediction bound is formally equivalent to an ordinary generalization bound under the identifications $Z \leftrightarrow Z^m$, $C \leftrightarrow \mathcal{A}(C,Z)$, $l \leftrightarrow \mathbf{l}$, $A \leftrightarrow \mathbf{A}$, $B \leftrightarrow \Pi$.

Meta-Estimators. Given an estimator $\mathbf{l} : \mathcal{A}(C,Z) \times (Z^m) \to [0,M]$ the *empirical meta-estimator* $\mathbf{l}_{emp}$ is the function

$$\mathbf{l}_{emp} : \mathcal{A}(\mathcal{A}(C,Z), Z^m) \times (Z^m)^n \to [0,M]$$

defined for $\mathbf{A} \in \mathcal{A}(\mathcal{A}(C,Z), Z^m)$ and $\mathbf{S} = (S_1, ..., S_n) \in (Z^m)^n$ by

$$\mathbf{l}_{emp}(\mathbf{A}, \mathbf{S}) = \frac{1}{n} \sum_{i=1}^{n} \mathbf{l}(\mathbf{A}(\mathbf{S}), S_i).$$

The meta-estimator $\mathbf{l}_{loo}$ is defined analogously. These definitions depend on the choice of the estimator $\mathbf{l}$ itself. For example if $\mathbf{l}$=l_{loo} then

$$(l_{loo})_{emp}(\mathbf{A}, \mathbf{S}) = \frac{1}{n} \sum_{i=1}^{n} l_{loo}(\mathbf{A}(\mathbf{S}), S_i).$$

The table below relates the descriptions of ordinary and meta-learning tasks.

	Ordinary learning	Meta learning
Data	$z \in Z$	$S = (z_1, ..., z_m) \in Z^m$
Samples	$S = (z_1, ..., z_m) \in Z^m$	$\mathbf{S} = (S_1, ..., S_n) \in (Z^m)^n$
Hypotheses	$c \in C$	$A \in \mathcal{A}(C,Z)$
Algorithms	$A \in \mathcal{A}(C,Z)$	$\mathbf{A} \in \mathcal{A}(\mathcal{A}(C,Z), Z^m)$
Loss function	$l : C \times Z \to [0,M]$	$\mathbf{l} : \mathcal{A}(C,Z) \times Z^m \to [0,M]$, where $\mathbf{l} = l_{emp}$ or l_{loo}
Learning Task	$D \in M_1(Z)$	$\mathbf{D} \in M_1(Z^m)$, typically $\mathbf{D} = \mathbf{D}_{\mathcal{E}}$ where $\mathbf{D}_{\mathcal{E}}$ is induced by an environment $\mathcal{E} \in M_1(M_1(Z))$ (see(9))
Empirical estimator	$l_{emp}(A,S) = \frac{1}{m}\sum_{i=1}^{m} l(A(S), z_i)$	$\mathbf{l}_{emp}(\mathbf{A},\mathbf{S}) = \frac{1}{n}\sum_{i=1}^{n} \mathbf{l}(\mathbf{A}(\mathbf{S}), S_i)$
Risk	$R(c,D) = E_{z\sim D}[l(c,z)]$	$E_{S\sim\mathbf{D}}[\mathbf{l}(A,S)]$
Bound	Generalization error	Estimator prediction

An important object which is *not* mapped is the transfer risk $\mathbf{R}(A, \mathcal{E})$. Correspondingly an estimator prediction bound is *not* a generalization error bound for the transfer risk.

Covering Numbers (these definitions are taken from [1]). Let X be a set, $X_0 \subseteq X$. For $\epsilon > 0$ and a metric d on X the covering numbers $\mathcal{N}(\epsilon, X_0, d)$ are defined by

$$\mathcal{N}(\epsilon, X_0, d) = \min\left\{N \in \mathbb{N} : \exists (x_1, ..., x_N) \in X^N, \forall x \in X_0, \exists i, d(x, x_i) \leq \epsilon\right\}.$$

For a class $\mathcal{F}$ of real functions on X and $S = (x_1, ..., x_n) \in X^n$ define $\mathcal{F} \restriction_S \subseteq \mathbb{R}^n$ by

$$\mathcal{F}|_S = \{(f(x_1), ..., f(x_n)) : f \in \mathcal{F}\},$$

and define, for $\epsilon > 0$ and any given n,

$$\mathcal{N}_1(\epsilon, \mathcal{F}, n) = \sup_{S \in X^n} \mathcal{N}(\epsilon, \mathcal{F}|_S, d_1),$$

where d_1 is the metric on $\mathbb{R}^n$ defined by

$$d_1(x, y) = \frac{1}{n} \sum_{i=1}^{n} |x_i - y_i|.$$

Loss Function Classes. Let $\mathcal{H} \subseteq C$. The *loss function class* $\mathcal{F}(\mathcal{H}, l)$ is the family of real functions

$$\mathcal{F}(\mathcal{H}, l) = \{z \in Z \mapsto l(c, Z) : c \in \mathcal{H}\}.$$

For $\mathcal{F}(\mathcal{H}, l)$ we use the topology of pointwise convergence which it inherits as a subset of $[0, M]^Z$. A set $\mathcal{H} \subseteq C$ is called *closed* if $\mathcal{F}(\mathcal{H}, l)$ is closed in this topology (and therefore also compact by Tychonoffs theorem). If $\mathcal{H}$ is closed then any finite linear combination of functions $c \in \mathcal{H} \mapsto \sum_i \alpha_i l(c, z_i)$ attains minima and maxima in $\mathcal{H}$.

For $\mathbf{H} \subseteq \mathcal{A}(C, Z)$ and a given estimator $\mathbf{l} : \mathcal{A}(C, Z) \times Z^m \to [0, M]$ we define an analogous (meta-) loss function class

$$\mathcal{F}(\mathbf{H}, \mathbf{l}) = \{S \in Z^m \mapsto \mathbf{l}(A, S) : A \in \mathbf{H}\}.$$

3 Estimator Prediction Bounds

In this section we give examples of estimator prediction bounds obtained from established results of statistical learning theory.

Selection from a Finite Set. Set the bound on the loss function M to be equal to 1 for simplicity and suppose that there is a *finite* set of hypotheses $\mathcal{H} = \{c_1, ..., c_K\} \subseteq C$. Define the algorithm A for a sample $S = (z_1, ..., z_m) \in Z^m$ by

$$A(S) = \arg\min_{c \in \mathcal{H}} \frac{1}{m} \sum_{j=1}^{m} l(c, z_j).$$

A well known application of Hoeffdings inequality and a union bound (see e.g. [1]) give, for any $\delta > 0$,

$$\forall D, D^m \left\{ S : \sup_{c \in \mathcal{H}} \left| R(c, D) - \frac{1}{m} \sum_{j=1}^{m} l(c, z_j) \right| \leq \sqrt{\frac{\ln(K/\delta)}{2m}} \right\} \geq 1 - \delta, \quad (11)$$

which gives the following generalization error bound for A:

$$\forall D \in M_1(Z), \forall \delta > 0, D^m \{ S : R(A(S), D) \leq B(\delta, S) \} \geq 1 - \delta$$

with

$$B(\delta, S) = l_{emp}(A, S) + \sqrt{\frac{\ln K + \ln(1/\delta)}{2m}}.$$

Note that this bound also holds for every algorithm searching a finite set of hypotheses of cardinality at most K, that is for every algorithm with $A(S) \in \mathcal{H}$ for all S and some set $\mathcal{H}$ with $|\mathcal{H}| \leq K$.

We now use the table at the end of the previous section. Substituting Z^m for Z, $\mathcal{A}(C, Z)$ for C, $\mathbf{l} = l_{emp}$ or $\mathbf{l} = l_{loo}$ for l and a finite set of algorithms $\{A_1, ..., A_K\}$ for $\{c_1, ..., c_K\}$, we arrive at the following statement:

Every meta algorithm $\mathbf{A}$ that such $\mathbf{A}(\mathbf{S}) \in \{A_1, ..., A_K\}$ for all $\mathbf{S} = (S_1, ..., S_n)$ has the estimator prediction bound

$$\forall \mathbf{D} \in M_1(Z^m), \forall \delta > 0, \mathbf{D}^n \{ \mathbf{S} : E_{S \sim \mathbf{D}} [\mathbf{l}(\mathbf{A}(\mathbf{S}), S)] \leq \Pi(\delta, \mathbf{S}) \} \geq 1 - \delta$$

with

$$\Pi(\delta, \mathbf{S}) = \mathbf{l}_{emp}(\mathbf{A}, \mathbf{S}) + \sqrt{\frac{\ln K + \ln(1/\delta)}{2n}}. \quad (12)$$

Selection from a Set of Bounded Complexity. Again with $M = 1$ consider a subset $\mathcal{H} \subseteq C$. It follows from the analysis in chapter 17 in [1] and Theorem 21.1 of [1], that the following holds for every $0 < \epsilon < 1$ and every distribution D on Z:

$$D^m \left\{ S \in Z^m : \forall c \in \mathcal{H}, \left| E_{z \sim D} [l(c, z)] - \frac{1}{m} \sum_{j=1}^{m} l(c, z_j) \right| \leq \epsilon \right\}$$
$$\geq 1 - 4\mathcal{N}_1 \left(\frac{\epsilon}{8}, \mathcal{F}(\mathcal{H}, l), 2m \right) e^{\frac{-\epsilon^2 m}{32}}. \quad (13)$$

which implies the following generalization error bound, valid for every algorithm A searching only the hypothesis space $\mathcal{H}$:

$$B(\delta, S) = l_{emp}(A, S) + \inf \left\{ t : 4\mathcal{N}_1 \left(\frac{t}{8}, \mathcal{F}(\mathcal{H}, l), 2m \right) e^{\frac{-t^2 m}{32}} \leq \delta \right\}. \quad (14)$$

Suppose now that $\mathbf{H} \subseteq \mathcal{A}(C, Z)$ is a space of algorithms and fix an estimator $\mathbf{l} = l_{loo}$ or $\mathbf{l} = l_{emp}$. Substituting Z^m for Z, $\mathcal{A}(C, Z)$ for C, $\mathbf{l}$ for l and $\mathbf{H}$ for $\mathcal{H}$, and $\mathcal{F}(\mathbf{H}, \mathbf{l})$ for $\mathcal{F}(\mathcal{H}, l)$ in the above, we obtain analogous to (13):

For every $0 < \epsilon < 1$ and every distribution $\mathbf{D}$ on Z^m:

$$\mathbf{D}^m \left\{ \mathbf{S} \in (Z^m)^n : \forall A \in \mathbf{H}, \left| E_{S \sim \mathbf{D}} [\mathbf{l}(A, S)] - \frac{1}{n} \sum_{j=1}^{n} \mathbf{l}(A, S_j) \right| \leq \epsilon \right\}$$
$$\geq 1 - 4\mathcal{N}_1 \left(\frac{\epsilon}{8}, \mathcal{F}(\mathbf{H}, \mathbf{l}), 2n \right) e^{\frac{-\epsilon^2 n}{32}}.$$

Every meta-algorithm $\mathbf{A}$ such $\mathbf{A}(\mathbf{S}) \in \mathbf{H}$ for all $\mathbf{S}$ has thus the estimator prediction bound

$$\Pi(\delta, \mathbf{S}) = \mathbf{l}_{emp}(\mathbf{A}, \mathbf{S}) + \inf \left\{ t : 4\mathcal{N}_1 \left(\frac{t}{8}, \mathcal{F}(\mathbf{H}, \mathbf{l}), 2n \right) e^{\frac{-t^2 n}{32}} \leq \delta \right\}. \quad (15)$$

Uniformly Stable Algorithms. Now let $M > 0$ be arbitrary. Bousquet and Elisseeff ([4]) prove that uniformly β-stable algorithms have a generalization error bound with sample-independent bound on the estimation error:

Theorem 2 *Let $A \in \mathcal{A}(C, Z)$ be uniformly β-stable. Then for any learning task $D \in M_1(Z)$ and any positive integer m, with probability greater $1 - \delta$ in a sample S drawn from D^m*

$$l(A(S), D) \leq l_{loo}(A, S) + \beta + (4m\beta + M) \sqrt{\frac{\ln \frac{1}{\delta}}{2m}}$$

and

$$l(A(S), D) \leq l_{emp}(A, S) + 2\beta + (4m\beta + M) \sqrt{\frac{\ln \frac{1}{\delta}}{2m}}.$$

These bounds are good if we can show uniform β-stability with $\beta \approx 1/m^a$, with $a > 1/2$. The notion of uniform stability easily transfers to meta-algorithms to give estimator prediction bounds. Fix an estimator $\mathbf{l} = l_{loo}$ or $\mathbf{l} = l_{emp}$ and suppose that the meta-algorithm satisfies the following condition:

For every meta sample $\mathbf{S} = (S_1, ..., S_n)$, if $\mathbf{S}'$ is the same as $\mathbf{S}$ except that one of the S_i has been deleted, and for every ordinary sample S we have

$$|\mathbf{l}(\mathbf{A}(\mathbf{S}), S) - \mathbf{l}(\mathbf{A}(\mathbf{S}'), S)| \leq \beta.$$

Theorem 2 then gives the estimator prediction bounds

$$\Pi_{loo}(\delta, \mathbf{S}) = \mathbf{l}_{loo}(\mathbf{A}, \mathbf{S}) + \beta + (4n\beta + M) \sqrt{\frac{\ln \frac{1}{\delta}}{2n}} \quad (16)$$

and

$$\Pi_{emp}(\delta, \mathbf{S}) = \mathbf{l}_{emp}(\mathbf{A}, \mathbf{S}) + 2\beta + (4n\beta + M) \sqrt{\frac{\ln \frac{1}{\delta}}{2n}}. \quad (17)$$

4 Transfer Risk Bounds for Meta Algorithms

To derive the results in this section we need the following simple lemma, which can also be found in [4].

Lemma 3 *Let $A \in \mathcal{A}(C, Z)$. Then for any learning task $D \in M_1(Z)$*

1. We have $E_{S\sim D^m}[l_{loo}(A,S)] = E_{S'\sim D^{m-1}}[R(A(S'),D)]$.

2. If A is uniformly β-stable then $|E_{S\sim D^m}[l_{emp}(A,S)] - E_{S\sim D^m}[l_{loo}(A,S)]| \leq \beta$.

Proof. Using the permutation symmetry of A and of the measure D^m we get

$$\begin{aligned} E_{S\sim D^m}[l_{loo}(A,S)] &= \frac{1}{m}\sum_{i=1}^{m} E_{S\sim D^m}\left[l\left(A\left(S^{\setminus i}\right), z_i\right)\right] \\ &= \frac{1}{m}\sum_{i=1}^{m} E_{S'\sim D^{m-1}}\left[E_{z\sim D}\left[l\left(A\left(S'\right), z\right)\right]\right] \\ &= E_{S'\sim D^{m-1}}\left[R\left(A\left(S'\right), D\right)\right]. \end{aligned}$$

Also

$$\begin{aligned} &|E_{S\sim D^m}[l_{emp}(A,S) - l_{loo}(A,S)]| \\ &\leq \frac{1}{m}\sum_{i=1}^{m}\left|E_S\left[l\left(A\left(S\right), z_i\right) - l\left(A\left(S^{\setminus i}\right), z_i\right)\right]\right| \\ &\leq \frac{1}{m}\sum_{i=1}^{m}|E_S[\beta]| = \beta. \end{aligned}$$

■

Suppose now that we have an estimator prediction bound Π for the meta-algorithm $\mathbf{A}$ with respect to the estimator $\mathbf{l}$, so that, for all $\delta > 0$,

$$\forall \mathbf{D} \in M_1(Z^m), \mathbf{D}^n\left\{\mathbf{S} : E_{S\sim\mathbf{D}}[\mathbf{l}(\mathbf{A}(\mathbf{S}), S)] \leq \Pi(\delta, \mathbf{S})\right\} \geq 1-\delta, \tag{18}$$

where the estimator $\mathbf{l} : \mathcal{A}(C,Z) \times Z^m \to [0, M]$ refers to either l_{emp} or l_{loo}. We have outlined several ways to obtain such bounds in section 3.

When $\mathbf{l} = l_{loo}$ the bound (18) is already powerful by itself. By the definition of $\mathbf{D}_{\mathcal{E}}$ and the first conclusion of Lemma 3 we have

$$\begin{aligned} E_{S\sim\mathbf{D}_{\mathcal{E}}}[l_{loo}(\mathbf{A}(\mathbf{S}), S)] &= E_{D\sim\mathcal{E}}[E_{S\sim D^m}[l_{loo}(\mathbf{A}(\mathbf{S}), S)]] \\ &= E_{D\sim\mathcal{E}}[E_{S'\sim D^{m-1}}[R(\mathbf{A}(\mathbf{S})(S'), D)]]. \end{aligned}$$

Substituting $\mathbf{D}_{\mathcal{E}}$ for $\mathbf{D}$ in (18) we conclude

Theorem 4 *If the meta-algorithm $\mathbf{A}$ satisfies the estimator prediction bound (18) with $\mathbf{l} = l_{loo}$ then for every environment $\mathcal{E}$, with probability greater than $1-\delta$ in the meta sample drawn from $(\mathbf{D}_{\mathcal{E}})^n$ we have*

$$E_{D\sim\mathcal{E}}[E_{S\sim D^{m-1}}[R(\mathbf{A}(\mathbf{S})(S), D)]] \leq \Pi(\delta, \mathbf{S}). \tag{19}$$

The left side of (19) is not quite equal to the transfer risk $\mathbf{R}(A, \mathcal{E})$. Here is a first application of this bound: Let $\{A_1, ..., A_K\}$ be a finite collection of algorithms. For any meta sample $\mathbf{S} = (S_1, ..., S_n)$ define $\mathbf{A}(\mathbf{S})$ to be

$$\mathbf{A}(\mathbf{S}) = \arg \min_{A \in \{A_1, ..., A_K\}} \frac{1}{n} \sum_{i=1}^{n} l_{loo}(A, S_i).$$

The meta-algorithm $\mathbf{A}$ selects the algorithm with the lowest leave-one-out error on average over the meta-sample. Applying the estimator prediction bound (12) for this type of algorithm in combination with (19) above then gives, for any $\mathcal{E}$ and with probability greater than $1 - \delta$ in the meta sample drawn from $(\mathbf{D}_{\mathcal{E}})^n$,

$$E_{D \sim \mathcal{E}}\left[E_{S \sim D^{m-1}}\left[R(\mathbf{A}(\mathbf{S})(S), D)\right]\right] \leq \frac{1}{n} \sum_{i=1}^{n} l_{loo}(\mathbf{A}(\mathbf{S}), S_i) + \sqrt{\frac{\ln(K/\delta)}{2n}}. \quad (20)$$

A similar result should hold if l_{loo} is replaced by any other, nearly unbiased estimator. A popular procedure, for example, is dividing the samples $S \in \mathbf{S}$ into training- and test-samples to estimate the generalization performance of an algorithm. If we chose from a finite set of candidates the algorithm $\mathbf{A}(\mathbf{S})$ which performs best on average over the test data in $\mathbf{S}$, when trained with the training data in $\mathbf{S}$, then we are implementing a version of the above meta-algorithm, and a corresponding version of (20) gives a probable performance guarantee for $\mathbf{A}(\mathbf{S})$ on future learning tasks drawn from the same environment as $\mathbf{S}$.

For more sophisticated meta-algorithms we need to consider the case $\mathbf{l} = l_{emp}$. In this case an estimator prediction bound only bounds the expected empirical error $l_{emp}(\mathbf{A}(\mathbf{S}), S)$ of $\mathbf{A}(\mathbf{S})$ for a sample S drawn from $\mathbf{D}_{\mathcal{E}}$, but it does not give any generalization guarantee for the hypothesis $\mathbf{A}(\mathbf{S})(S)$. For example $\mathbf{A}(\mathbf{S})$ could be some single-nearest-neighbour algorithm for which we would have $l_{emp}(\mathbf{A}(\mathbf{S}), S) = 0$ for almost all S, but $\mathbf{A}(\mathbf{S})$ would have poor generalization performance.

Recall the decomposition of the transfer risk (6) in the introduction:

$$\begin{aligned} &\mathbf{R}(\mathbf{A}(\mathbf{S}), \mathcal{E}) \\ &\leq E_{S \sim \mathbf{D}_{\mathcal{E}}}\left[\mathbf{l}(\mathbf{A}(\mathbf{S}), S)\right] + \sup_{D, \mathbf{S}'} \left|E_{S \sim D^m}\left[R(\mathbf{A}(\mathbf{S}'), D) - \mathbf{l}(\mathbf{A}(\mathbf{S}'), S)\right]\right|. \end{aligned}$$

The estimator prediction bound controls the first term above, so it remains to bound the second term which is independent of $\mathbf{S}$. We need to bound the expected estimation error of the estimator $\mathbf{l}$ uniformly for all distributions D and all algorithms $\mathbf{A}(\mathbf{S})$ for all meta-samples $\mathbf{S}$.

Theorem 5 *Suppose the meta-algorithm* $\mathbf{A}$ *has an estimator prediction bound* Π *with respect to the estimator* $\mathbf{l} = l_{emp}$ *, and that for every* $\eta > 0$ *there is a number* $B(\eta)$ *such that for every distribution* $D \in M_1(Z)$*, and every meta-sample* $\mathbf{S}$ *we have*

$$D^m\left\{S : |R(\mathbf{A}(\mathbf{S})(S), D) - l_{emp}(\mathbf{A}(\mathbf{S}), S)| \leq B(\eta)\right\} \geq 1 - \eta. \quad (21)$$

Let $\epsilon = \inf_\eta (B(\eta) + M\eta)$. *Then for every environment* $\mathcal{E}$, *with probability greater than* $1-\delta$ *in* $\mathbf{S}$ *as drawn from* $(\mathbf{D}_\mathcal{E})^n$ *we have*

$$\mathbf{R}(\mathbf{A}(\mathbf{S}), \mathcal{E}) \leq \Pi(\delta, \mathbf{S}) + \epsilon.$$

Proof. For any D, $\mathbf{S}$ and arbitrary η we have

$$\begin{aligned}
&E_{S\sim D^m}[R(\mathbf{A}(\mathbf{S})(S), D)] \\
&\leq E_{S\sim D^m}[l_{emp}(\mathbf{A}(\mathbf{S}), S)] + E_{S\sim D^m}[|R(\mathbf{A}(\mathbf{S})(S), D) - l_{emp}(\mathbf{A}(\mathbf{S}), S)|] \\
&\leq E_{S\sim D^m}[l_{emp}(\mathbf{A}(\mathbf{S}), S)] + B(\eta) + M\eta,
\end{aligned}$$

where (21) was used in the last inequality. Taking the expectation $D \sim \mathcal{E}$ gives

$$\begin{aligned}
\mathbf{R}(\mathbf{A}(\mathbf{S}), \mathcal{E}) &= E_{D\sim\mathcal{E}}[E_{S\sim D^m}[R(\mathbf{A}(\mathbf{S})(S), D)]] \\
&\leq E_{D\sim\mathcal{E}}[E_{S\sim D^m}[l_{emp}(\mathbf{A}(\mathbf{S}), S)]] + \epsilon \\
&= E_{S\sim\mathbf{D}_\mathcal{E}}[l_{emp}(\mathbf{A}(\mathbf{S}), S)] + \epsilon \\
&\leq \Pi(\delta, \mathbf{S}) + \epsilon,
\end{aligned}$$

where the last inequality holds with probability greater than $1-\delta$ in the meta-sample $\mathbf{S}$ as drawn from $(\mathbf{D}_\mathcal{E})^n$ by virtue of the estimator prediction bound (18) applied with $\mathbf{D}_\mathcal{E}$ in place of $\mathbf{D}$. ■

The condition (21) is often satisfied, typically with $B(\delta)$ decreasing as $\ln(1/\delta)$ in δ and as $m^{-1/2}$ in m, so we should get a bound ϵ decreasing about as quickly as $\sqrt{\ln(m)/m}$. Using the results in section 3, now on the level of ordinary learning, we see that the above theorem can be applied

- if every $\mathbf{A}(\mathbf{S})$ selects a hypothesis from a finite set $\mathcal{H}(\mathbf{S})$ of choices with $|\mathcal{H}(\mathbf{S})| \leq K$ for all $\mathbf{S}$. This follows from (11) The $\mathcal{H}(\mathbf{S})$ may of course be different for different $\mathbf{S}$..
- if every $\mathbf{A}(\mathbf{S})$ selects a hypothesis from a set $\mathcal{H}(\mathbf{S}) \subseteq C$ with uniformly bounded complexities. Here we use (13). An application is given in section 5.2.
- if every $\mathbf{A}(\mathbf{S})$ is uniformly β-stable with $\beta \approx 1/m$. This follows from Theorem 2.

In the last case we can give a much better bound, where the additional error term ϵ is often of order $1/m$:

Theorem 6 *Suppose the meta-algorithm* $\mathbf{A}$ *has an estimator prediction bound* Π *with respect to the estimator* $\mathbf{l} = l_{emp}$, *and that for some* β *the algorithms* $\mathbf{A}(\mathbf{S})$ *are uniformly* β*-stable for every meta-sample* $\mathbf{S}$. *Then for any environment* $\mathcal{E}$ *and* $\delta > 0$, *with probability greater than* $1-\delta$ *in* $\mathbf{S}$ *as drawn from* $(\mathbf{D}_\mathcal{E})^n$

$$\mathbf{R}(\mathbf{A}(\mathbf{S}), \mathcal{E}) \leq \Pi(\delta, \mathbf{S}) + 2\beta.$$

Proof. We have

$$\begin{aligned} E_{S\sim D^m}\left[R\left(\mathbf{A}\left(\mathbf{S}\right)\left(S\right),D\right)\right] &\leq E_{S'\sim D^{m-1}}\left[R\left(\mathbf{A}\left(\mathbf{S}\right)\left(S'\right),D\right)\right]+\beta \\ &= E_{S\sim D^m}\left[l_{loo}\left(\mathbf{A}\left(\mathbf{S}\right),S\right)\right]+\beta \\ &\leq E_{S\sim D^m}\left[l_{emp}\left(\mathbf{A}\left(\mathbf{S}\right),S\right)\right]+2\beta, \end{aligned}$$

where the first inequality follows directly from uniform stability and the next lines follow from Lemma 3. Taking the expectation $D\sim\mathcal{E}$ and using the estimator prediction bound (18) with $\mathbf{D}_{\mathcal{E}}$ in place of $\mathbf{D}$ gives the result in just as in the proof of the previous theorem. ■

Theorem 1 now follows immediately from Theorem 6 and from the estimator prediction bound (17) in section 3. In section 6 an application of this theorem to a practical meta-learning algorithm is discussed.

The estimator prediction bound $\Pi\left(\delta,\mathbf{S}\right)$ will typically depend on the size n of the meta-sample $\mathbf{S}=\left(S_1,...,S_n\right)$, and not on the size m of the constituting samples S_i. One may therefore wonder, how we can have an m-dependence of the estimation error as 2β (often order $1/m$), while in Theorem 2 [4] it is $2\beta+O\left(\sqrt{1/m}\right)$. The reason for this difference is that to bound the transfer-risk in the above proof we only need to bound the expectation in S of the random variable $R\left(\mathbf{A}\left(\mathbf{S}\right)\left(S\right),D\right)$, whereas the proof of Theorem 2 in [4] needs to use McDiarmid's concentration inequality to bound this random variable itself with high probability in S, which is where the $O\left(\sqrt{1/m}\right)$ term comes from.

5 Comparison to Other Results

In this section we relate our results to others, beginning with a comparison to ordinary generalization bounds. Then we compare our method to the approach taken by J.Baxter in [3] where the generalization of meta-algorithms is also studied.

5.1 Comparison to Ordinary Generalization Error Bounds

Are our results better or worse than ordinary generalization error bounds? This question is at the same time very important and very imprecise, because the two kinds of results refer to different objects and situations.

The ordinary generalization error bound (examples in section 3) applies to a situation where a sample S has already been drawn from an unknown task D and the estimator $l_{emp}\left(A,S\right)$ already has a definite value. It typically has the structure

$$\forall D,D^m\left\{S:R\left(A\left(S\right),D\right)\leq l_{emp}\left(A,S\right)+\epsilon_0\right\}\geq 1-\delta$$

where ϵ_0 is a bound on the estimation error. Often $\epsilon_0\approx\sqrt{1/m}$.

Our bounds on the other hand apply to a situation where only the meta-sample $\mathbf{S}$ is known, and typically have the structure

$$\forall \mathcal{E}, (\mathbf{D}_{\mathcal{E}})^n \{\mathbf{S} : \mathbf{R}(\mathbf{A}(\mathbf{S}), \mathcal{E}) \leq \Pi(\delta, \mathbf{S}) + \epsilon_0'\} \geq 1 - \delta$$

where $\Pi(\delta, \mathbf{S})$ is the estimator prediction bound and ϵ_0' is again a bound on the estimation error, uniformly valid for all algorithms $A = \mathbf{A}(\mathbf{S})$ for any $\mathbf{S}$.

To get ϵ_0' our method always requires some condition (uniform bounds on estimation errors, β-stability) on the algorithms $\mathbf{A}(\mathbf{S})$, which is also sufficient to prove an ordinary generalization error bound for such algorithms $\mathbf{A}(\mathbf{S})$. The corresponding estimation errors are about the same in our bounds and in the ordinary generalization error bounds. In case of Theorem 5 our ϵ_0' is slightly worse than that of the ordinary bound (i.e. $\sqrt{\ln(m)/m}$ vs $\sqrt{1/m}$), in case of Theorem 6 it is actually better (2β vs $2\beta + O\left(\sqrt{1/m}\right)$). Let's ignore these differences and put $\epsilon_0 = \epsilon_0'$. Comparing the two bounds therefore involves a comparison of the estimator prediction bound $\Pi(\delta, \mathbf{S})$ to a 'generic' value of the estimator $l_{emp}(A, S)$.

Our bound $\Pi(\delta, \mathbf{S})$ has the disadvantage that it contains an additional error of meta-estimation. But as the size n of the meta-sample $\mathbf{S}$ becomes large, corresponding to an experienced meta-learner, this additional term tends to zero, and $\Pi(\delta, \mathbf{S})$ is likely to win over the 'generic' $l_{emp}(A, S)$, because $\mathbf{A}(\mathbf{S})$ is likely to outperform the 'generic' algorithm A on the meta-sample $\mathbf{S}$. To make this precise we have to give more meaning to the word 'generic'.

While it is easy to define a generic value of S (simply taking $S \sim \mathbf{D}_{\mathcal{E}}$ if some environment $\mathcal{E}$ is given), it is not so clear how we should pick a generic algorithm A. For simplicity consider a finite set of algorithms $\{A_1, ..., A_K\}$. We should select A uniformly at random from this set to obtain a generic algorithm. The generic value of $l_{emp}(A, S)$ is then

$$\Gamma = E_{S \sim \mathbf{D}_{\mathcal{E}}} \left[\frac{1}{K} \sum_{k=1}^{K} l_{emp}(A_k, S) \right].$$

The meta algorithm to consider for comparison is

$$\mathbf{A}(\mathbf{S}) = \arg \min_{A \in \{A_1, ..., A_K\}} \frac{1}{n} \sum_{S \in \mathbf{S}} l_{emp}(A, S)$$

with the estimator prediction bound

$$\begin{aligned} E_{S \sim \mathbf{D}_Q} [l_{emp}(\mathbf{A}(\mathbf{S}), S)] &\leq \min_{k=1}^{K} \frac{1}{n} \sum_{S_i \in \mathbf{S}} l_{emp}(A_k, S_i) + \sqrt{\frac{\ln(K/\delta_M)}{2n}} \\ &= \Pi(\delta_M, \mathbf{S}), \end{aligned} \tag{22}$$

where δ_M is the confidence parameter associated with the draw of the meta-sample $\mathbf{S}$. Now let

$$\Delta(\mathbf{S}) = \frac{1}{K} \sum_{k=1}^{K} \frac{1}{n} \sum_{S \in \mathbf{S}} l_{emp}(A_k, S) - \min_{k=1}^{K} \frac{1}{n} \sum_{S \in \mathbf{S}} l_{emp}(A_k, S).$$

$\Delta(\mathbf{S})$ will be positive unless all algorithms behave the same on the meta-sample, in which case it is zero and meta-learning is indeed pointless (essentially an empirical instantiation of the NFLT). With the bound M on the loss function equal to 1, an application of Hoeffding's inequality gives, with probability greater than $1-\delta_M$ in a meta sample $\mathbf{S}$ drawn from $(\mathbf{D}_{\mathcal{E}})^n$,

$$\frac{1}{n}\sum_{S\in\mathbf{S}}\frac{1}{K}\sum_{k=1}^{K} l_{emp}(A_k,S) \leq \Gamma + \sqrt{\frac{\ln(1/\delta_M)}{2n}},$$

so with probability greater than $1-2\delta_M$ in the meta-sample $\mathbf{S}$ we have

$$\Gamma - \Pi(\delta_M,\mathbf{S}) \geq \Delta(\mathbf{S}) - \frac{\sqrt{\ln(1/\delta_M)}+\sqrt{\ln K+\ln(1/\delta_M)}}{\sqrt{2n}}, \tag{23}$$

in addition to validitiy of our bound (22). So for large meta-samples $\mathbf{S}$ our bounds will very probably be true and better than the generic value of ordinary generalization bounds by a margin of roughly $\Delta(\mathbf{S})$.

For a practical perspective consider image recognition, when the tasks in the support of $\mathcal{E}$ share a certain invariance property (say image rotation), and there is only one algorithm in $\{A_1,...,A_K\}$ having this invariance property. We can then expect the wrong algorithms to have fairly large losses for a given meta sample $\mathbf{S}$, so that $\Delta(\mathbf{S})$ will have order ≈ 1.

5.2 Comparison to the Bias Learning Model

The approach taken in [3] can be partially reformulated in our framework. We will consider only ERM-algorithms in $\mathcal{A}(C,Z)$ which have the form

$$A_{\mathcal{H}}(S) = \arg\min_{c\in\mathcal{H}} \frac{1}{m}\sum_{z_i\in S} l(c,z_i), \tag{24}$$

for some closed set $\mathcal{H}\subseteq C$ (the assumption of closure ensures existence of the minimum). Actually [3] allows any algorithm searching the set $\mathcal{H}$, such as regularised algorithms, but the analysis in [3] does not exploit the advantages of regularisation and we stick to ERM for definiteness and motivation.

The traditional method to give generalization error bounds for such algorithms is described in [1] or [16] and involves the study of the complexity of the function space $\mathcal{F}_{\mathcal{H}} = \{z\mapsto l(c,z) : c\in\mathcal{H}\}$ in terms of covering numbers or related quantities, and proceeds to prove a uniform bound on the estimation error, such as (13) in section 3, valid for all $c\in\mathcal{H}$, and with high probability in the sample S. This leads to corresponding generalization error bounds. We have sketched a version of this approach which can be applied both to ordinary and to meta algorithms in section 3.

The choice of the *hypothesis space* $\mathcal{H}$ completely defines the algorithm (24). A collection of such algorithms can therefore be viewed as a family $\mathbb{H}$ of closed

subsets $\mathcal{H} \subseteq C$ which define the algorithms $A_{\mathcal{H}}$ by virtue of formula (24). A corresponding meta-algorithm takes a meta-sample $\mathbf{S}$, sampled from an environment $\mathcal{E}$ as usual, and returns an algorithm $\mathbf{A}(\mathbf{S}) = A_{\mathcal{H}(\mathbf{S})}$ for some hypothesis space $\mathcal{H}(\mathbf{S}) \in \mathbb{H}$. The meta-algorithm can thus be equivalently considered as a map $\mathbf{S} \to \mathcal{H}(\mathbf{S})$ or

$$\mathcal{H} : \bigcup_{n=1}^{\infty} (Z^m)^n \to \mathbb{H}.$$

Such a meta-algorithm effectively *learns the hypothesis space* $\mathcal{H}(\mathbf{S})$, and in [3] it is called a *bias learner*. For the remainder of this section take $\mathbb{H}$ to be fixed and let $\mathbf{A}$ be any meta-algorithm defined by the ERM formula $\mathbf{A}(\mathbf{S}) = A_{\mathcal{H}(\mathbf{S})}$ for some map $\mathbf{S} \mapsto \mathcal{H}(\mathbf{S}) \in \mathbb{H}$. We also assume the bound M on the loss function to be equal to 1.

In our framework it is natural to study covering numbers for the space of algorithms

$$\mathbf{H}_{\mathbb{H}} = \{A_{\mathcal{H}} : \mathcal{H} \in \mathbb{H}\}$$

and use them to derive an estimator prediction bound (15) as outlined in section 3. Imposing a uniform bound on the complexities of the hypothesis spaces in $\mathbb{H}$ then allows the application of Theorem 5. Putting together the estimator prediction bound (15), the uniform bound on the estimation error (13) and Theorem 5, we arrive at

Corollary 7 *Let*

$$\epsilon_0 = \inf_{\gamma > 0} \left\{ \gamma + 4 \sup_{\mathcal{H} \in \mathbb{H}} \mathcal{N}_1 \left(\frac{\gamma}{8}, \mathcal{F}(\mathcal{H}, l), 2m \right) e^{-\gamma^2 m/32} \right\}$$

and, for $\delta > 0$*,*

$$\epsilon_1 = \inf \left\{ t : 4\mathcal{N}_1 \left(\frac{t}{8}, \mathcal{F}(\mathbf{H}, \mathbf{l}), 2n \right) e^{\frac{-t^2 n}{32}} \leq \delta \right\}.$$

Then for any environment $\mathcal{E}$*, with probability at least* $1 - \delta$ *in the draw of a meta-sample* $\mathbf{S}$ *from* $(\mathbf{D}_{\mathcal{E}})^n$*, we have*

$$\mathbf{R}\left(A_{\mathcal{H}(\mathbf{S})}, \mathcal{E}\right) \leq \frac{1}{n} \sum_{S_i \in \mathbf{S}} l_{emp}\left(A_{\mathcal{H}(\mathbf{S})}, S_i\right) + \epsilon_1 + \epsilon_0.$$

For convenience of comparison we give implicit bounds on the sample complexities, which are easily derived using $\epsilon_0 = \epsilon_1 = \epsilon/2$ and $\gamma = \epsilon/4$:

Corollary 8 *For any* $0 < \epsilon < 1$*,* $\delta > 0$*, if*

$$n \geq \frac{128}{\epsilon^2} \ln \left(\frac{4\mathcal{N}_1\left(\frac{\epsilon}{16}, \mathcal{F}(\mathbf{H}_{\mathbb{H}}, l_{emp}), 2n\right)}{\delta} \right) \tag{25}$$

and

$$m \geq \frac{512}{\epsilon^2} \ln \left(\frac{4 \sup_{\mathcal{H} \in \mathbb{H}} \mathcal{N}_1 \left(\frac{\epsilon}{32}, \mathcal{F}\left(\mathcal{H}, l\right), 2m \right)}{\epsilon} \right), \tag{26}$$

then for any environment $\mathcal{E}$ *, with probability greater than* δ *in the draw of a meta-sample* $\mathbf{S}$ *from* $\left(\mathbf{D}_{\mathcal{E}}\right)^n$*, we have*

$$\mathbf{R}\left(A_{\mathcal{H}(\mathbf{S})}, \mathcal{E}\right) \leq \frac{1}{n} \sum_{S_i \in \mathbf{S}} l_{emp}\left(A_{\mathcal{H}(\mathbf{S})}, S_i\right) + \epsilon.$$

J.Baxter in [3] also defines capacities for $\mathbb{H}$, but aims at giving a bound on

$$\sup_{\mathcal{H} \in \mathbb{H}} \left| E_{D \sim \mathcal{E}} \left[\inf_{c \in \mathcal{H}} R\left(c, D\right) \right] - \frac{1}{n} \sum_{S_i \in \mathbf{S}} l_{emp}\left(A_{\mathcal{H}}, S_i\right) \right|$$

valid with high probability in $\mathbf{S}$ as drawn from $\left(\mathbf{D}_{\mathcal{E}}\right)^n$ for any $\mathcal{E}$. A corresponding bound on

$$\mathrm{er}_{\mathcal{E}}\left(\mathcal{H}\left(\mathbf{S}\right)\right) := E_{D \sim \mathcal{E}} \left[\inf_{c \in \mathcal{H}(\mathbf{S})} R\left(c, D\right) \right] \tag{27}$$

(which in [3] is called the *generalization error of the bias learner*), results. This is Theorem 2 in [3]. The expression (27) is the expected risk of the optimal hypothesis in $\mathcal{H}\left(\mathbf{S}\right)$ as D is drawn from the environment.

The inequality

$$\begin{aligned} \mathrm{er}_{\mathcal{E}}\left(\mathcal{H}\left(\mathbf{S}\right)\right) &= E_{D \sim \mathcal{E}} \left[E_{S \sim D^m} \left[\inf_{c \in \mathcal{H}(\mathbf{S})} R\left(c, D\right) \right] \right] \\ &\leq E_{D \sim \mathcal{E}} \left[E_{S \sim D^m} \left[R\left(A_{\mathcal{H}(\mathbf{S})}\left(S\right), D \right) \right] \right] \\ &= \mathbf{R}\left(A_{\mathcal{H}(\mathbf{S})}, \mathcal{E}\right) \end{aligned} \tag{28}$$

shows that our bounds on the transfer risk also provide bounds on (27). Note however that a bound on (27) does not itself guarantee generalization, because we may not find the optimal hypothesis from a finite future sample. This is similar to the estimator prediction bounds in our approach and contrary to our bounds on the transfer risk.

In Theorem 3 of [3] the capacity of a given $\mathcal{H}$ is used to formulate a uniform bound on the estimation error of the hypotheses in $\mathcal{H}$ similar to (13). If corresponding capacity bounds held for *all* hypothesis spaces $\mathcal{H} \in \mathbb{H}$, a bound on the transfer risk $\mathbf{R}\left(A_{\mathcal{H}(\mathbf{S})}, \mathcal{E}\right)$ would result from the bound on (27) in a way parallel to our approach (in [3] a bound on the transfer risk comparable to our bounds is never stated). In this case the results become comparable and the bounds on the sample complexities look similar. This is not surprising since both derivations of bounds are rooted in the same classical method (see e.g. [16]).

The sample complexity bounds on the m-sample depending on the uniform capacity bound are then essentially the same in [3] as in (26) (if we disregard that [3] imposes additional conditions on m in Theorem 2). For a comparison we therefore focus on the sample complexity bounds on the size n of the meta-sample. In [3], Theorem 2, to get

$$\mathrm{er}_{\mathcal{E}}\left(\mathcal{H}\left(\mathbf{S}\right)\right) \leq \frac{1}{n}\sum_{S_i \in \mathbf{S}} l_{emp}\left(A_{\mathcal{H}(\mathbf{S})}, S_i\right) + \epsilon$$

with probability at least $1-\delta$ in $\mathbf{S}$, it is required that

$$n \geq \frac{256}{\epsilon^2}\ln\frac{8C\left(\frac{\epsilon}{32}, \mathbb{H}^*\right)}{\delta}, \tag{29}$$

and there is an additional condition on m.

To compare (29) with our bound (25), we disregard the constants (which are better in (25)) and concentrate on a comparison of the complexity measures $C\left(\epsilon, \mathbb{H}^*\right)$ and $\mathcal{N}_1\left(\epsilon, \mathcal{F}\left(\mathbf{H}_{\mathbb{H}}, l_{emp}\right), n\right)$.

In [3] the capacity $C\left(\epsilon, \mathbb{H}^*\right)$ is defined as follows: For $\mathcal{H} \in \mathbb{H}$ define a real function $\mathcal{H}^*$ on $M_1\left(Z\right)$ by

$$\mathcal{H}^*\left(D\right) = \inf_{c \in \mathcal{H}} R\left(c, D\right).$$

In [3] there are assumptions to guarantee that $\mathcal{H}^*$ is measurable on $M_1\left(Z\right)$, and since it is obviously bounded we have $\mathcal{H}^* \in L_1\left(M_1\left(Z\right), \mathbf{Q}\right)$ for any probability measure $\mathbf{Q} \in M_1\left(M_1\left(Z\right)\right)$. Use $d_{\mathbf{Q}}$ to denote the metric in $L_1\left(M_1\left(Z\right), \mathbf{Q}\right)$ and denote

$$\mathbb{H}^* = \left\{\mathcal{H}^* : \mathcal{H} \in \mathbb{H}\right\}.$$

Then

$$C\left(\epsilon, \mathbb{H}^*\right) = \sup_{\mathbf{Q} \in M_1\left(M_1\left(Z\right)\right)} \mathcal{N}\left(\epsilon, \mathbb{H}^*, d_{\mathbf{Q}}\right).$$

It turns out that our complexity measures are bounded by those in [3].

Proposition 9 *For all ϵ, n*

$$\mathcal{N}_1\left(\epsilon, \mathcal{F}\left(\mathbf{H}_{\mathbb{H}}, l_{emp}\right), n\right) \leq C\left(\epsilon, \mathbb{H}^*\right).$$

Proof. For a sample $S = \left(z_1, ..., z_m\right) \in Z^m$ use D_S to denote the empirical distribution $D_S \in M_1\left(Z\right)$ induced by S:

$$D_S = \frac{1}{m}\sum_{i=1}^{m} \delta_{z_i},$$

where δ_z is the unit mass concentrated at $z \in Z$. Note that for $\mathcal{H} \in \mathbb{H}$ we have

$$\mathcal{H}^*\left(D_S\right) = \inf_{c \in \mathcal{H}} \frac{1}{m}\sum_{i=1}^{m} l\left(c, z_i\right) = l_{emp}\left(A_{\mathcal{H}}, S\right).$$

For a meta-sample $\mathbf{S} = (S_1, ..., S_n) \in (Z^m)^n$ use $\mathbf{Q_S}$ to denote the empirical distribution $\mathbf{Q_S} \in M_1(M_1(Z))$ induced by $\mathbf{S}$:

$$\mathbf{Q_S} = \frac{1}{n} \sum_{i=1}^{n} \delta_{D_{S_i}},$$

where δ_D is the unit mass concentrated at $D \in M_1(Z)$.

Now take any meta-sample $\mathbf{S} = (S_1, ..., S_n) \in (Z^m)^n$ and let $N = \mathcal{N}(\epsilon, \mathbb{H}^*, d_{\mathbf{Q_S}})$. Then there is a set of functions $\{\Psi_1, ..., \Psi_N\} \subseteq L_1(M_1(Z))$ such that for every $\mathcal{H} \in \mathbb{H}$ there is some i such that

$$\begin{aligned} \epsilon &\geq d_{\mathbf{Q_S}}(\mathcal{H}^*, \Psi_i) \\ &= \frac{1}{n} \sum_{j=1}^{n} \left| \mathcal{H}^*\left(D_{S_j}\right) - \Psi_i\left(D_{S_j}\right) \right| \\ &= \frac{1}{n} \sum_{j=1}^{n} \left| l_{emp}(A_{\mathcal{H}}, S_j) - \Psi_i\left(D_{S_j}\right) \right|. \end{aligned} \tag{30}$$

On the other hand we have

$$\mathcal{F}(\mathbf{H}_{\mathbb{H}}, l_{emp})|_{\mathbf{S}} = \{(l_{emp}(A_{\mathcal{H}}, S_1), ..., l_{emp}(A_{\mathcal{H}}, S_n)) : \mathcal{H} \in \mathbb{H}\},$$

so, setting $x_i \in \mathbb{R}^n$ with $(x_i)_j = \Psi_i\left(D_{S_j}\right)$, we see from (30) that every member of $\mathcal{F}(\mathbf{H}_{\mathbb{H}}, l_{emp})|_{\mathbf{S}}$ is within d_1-distance ϵ of some x_i. It follows that

$$\mathcal{N}(\epsilon, \mathcal{F}(\mathbf{H}_{\mathbb{H}}, l_{emp})|_{\mathbf{S}}, d_1) \leq \mathcal{N}(\epsilon, \mathbb{H}^*, d_{\mathbf{Q_S}}),$$

whence

$$\begin{aligned} \mathcal{N}_1(\epsilon, \mathcal{F}(\mathbf{H}_{\mathbb{H}}, l_{emp}), n) &= \sup_{\mathbf{S} \in (Z^m)^n} \mathcal{N}(\epsilon, \mathcal{F}(\mathbf{H}_{\mathbb{H}}, l_{emp})|_{\mathbf{S}}, d_1) \\ &\leq \sup_{\mathbf{S} \in (Z^m)^n} \mathcal{N}(\epsilon, \mathbb{H}^*, d_{\mathbf{Q_S}}) \\ &\leq \sup_{\mathbf{Q} \in M_1(M_1(Z))} \mathcal{N}(\epsilon, \mathbb{H}^*, d_{\mathbf{Q}}) \\ &= \mathcal{C}(\epsilon, \mathbb{H}^*) \end{aligned}$$

■

We can conclude that our bounds are normally applicable when those in [3] are. It may however happen, that our covering numbers increase polynomially in n, in which case we still get tight bounds, but the capacities in [3] are infinite.

6 A Meta-Algorithm for Regression

In this section we present a meta-learning algorithm for function estimation. The algorithm is based on *regularized least-squares regression*, or *ridge regression* (as in [4] or [6]) and preliminary experiments appear promising.

To implicitely also define a 'kernelised' version of the algorithm, we describe it in a setting where the *input space* is a subset $\mathcal{X}$ of the unit ball $\{\|x\| \leq 1\}$ in a separable, possibly infinite dimensional Hilbert space H, with an appropriately defined inner product.

The *output space* $\mathcal{Y}$ is the interval $[0,1]$, the data space Z is given by $Z = \mathcal{X} \times \mathcal{Y} \subseteq \{\|x\| \leq 1\} \times [0,1]$ and a learning task is given by a distribution $D \in M_1(\mathcal{X} \times \mathcal{Y})$. Then $D(x,y)$ is interpreted as the probability of finding the input value x associated with the output value y in the context of the task D.

As a hypothesis or concept space we consider the bounded linear functionals h on H which can be identified with members $h \in H$ via the action of the inner product $h(x) = \langle h, x \rangle$ in H.

As a loss function we use $l : H \times Z \to \mathbb{R}_+$ given by

$$l(h,(x,y)) = (\langle h, x \rangle - y)^2 .$$

This loss function is unbounded contrary to what is generally required in this paper. It will however turn out that the effective hypothesis space searched by the algorithms in this section is the ball $\left\{\|h\| \leq \lambda^{-1/2}\right\}$ where λ is the regularization parameter introduced below.

6.1 Regularized least squares Regression

A standard algorithm $A \in A(H,Z)$ for this type of problem is defined as follows: Let $S = (z_1,...,z_m) = ((x_1,y_1),...,(x_m,y_m)) \in Z^m$ be a sample. We write, for $h \in H$,

$$L(h) = \frac{1}{m}\sum_{i=1}^{m} (\langle h, x_i \rangle - y_i)^2 + \lambda \|h\|^2$$

and define

$$A(S) = \arg\min_{h \in H} L(h) . \tag{31}$$

Note that $\lambda \|A(S)\|^2 \leq L(A(S)) \leq L(0) \leq 1$ so $\|A(S)\| \leq \lambda^{-1/2}$. The effective hypothesis space is then $\left\{\|h\| \leq \lambda^{-1/2}\right\}$, as claimed above. Thus $|\langle h, x \rangle| \leq \lambda^{-1/2}$ and the loss function is bounded by λ^{-1}.

Any component of h perpendicular to all the x_i will only increase L, so we may assume that $A(S)$ is in the subspace generated by $\{x_1,...,x_m\}$, in other words

$$A(S) = \sum_{i=1}^{m} \alpha_i x_i \tag{32}$$

for some (possibly non-unique) vector $\alpha \in \mathbb{R}^m$. To find α we substitute (32) in L and equate the gradient to zero. The result of this well known computation is the formula

$$(G + m\lambda I)\alpha = y \tag{33}$$

where $G_{ij} = \langle x_i, x_j \rangle$ is the *Gramian matrix*, here considered as an operator on $\mathbb{R}^m$, $I = \delta_{ij}$ is the identity, and $y = (y_1,...,y_m)$ the set of target values in the

sample, here considered as a vector $y \in \mathbb{R}^m$. Equation (33) can be efficiently solved for α using the Cholesky decomposition method. The formula for the empirical loss of $A(S)$ is, using (32) and (33)

$$\begin{aligned} l_{emp}(A,S) &= \frac{1}{m}\sum_{i=1}^{m}\left((G\alpha)_i - y_i\right)^2 \\ &= \frac{1}{m}\sum_{i=1}^{m}\left(\left(\left(G+m\lambda I\right)\alpha\right)_i - y_i - m\lambda\alpha_i\right)^2 \\ &= \frac{1}{m}\sum_{i=1}^{m}\left(-m\lambda\alpha_i\right)^2 \\ &= m\lambda^2\sum_{i=1}^{m}\alpha_i^2. \end{aligned} \tag{34}$$

It follows from example 3 in [4] that the algorithm A so defined is β-stable with $\beta = 2/(\lambda m)$.

6.2 A Meta-Algorithm

Consider now a meta sample $\mathbf{S} = (S_1, ..., S_n)$, drawn from $(\mathbf{D}_{\mathcal{E}})^n$ for some environment $\mathcal{E}$, and suppose that we have used some 'primer' algorithm A_0 (for example the regression algorithm above for an appropriate value of $\lambda = \lambda_0$) to train corresponding regression functions $h_k = A_0(S_k) \in H$. The sequence of vectors $(A_0(S_1), ..., A_0(S_n)) = (h_1, ..., h_n)$ in some way contains our experiences with the environment $\mathcal{E}$. The idea of the meta-algorithm is now to use the h_k as *additional features* to describe a given new data-point x. We do this by combining the n-dimensional vector $(h_1(x), ..., h_n(x))$ with the existing description $x \in H$.

The intuitive motivation is that we expect the h_i to already describe relevant properties (symmetries, elimination of irrelevant features) of the environment, that we rely on, in particular if the sample-sizes are rather small. Imagine the classification (by thresholding of a regression functions) of character-images of a new character set, say the greek characters, after having learnt other character sets (roman, gothic etc). We could attempt to describe the image of the character α by saying that 'it looks *a little bit like* an x and *a lot like* an $\mathbf{a}$, but rather *unlike* an l'. On the basis of this description a person might recognize the character α, without any previous *visual* training data for α.

The terms *a little bit like*, *a lot like* and *unlike* are quantifications given by previously learnt regression functions for x, $\mathbf{a}$ and l, which may already have a certain robustness relative to deformations, changes in scaling or variations in line thickness. If the sample-size m is large we can derive such robustness more directly and reliably from the training data for α itself, but for a very small sample-size we expect the new features to be helpful. The whole idea is strongly related to the *Chorus of Prototypes* introduced by Edelman in [8], so we will call our algorithm *CP-Regression*.

To formally define the algorithm, consider a 'primer' algorithm $A_0 \in \mathcal{A}(H, Z)$ such that $\|A_0(S)\| \leq \kappa$ for all $S \in Z^m$. For example we could take for A_0 the regularized least squares regression, as defined above, with a regularization parameter λ_0, in which case we would have $\kappa = \lambda_0^{-1/2}$. Fix a mixture parameter $\mu \in [0, 1]$ which will be used to interpolate between the old and the new features and a regularization parameter $\lambda > 0$.

Now let the meta-sample $\mathbf{S} = (S_1, ..., S_n)$ be given. We have to define an algorithm $\mathbf{A}(\mathbf{S}) \in \mathcal{A}(H, Z)$. On the vectorspace H we define a new inner product $\langle ., . \rangle_{\mathbf{S}}$ by

$$\langle x_1, x_2 \rangle_{\mathbf{S}} = (1 - \mu) \langle x_1, x_2 \rangle + \frac{\mu}{\kappa^2 n} \sum_{k=1}^{n} \langle A_0(S_k), x_1 \rangle \langle A_0(S_k), x_2 \rangle, \tag{35}$$

which is positive definite for $0 \leq \mu < 1$ (in the case $\mu = 1$ we can use a quotient construction to replace H, which then becomes n'-dimensional with $n' \leq n$). We will use $\|.\|_{\mathbf{S}}$ to denote the norm corresponding to $\langle ., . \rangle_{\mathbf{S}}$.

Let $S \in Z^n$ be any sample, $S = (z_1, ..., z_m) = ((x_1, y_1), ..., (x_m, y_m))$ with $x_i \in H$, $\|x_i\| \leq 1$, $y_i \in [0, 1]$. We define

$$\mathbf{A}(\mathbf{S})(S) = \arg\min_{h \in H} \frac{1}{m} \sum_{i=1}^{m} (\langle h, x_i \rangle_{\mathbf{S}} - y_i)^2 + \lambda \|h\|_{\mathbf{S}}^2$$

and the corresponding regression function

$$\mathbf{A}(\mathbf{S})(S)(x) = \langle \mathbf{A}(\mathbf{S})(S), x \rangle_{\mathbf{S}}.$$

Note that

$$\begin{aligned} \|x\|_{\mathbf{S}}^2 &= (1 - \mu) \|x\|^2 + \frac{\mu}{\kappa^2 n} \sum_{k=1}^{n} \langle A_0(S_k), x \rangle^2 \\ &\leq (1 - \mu) \|x\|^2 + \frac{\mu}{\kappa^2 n} \sum_{i=1}^{n} \kappa^2 \|x\|^2 = \|x\|^2, \end{aligned}$$

so $\mathcal{X} \subseteq \{\|x\|_{\mathbf{S}} \leq 1\}$. Therefore $\mathbf{A}(\mathbf{S})$ is ordinary regularised least squares regression with the modified inner product $\langle ., . \rangle_{\mathbf{S}}$. It follows from the analysis in [4] that the algorithms $\mathbf{A}(\mathbf{S})$ are uniformly β-stable with $\beta = 2/(m\lambda)$, for every meta-sample $\mathbf{S}$, with respect to the square loss function we use.

The implementation of $\mathbf{A}$ is straightforward: Given $\mathbf{S} = (S_1, ..., S_n)$ one computes the vectors $h_k = A_0(S_k)$. Now for any new m-sample S the Gramian

$$(G_{\mathbf{S}})_{ij} = \langle x_i, x_j \rangle_{\mathbf{S}} = (1 - \mu) \langle x_i, x_j \rangle + \frac{\mu}{\kappa^2 n} \sum_{k=1}^{n} \langle h_k, x_i \rangle \langle h_k, x_2 \rangle$$

is determined, and the equation $(G_{\mathbf{S}} + m\lambda I)\,\alpha = y$ is solved for α using Cholesky decomposition. We then get the regression function

$$\begin{aligned} x &\mapsto \sum_{i=1}^{m} \alpha_i \langle x_i, x\rangle_{\mathbf{S}} = \\ &= (1-\mu)\sum_{i=1}^{m} \alpha_i \langle x_i, x\rangle + \mu \sum_{k=1}^{n} \gamma_k \langle h_k, x\rangle \end{aligned}$$

with

$$\gamma_k = \frac{1}{\kappa^2 n}\sum_{i=1}^{m} \alpha_i \langle h_k, x_i\rangle .$$

In a nonlinear case, when the inner product in H is defined by a complicated kernel, this regression function may be cumbersome to compute since all the computations of $\langle h_k, x\rangle$ will each again involve m computations of the kernel. Also the entire meta-sample $\mathbf{S}$ has then to be present in memory. In a linear case, when the vectorspace operations in H can be performed explicitly, the computational burden is significantly reduced to the computation of a single inner product $\langle h, x\rangle$ of x with the vector

$$h = (1-\mu)\sum_{i=1}^{m} \alpha_i x_i + \mu \sum_{k=1}^{n} \gamma_k h_k$$

which is determined once during training.

6.3 Analysis of CP-Regression

As already noted the algorithms $\mathbf{A}(\mathbf{S})$ are uniformly β-stable with $\beta = 2/(m\lambda)$, for every meta-sample $\mathbf{S}$, with respect to square loss. This gives condition 2 for the application of Theorem 1.

The first condition, essential for the estimator prediction bound, is satisfied by virtue of the following proposition which is proven in the next subsection:

Corollary 10 *The algorithm* $\mathbf{A}$ *is uniformly* β'*-stable w.r.t.* l_{emp} *in the sense that, if* $\mathbf{S} = (S_1, ., S_k, .., S_n)$ *is a meta sample and* $\mathbf{S}' = (S_1, ., S_{k-1}, S_{k+1}.., S_n)$ *is the same as* $\mathbf{S}$*, with only some* S_k *deleted, then*

$$|l_{emp}(\mathbf{A}(\mathbf{S}), S) - l_{emp}(\mathbf{A}(\mathbf{S}'), S)| \leq \beta'$$

for every sample $S \in Z^m$*, with*

$$\beta' = \frac{4\mu}{\lambda(n-1)}.$$

Substitution in Theorem 1 gives, for every environment $\mathcal{E}$ with probability at least $1-\delta$ in a meta-sample $\mathbf{S}$ drawn from $(\mathbf{D}_{\mathcal{E}})^n$,

$$\mathbf{R}\left(\mathbf{A}\left(\mathbf{S}\right),\mathcal{E}\right) \leq \frac{1}{n}\sum_{S_i\in\mathbf{S}} l_{emp}\left(\mathbf{A}\left(\mathbf{S}\right),S_i\right) + \\ + \frac{8\mu}{\lambda\left(n-1\right)} + \left(\frac{16\mu n}{\lambda\left(n-1\right)}+\frac{1}{\lambda}\right)\sqrt{\frac{\ln\left(1/\delta\right)}{2n}} + \frac{4}{m\lambda}. \tag{36}$$

The bound gives a performance guarantee of the algorithm applied to future tasks on the basis of the empirical term

$$\left(l_{emp}\right)_{emp}\left(\mathbf{A},\mathbf{S}\right) = \frac{1}{n}\sum_{S_i\in\mathbf{S}} l_{emp}\left(\mathbf{A}\left(\mathbf{S}\right),S_i\right). \tag{37}$$

If $\mu = 0$, corresponding to no meta-learning at all, the bound (36) becomes more attractive to look at, but we expect the empirical term to be larger. For small n it is better to take small μ, while for very large values of n the value of μ which results in the smallest empirical term is best. It is tempting to minimize the bound with respect to μ. Unfortunately (36) applies only if the parameters of $\mathbf{A}$ have been fixed in advance, it does not justify the selection of the parameters λ, μ or the choice of the primer algorithm A_0 which enters the bound only indirectly through the term (37)). Although this problem can be partially eliminated (see the method of sieves as used in [1]), it remains a major weakness of our algorithm. A more principled approach would involve the direct minimization of

$$\frac{1}{n}\sum_{S_i\in\mathbf{S}} l_{emp}\left(A,S_i\right) + N\left(A\right)$$

where $N\left(A\right)$ would be some meta-regularizer. Our algorithm attempts to decrease the quantity (37) only indirectly by the passage to (presumably) more reliable features.

6.4 Stability of CP-Regression

In this subsection we prove Proposition 10. For a bounded operator T on a real Hilbert space H we use $\left\|T\right\|_{\infty}$ to denote its operator norm

$$\left\|T\right\|_{\infty} = \sup_{\left\|x\right\|\leq 1}\left\|Tx\right\| = \sup_{\left\|x\right\|,\left\|y\right\|\leq 1}\left|\left\langle Tx,y\right\rangle\right|$$

and use T^t, $Ker\left(T\right)$ and $Ran\left(T\right)$ to denote its transpose, nullspace and range respectively. A symmetric operator satisfies $\left\langle Tx,y\right\rangle = \left\langle x,Ty\right\rangle$ for all x and y (i.e. $T = T^t$), and a positive operator is a symmetric operator also satisfying $\left\langle Tx,x\right\rangle \geq 0$ for all x.

Lemma 11 *Let G_1 and G_2 be positive operators and $\lambda > 0$. Then*

1. $G_i + \lambda I$ *is invertible,*
2. $\left\|(G_i + \lambda I)^{-1}\right\|_\infty \leq 1/\lambda$ *and*
3. we have

$$\left\|(G_1 + \lambda I)^{-1} - (G_2 + \lambda I)^{-1}\right\|_\infty \leq \frac{1}{\lambda^2} \|G_1 - G_2\|_\infty .$$

4. Let x_1 *and* x_2 *satisfy* $(G_i + \lambda I) x_i = y$*. Then*

$$\left|\|x_1\|^2 - \|x_2\|^2\right| \leq 2\lambda^{-3} \|G_1 - G_2\|_\infty \|y\|^2 .$$

Proof. 1. If $(G_i + \lambda I) x = 0$ then $-\lambda \|x\| = \langle G_i x, x\rangle \geq 0$ so $x = 0$. Thus $G_i + \lambda I$ is 1-1, and since $Ran\,(G_i + \lambda I) = Ran\left((G_i + \lambda I)^t\right) = Ker\,(G_i + \lambda I)^\perp = \{0\}^\perp$ it is also onto.

2. Suppose $(G_i + \lambda I) x = y$. Then

$$\begin{aligned} \lambda^2 \|x\|^2 &= \|y - G_i x\|^2 = \|y\|^2 - 2\langle G_i x, y\rangle + \|G_i x\|^2 \\ &= \|y\|^2 - 2\langle G_i x, G_i x + \lambda x\rangle + \|G_i x\|^2 \\ &= \|y\|^2 - \|G_i x\|^2 - 2\lambda \langle x, G_i x\rangle \leq \|y\|^2 , \end{aligned}$$

which proves the second conclusion.

3. We have

$$\begin{aligned} &\left((G_1 + \lambda I)^{-1} - (G_2 + \lambda I)^{-1}\right)(G_2 + \lambda I) \\ &= (G_1 + \lambda I)^{-1}(G_1 + \lambda I + G_2 - G_1) - (G_2 + \lambda I)^{-1}(G_2 + \lambda I) \\ &= (G_1 + \lambda I)^{-1}(G_2 - G_1) , \end{aligned}$$

so, using the second conclusion,

$$\begin{aligned} &\left\|(G_1 + \lambda I)^{-1} - (G_2 + \lambda I)^{-1}\right\|_\infty \\ &= \left\|(G_1 + \lambda I)^{-1}(G_2 - G_1)(G_2 + \lambda I)^{-1}\right\|_\infty \\ &\leq \left\|(G_1 + \lambda I)^{-1}\right\|_\infty \|G_2 - G_1\|_\infty \left\|(G_2 + \lambda I)^{-1}\right\|_\infty \\ &\leq \lambda^{-2} \|G_1 - G_2\|_\infty . \end{aligned}$$

Finally, using the first three conclusions, if $x_i = (G_i + \lambda I)^{-1} y$, then

$$\begin{aligned} \left|\|x_1\|^2 - \|x_2\|^2\right| &= |\langle x_1 + x_2, x_1 - x_2\rangle| \\ &\leq (\|x_1\| + \|x_2\|)\|x_1 - x_2\| \\ &\leq \left(2\lambda^{-1}\|y\|\right)\left(\lambda^{-2}\|G_1 - G_2\|_\infty \|y\|\right) . \end{aligned}$$

■

Proof. of Proposition 10. Suppose $\mathbf{S} = (S_1, ., S_{k_0}, .., S_n)$ is a meta sample and that $\mathbf{S}' = (S_1, ., S_{k_0-1}, S_{k_0+1}.., S_n)$ is the same as $\mathbf{S}$, with only some S_{k_0} deleted. We have to show that

$$|l_{emp}(\mathbf{A}(\mathbf{S}), S) - l_{emp}(\mathbf{A}(\mathbf{S}'), S)| \leq \frac{4\mu}{\lambda(n-1)}$$

for every sample $S = (z_1, ..., z_m) = ((x_1, y_1), ..., (x_m, y_m)) \in Z^m$.

Let G and G' be the gramian matrices arising from the vectors x_i and the inner products $\langle ., . \rangle_{\mathbf{S}}$ and $\langle ., . \rangle_{\mathbf{S}'}$ respectively, that is

$$G_{ij} = \langle x_i, x_j \rangle_{\mathbf{S}} \text{ and } G'_{ij} = \langle x_i, x_j \rangle_{\mathbf{S}'}.$$

We regard G and G' as operators on $\mathbb{R}^m$ and use $\|.\|_m$ and $\langle ., . \rangle_m$ for the canonical norm and inner product in $\mathbb{R}^m$ respectively.

We have, using (35) and denoting $h_k = A_0(S_k)$,

$$G_{ij} - G'_{ij} = \frac{-\mu}{\kappa^2 n(n-1)} \sum_{k \neq k_0} \langle h_k, x_i \rangle \langle h_k, x_j \rangle + \frac{\mu}{\kappa^2 n} \langle h_{k_0}, x_i \rangle \langle h_{k_0}, x_j \rangle$$

so, if η and γ are any two unit vectors in $\mathbb{R}^m$, we have, with $v = \sum_{i=1}^m \eta_i x_i$ and $w = \sum_{i=1}^m \gamma_j x_j$,

$$\begin{aligned} |\langle (G - G')\eta, \gamma \rangle_m| &= \frac{-\mu}{\kappa^2 n(n-1)} \sum_{k \neq k_0} \langle h_k, v \rangle \langle h_k, w \rangle + \frac{\mu}{\kappa^2 n} \langle h_{k_0}, v \rangle \langle h_{k_0}, w \rangle \\ &\leq \frac{\mu}{\kappa^2 n(n-1)} \sum_{k \neq k_0} \|h_k\|^2 \|v\| \|w\| + \frac{\mu}{\kappa^2 n} \|h_{k_0}\|^2 \|v\| \|w\| \\ &\leq \frac{2\mu}{n-1} \|v\| \|w\| \end{aligned}$$

Now using the triangle and Cauchy Schwarz inequalities

$$\|v\| = \left\| \sum_{i=1}^m \eta_i x_i \right\| \leq \sum_{i=1}^m |\eta_i| \|x_i\| \leq \|\eta\|_m \left(\sum_{i=1}^m \|x_i\|^2 \right)^{1/2} \leq m^{1/2}$$

and similarly

$$\|w\| \leq m^{1/2},$$

so that $|\langle (G - G')\eta, \gamma \rangle_m| \leq (2\mu m)/(n-1)$. Since η and γ were arbitrary unit vectors we have

$$\|G - G'\|_\infty \leq \frac{2\mu m}{n-1}. \tag{38}$$

Now if α and α' are vectors in $\mathbb{R}^m$ which are solutions of $(G - m\lambda I)\alpha = y$ and $(G' - m\lambda I)\alpha' = y$ respectively, and $y \in \mathbb{R}^m$ is a vector with $|y_i| \leq 1$, then, using the last conclusion of Lemma 11 together with (38),

$$\begin{aligned} \left| \|\alpha\|_m^2 - \|\alpha'\|_m^2 \right| &\leq 2(m\lambda)^{-3} \|G - G'\|_\infty \|y\|_m^2 \\ &\leq 4m^{-2}\lambda^{-3}\mu \|y\|_m^2 / (n-1) \\ &\leq 4m^{-1}\lambda^{-3}\mu / (n-1). \end{aligned}$$

Using the formula (34) for the empirical error in regularised least squares regression then gives

$$\begin{aligned} \left|l_{emp}\left(\mathbf{A}\left(\mathbf{S}\right),S\right)-l_{emp}\left(\mathbf{A}\left(\mathbf{S}'\right),S\right)\right| &= m\lambda^2\left|\left\|\alpha\right\|_m^2-\left\|\alpha'\right\|_m^2\right| \\ &\leq \frac{4\mu}{\lambda\left(n-1\right)}. \end{aligned}$$

■

7 Conclusion

We have employed established analytical tools of statistical learning theory to analyze transfer learning. The notion of uniform algorithmic stability has proven to be particularly useful. Many interesting problems remain, of which we mention only two:

1. The unnatural requirement, that all sample-sizes be equal to the meta-learner, should be eliminated.

2. CP-Regression could be implemented and more systematically tested with a nonlinear kernel.

References

[1] M.Anthony, P.Bartlett, *Learning in Neural Networks: Theoretical Foundations*, Cambridge University Press 1999

[2] J.Baxter, Theoretical Models of Learning to Learn, in *Learning to Learn*, S.Thrun, L.Pratt Eds. Springer 1998

[3] J.Baxter, A Model of Inductive Bias Learning, *Journal of Artificial Intelligence Research* 12: 149-198, 2000

[4] O.Bousquet, A. Elisseeff, "Stability and Generalization", *Journal of Machine Learning Research*, 2: 499-526, 2002.

[5] R.Caruana, Multitask Learning, in *Learning to Learn*, S.Thrun, L.Pratt Eds. Springer 1998

[6] N.Christianini, J.S. Taylor, *Support Vector Machines*, Cambridge University Press 2000

[7] Luc Devroye, László Györfi, Gábor Lugosi, *A Probabilistic Theory of Pattern Recognition*. Springer, 1996.

[8] S.Edelman, Representation, similarity and the chorus of prototypes. *Minds and Machines*, 45-68, 1995

[9] Wassily Hoeffding, "Probability inequalities for sums of bounded random variables", *Journal of the American Statistical Association*, 58:13-30, 1963.

[10] S.Kutin, Partha Niyogi, Almost-everywhere algorithmic stability and generalization performance, Technical report , Department of Computer Science, University of Chicago, 2002.

[11] David McAllester, "Some PAC-Bayesian Theorems", *Proceedings of the Eleventh Annual Conference In Computational Learning Theory*, 230-234, 1998.

[12] Colin McDiarmid, "Concentration", in *Probabilistic Methods of Algorithmic Discrete Mathematics*, p. 195-248. Springer, Berlin, 1998.

[13] A.Robins, Transfer in Congnition, in *Learning to Learn*, S.Thrun, L.Pratt Eds. Springer 1998

[14] S.Thrun, *Explanation-Based Neural Network Learning*, Kluwer 1996

[15] S.Thrun, Lifelong Learning Algorithms, in *Learning to Learn*, S.Thrun, L.Pratt Eds. Springer 1998

[16] V.Vapnik, *The Nature of Statistical Learning Theory*, Springer 1995

[17] D.H.Wolpert, *The Mathematics of Generalization*, Addison Wesley, 1995