STAGED LEARNING

JOHN LANGFORD

ABSTRACT. There are many examples of machine learning problems which consist of hierarchically related subproblems. Given that this extra labeling information is available, how can we take advantage of it? The Staged Learning model allows learning on hierarchies of related tasks of arbitrary depth which can be understood as a generalization of Jon Baxter's [1] Learning to Learn model. Upper bounds on the number of examples, tasks, etc... are presented and proved. In the process, a new, more flexible bound is given for the Learning to Learn model. All of these bounds are accomplished by a new technique of mapping metalearning problems down onto learning problems. This is made possible by moving from consideration of the hypothesis space to consideration of the hypothesis loss space.

1. Introduction

The basic PAC model lays out a theoretical framework for machine learning which has been qualitatively useful. Unfortunately, there are many forms of a priori information about a particular problem which cannot be inserted easily into the PAC framework. In the PAC framework, every learning problem is approached *tabula rasa* - without incorporating information from prior learning problems into the learning of new learning problems. By removing the *tabula rasa* aspect of PAC learning and allowing information from other tasks to aid in the learning of the current task, empirical performance improvement has been shown [4][9]. The improved empirical performance can be understood as stemming from a refined, stronger, bias for learning implying fewer needed examples per task. Jon Baxter worked through the details of a proof that fewer examples per task are needed when the tasks are "related"[1].

Several people have encountered hierarchical learning problems such as http or usenet posting[7] classification in which a hierarchy of related tasks exists naturally. Baxter's work can't use all of the information contained in a hierarchy of related tasks with a depth greater than 2. This paper provides a generalization of Baxter's work to arbitrary depth hierarchies, allowing a theoretical analysis of hierarchies of related tasks. In the process, we give a more flexible proof of Baxter's Learning to Learn results.

The Staged Learning model will be defined formally later. Informally, the model consists of examples, tasks, groups of tasks, etc... with each element drawn according to some unknown distribution.

1.1. An example: Character recognition.

The character recognition problem naturally falls into a relatively deep hierarchy which can be understood by considering the set of subproblems or the set of invariants which each level of the hierarchy introduces.

- all character recognition problems
- all English character recognition problems
- all handwritten English character recognition problems
- Recognizing handwritten English letters by John Langford
- Recognizing the letter 'a' handwritten by John Langford

At each of these level certain invariants would hopefully be picked out.

- scale invariance, location invariance, curve detection
- No more than 3 vertical or horizontal lines. No more than 2 curves.
- Start and stop point for each curve.
- difficult to state. I suspect invariants exist because different characters in my handwriting seem 'similar'.
- Exact task

The Staged Learning model suggests the use of chained function approximators, one for each level in the hierarchy providing some theoretical motivation for neural networks of a depth greater than 2. Some current character recognition efforts [3] use a deep neural net as suggested by this model.

1.2. Main results.

The contributions of this paper are:

1. A bound relevant to the Multitask Learning [4] setting
2. A technique for mapping metalearning problems onto learning problems using the associated loss space.
3. Improvements in the flexibility of bounds in Baxter's Learning to Learn model
4. A model for Staged Learning
5. Upper bounds for the sample complexity (in number of examples, tasks, etc...) of learning in the Staged Learning model

The Multitask Learning bound provides a more flexible bound then could previously be applied to the Multitask Learning setting. The technique for mapping metalearning problems onto learning problems simplifies the proof of Learning to Learn and Staged Learning bounds.

The improved flexibility of bounds for the Learning to Learn model simply allow a tighter application of the Learning to Learn model by trading off the number of task and number of examples/task.

The Staged Learning model is applicable to several real world learning problems involve hierarchies of related tasks. A model for such situations leads to an improved understanding of when learning can occur. In addition, the Staged Learning model provides some theoretical justification for the use of deep (depth greater than 2) neural nets which have sometimes been found useful experimentally for learning multiple tasks.

The upper bounds for the Staged Learning model are significant for research in learning on hierarchies. Like most distribution independent statistical bounds, the results are not tight enough for quantitative use. However, the bounds do provide a useful qualitative understanding of what is necessary for learning to occur within the model.

1.3. **Paper organization.** The results will be introduced incrementally. In section 2, I will introduce a generalized version of a theorem from Baxter [1] which will be used in all the results. In section 3, I will reprove Baxter's Learning to Learn results with a more flexible bound. Since the Staged Learning with a depth of 2 is Learning to Learn, I will next introduce the full Staged Learning model and prove a convergence theorem in the model. In section 5, I will apply the bound to a hierarchical learning problem and discuss other settings in which the bound is applicable. Section 6 will discuss open work and future directions of research. Appendix A contains a reference table for all of the notation used. Appendix B contains the details for the proof of the theorem introduced in section 2.

2. Preliminaries

2.1. **PAC framework.** The notation used is slightly different from standard PAC notation. This is necessary because there is no easy generalization of the standard notation to a hierarchical model. After introducing the notation, a generalization of Haussler's[5] sample complexity bounds will be proved. This generalized theorem will be used as a building block in proving a more flexible bound for Baxter's Learning to Learn setting and in the convergence bound for the Staged Learning model.

The initial framework will follow the setup of Haussler[5] with notation revised for easy extension into the Staged Learning model. The basic PAC model is characterized by a hypothesis space, H_F, composed of hypotheses, $H_f : x \to y$ which take an input $x \in X$, and produce an output, $y \in Y$. Let $z = (x, y)$ denote a labeled example and let $Z = X \times Y$. The loss function, l, describes the penalty for predicting some output y_1 when the correct answer is y_2. Instead of viewing l as a function from $Y \times Y \to [0, 1]$, we will find it more convenient to think of l as a function from $(X \to Y) \times Z \to [0, 1]$ where $l(H_f, (x, y))$ gives the loss of predicting $H_f(x)$ when the correct answer is y. Let us create a loss space, $H^l_F = \{l(H_f, \cdot) : H_f \in H_F\}$ by partially applying the loss function to each hypothesis. Individual elements, $H^l_f : Z \to [0, 1]$, of the loss space H^l_F, are functions from a labeled example to some loss in $[0, 1]$. Some number, N, of labeled examples, $z^N \in Z^N$, is generated according to an unknown distribution, P, on the example space, Z. Let z_n denote the nth labeled example in z^N. A learning algorithm, $L(H_F, z^N, l)$, in the PAC setting can be thought of as taking a hypothesis space, H_F, a set of examples, z^N, and a loss function, $l : (X \to Y) \times Z \to [0, 1]$ to produce a hypothesis, H_f. We will think of the empirical minimization algorithm: [1] evaluate every hypothesis and pick the one with the lowest average loss. An upper bound which applies to the empirical risk minimization algorithm will apply to any other algorithm which outputs a hypothesis in a subset of H_F in the same model.

We now recap the setup of Haussler's agnostic sample complexity bounds [5]. The goal is to prove that for all hypotheses, H_f, the empirical error, $\hat{E}(H^l_f, z^N) = \frac{1}{N}\sum_{n=1}^{N} H^l_f(z_n)$, is close (in a way defined later) to the expected error, $E(H^l_f, P) = \int_Z H^l_f(z)dP(z)$, on a random example drawn according to the distribution, P, over examples. In order to achieve this goal, it is necessary to have some measure of

[1] Other choice algorithms can be used. For example the empirical risk maximization algorithm or the algorithm which chooses the hypothesis with error nearest to 1/2.

the size of the hypothesis space. We first use the example distribution, P, to induce a metric, d_P on the hypothesis space:

$$d_P(H^l_f, H^l_g) = \int_X \sup_{y \in Y} |H^l_f((x,y)) - H^l_g((x,y))| dP_{|X}(x)$$

Here, $P_{|X}(x) = \int_Y P(x,y)dy$ is the marginal distribution on X.

With this metric, we can define the size, $N(\epsilon, H^l_F, d_P)$, of the loss space, H^l_F, as the minimum size of a subset of the elements such that all portions of the space are ϵ close to some element in the subset as measured by d_P. Formally, $N(\epsilon, H^l_F, d_P) = \inf_{S \subseteq H^l_F : \forall H^l_f \in H^l_F \exists s \in S : d_P(H^l_f, s) \leq \epsilon} |S|$. Since we do not know the distribution, we further define a capacity, $C(\epsilon, H^l_F)$, of the loss space as the worst value of $N(\epsilon, H^l_F, d_P)$ for all P. Formally, $C(\epsilon, H^l_F) = \sup_P N(\epsilon, H^l_F, d_P)$.

We measure the distance between the empirical and expected error using the d_ν metric defined by, $d_\nu(a,\ b) = \frac{|a-b|}{\nu+a+b}$.

2.2. **Extension of PAC framework to multiple tasks.** There is a simple extension of the PAC framework to multiple tasks which centers around the observation that Hoeffding's inequality holds for independent random variables with *different* means [8]. Similar results have been derived by Baxter [1]. This variant allows for tasks with different numbers of examples and requires less notational machinery.

The learning function can return a chain of hypotheses $H_{f_1}, ..., H_{f_j}$, each of which is applied to some (predetermined) portion of the examples. Let H^M_f denote a chain of M hypothesis. The standard PAC bound will hold except that the size of the hypothesis space drawn from will increase to $H = H_{F_1} \times ... \times H_{F_j}$. With this bound, we will be able to measure how well a hypothesis space learns a group of tasks (by a "task", we mean the agnostic version of a target concept) without needing enough examples to make *each* task's empirical error converge to it's true error for all hypothesis. As a side effect we will also remove a requirement for the same number of examples/task.

Let:

- M = the number of tasks
- N_m = the number of examples for task m
- $N_{total} = \sum_{m=1}^M N_m$ = the total number of examples for all tasks.
- $\hat{E}(H^{lM}_f, z^{N_{total}}) = \frac{1}{N_{total}} \sum_{m=1}^M \sum_{n=1}^{N_m} l(H_{f_m}, z_{nm})$ = empirical loss on the given sample.
- $E(H^{lM}_f, P^M) = \frac{1}{N_{total}} \sum_{m=1}^M N_m \int_Z l(H_{f_m}, z_m) dP_m(z_m)$ = expected loss when the mth hypothesis is weighted in proportion to N_m.

With this notation in hand, we are ready to state the first theorem.

2.3. **Multitask Learning Theorem.** The Multitask Learning Theorem essentially states that we can guarantee convergence between the average empirical error and average true error with fewer examples then would be necessary to guarantee convergence between empirical and true error on all tasks individually.

Theorem 1. *Let $H^l \subseteq H^l_{F_1} \times ... \times H^l_{F_M}$ be a permissible set of chains of M functions each mapping $Z \to [0,1]$. Let each set of examples $z_{1m}...z_{N_m m}$ be generated according to a probability distribution, P_m with $N_{total} > \frac{2}{\alpha^2 \nu}$.* $\forall \nu > 0, 0 < \alpha < 1$,

$$P(z^N : \sup_{H^l} d_\nu(\hat{E}(H^{lM}_f, z^{N_{total}}), E(H^{lM}_f, P^M) > \alpha)$$

$$\leq 4C(\frac{\alpha\nu}{8}, H^l) e^{\frac{-\alpha^2 \nu N_{total}}{8}}$$

Proof. See appendix B. ∎

This bound is quite useful in it's own right. The bound does not require a minimum number of examples/task or a minimum number of tasks - the only constraint concerns the total number of examples. The bound is useful when trying to prove the aggregate convergence of the empirical error across multiple tasks to the true error across multiple tasks. The bound is tighter (by a constant factor) than requiring each task's empirical error converge to the true error.

3. Learning to Learn

The Staged Learning model is a generalization of the Learning to Learn model. I will first introduce the Learning to Learn model and prove a bound related to this model. The proof is simplified version of Baxter's proof which relies upon the Multitask Learning theorem with a parameterization which allows trading off between the number of examples per task and number of tasks.

In order to perform the analysis, we want to define mappings between the Learning to Learn model and the Multitask Learning Theorem.

3.1. **Model.** The Learning to Learn model as introduced by Baxter[1] is a two level hierarchy. The model covered here is the same, although the notation differs for the ease of the later description of the Staged Learning model.

At the top level, there is a space of hypothesis spaces, H_{FG}, composed of hypothesis spaces, H_{Fg}, which contain individual hypotheses, $H_{fg} : X \to Y$. In the Learning to Learn model, there is a probability distribution, Q, over all possible tasks (a task is a target function and data distribution). Drawing M times from Q produces a set of probability distributions, P^M, which represent M unknown tasks. From each task, P_m, N labeled examples (N_m = N in the previous theorem), z^N are drawn. The input to a learning to learn algorithm, $L_2(H_{FG}, z^{NM}, L, l)$, is a space of hypothesis spaces, H_{FG}, a set of sets of labeled examples, z^{NM}, a learning algorithm, $L(H_{Fg}, z_m^N, l)$, and a loss function, $l : (X \to Y) \times (X \times Y) \to [0, 1]$, for a hypothesis on an example. Output is a hypothesis space, H_{Fg}, on which future tasks will have an expected low loss.

The loss function for the higher level learning problem is defined in terms of the lower level loss function and the learning algorithm.

$$l_2(L, l, H_{Fg}, z_m^N) =$$

$$\hat{E}(l(L(H_{Fg}, z^N, l), z_{nm}) = \frac{1}{N}\sum_{n=1}^{N} l(L(H_{Fg}, z_m^N, l), z_{nm})$$

The higher level loss function allows us to compare hypothesis spaces based on their empirical loss on a particular task. With this loss function, we can define the empirical minimization algorithm at the higher level as: evaluate each hypothesis space on each task and pick the hypothesis space with the lowest average error across all tasks.

Now, we can define loss spaces at both the lower, $H_{Fg}^l = \{l(H_{fg}, \cdot) : H_{fg} \in H_{Fg}\}$, and upper, $H_{FG}^{l_2} = \{l_2(L, l, H_{Fg}, \cdot) : H_{Fg} \in H_{FG}\}$, levels in the Learning to Learn model by partially evaluating the loss functions. Proving a bound for the Learning to Learn model will involve two mappings onto the Multitask Learning theorem, once for the choice of hypothesis space, and once for the M hypothesis choices within a hypothesis space.

3.2. **Mapping.** This table gives the mappings of Learning to Learn problem down onto the PAC problem, which is included for comparison.

	PAC	L. to L., level 1	L. to L., level 2
hypothesis space	$(H_F)^M$	$\cup_g(H_{Fg})^M$	H_{FG}
loss function	l	l	l_2
loss space	$(H_F^l)^M$	$\cup_g(H_{Fg}^l)^M$	$H_{FG}^{l_2}$
example space	$Z^{N_{total}}$	Z^{NM}	$(Z^N)^M$
training example	z_n	z_{nm}	z_m^N
example quantity	N_{total}	NM	M
probability distribution	P^M	P^M	Q

These mappings will be explained in more detail below.

3.3. **Learning to Learn.** Let's recover Baxter's Learning to Learn results. The Learning to Learn algorithm is: for each task, learn on each hypothesis space. Choose the hypothesis space which produces the lowest empirical error across all tasks learned on it.

We want to know that the two Learning to Learn mappings (level 1 and level 2) satisfy the assumptions of the Multitask Learning theorem.

Lemma 1. *The mapping for Learning to Learn, level 1: $H^l = \cup_g(H_{Fg}^l)^M$, $Z = Z$, $N_m = N$, $M = M$, $P_m = P_m$, and $N_{total} = NM$ is applicable to the Multitask Learning theorem.*

Proof. The proof is by assertion. We require the Learning to Learn model to be such that $\forall g : H_{Fg}^l$ is permissible. Z is unchanged from the Multitask Learning model, and each P_m is a probability distribution on Z. Let $H^l = \cup_j(H_{Fg})^M \subset (\cup_j H_{Fg})^M$. $N_{total} = NM$. All of the preconditions of the Multitask Learning theorem have been met. □

Now, we want to prove that we can learn an average empirical error which is a good estimate of the average true error. We require the extra constraint that $\forall m : \ N_m = \frac{N_{total}}{M}$ because the learning function at the next level will be using each *task* as an example and we need each example to have equal weight in the sum.

Corollary 1. $\forall a_1, b_1 \in (0,1)$ *if* $NM \geq \frac{8}{\alpha^2 a_1^2 \nu} \ln \frac{4C(\frac{a_1 \alpha \nu}{8}, \cup_j (H_{Fg}^l)^M)}{b_1 \delta}$ *then with probability at least* $1 - \delta b_1$*, the difference between the empirical and true error of any hypothesis chain, $(H_{fg}^l)^M$ in the loss space, $\cup_j(H_{Fg}^l)^M$, will satisfy*

$$d_\nu(\hat{E}((H_{fg}^l)^M, P^M), E((H_{fg}^l)^M, z^{NM}) \leq \alpha a_1$$

Proof. Application of the Multitask Learning theorem to the mappings in the lemma □

We need another constraint because it is possible to overfit our choice of hypothesis space. The necessary constraint can be found by reducing the metalearning procedure to a learning procedure which uses the base level learning procedure as a subroutine. For the moment, let's assume the learning subroutine has no error, $E(H_{fg}^l, P_m) = \hat{E}(H_{fg}^l, z_m^N)$. The choice of the "best" hypothesis space is governed by the learning problem indicated by the 'Learning to Learn' mapping above. The second mapping views each draw from Q as giving an example, z^N, on which some loss in $[0,1]$ is suffered. Each draw from Q is independent, so we can apply the generalized Multitask Learning theorem to each example.

Lemma 2. *The mapping for Learning to Learn, level 2: $H^l = H_{FG}^{l_2}$, $Z = Z^N$, $N_1 = M$, $M = 1$, $P_1 = Q$, and $N_{total} = M$ is applicable to the Multitask Learning theorem.*

Proof. Again, this is a proof by assertion. We require that $H_{FG}^{l_2}$ be permissible. Q induces a probability for each example, z^N. $H^l = H_{FG}^{l_2} \subset H_{FG}^{l_2}$. □

Corollary 2. $\forall a_2, b_2 \in (0,1)$ *if*

$$M \geq \frac{8}{a_2^2 \alpha^2 \nu} \ln \frac{4C(\frac{a_2 \alpha \nu}{8}, H_{FG}^{l_2})}{\delta b_2}$$

then with probability at least $1-\delta b_2$, the difference between the empirical and true error of any hypothesis space on the hypothesis space space, H_{FG}, will satisfy:

$$d_v(E(H^{l_2}_{Fg}, P), \hat{E}(H^{l_2}_{Fg}, (z^N)^M)) \leq \alpha a_2$$

Proof. Application of the Multitask Learning theorem to the mappings in the lemma □

Theorem 2. *Assume that H_{Fg} and the loss function, l, are such that H^l_{Fg} is permissible for all $H_{Fg} \in H_{FG}$. Also assume that H_{FG} and the loss function, l_2, are such that $H^{l_2}_{FG}$ is permissible. Let Q be any probability distribution over probability distributions on Z. Let $\nu > 0$, $0 < \delta, a_1, b_1, a_2, b_2 < 1$, $a_1 + a_2 \leq 1$, $b_1 + b_2 \leq 1$, and $0 < \alpha < 1$. Also let $(z^N)^M$ be generated by M independent draws according to Q each draw of which is used to draw N examples. If $N \geq \frac{1}{M}\frac{8}{a_1^2\alpha^2\nu}\ln\frac{4C(\frac{a_1\alpha\nu}{8}, \cup_g(H^l_{Fg})^M)}{\delta b_1}$ and $M \geq \frac{8}{a_2^2\alpha^2\nu}\ln\frac{4C(\frac{a_2\alpha\nu}{8}, H^{l_2}_{FG})}{\delta b_2}$ then*

$$P(\exists H_{Fg} \in H_{FG} : d_\nu(E(H^{l_2}_{Fg}, P), \hat{E}(H^{l_2}_{Fg}, z^N)) > \alpha) \leq \delta$$

Proof. Just apply the two lemmas above with the triangle inequality. □

3.4. **Comparison.** Learning to Learn can be a big win in some situations, but there is a constant overhead (in number of examples required) when it is applied to degenerate cases where a PAC bound could be applied.

3.4.1. *Comparison with previous work.* It's worth comparing this result with Baxter's work. If we set $a_1 = a_2 = b_1 = b_2 = \frac{1}{2}$, we recover exactly Baxter's Learning to Learn bound. The extra parameters allow us to trade off between the number of tasks and the number of examples/task for any particular application.

3.4.2. *Comparison with extremes.* Several interesting extreme cases arise in Learning to Learn.

One interesting case is where each H_{Fg} contains just one hypothesis. Then the total number of examples required is $N_{total} = NM \geq \frac{8}{a_1^2\alpha^2\nu}\ln\frac{4C(\frac{\alpha a_1\nu}{8}, H^{l_2}_{FG})}{\delta b_1}$. The same problem can be analyzed in the PAC framework which requires only $N_{total} \geq \frac{8}{\alpha^2\nu}\ln\frac{4C(\frac{\alpha\nu}{8}, H^l_G)}{\delta}$ examples, a savings by a factor of approximately $\frac{\ln\frac{1}{b_1}}{a_1^2}$ neglecting changes in capacity. There is some cost to misusing the learning to learn model.

On the other hand, consider a case in which there exist many tasks known to come from a large space, $\cup_g H_{Fg}$, and also known to be 'related' (which in this case means all the tasks can be learned well on a small subset of the hypothesis space). If one carves up the large space, $\cup_g H_{Fg}$, into many equal size smaller spaces, H_{Fg}, and apply the Learning to Learn theorem. In this case, with $N_{total} = NM \geq \frac{8}{\alpha^2 a_1^2\nu}\ln\frac{4C(\frac{\alpha a_1\nu}{8}, \cup_g(H^l_{Fg})^M)}{\delta b_1}$ we can be confident that the next task will be easily learnable with $N \sim \frac{\ln C(\epsilon, \cup_g(H^l_{Fg})^M)}{\ln C(\epsilon, H^l_{FG})}$ examples. In the PAC case, we will always have to use the capacity of the larger space, $N_{total} \geq \frac{8}{\alpha^2\nu}\ln\frac{4C(\frac{\alpha\nu}{8}, \cup_g H^l_{Fg})}{\delta}$ to learn. Ignoring constant factors, the number of examples required per task can be much larger in the PAC setting then in the Learning to Learn setting.

4. Staged Learning

4.1. **Model.** Staged Learning is the natural extension to Learning to Learn to Learn ... to Learn. In the Staged Learning setting, one is given a set of set of ... sets of hypotheses, $H_{FG...HI}$, and wish to guarantee that any set of ... sets of hypotheses, $H_{FG...Hi}$, will have an empirical error close, (α, δ), to the true expected error on future sets of ... sets of examples, $x_k^{NML....}$. Labeled examples, $z^{NML...K}$, are drawn from a distribution, R over distributions over ... distributions over examples, which is used to draw a set, S^L, over sets of ... sets of examples. The S_l are each used to draw a T^K, and the process is repeated inductively until examples are drawn at the base level. 'Staged Learning' was chosen as a name because the learning is done in stages where the entropy of the choice of a hypothesis function is gradually reduced to 0. In the Staged Learning setup, the learning algorithm at the highest level, $L_D(H_{FG...I}, z^{NML...K}, l, L, ...)$, takes a set of set of ... sets of hypotheses, a set of set of ... sets of examples, a loss function of a hypothesis applied to a labeled example, and learning algorithms for each level in the hierarchy below the top.

The loss function at the lowest level, 1, and the next level, 2, are the same as in the Learning to Learn case,

$$l_2(L, l, H_{Fg...i}, z^N_{ml...k}) = \hat{E}(l(L(H_{Fg...i}, z^N_{ml...k}, l), z_{nml....k})$$

At the next level up in the hierarchy, the loss function,

$$l_3(L_2, L, l, H_{FG...i}, z^{NM}_{l...k}) = \hat{E}(l_2(L, l, L_2(H_{FG...i}, z^{NM}_{l...k}, l, L), z^N_{ml...k}))$$

is the average empirical loss across a set of set of tasks. The loss functions for the higher functions are defined recursively in this manner until the dth loss function,

$$l_D(L_{D-1}, ..., L, l, H_{FG....i}, z^{NML...}_k)$$

$$= \hat{E}(l_{D-1}(L_{D-2}, ..., L, l, L_{D-1}(H_{FG...i}, z^{NML...}_k, l, L, ..., L_{D-2}), z^{NML...}_k))$$

is reached. For the purposes of the proof, we will once again partially apply the loss function to the hypothesis space in order to get a loss space at level 1,

$$H^l_{Fg...i} = \{l(H_{fg...i}, \cdot) : H_{fg...i} \in H_{Fg...i}\}$$

level 2,

$$H^{l_2}_{FG...i} = \{l_2(L, l, H_{Fg...i}, \cdot) : H_{Fg...i} \in H_{FG...i}\}$$

..., and level d,

$$H^{l_D}_{FG...I} = \{l_D(l, L_2, ..., H_{FG...i}, \cdot) : H_{FG...i} \in H_{FG...I}\}$$

4.2. **Mapping.** The mappings are straightforward now.

	Staged, level 1	Staged, level d	level D
hyp. space	$\cup_i(... \cup_h (\cup_g(H_{Fg...i})^M)^L...)^K$	$\cup_i(... \cup_h (H_{F...Gh...i})^L...)^K$	$H_{FG...I}$
loss fun.	l	l_d	l_D
loss space	$\cup_i(... \cup_h (\cup_g(H^l_{Fg...i})^M)^L...)^K$	$\cup_i(... \cup_h (H^{l_d}_{F...Gh...i})^L...)^K$	$H^{l_D}_{FG...I}$
ex. space	$Z^N_{ml...k}$	$Z^{N...M}_{l...k}$	$Z^{NML...K}$
example	$z_{nml...k}$	$z^{N...}_{ml...k}$	$z^{NML...}_k$
# ex.	$NML...K$	$ML...K$	K
dist.	$P^{M...K}$	$Q^{L...K}$	R

4.3. **Distribution convergence theorem.**

Lemma 3. *Each of the mappings defined above:* $H^l = \cup_i(... \cup_h (H^{l_d}_{F...Gh...i})^L...)^K$, $Z = Z^{N...M}_{l...k}$, $N_m = M$, $M = L...K$, $P_m = Q_{l...k}$, *and* $N_{total} = ML...K$ *can be applied to the Multitask Learning theorem.*

Proof. Again, by assertion. We will require that all hypothesis loss spaces be such that they are permissible. N_{total} will be the example quantity given. In all cases H^l is the loss space given in the map. Notice that the loss space is always a subset of $(\cup_{g...i} H_{Fg...i})^{M...K}$. □

Theorem 3. *With a hierarchical nesting of hypothesis spaces,* $H_{FG...I}$, *a loss function,* l, *and algorithms for learning at lower levels,* L, L_2, ..., L_{D-1} *such that all loss spaces,* $H^{l_D}_{F..Gh...i}$ *are permissible,* R *a probability distribution over distributions over ... over the space of labeled examples,* Z, $\forall\ \nu > 0$, $0 < \delta < 1$,

$0 < \alpha < 1$, $a_i, b_i \in [0,1] : \sum_{i=1}^{D} a_i \leq 1, \sum_{i=1}^{D} b_i \leq 1$ *and* $z^{NML...K}$ *generated according to the Staged Learning model, if*

$$N \geq \frac{1}{ML...K} \frac{8}{\alpha^2 a_1^2 \nu} \ln \frac{4C(\frac{\alpha a_1 \nu}{8}, \cup_i (... \cup_h (\cup_g (H^l_{Fgh...i})^M)^L ...)^K)}{\delta b_1}$$

$$M \geq \frac{1}{L...K} \frac{8}{\alpha^2 a_2^2 \nu} \ln \frac{4C(\frac{\alpha a_2 \nu}{8}, \cup_i (... \cup_h (H^{l_2}_{FGh...i})^L ...)^K)}{\delta b_2}$$

..., and

$$K \geq \frac{8}{\alpha^2 a_D^2 \nu} \ln \frac{4C(\frac{\alpha a_D \nu}{8}, H^{l_D}_{FG...I})}{\delta b_D}$$

then

$$P(\exists H_{F...Gi} \in H_{F...GI} : b_\nu(E(H^{l_D}_{F...Gi}, P), \hat{E}(H^{l_D}_{F...Gi}, z^{NML...K})) > \alpha) \leq \delta$$

Proof. A chain of $D-1$ applications of the triangle inequality for d_ν. □

4.4. **Generalizations.** All of the bounds here apply to the agnostic case: there is no need to guarantee that there exists a 'perfect' (zero loss) choice. If such a guarantee can be made, then all of the arguments above can be made with a Multitask Learning theorem for the realizable case which allowing tighter bounds.

5. Examples

Examples of situations where Learning to Learn can be useful are fairly common - any time there is a group of tasks which exploit some common substructure Learning to Learn may be useful. Many other people[9] have covered these examples. When is the full Staged Learning model necessary?

5.1. **Species recognition.** Given the hierarchy present from kingdom to species in nature, it wouldn't be unreasonable to hope that starting with an 'ape family' biased hypothesis space significantly better generalization on the task of 'recognizing a chimpanzee' could be achieved. In this application, the tradeoff between the number of examples/task, tasks, ... would be very useful because typically the number of members of a species is much greater than the number of species in a family.

5.2. **Information hierarchies.** One of the more practical uses of Staged Learning may lie in the direction of using information hierarchies (indices) which are already existent on the Internet. Many indices are organized in a hierarchical fashion. Constructing a recognizer for each leaf of the hierarchy would be useful because it is rather expensive to build human made hierarchies of information. One approach to this is the use of a naive Bayes model by McCallum[7] and others. Essentially, one starts with some hierarchically organized documents. At each node in the hierarchy, the word frequency in all children is calculated. Then a set of mixture weights, λ_1, ..., λ_{root}, $\lambda_{uniform}$ s.t. $\sum_i \lambda_i = 1$ is calculated using an EM method giving a final probability of a word in a document which is:

$$P(word|class) =$$

$$\lambda_1 P(word|N_1) + \lambda_2 P(word|N_2) + ... + \lambda_{root} P(word|root) + \lambda_{uniform} \frac{1}{|dict|}$$

Here N_i = the node at level i on a path from the class to the root.

A classification is made by multiplying all probabilities of all occurring words in each document and choosing the class with the highest probability. If we assume that $\lambda_i > \epsilon > 0$ for some $\epsilon > 0$ (in practice none of the $\lambda_i = 0$) then this method fits into the Staged Learning model. To see this, start at the root of the hierarchy. The classifier can have $P(word|class) \in [\lambda_{uniform} P_{uniform}(word), (1-\lambda_{uniform})(1-P_{uniform}(word))]$. At the next level down, the range of possible classifiers becomes further constrained, and so on until the final class is reached. We can then view the learning problem as choosing between a hierarchy of classes where each level of the hierarchy is parameterized by the relative frequency of words in the child classes. $H_{F...Gh...i}$ will range over all the possible word frequencies of children of a node at level k.

5.3. **Boolean variables.** Restricting to the discrete case on boolean variables a simple complete analysis can be done. Assume we have a depth of 3 and the input is n variables. We know that each task (target concept) depends on only 3 variables. The tasks are grouped into families where the tasks of each family depend on 10 variables associated with that family. There are several families of tasks. How many families of examples/task, tasks/family, and families do we need to learn which set of 10 variables the families depend on? We will find that the number of tasks per task family needed to learn a hypothesis space will be noticeably smaller when using the staged learning model then when using the Learning to Learn model on each task family.

There are 2^{2^n} functions on n variables, $2^{2^{10}}$ functions on 10 variables, and $2^{2^3} = 256$ functions on 3 variables. There are n choose $10 \leq (\frac{ne}{10})^{10}$ possible sets of 10 variables in n and 10 choose $3 = 120$ possible sets of 3 variables in 10.

- Let H_{Fgh} = all 256 functions over a set of 3 variables determined by g and h.
- Let H_{FGh} = a set composed of the possible settings of 3 variables in 10 determined by h.
- Let H_{FGH} = a set composed of all choices of 10 variables.

Then we have:

- $|H_{Fgh}| \leq 2^8$
- $|\cup_g (H_{Fgh})^M| \leq 3 * 5 * 2^{8M+3}$
- $|\cup_h (\cup_g (H_{Fgh})^M)^K| \leq 3^K * 5^K * 2^{8MK+3K}(\frac{ne}{10})^{10}$
- $|\cup_h (H_{FGh})^K| \leq 3^K * 5^K * 2^{3K}(\frac{ne}{10})^{10}$
- $|H_{FGH}| \leq (\frac{ne}{10})^{10}$

With the capacities defined, we only need to set the values of $a_1 = a_2 = a_3 = b_1 = b_2 = b_3 = \frac{1}{3}$ to apply the bound.

If

$$N \geq \frac{1}{MK} \frac{9*8}{\alpha^2 \nu} \ln \frac{3^{K+1} 5^K 2^{8MK+3K+2} (\frac{ne}{10})^{10}}{\delta}$$

$$M \geq \frac{1}{K} \frac{9*8}{\alpha^2 \nu} \ln \frac{3^{K+1} 5^K 2^{3K+2} (\frac{ne}{10})^{10}}{\delta}$$

and

$$K \geq \frac{9*8}{\alpha^2 \nu} \ln \frac{2^2 3 (\frac{ne}{10})^{10}}{\delta}$$

then

$$P(\exists H_{FGh} \in H_{FGH} : b_\nu(E(H^{l_D}_{FGh}, P), \hat{E}(H^{l_D}_{FGh}, z^{NMK})) > \alpha) \leq 1 - \delta$$

Working the numbers out more explicitly we need $K \geq \frac{3^2 2^3}{\alpha^2 \nu}(\ln \frac{2^2 3}{\delta} + 10 \ln \frac{ne}{10})$, $M \geq \frac{1}{K} \frac{9*8}{\alpha^2 \nu}(\ln \frac{2^2 3}{\delta} + K \ln 3 * 5 * 2^3 + 10 \ln \frac{ne}{10})$, and $N \geq \frac{1}{MK} \frac{9*8}{\alpha^2 \nu}(\ln \frac{2^2 3}{\delta} + 10 \ln \frac{ne}{10} + K \ln 3 * 5 + K(8M+3)\ln 2)$.

How does this compare with the PAC bound? The most straightforward application of the PAC bound will require making a choice from a very large space of size 2^{2^n}. The log capacity of this space is 2^n. A smarter application of the PAC bound would note that only $\binom{n}{3} 2^{2^3} \leq 2^8 n^3$ hypothesis are possible with a log capacity bounded by $8 \ln 2 + 3 \ln n$.

Learning to Learn can do much better. The Learning to Learn algorithm must choose between $\binom{n}{3} \leq n^3$ hypothesis spaces with a log capacity bounded by $3 \ln n$. After this choice is made, individual hypothesis are chosen from a set of 2^8 choices with a log capacity of $8 \ln 2$, much lower then the results of the PAC bound.

Depth 3 staged learning lowers the number of tasks required to learn which 3 variables are important. We must make a choice between a set of hypothesis sets of size $\binom{n}{10} \leq n^{10}$ with a log capacity of $10 \ln n$. After this choice is made, Learning to Learn can be applied to $O(10^3)$ hypothesis spaces with log capacity of $3 \ln 10$, a noticeably improvement over the pure Learning to Learn case.

Learning to Learn reduces the number of examples required per task and Staged Learning reduces the number of tasks required in Learning to Learn.

5.4. **Deep neural nets.** As mentioned in the introduction, the Staged Learning model suggests the use of deep feed forward neural nets or, more generically, chained nonlinear feature detectors. Assume that a neural net implements a staged learning model by freezing weights to restrict the available hypothesis space as you move from the root to a leaf. The choice of which weights to freeze is arbitrary and the analysis can work for unnatural choices of weights. However, the natural choice of which weights to freeze corresponds to freezing weights a level or two at a time starting at the first hidden layer and proceeding towards the output as you move through the hierarchy from the root to a particular task.

In light of this, it is interesting to notice that top efforts in character recognition [3] employ a deep feed forward net. The first couple layers of this feed forward net can be useful in recognizing a new language [2].

6. Conclusion and Discussion

Hierarchies of related tasks arise naturally in many domains, but previous bounds were unable to take the information contained in the hierarchy fully into account. We have defined and proved a bound for the Staged Learning model. Along the way, we also proved a more flexible bound for the Learning to Learn model and proved a theorem relevant to Multitask learning. All of these proofs were made eaiser by working explicitly in the loss space.

There is another problem suggested by Staged Learning, that of automatically deducing hierarchies. The extra information contained in an a. priori hierarchy can be rather expensive to produce. It would be nice to create hierarchies automatically by unsupervised learning techniques and a theoretical understanding of this process would be very useful. An EM analysis between these two techniques could prove fruitful.

The analysis in this paper is entirely theoretical. One nice aspect of the theoretical analysis is no particular learning algorithm is required at any level in the hierarchy (in fact, a different learning algorithm could be used at every level in the hierarchy). Unfortunately, the theoretical analysis is only an upper bound. I believe that a lower bound on the number of tasks required can not be proved without some definition of 'relatedness'. Consider the single task distribution which draws the same task every time. Clearly, only one task need be drawn, although many examples from the one task may be required. On the other hand, A distribution over many tasks could require many samples in order to find a confident estimate of the learnability of the task distribution by a particular hypothesis space. A lower bound on the number of examples required has been proved [1].

It would be nice to remove the requirement that each task have the same minimum number of examples and get a bound which is more adaptive. I'm convinced that this is possible using variables with nonidentical ranges in the Hoeffding bounds.

Acknowledgements: I would like to thank Jon Baxter, Avrim Blum, and Sebastian Thrun whose comments where indispensable in making this paper coherent.

Notation Appendix

In general, superscripts denote cross products and subscripts denote "modifier" or "member of". Capitols denote a full range and lower case denotes a member of a range.

Here are some examples:

- x^N, a set (of input examples)
- x_n, a member of a set (of input examples)
- H_F, a set (of hypothesis)
- H_f, a member (hypothesis) of a set (of hypothesis)

To get nested sets, I add extra indices:

- $x^{N_{total}}$, a set (of input examples possibly drawn from multiple tasks)
- x_m^N, a set (of input examples for the mth task)
- x^{NM} a set of sets of input examples.
- P^M, a set (of probability distributions over examples)
- H_{FG}, a set (of hypothesis sets) of sets (of hypothesis)
- H_{Fg}, a member (hypothesis space) of a set of sets (of hypothesis)

An arbitrary nesting looks like:

- $H_{FG...I}$, a set of sets of ... of sets (of hypothesis)
- $H_{FG...Hi}$, a member of a set of sets of ... of sets (of hypothesis)

I associate a particular level in the hierarchy of nested sets with a particular position in the indices. The first index corresponds to the deepest level. Thus:

- $x_{nm...lk}$, the nth element of the ... of the mth element of the lth set of sets of ... of sets (of input examples)
- $H_{Fg...i}$, a member of ... a member of a set of sets of ... of sets (of hypothesis) $=$ a set (of hypothesis).

It is often useful to get the size of a particular finite set. To indicate the size of a set, I use the index. Thus, $N = |x^N|$ and $M = |(x^N)^M|$ denote the number of examples and example sets respectively.

The notation for multiple tasks:

- $N_m =$ number of examples for task m
- $N_{total} = \sum_{m=1}^{M} N_m =$ total number of examples for all tasks. 't' for "total"
- $\hat{E}(H_f^M, z^N) = \frac{1}{N} \sum_{m=1}^{M} \sum_{n=1}^{N_m} l(H_{f_m}, z_{nm})$
- $E(H_f^M, P^M) = \frac{1}{N} \sum_{m=1}^{M} N_m \int_Z l(H_{f_m}, z_m) dP_m(z_m)$

6.1. Learning depth.

The proofs often involve learning bias at some depth, d.

6.2. Learning algorithms.

The lower order learning algorithms are arguments to the higher order learning algorithms. It does not matter what the lower order learning algorithms are as long as they are deterministic.

- L, a learning algorithm
- L_2, a Learning to Learn algorithm
- L_D, a Staged Learning algorithm at depth d

6.3. Loss function.

Loss functions at various depths are necessary to measure the convergence of distributions.

- $l(H_f, z_n) = l(H_f, (x_n, y_n)) = l(H_f(x_n), y_n)$ PAC model loss function
- $l_2(L, l, H_{Fg}, z_m^N) = \hat{E}(l(L(H_{Fg}, z_m^N, l), z_{nm}) = \frac{1}{N} \sum_{n=1}^{N} l(L(H_{Fg}, z_m^N, l), z_{nm})$ Learning to Learn top level loss function

- $l_D(L_{D-1},...,L,l,H_{FG...Hi},z_k^{NM...L})$
$$= \hat{E}(l_{D-1}(L_{D-2},...,L,l,L_{D-1}(H_{FG...Hi},z_k^{NM...L},l,L,...,L^{d-2}),z_k^{NM...L}))$$
Staged Learning loss function at the highest level

6.4. **Loss spaces.** With a loss function and hypothesis space, we can define loss spaces.

- $H_F^l = \{l(H_f,\cdot) : H_f \in H_F\}$ loss hypothesis space for PAC model.
- $H_{FG}^{l_2} = \{l_2(L,l,H_{Fg},\cdot) : H_{Fg} \in H_{FG}\}$ loss hypothesis space for top level Learning to Learn model.
- $H_{FG...I}^{l_D} = \{l_D(l,L_2,...,H_{FG...Hi},\cdot) : H_{FG...Hi} \in H_{FG...HI}\}$ loss hypothesis space for the highest level in a Staged Learning model.

6.5. **Error.** There are two kinds of error at each level in the hierarchy.

- $\hat{E}(H_{F...Gh...i}^{l_D}, z_{l...k}^{N...M}) = \frac{1}{M}\sum_{m=1}^{M} H_{F...Gh...i}^{l_D}(z_{ml...k}^{N...})$ empirical error
- $E(H_{F...Gh...i}^{l_D}, P) = \int_Z H_{F...Gh...i}^{l_D} dP(z_{ml...k}^{N...})$ expected error

Proof of Multitask Learning Theorem

The proof of the Multitask Learning theorem is broken up into 3 lemmas similar to Haussler[5] and Baxter[1].

Make two identical draws of $z^{N_{total}}$, labeled as $z_1^{N_{total}}$ and $z_2^{N_{total}}$.

Lemma 4. *Let H^l be a permissible set of chains of M functions. Let P^M be M probability measures on Z. $\forall \nu > 0,\ 0 < \alpha < 1$ with $N_{total} \geq \frac{2}{\alpha^2 v}$,*
$$P(z^{N_{total}} : d_\nu(\hat{E}(H_f^{lM}, z^{N_{total}}), E(H_f^{lM}, P^M)) > \alpha)$$
$$\leq 2P(z_1^{N_{total}}, z_2^{N_{total}} : d_\nu(\hat{E}(H_f^{lM}, z_1^{N_{total}}), \hat{E}(H_f^{lM}, z_2^{N_{total}})) > \frac{\alpha}{2})$$

Proof. permissibility implies measurability of H^l guarantees the measurability of suprema over H^l. The triangle inequality for d_ν implies
$$P(z_1^{N_{total}}, z_2^{N_{total}} : d_\nu(\hat{E}(H_f^{lM}, z_1^{N_{total}}), \hat{E}(H_f^{lM}, z_2^{N_{total}})) > \frac{\alpha}{2})$$
$$\geq P(z_1^{N_{total}}, z_2^{N_{total}} : d_\nu(\hat{E}(H_f^{lM}, z_1^{N_{total}}), E(H_f^{lM}, P^M)) > \alpha \text{ and}$$
$$d_\nu(\hat{E}(H_f^{lM}, z_2^{N_{total}}), E(H_F^{lM}, P^M)) < \frac{\alpha}{2})$$

By Chebyshev's inequality, for a fixed H_f^{lM},
$$P(z^{N_{total}} : d_\nu(\hat{E}(H_f^{lM}, z^{N_{total}}), E(H_f^{lM}, P^M)) < \frac{\alpha}{2})$$
$$\geq P(z^{N_{total}} : \frac{|\hat{E}(H_f^{lM}, z^{N_{total}}) - E(H_f^{lM}, P^M)|}{\nu} < \frac{\alpha}{2})$$
$$\geq 1 - \frac{\sum_{m=1}^{M} N_m E(H_{f_m}^l, P_m)(1 - E(H_{f_m}^l, P_m))}{N_{total}^2 \nu \frac{\alpha^2}{4}}$$
$$\geq 1 - \frac{1}{N_{total}\nu\alpha^2}$$

$$\geq \frac{1}{2}$$

if $N_{total} \geq \frac{2}{\alpha^2 \nu}$ □

Next we need to define a permutation group between the elements of $z_1^{N_{total}}$ and $z_2^{N_{total}}$ so that we can bound the distance between the loss of these two samples under the permutation group. We will first define the permutation group between the examples of each task and then define the permutation group on all tasks, $\Gamma_{2N_{total}}$, as the direct product of each of the permutation groups of each task.

For each task with N_m examples, define Γ_{2N_m} as the permutation group between each task. Γ_{2N_m} is in turn defined as the N_m way direct product of Γ_2, the permutation group on 2 elements. Γ_2 has 2 elements, $\sigma_0 : \{0,1\} \rightarrow \{0,1\}$ and $\sigma_1 : \{0,1\} \rightarrow \{0,1\}$ defined by $\sigma_1(i) = 1 - i$ and $\sigma_0(i) = i$. Now, we define Γ_{2N_m} $= \prod_{n=1}^{N_m} \Gamma_2$ as the direct product of N_m Γ_2 groups and then we define $\Gamma_{2N_{total}} = \prod_{m=1}^{M} \Gamma_{2N_m}$ as the direct product between all of the N_m's.

Let $\sigma \in \Gamma_{2N_{total}}$ denote an element from the permutation group and $z_{\sigma(1)}^{N_{total}}, z_{\sigma(2)}^{N_{total}}$ a permutation applied between the elements of $z_1^{N_{total}}$, $z_2^{N_{total}}$.

Define: H_f^{lM} as a set of M functions, one drawn from each $H_{F_m}^l$.

Define: $d_{z^{2N_{total}}}(H_f^{lM}, H_g^{lM})$

$$= \frac{1}{2N_{total}} \sum_{m=1}^{M} \sum_{n=1}^{N_m} |H_{f_m}^l(z_{1nm}) - H_{g_m}^l(z_{1nm})| + |H_{f_m}^l(z_{2nm}) - H_{g_m}^l(z_{2nm})|$$

Lemma 5. *Let $H^l \subset H_{F_1}^l \times ... \times H_{F_M}^l$ be a permissible set of chains of functions as in the Multitask Learning Theorem. Fix $z_1^{N_{total}}, z_2^{N_{total}}$ and let $C = \{g_1^l, ..., g_K^l\}$ be an $\frac{\alpha\nu}{8}$ cover for $(H^l, d_{z^{2N_{total}}})$ then*

$$P(\sigma : \sup_H d_\nu[\hat{E}(H_f^{lM}, z_1^{N_{total}}), \hat{E}(H_f^{lM}, z_2^{N_{total}})] > \frac{\alpha}{2})$$

$$\leq \sum_{k=1}^{K} P(\sigma : d_\nu[\hat{E}(g_f^{lM}, z_1^{N_{total}}), \hat{E}(g_f^{lM}, z_2^{N_{total}})] > \frac{\alpha}{4})$$

with each σ chosen uniformly from $\Gamma_{2N_{total}}$.

Proof. Fix σ and let H_f^{lM} be such that $d_\nu[\hat{E}(H_f^{lM}, z_1^{N_{total}}), \hat{E}(H_f^{lM}, z_2^{N_{total}})] > \frac{\alpha}{2}$. Choose $f_k = g_f^{lM} \in C$ such that $d_{z^{2N_{total}}}(H_f^{lM}, g_f^{lM}) \leq \frac{\alpha\nu}{8}$. In this case:

$$\frac{2}{\nu} d_{z^{2N_{total}}}(H_f^{lM},\ g_f^{lM})$$

$$= \frac{\sum_{m=1}^{M} \sum_{n=1}^{N_m} |H_{f_m}^l(z_{1nm}) - g_{f_m}^l(z_{1nm})| + |H_{f_m}^l(z_{2nm}) - g_{f_m}^l(z_{2nm})|}{\nu N_{total}}$$

$$= \frac{\sum_{m=1}^{M} \sum_{n=1}^{N_m} |H_{f_m}^l(z_{\sigma(nm)}) - g_{f_m}^l(z_{\sigma(nm)})| + |H_{f_m}^l(z_{\sigma(nm)}) - g_{f_m}^l(z_{\sigma(nm)})|}{\nu N_{total}}$$

$$= \frac{\sum_{m=1}^{M} \sum_{n=1}^{N_m} |H_{f_m}^l(z_{\sigma(nm)}) - g_{f_m}^l(z_{\sigma(nm)})| + \sum_{m=1}^{M} \sum_{n=1}^{N_m} |H_{f_m}^l(z_{\sigma(nm)}) - g_{f_m}^l(z_{\sigma(nm)})|}{\nu N_{total}}$$

$$\geq \frac{\sum_{m=1}^{M}\sum_{n=1}^{N_m}|H^l_{f_m}(z_{\sigma(nm)}) - g^l_{f_m}(z_{\sigma(nm)})|}{\nu N_{total} + \sum_{m=1}^{M}\sum_{n=1}^{N_m}|H^l_{f_m}(z_{\sigma(nm)}) - g^l_{f_m}(z_{\sigma(nm)})|}$$

$$+\frac{\sum_{m=1}^{M}\sum_{n=1}^{N_m}|H^l_{f_m}(z_{\sigma(nm)}) - g^l_{f_m}(z_{\sigma(nm)})|}{\nu N_{total} + \sum_{m=1}^{M}\sum_{n=1}^{N_m}|H^l_{f_m}(z_{\sigma(nm)}) - g^l_{f_m}(z_{\sigma(nm)})|}$$

$$= d_\nu(\hat{E}(H^{lM}_f, z^{N_{total}}_{\sigma(1)}), \hat{E}(g^{lM}_f, z^{N_{total}}_{\sigma(1)})) + d_\nu(\hat{E}(H^{lM}_f, z^{N_{total}}_{\sigma(2)}), \hat{E}(g^{lM}_f, z^{N_{total}}_{\sigma(2)})$$

By the triangle inequality for d_ν,

$$\frac{2}{\nu} d_{z^{2N_{total}}}(H^{lM}_f, g^{lM}_f) + d_\nu(\hat{E}(g^{lM}_f, z^{N_{total}}_{\sigma(1)}), \hat{E}(g^{lM}_f, z^{N_{total}}_{\sigma(2)}))$$

$$\geq d_\nu(\hat{E}(H^{lM}_f, z^{N_{total}}_{\sigma(1)}), \hat{E}(g^{lM}_f, z^{N_{total}}_{\sigma(1)}))$$

$$+ d_\nu(\hat{E}(H^{lM}_f, z^{N_{total}}_{\sigma(2)}), \hat{E}(g^{lM}_f, z^{N_{total}}_{\sigma(2)}))$$

$$+ d_\nu(\hat{E}(g^{lM}_f, z^{N_{total}}_{\sigma(1)}), \hat{E}(g^{lM}_f, z^{N_{total}}_{\sigma(2)}))$$

$$\geq d_\nu(\hat{E}(H^{lM}_f, z^{N_{total}}_{\sigma(1)}), \hat{E}(H^{lM}_f, z^{N_{total}}_{\sigma(2)}))$$

Since $\frac{2}{\nu} d_{z^{2N_{total}}}(H^{lM}_f, g^{lM}_f) \leq \frac{\alpha}{4}$ and $b_\nu(\hat{E}(H^{lM}_f, z^{N_{total}}_{\sigma(1)}), \hat{E}(H^{lM}_f, z^{N_{total}}_{\sigma(2)})) > \frac{\alpha}{2}$ we must have $d_\nu(\hat{E}(g^{lM}_f, z^{N_{total}}_{\sigma(1)}), \hat{E}(g^{lM}_f, z^{N_{total}}_{\sigma(2)})) > \frac{\alpha}{4}$ implying that if a "bad" H^{lM}_f exists there must exist a "bad" g^{lM}_f. □

The next lemma is essentially a statement that Hoeffding bounds apply to the second equation in the lemma above.

Lemma 6. *Let f be any chain of functions.* $\forall z_1^{N_{total}}, z_2^{N_{total}}$,

$$P(\sigma : d_\nu[\hat{E}(g^{lM}_f, z_1^{N_{total}}), \hat{E}(g^{lM}_f, z_2^{N_{total}})] > \frac{\alpha}{4}) \leq 2e^{-\frac{\alpha^2 \nu N_{total}}{8}}$$

where each σ is chosen uniformly at random

Proof. For any σ,

$$d_\nu[\hat{E}(g^{lM}_f, z_1^{N_{total}}), \hat{E}(g^{lM}_f, z_2^{N_{total}})] = \frac{\sum_{m=1}^{M}\sum_{n=1}^{N_m}[g^l_{f_m}(z_{1\sigma(nm)}) - g^l_{f_m}(z_{2\sigma(nm)})]}{\nu N_{total} + \sum_{m=1}^{M}\sum_{n=1}^{N_m}[g^l_{f_m}(z_{1nm}) + g^l_{f_m}(z_{2nm})]}$$

Each pair, $g_{f_m}(z_{1\sigma(nm)}) - g_{f_m}(z_{2\sigma(nm)})$ is a 0 mean random variable over the choice of σ. Hoeffding's bound for 0 mean independent random variables, $X_1, ..., X_K$ with ranges $a_k \leq X_k \leq b_k$ is:

$$P(|\sum_{k=1}^{K} X_k| \geq \epsilon) \leq 2e^{\frac{-2\epsilon^2}{\sum_{k=1}^{K}(b_k - a_k)^2}}$$

Which, when applied to our N_{total} random variables gives

$$P(\sigma : d_\nu[\hat{E}(g^{lM}_f, z_1^{N_{total}}), \hat{E}(g^{lM}_f, z_2^{N_{total}})] > \frac{\alpha}{4})$$

$$\leq 2e^{-\frac{\alpha^2(\nu N_{total}+\sum_{m=1}^{M}\sum_{n=1}^{N_m}[g^l_{f_m}(z_{1nm})+g^l_{f_m}(z_{2mn})])^2}{32\sum_{m=1}^{M}\sum_{n=1}^{N_m}[g^l_{f_m}(z_{1\sigma(nm)})+g^l_{f_m}(z_{2\sigma(nm)})]^2}}$$

Letting $\gamma = \sum_{m=1}^{M}\sum_{n=1}^{N_m}[g^l_{f_m}(z_{1nm}) + g^l_{f_m}(z_{2nm})]$, this becomes:

$$P(\sigma : d_\nu[\hat{E}(g_f^{lM}, z_1^{N_{total}}), \hat{E}(g_f^{lM}, z_2^{N_{total}})] > \frac{\alpha}{4}) \leq 2e^{-\frac{\alpha^2(\nu N_{total}+\gamma)^2}{32\gamma}}$$

Which is worst for $\gamma = \nu N_{total}$, implying the lemma. □

The three lemmas above give a proof for the Multitask Learning Theorem.

References

[1] Jon Baxter. A Model of Inductive Bias Learning. 1998. http://syseng.anu.edu.au/ jon/papers/bias.ps.gz

[2] Jon Baxter and Peter Bartlett. The Canonical Distortion Measure in Feature Space and 1-NN Classification. NIPS97. http://syseng.anu.edu.au/ jon/papers/NIPS97.ps.gz

[3] Y. Bengio, Y. LeCun, C. Nohl, and C. Burges. Lerec: A nn/hmm hybrid for on-line handwriting recognition. Neural Computation, 7(6):1289-1303, November 1995

[4] Rich Caruana, "Multitask Learning," MACHINE LEARNING, 1997.

[5] David Haussler. “Decision Theoretic Generalizations of the PAC Model for Neural Net and Other Learning Applications”, Information and Computation 100, 78-150 (1992)

[6] Thorsten Joachims, Tom Mitchell, Dayne Freitag, and Robert Armstrong, “WebWatcher: Machine Learning and Hypertext”, 1995

[7] Andrew McCallum, Ronald Rosenfeld, Tom Mitchell, and Andrew Ng, “Improving Text Classification by Shrinkage in a Hierarchy of Classes”, ICML98, 359-367, 1998

[8] Rajeev Motwani and Prabhakar Raghavan, “Randomized Algorithms”, Cambridge University Press, p. 70, 1995

[9] S. Thrun and L.Y. Pratt (eds), 1997. Learning To Learn. Kluwer Academic Publishers.

E-mail address: `jcl+@cs.cmu.edu`