Hierarchical Multi-Task Learning: a Cascade Approach Based on the Notion of Task Relatedness

Alon Zweig ALON.ZWEIG@MAIL.HUJI.AC.IL
Daphna Weinshall DAPHNA@CS.HUJI.AC.IL
School of Computer Science and Engineering, The Hebrew University of Jerusalem, Jerusalem, Israel, 91904

Abstract

Multi-task learning can be shown to improve the generalization performance of single tasks under certain conditions. Typically, the algorithmic and theoretical analysis of multi-task learning deals with a two-level structure, including a group of tasks and a single task. In many situations, however, it is beneficial to consider varying degrees of relatedness among tasks, assuming that some tasks are closely related and should contribute more to the learning process, while other tasks are less related but can still contribute some information to the learning process. The extension of current approaches to the multi-level setting may not be trivial.

We propose a general framework for a full hierarchical multi-task setting. We define an explicit notion of hierarchical tasks relatedness, where at each level we assume that some aspects of the learning problem are shared. We suggest a cascade approach, where at each level of the hierarchy a learner learns jointly the uniquely shared aspects of the tasks by finding a single shared hypothesis. This shared hypothesis is used to bootstrap the preceding level in the hierarchy, forming a hypothesis search space. We analyze sufficient conditions for our approach to reach optimality, and provide generalization guarantees in an empirical risk minimization setting.

1. Introduction

Multi-task learning is typically studied in a flat scenario, where all related tasks are similarly treated. Here we formulate a hierarchal multi-task learning setting which breaks this symmetry between the tasks, analyzing the potential gain in jointly learning shared aspects of the tasks at each level of the hierarchy. We also analyze the sample complexity of the hierarchical multi-task learning approach.

In general, multi-task learning seeks to improve the generalization performance of a single task by exploiting information from other related tasks. Various algorithmic approaches have been proposed to exploit information from related tasks, such as the sharing of hidden nodes in neural networks (Caruana, 1997; Baxter, 2000), multi-task regularization of model parameters in kernel methods (Evgeniou et al., 2006), formulating task relationships based on a task covariance matrix which is used to regularize the parameters of a single task (Zhang and Yeung, 2010), and hierarchical Bayesian approaches for modeling the sharing among tasks based on a common prior (Xue et al., 2007; Daume III, 2009).

Theoretical studies in this field focus on the reduction of the sample complexity of a single task given other related tasks. (Baxter, 2000) analyzed the potential gain under the assumption that all tasks share a common inductive bias, reflected by a common near-optimal hypothesis class. (Ben-David and Borbely, 2008) formulated a specific notion of task relatedness as a family of transformations of the sample generating distributions of the tasks.

The motivation behind mutli-task approaches is that there exists some notion of sharing between a group of tasks and a single task within this group. Almost all of the present multi-task approaches consider sharing among tasks organized in a two level hierarchy - a group of tasks and a single task. In many scenarios it is not clear how they might extend to handle a richer hierarchal setting, where the amount of shared information may vary. An exception is the hierarchical Bayesian method (Berger, 1985), where sharing is done based on a common prior which can be decomposed hierarchically. This method is applied to multi-task setting in methods such as those proposed by (Xue

Presented in the Theoretically Grounded Transfer Learning workshop, held at ICML 2013

et al., 2007) and (Daume III, 2009).

In this work we consider a general hierarchical multi-task paradigm extending the common two level approach. We build our notion of shared information among tasks on the concept of task transformations, unlike the shared priors used in the hierarchical Bayesian method. For this we adopt the task relatedness framework of (Ben-David and Borbely, 2008) and extend it to the hierarchical setting. Motivated by the potential in hierarchically grouping tasks, where levels representing a looser notion of task relatedness correspond to many tasks and levels representing a stricter notion correspond to fewer tasks, we propose a cascade approach, where at each level of the cascade the uniquely shared aspects are learnt jointly.

In our proposed learning cascade, at each step a single hypothesis is jointly learned by considering the union of all tasks, and then used to create the inductive bias for the next level in the hierarchy. Joint learning approaches, where shared aspects of the learning problem are learnt from all the tasks together, are widely used in hierarchical learning settings such as the well known CART algorithm of (Brieman et al., 1984) (see (Silla and Freitas, 2010) for a recent review). The rest of this paper is organized as follows. In Section 2 we review the task relatedness framework of (Ben-David and Borbely, 2008) and extend it to a hierarchy of task relations. We present IMT-ERM (Iterative Multi Task-ERM), which is a hierarchical generalization of MT-ERM. We then describe and analyze a specific learning approach based on a learning cascade, CMT-ERM (Cascade Multi Task-ERM).

The optimality analysis of a single step in the cascade is presented in Section 3. In Section 4 we extend our analysis to the ERM setting, providing generalization error guarantees for a single stage in the hierarchy (Theorem 2). In Theorem 3 this analysis is extended recursively to the whole hierarchy. In Section 5 we present an experiment demonstrating the effectiveness of our approach. All proofs are omitted for lack of space.

2. Hierarchical Multi-Task Paradigm

We now define our hierarchical multi-task paradigm. We start by reviewing the non-hierarchical multi-task paradigm in Section 2.1. In Section 2.2 we extend this paradigm to a hierarchical one. In Section 2.3 we describe a cascade approach to the implementation of this paradigm, and outline some important ways which it differs from the basic approach.

2.1. Task Relatedness Background

We start by reviewing the multi-task learning scenario and notion of relatedness presented in (Ben-David and Borbely, 2008), following the same notations and stating the relevant definitions. In this approach to multitask learning one wishes to learn a single task, and the role of additional related tasks is only to aid the learning of this task. Formally, the multi-task learning scenario can be stated as follows: Given domain $\mathcal{X}$, n tasks $1, ..., n$ and unknown distributions P_1,...P_n over $\mathcal{X} \times \{0, 1\}$, a learner is presented with a sequence of random samples S_1,...S_n drawn from these P_i's respectively. The learner seeks a hypothesis $h : \mathcal{X} \to \{0, 1\}$ such that, for (x, b) drawn randomly from P_1, $h(x) = b$ with high probability. We focus on the extent to which the samples S_i, for $i \neq 1$ can be utilized to help find a good hypothesis for predicting the labels of task 1.

Task relatedness is defined based on a set $\mathcal{F}$ of transformations $f : \mathcal{X} \to \mathcal{X}$ (following definitions 1 and 2 in (Ben-David and Borbely, 2008)). Tasks 1 and 2 are said to be $\mathcal{F}$-related if $P_1(x, b) = P_2(f(x), b)$ or $P_2(x, b) = P_1(f(x), b)$. Given a hypothesis space $\mathbb{H}$ over domain $\mathcal{X}$, we assume $\mathcal{F}$ acts as a group over $\mathbb{H}$, namely $\mathcal{F}$ is closed under function composition and $\mathbb{H}$ is closed under transformations from $\mathcal{F}$. Two hypothesis $h_1, h_2 \in \mathbb{H}$ are said to be equivalent under $\mathcal{F}$ iff there exists $f \in \mathcal{F}$ such that $h_2 = h_1 \circ f$, and hypothesis equivalence under $\mathcal{F}$ is denoted by $\sim_{\mathcal{F}}$. $[h]_{\sim_{\mathcal{F}}} = \{h \circ f : f \in \mathcal{F}\}$ denotes a set of hypotheses which are equivalent up to transformations in $\mathcal{F}$. $\mathbb{H}/\sim_{\mathcal{F}}$ denotes the family of all equivalence classes of $\mathbb{H}$ under $\sim_{\mathcal{F}}$, namely, $\mathbb{H}/\sim_{\mathcal{F}} = \{[h]_{\sim_{\mathcal{F}}} : h \in \mathbb{H}\}$.

The learning scenario assumes that the learner gets samples $\{S_i : i \leq n\}$, where each S_i is a set of samples drawn iid from P_i. The probability distributions are assumed to be pairwise $\mathcal{F}$-related. The learner knows the set of indices of the distributions, $\{1, ..., n\}$ and the family of functions $\mathcal{F}$, but does not know the data-generating distribution nor which specific function f relates any given pair of distributions. In this setting, (Ben-David and Borbely, 2008) proposed to exploit the relatedness among tasks by first finding all aspects of the tasks which are invariant under $\mathcal{F}$, and then focus on learning the specific $\mathcal{F}$-sensitive elements of the target task. The potential benefit lies both in the reduction of the search space from the original $\mathbb{H}$ to a smaller $\mathcal{F}$-sensitive subspace $[h]_{\sim_{\mathcal{F}}}$, and from the bigger sample size available to learn the $\mathcal{F}$-invariant aspects of the task. However, the second potential benefit is not guaranteed and depends on the complexity of finding a single $[h]_{\sim_{\mathcal{F}}}$ in $\mathbb{H}/\sim_{\mathcal{F}}$. The complexity of finding $[h]_{\sim_{\mathcal{F}}}$ is formalized using the notion of gen-

eralized VC-dimension from (Baxter, 2000).

2.2. Hierarchical Task Relatedness

Next we extend the multi-task learning setting to a hierarchical multi-task learning setting. In the hierarchical setting our objective is the same as in the original one - the learning of a single task by exploiting additional related tasks. Our approach extends the original one by assuming that the group of tasks $\{1, ..., n\}$, and the corresponding family of transformation functions $\mathcal{F}$, can be decomposed hierarchically. We denote by l a single level in the hierarchy, $0 \leq l \leq L$ and $\mathcal{T}_l \subseteq \{1, ..., n\}$ the group of related tasks in the l's level of the hierarchy. $\mathcal{F}_l \subset \mathcal{F}$ denotes a family of transformations for which all task in $\mathcal{T}_l$ are pairwise $\mathcal{F}_l$-related.

We assume that the set of transformations for each level $0 \leq l \leq L-1$ can be written as a concatenation of the set of domain transformations corresponding to the preceding level, $\mathcal{F}_{l+1}$, and a set of domain transformations $\mathcal{G}_{l+1}$, hence $F_l = \{g \circ f : g \in \mathcal{G}_{l+1}, f \in \mathcal{F}_{l+1}\}$. We call the set of transformations $\mathcal{G}_l$ the set of shared transformations among tasks in $\mathcal{T}_l$.

Definition 1 We say that $\{\mathcal{T}_l, \mathcal{F}_l, \mathcal{G}_l\}_{l=0}^{L}$ is a hierarchical decomposition of a set of $\mathcal{F}$-related tasks $\{1, ..., n\}$ iff:

1. $\mathcal{T}_0 = \{1, .., n\}$
2. $\mathcal{T}_L = \{1\}$, hence $\mathcal{T}_L$ represents the target task.
3. $\mathcal{F}_L = \{f\}$, where f is the identity transformation, hence $f(x) = x$.
4. for all $0 \leq l \leq L-1$:
   (a) $\mathcal{T}_{l+1} \subset \mathcal{T}_l$
   (b) $\forall i, j \in \mathcal{T}_l$, there exists $f \in \mathcal{F}_l$ such that $P_i(x, b) = P_j(f(x), b)$
   (c) $\mathcal{T}_{l+1}$ shares the set of transformations $\mathcal{G}_{l+1}$
   (d) $\mathcal{F}_l = \{g \circ f : g \in \mathcal{G}_{l+1}, f \in \mathcal{F}_{l+1}\}$.
5. $\mathcal{F}_l$ and $\mathcal{G}_l$ act as a group over $\mathbb{H}$, for all $0 \leq l \leq L$.

See Fig. 1 for an illustration of the hierarchical decomposition.

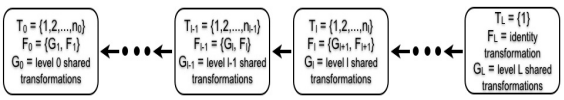

Figure 1. An illustration of the hierarchical decomposition $\{\mathcal{T}_l, \mathcal{F}_l, \mathcal{G}_l\}_{l=0}^{L}$. We assume an indexed set of tasks, $\{1,..,n_0\}$, where 1 is the objective task. The size of each group of tasks decreases as the level increases, thus: $n_0 > n_{l-1} > n_l > 1$. An arrow denotes inclusion relations, $\mathcal{T}_L \subset \mathcal{T}_l \subset \mathcal{T}_{l-1} \subset \mathcal{T}_0$ and $\mathcal{F}_L \subset \mathcal{F}_l \subset \mathcal{F}_{l-1} \subset \mathcal{F}_0$.

From the definition of the hierarchical decomposition (Step 4d) we see that the set of transformations for level l can be obtained by concatenating the set of shared transformations of levels $l+1$ till L. This point will be crucial in understanding the benefit of the cascade hierarchical approach.

Lemma 1 $\mathcal{F}_{l+1} \subset \mathcal{F}_l$, for all $0 \leq l \leq L-1$.

Lemma 2 Given $h \in \mathbb{H}$, $[h]_{\sim_{\mathcal{F}_{l+1}}} \subset [h]_{\sim_{\mathcal{F}_l}}$, for all $0 \leq l \leq L-1$.

2.3. Hierarchical Learning Paradigm - the Cascade Approach

In this section we present and motivate our cascade approach for hierarchical multi-task learning. We start by presenting the IMT-ERM (Iterative MT-ERM) which generalizes the MT-ERM (Multi-Task Empirical Risk Minimization) learning paradigm defined in (Ben-David and Borbely, 2008) to the hierarchical setting. This formalism serves as a basis for our discussion, motivating our proposed cascade. The cascade method is a more constructive approach for which we can provide precise generalization guarantees; this comes at the cost of restricting the learning scope.

We follow standard notations and denote the empirical error of a hypothesis given sample S as:

$$\hat{Er}^S(h) = \frac{|\{(x, b) \in S : h(x) \neq b\}|}{|S|}.$$

The true error of a hypothesis is:

$$Er^P(h) = P(\{(x, b) \in \mathcal{X} \times \{0, 1\} : h(x) \neq b\}).$$

We define the error of any hypothesis space $\mathbb{H}$ as:

$$Er^P(\mathbb{H}) = \inf_{h \in \mathbb{H}} Er^p(h).$$

For notation convenience we shall denote the i'th task in $\mathcal{T}_l$ by l_i.

2.3.1. Iterative MT-ERM

Definition 2 Given $\mathbb{H}$, n tasks hierarchically decomposed by $\{\mathcal{T}_l, \mathcal{F}_l, \mathcal{G}_l\}_{l=0}^{L}$ and their sequence of labeled sample sets $S_1, .., S_n$, the IMT-ERM paradigm works as follows:

1. $\mathbb{H}^0 = \mathbb{H}$.
2. for $l = 0..L$
   (a) Pick $h = \arg\min_{h \in \mathbb{H}^l} \inf_{h_{l_1}, ..., h_{l_{|\mathcal{T}_l|}} \in [h]_{\sim_{\mathcal{F}_l}}} \sum_{i=1}^{|\mathcal{T}_l|} \hat{Er}^{S_{l_i}}(h_{l_i})$

(b) $\mathbb{H}^{l+1} = [h]_{\sim_{\mathcal{F}_l}}$

3. output $h^\diamond$ the single hypothesis in $[h]_{\sim_{\mathcal{F}_L}}$ as the learner's hypothesis.

Note that the fact that $h^\diamond$ is the single hypothesis in $[h]_{\sim_{\mathcal{F}_L}}$ follows directly from the definition of $\mathcal{F}_L$ as containing only the identity transformation.

Following the definition of hierarchical decomposition one can readily see that for $L = 1$ the IMT-ERM is exactly the MT-ERM.

The learning complexity of Step 2a, picking $[h]_{\sim_{\mathcal{F}_l}} \in \mathbb{H}^l/\sim_{\mathcal{F}_l}$, is analyzed in (Ben-David and Borbely, 2008); it uses the notion of generalized VC-dimension from (Baxter, 2000), denoted by $d_{\mathbb{H}^l/\sim_{\mathcal{F}_l}}(n)$, where n refers to the number of tasks, $|\mathcal{T}_l|$ in our setting.

The generalized VC-dimension determines the sample complexity; it has a lower bound of $\sup\{VCdim([h]_{\sim_{\mathcal{F}_l}}) : [h]_{\sim_{\mathcal{F}_l}} \in \mathbb{H}^l/\sim_{\mathcal{F}_l}\}$ and in the general case an upper-bound of $VCdim(\mathbb{H}^l)$. This analysis captures the interrelation between the set of transformations $\mathcal{F}_l$ and the hypothesis space $\mathbb{H}^l$. From Lemma 1 and Lemma 2 we know that both $\sup\{VCdim([h]_{\sim_{\mathcal{F}_l}}) : [h]_{\sim_{\mathcal{F}_l}} \in \mathbb{H}^l/\sim_{\mathcal{F}_l}\}$ and $VCdim(\mathbb{H}^l)$ monotonically decrease with l, which determines the potential of the hierarchical approach.

The MT-ERM paradigm and its hierarchical extension IMT-ERM do not provide a constructive way of performing Step 2a. In the following we shall consider the conditions under which Step 2a can be replaced by choosing a single hypothesis from the equivalence class derived by the shared transformation at each level: $[h]_{\sim_{\mathcal{G}_l}}$. This yields a constructive approach with an explicit search complexity of $VCdim([h]_{\sim_{\mathcal{G}_l}})$, for each level l in the hierarchy.

2.3.2. Cascade Approach

Following Definition 1, we see that each transformation $f \in \mathcal{F}_l$ can be expressed as a concatenation of transformations from all of the shared transformation families $\{\mathcal{G}_i\}_{i=l+1}^L$. We propose a learning approach where we exploit the fact that the transformations can be decomposed into shared transformations and learn together the shared transformations at each level of the cascade. In order to perform this shared learning we consider all tasks sharing a set of transformations as a single unified task. For each such unified task representing level l, we search for a single hypothesis in an equivalence class $[h]_{\sim_{\mathcal{G}_l}}$ where $\mathcal{G}_l$ is the shared family of transformations and h is the hypothesis chosen in the previous stage. The unified task is defined next.

For each level in the hierarchy $0 \le l \le L$, we define the task representing the level as the union of all tasks in $\mathcal{T}_l$. As before we call the i'th task in $\mathcal{T}_l$ $l_i \in \{1..n\}$. Let P_1,...P_n be the probability distributions over $\mathcal{X} \times \{0,1\}$ of tasks $\{1..n\}$ respectively. We shall now define the probability distribution over $\mathcal{X} \times \{0,1\}$, which describes the single task $\mathbf{a}_l$ representing the union of all tasks in $\mathcal{T}_l$ as the average of the distribution of tasks in $\mathcal{T}_l$. This is equivalent to defining a joint probability space where each task is given the same uniform prior. Hence:

$$P_{\mathbf{a}_l}(x,b) = \frac{1}{|\mathcal{T}_l|}\sum_{i=1}^{|\mathcal{T}_l|} P_{l_i}(x,b) \tag{1}$$

Lemma 3

$$Er^{P_{\mathbf{a}_l}}(h) = \frac{1}{|\mathcal{T}_l|}\sum_{i=1}^{|\mathcal{T}_l|} Er^{P_{l_i}}(h) \tag{2}$$

This follows from the definition of $P_{\mathbf{a}_l}$ and Er^P.

We denote by S_{a_l} the union of all samples from all tasks belonging to $\mathcal{T}_l$, $0 \le l \le L$. For a sufficiently large iid sample, $\hat{Er}^{S_{a_l}}(h)$ converges to $Er^{P_{\mathbf{a}_l}}(h)$.

Next we define the CMT-ERM paradigm (Cascade Multi-Task ERM). Here for each level $0 \le l \le L$ of the hierarchy we search for an optimal hypothesis for the unified task $\mathbf{a}_l$. The algorithm iterates through two steps. The first step in iteration l defines the search space as the equivalence class $[h]_{\sim_{\mathcal{G}_l}}$ of the best hypothesis h from the previous iteration. In the second step we search this equivalence class for a single best hypothesis given task $\mathbf{a}_l$.

Definition 3 Given $\mathbb{H}$, n tasks hierarchically decomposed by $\{\mathcal{T}_l, \mathcal{F}_l, \mathcal{G}_l\}_{l=0}^L$ and the sequence of labeled sample sets corresponding to each unified task $S_{a_0}, .., S_{a_L}$, the CMT-ERM paradigm works as follows:

1. Pick $h \in \mathbb{H}$ that minimizes $\hat{Er}^{S_{a_0}}(h)$
2. for $l = 0..L$
   (a) $\mathbb{H}^l = [h]_{\sim_{\mathcal{G}_l}}$
   (b) Pick $h \in \mathbb{H}^l$ that minimizes $\hat{Er}^{S_{a_l}}(h)$
3. output $h^\diamond = h$.

Note that unlike the IMT-ERM or MT-ERM paradigms, the learning stage corresponding to the shared transformations from all tasks in Step 2b is a single well defined ERM problem. The parameter governing the sample complexity of the shared learning (learning task $\mathbf{a}_l$) can also be precisely defined -

$VCdim([h]_{\sim_{\mathcal{G}_l}})$. The potential gain in such an approach lies in the increased number of samples available for the unified tasks and the decrease in search complexity as compared to the original $VCdim(\mathbb{H})$.

In Step 1 we search over all $\mathbb{H}$. This step corresponds to task $\mathbf{a}_0$, which is the union over all tasks, for which we assume a sufficient amount of samples is available. Under certain conditions the search for an optimal hypothesis in Step 1 can be avoided; for instance, we may choose a hypothesis which has minimal false-negatives irrespective of the number of false positives.

3. Cascade Optimality

In the cascade approach we compute the final hypothesis in stages defined by the concatenation of transformations, solving at each step a simpler problem with a larger sample. In this section we analyze the conditions under which an optimal hypothesis of a single task can be found by searching through a cascade of shared transformations, where each transformation fits a single level in the hierarchical decomposition described above. We start by analyzing specific properties of the optimal transformations given our hierarchical decompositions in Section 3.1. In Sections 3.2 and 3.3 we present two properties of a cascade of shared transformations, which are the basis of the assumptions under which the cascade approach can reach optimality. Finally, in Section 3.4 we state the assumptions and prove that the cascade approach can reach optimality.

3.1. Hierarchical Task Relatedness Properties

We recall that from Lemma 2 in (Ben-David and Borbely, 2008) we can deduce that for any task $j \in \mathcal{T}_l$

$$Er^{P_{l_j}}([h]_{\sim_{\mathcal{F}_l}}) = \inf_{h_1,\ldots h_{|\mathcal{T}_l|} \in [h]_{\sim_{\mathcal{F}_l}}} \frac{1}{|\mathcal{T}_l|} \sum_{i=1}^{|\mathcal{T}_l|} Er^{P_{l_i}}(h_i) \quad (3)$$

In the definition of the hierarchical decomposition above (Definition 1), we assumed that tasks in $\mathcal{T}_l$ share the set of transformations $\mathcal{G}_l$. In the following we show that any $g \in \mathcal{G}_l$ which is optimal for a single task $j \in \mathcal{T}_l$ in the sense that it minimizes $Er^{P_{l_j}}([h \circ g]_{\sim_{\mathcal{F}_l}})$, is optimal for all other tasks in $\mathcal{T}_l$.

Lemma 4 For each $j \in \mathcal{T}_l$, $g^* = \arg\min_{g \in \mathcal{G}_l} Er^{P_{l_j}}([h \circ g]_{\sim_{\mathcal{F}_l}}) \Leftrightarrow \forall i \in \mathcal{T}_l$ and $\forall g \in \mathcal{G}_l$ $Er^{P_{l_i}}([h \circ g^*]_{\sim_{\mathcal{F}_l}}) \leq Er^{P_{l_i}}([h \circ g]_{\sim_{\mathcal{F}_l}})$.

Lemma 5 If $g' = \arg\min_{g \in \mathcal{G}} Er^P([h \circ g]_{\sim_{\mathcal{F}}})$ and $f' = \arg\min_{f \in \mathcal{F}} Er^P(h \circ g' \circ f)$, then $g' = \arg\min_{g \in \mathcal{G}} Er^P(h \circ g \circ f')$

3.2. Transformation-Multiplicativity

Definition 4 Two transformations taken from two families of transformations $g \in \mathcal{G}$ and $f \in \mathcal{F}$ are multiplicative with respect to hypothesis $h \in \mathbb{H}$ iff $h \circ g \circ f(x) = h \circ g(x) \cdot h \circ f(x)$.

It is easy to see that if $g \in \mathcal{G}$ and $f \in \mathcal{F}$ are multiplicative then $\{x | h \circ g \circ f(x) = 1\} = \{x | h \circ g(x) = 1\} \cap \{x | h \circ f(x) = 1\}$. In other words, the support of the concatenated transformation is contained in the support of each of the transformations under hypothesis h. The *transformation-multiplicativity* property lets us write $Er^P(h \circ g \circ f)$ as $Er^P(h \circ g)$ plus a residual term, describing the gain obtained by adding the transformation from $\mathcal{F}$. We shall denote the residual term by R^{gf}, where

$$\begin{aligned} R^{gf} = \\ P(\{(x,b) \in \mathcal{X} \times \{0,1\} : b = 1, h \circ g(x) = 1, h \circ f(x) = 0\}) \\ -P(\{(x,b) \in \mathcal{X} \times \{0,1\} : b = 0, h \circ g(x) = 1, h \circ f(x) = 0\}) \end{aligned} \quad (4)$$

This term measures the volume of new errors introduced when adding transformation f, while eliminating the volume of those errors of g which are corrected for by f. Under the *transformation-multiplicativity* assumption, adding a transformation f can change the overall classification value only for points in the support of $h \circ g$. In the following, for the general case (not necessarily assuming *transformation-multiplicativity*) we shall refer to this as the amount by which f 'corrects' the support of g.

Lemma 6 If transformations $g \in \mathcal{G}$ and $f \in \mathcal{F}$ are multiplicative with respect to hypothesis $h \in \mathbb{H}$ then:

$$Er^P(h \circ g \circ f) = Er^P(h \circ g) + R^{gf} \quad (5)$$

Choosing a transformation $f \in \mathcal{F}$ which minimizes $Er^P(h \circ g \circ f)$ implies that $R^{gf} \leq 0$ as it can only decrease the overall error. This is stated formally in the following lemma.

Lemma 7 $f = \arg\min_{f' \in \mathcal{F}} Er^P(h \circ g \circ f') \Rightarrow R^{gf} \leq 0$

Lemma 6 thus shows that under the *transformation-multiplicativity* assumption, the error $Er^P(h \circ g \circ f)$ can be decomposed into 2 terms: the error obtained by choosing $g \in \mathcal{G}$, $Er^P(h \circ g)$, and a residual term R^{gf} referring to the change in the overall classification value of points in the support of $h \circ g$ when adding a transformation $f \in \mathcal{F}$.

3.3. The property of Indifference

We say transformation $f \in \mathcal{F}$ is *indifferent* if for two transformations from a different family of transformations $g, g^* \in \mathcal{G}$, where g^* is optimal, the difference between the amount f 'corrects' the support of g and the amount f 'corrects' the support of g^* is bounded by the difference of the errors between g and g^*. Formally,

Definition 5 A transformation $f \in \mathcal{F}$ is said to be *indifferent* with respect to distribution P, hypothesis h and transformation $g \in \mathcal{G}$, if for $g^* = \arg\min_{g\in\mathcal{G}} Er^P(h \circ g)$ the following holds:

$$R^{g^* f} - R^{gf} \quad \leq \quad Er^P(h \circ g) - Er^P(h \circ g^*) \qquad (6)$$

3.4. Optimality

Now we are ready to state the following two assumptions under which the optimal transformation $g^l \in \mathcal{G}_l$ for each of the tasks in $\mathcal{T}_l$ is the same as the optimal transformation $g^{\mathbf{a}_l} \in \mathcal{G}_l$ for task $\mathbf{a}_l$.

We use the following notations:

- $g^{\mathbf{a}_l} = \arg\min_{g\in\mathcal{G}_l} Er^{P_{\mathbf{a}_l}}(h \circ g)$, the optimal transformation with respect to the task $\mathbf{a}_l$, representing the union of tasks in T_l.
- $g^l = \arg\min_{g\in\mathcal{G}_l} Er^{P_{l_i}}([h \circ g]_{\sim_{\mathcal{F}_l}})$, $\forall i \in |\mathcal{T}_l|$, the optimal transformation which is shared by all of the tasks in $\mathcal{T}_l$ (it follows from Lemma 4 that g^l exists).
- $f^{l_i} = \arg\min_{f\in\mathcal{F}_l} Er^{P_{l_i}}(h \circ g^l \circ f)$, $\forall i \in |\mathcal{T}_l|$, the optimal transformation which is specific to each of the tasks in $\mathcal{T}_l$.

Assumption 1 $\forall i \in |\mathcal{T}_l|$, both $g^{\mathbf{a}_l}, f^{l_i}$ and g^l, f^{l_i} are *transformation-multiplicative*.

Assumption 2 $\forall i \in |\mathcal{T}_l|$, f^{l_i} is *indifferent* with respect to distribution $P_{\mathbf{a}_l}$, the given hypothesis h and $g^l \in \mathcal{G}_l$.

Theorem 1 Under assumptions 1 and 2:

$$\arg\min_{g\in\mathcal{G}_l} Er^{P_{\mathbf{a}_l}}(h \circ g) = \arg\min_{g\in\mathcal{G}_l} Er^{P_{l_i}}([h \circ g]_{\sim_{\mathcal{F}_l}}), \forall i \in \mathcal{T}_l$$

4. Multi-Task Cascade ERM

In the previous section we showed that the optimal transformation of a task can be found by considering a group of tasks together. This implies a learning scenario where searching for the optimal transformation of a single task can be done by considering the group of tasks sharing this transformation, thus benefiting from the bigger sample size contributed from the whole group.

Our discussion until now focused on the optimal solution. Now we extend this analysis to the case where we cannot guarantee optimality of the solution. For this we need to extend our assumptions to imply that a near optimal solution for the group of tasks considered together is also near optimal for each of the tasks considered separately, thus permitting the derivation of an ERM approach.

To begin with, instead of considering the *transformation-multiplicativity* only for the optimal choice of transformations, we assume *transformation-multiplicativity* between $f^{l_i} \in \mathcal{F}_l$, the optimal transformation for task $l_i \in \mathcal{T}_l$, and any transformation $g \in \mathcal{G}_l$ which is close to the optimal transformation, $g^{\mathbf{a}_l}$. Let $\mathcal{G}_l^\epsilon$ denote the set of all transformations in $\mathcal{G}_l$ which have an error bigger by at most ϵ when measured according to $P_{\mathbf{a}_l}$. Formally,

Definition 6 Given hypothesis $h \in \mathbb{H}^{l-1}$, $g \in \mathcal{G}_l^\epsilon$ iff:

$$Er^{P_{\mathbf{a}_l}}(h \circ g) \leq Er^{P_{\mathbf{a}_l}}(h \circ g^{\mathbf{a}_l}) + \epsilon$$

Next we extend the *indifference* assumption, adding a lower bound to the difference of residuals. Thus, the improvement achieved by adding the transformation to a near optimal transformation is close to the improvement when adding a transformation to an optimal transformation.

In order to extend our cascade approach from two levels to the $L+1$ levels in the hierarchy, we extend the assumptions to the whole hierarchy:

Cascade assumptions: for $l = 0..L$ and $h \in \mathbb{H}^{l-1}$ the chosen hypothesis for $P_{\mathbf{a}_{l-1}}$:

1. $\forall g \in \mathcal{G}_l^\epsilon$, $\forall i \in |\mathcal{T}_l|$ and $f^{l_i} = \arg\min_{f\in\mathcal{F}_l} Er^{P_{l_i}}(h \circ g \circ f)$, g and f^{l_i} are multiplicative with respect to h; therefore

$$h \circ g \circ f^{l_i}(x) = h \circ g(x) \cdot h \circ f^{l_i}(x)$$

2. For:
   - $g^{\mathbf{a}_l} = \arg\min_{g\in\mathcal{G}_l} Er^{P_{\mathbf{a}_l}}(h \circ g)$
   - $\hat{f^{l_i}} = \arg\min_{f\in\mathcal{F}_l} Er^{P_{l_i}}(h \circ g^{\mathbf{a}_l} \circ f)$, $\forall i \in [1..|\mathcal{T}_l|]$

   $\hat{f^{l_i}}$ is *indifferent* with respect to the distribution $P_{\mathbf{a}_l}$, hypothesis h and $\forall g \in \mathcal{G}_l^\epsilon$:
   $|\sum_{i=1}^{|\mathcal{T}_l|} R^{g^{\mathbf{a}_l} \hat{f^{l_i}}} - \sum_{i=1}^{|\mathcal{T}_l|} R^{g \hat{f^{l_i}}}| \leq \sum_{i=1}^{|\mathcal{T}_l|} Er^{P_{\mathbf{a}_l}}(h \circ g) - \sum_{i=1}^{|\mathcal{T}_l|} Er^{P_{\mathbf{a}_l}}(h \circ g^{\mathbf{a}_l})$

(Note that we consider $\mathbb{H}^{-1}$ to be the original hypothesis space $\mathbb{H}$.)

For simplicity, in the following we shall refer to the strong form of Assumptions 1,2 stated above as *transformation-multiplicativity* and *indifference* respectively.

We can now analyze the error bound of a hypothesis composed of some original hypothesis h, a shared transformation learnt via an ERM process $g^\diamond$, and the optimal transformation for each task f^{l_i} given the hypothesis $h \circ g^\diamond$.

Theorem 2 Let $d = VCdim([h]_{\sim_{\mathcal{G}_l}})$, $h^* = \arg\min_{h\in[h]_{\sim_{\mathcal{G}_l}}} Er^{P_{\mathbf{a}_l}}(h)$ and $h^\diamond \in [h]_{\sim_{\mathcal{G}_l}}$ denote the output of a standard ERM algorithm trained on task $\mathbf{a}_l$ with sample size $|S_{\mathbf{a}_l}|$. Then for every ϵ and $\delta > 0$ and if $|S_{\mathbf{a}_l}| \geq c_0(\frac{1}{\epsilon}log\frac{1}{\delta} + \frac{d}{\epsilon}log\frac{1}{\epsilon})$, with probability greater than $(1-\delta)$

$$\forall l_i \in \mathcal{T}_l, \quad Er^{P_{l_i}}([h^\diamond]_{\sim_{\mathcal{F}_l}}) \leq Er^{P_{l_i}}([h^*]_{\sim_{\mathcal{F}_l}}) + 2\epsilon$$

Recall the CMT-ERM paradigm presented in Section 2.3.2. Theorem 2 deals with the error accumulated in a single stage of the cascade, in which a near optimal shared hypothesis has been found. In the following theorem we extend this analysis to all levels in the hierarchy, and conclude the overall generalization analysis of the CMT-ERM paradigm. Theorem 3 states the generalization error for CMT-ERM when applied to the hierarchically decomposed multi-task learning setting:

Theorem 3 Let $\{\mathcal{T}_l, \mathcal{F}_l, \mathcal{G}_l\}_{l=0}^L$ be a hierarchical decomposition of a set of $\mathcal{F}$-related tasks for which the cascade assumptions hold. Let $h^* = \arg\min_{h\in\mathbb{H}} Er^{P_1}(h)$ and $h^\diamond$ the output of the CMT-ERM algorithm. For $l = -1..L$, let $d_l = VCdim([h]_{\sim_{\mathcal{G}_l}})$, ϵ_l the objective generalization error for each step in the algorithm, $\epsilon = 2\sum_{l=-1}^{L}\epsilon_l$ and $|S_{\mathbf{a}_l}|$ the sample size available at each step of the algorithm. If $\delta > 0$ and for every $l = -1..L$, $|S_{\mathbf{a}_l}| \geq c_0(\frac{1}{\epsilon_l}log\frac{L}{\delta} + \frac{d_l}{\epsilon_l}log\frac{1}{\epsilon_l})$, with probability greater than $(1-\delta)$

$$Er^{P_1}(h^\diamond) \leq Er^{P_1}(h^*) + \epsilon$$

5. Experiment

In order to demonstrate when our proposed framework achieves improved performance, we tested it in a controlled manner on a synthetic dataset we had created. For that we consider the hypothesis class of conjunction classifiers (Valiant, 1984; Haussler, 1988). We assume an input space of boolean features. Each classifier in this hypothesis class is expressed as a conjunction of a subset of the input boolean features.

The set of transformations $\mathcal{F}$ are all possible feature swaps of length L, denoted by $\mathbf{i}_1...\mathbf{i}_L$ - $\mathbf{j}_1...\mathbf{j}_L$. A feature swap of length L is defined as swapping the values of the features indexed by $\mathbf{i}_1...\mathbf{i}_L$ with the values of the features $\mathbf{j}_1...\mathbf{j}_L$. Each feature index appears at most once in either of the sets, or once in both sets when a feature is swapped with itself, i.e $\mathbf{i}_t = \mathbf{j}_t$. For example, a swap transformation $1,3$ - $2,4$ applied to the binary sample 0101 will result in the transformed binary sample 1010. We decompose the set of transformations $\mathcal{F}$ in the following way: for each $0 \leq l \leq L$, $\mathcal{F}_l$ corresponds to all swaps of length $L-l$ among the features that were not swapped yet. $\mathcal{G}_{l+1}$ corresponds to all swaps of length 1 among the non swapped features. It is easy to see that the sets of transformations- $\mathcal{F}$,$\mathcal{F}_l$ and $\mathcal{G}_{l+1}$ indeed follow definitions 1 and 2 in (Ben-David and Borbely, 2008). We note that each swap of length $L-l$ for $0 \leq l \leq L$ can be written as a swap of size L with l self swaps.

We consider the first hypothesis to be the hypothesis which accepts all samples. With the above notations we do this by concatenating L features valued '1' to all samples (both positive and negative) in the training set. The first hypothesis is the conjunction of these added L features.

We note that choosing L features for a conjunction classifier is clearly equivalent to search for a swap of length L given the defined conjunction over the L dummy features concatenated to the original feature representation. We also note that practically the search over the swap operations can be done on smaller search space considering only swaps between the original features and the concatenated L features.

Synthetic dataset The data defines a group of tasks related in a hierarchical manner: the binary features correspond to nodes in a tree-like structure, and the number of tasks sharing each feature decreases with the distance of the feature node from the tree root. Leaf nodes correspond to binary classification tasks. For all of the tasks, each negative example is assigned a binary feature vector uniformly at random. For a given task, each feature not on the path from the leaf (representing the task) to the root, is assigned a value of 1 or 0 with equal probability for the positive examples. All features on the path are assigned the value 1. The input to the algorithm includes the sets of examples and labels from all n tasks $\{\mathbf{S}^i\}_{i=1}^n$, as well as the hierarchical grouping of tasks.

We search for conjunctions of L features.

For each leaf task, we consider the set $\{\mathcal{T}_l\}_{l=0}^L$ to correspond to the grouping of tasks represented by the path

from the leaf to the root in the dataset thus defined. $\mathcal{T}_0$ corresponds to the root node representing all tasks. $\mathcal{T}_L$ corresponds to the leaf node representing the single objective task. The triplet $\{\mathcal{T}_l, \mathcal{F}_l, \mathcal{G}_l\}_{l=0}^{L}$ clearly obeys definition 1 being a hierarchical decomposition of a set of $\mathcal{F}$-related tasks.

We defined each set of samples $S_{a_l} \in S_{a_0}, .., S_{a_L}$ to correspond to the union of positive and negative samples from all tasks represented by the l's node on the path down from the root to the target task L.

The general cascade algorithm (definition 3) can be rewritten for this specific choice of $\mathbb{H}$ and $\mathcal{G}$ as follows:

1. Set h as the conjunction of the first L features.
2. Init Z the set of all features indices.
3. for $l = 0..L$
   (a) find optimal $h \in [h]_{\sim_{\mathcal{G}_l}}$:
      i. Swap the $l+1$ feature with a feature among all features in Z with index greater than $L+1$ that minimizes $\hat{Er}^{S_{a_l}}(h)$
      ii. Remove this feature from Z
4. output $h^{\diamond} = h$.

It is easy to see that for this specific experimental setting assumptions 1 and 2 of theorem 1 hold; proofs are omitted due to space limitations.

Results Figure 2 shows the performance of our approach compared to standard baselines. First we consider the original two step approach for learning conjunction classifiers, the Valiant step and Haussler step (Valiant, 1984; Haussler, 1988) denoted by 'VHConj'. For the 'VHConj' approach each task is trained on its own. As a multi-task baseline we consider the popular approach for training jointly a set of linear classifiers, by minimizing a loss function (hinge loss) together with a group-lasso regularization term. This approach is denoted by 'LinReg-L12'.

We consider a hierarchy of tasks with 64 tasks (leaves in the hierarchy tree). We test our approach on each task. We test performance as a function of sample size. We show the mean classification accuracy of all tasks averaged over 5 repetitions of the experiment.

With very few samples a conjunction classifier searching for exactly L features is too restrictive as we cannot assume enough support to discover correctly *all* features. To deal with such scenarios we consider two classifiers- the original conjunction classifier using all the discovered features and denoted by 'FullCascadeConj', and a more robust version which returns positive if $L-1$ features take the value '1' and denoted by 'PartialCascadeConj'. We note that the 'VConj' is not restricted to a pre-set number of features, thus this heuristic does not need to be applied to it.

We clearly see that learning benefits much from our cascade approach. The cascade conjunction approach outperforms significantly both the original conjunction baseline and the multi-task learning baseline. As expected for the very small sample scenario the 'PartialCascadeConj' performs best. For larger samples 'FullCascadeConj' improves significantly. We note that we were also able to obtain the same results without assuming prior knowledge of the hierarchy using an algorithm which can discover it automatically.

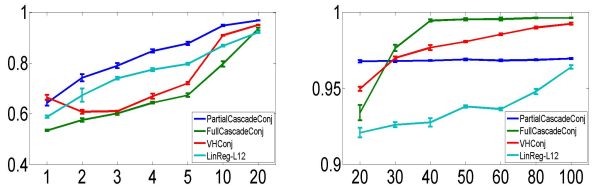

Figure 2. Accuracy results for the synthetic dataset experiment. 'Y-axis' corresponds to accuracy. 'X-axis' corresponds to sample size of the positive set (negative set has the same size). Left plot corresponds to a small sample scenario. Right plot corresponds to a large sample size scenario. 'PartialCascadeConj'- denotes the classifier classifying as positive if $L-1$ features take the value '1'; 'FullCascadeConj' denotes the original conjunction classifier using all the discovered features; 'VHConj' denotes the two step approach based on Valiant's step and Haussler's step; 'LinReg-L12' corresponds to the multi-task baseline where training is done using the group-lasso regularization term.

6. Summary

We presented a general hierarchical multi-task learning framework, extending the commonly used two-level hierarchy in multi-task settings to a multi-level hierarchy. A full hierarchical view of multi-task learning enables tasks of varying degrees of relatedness to contribute in the learning process.

We consider a top-down cascade learning approach, starting from the most abstract level, where at each level of the hierarchy a single optimal hypothesis for the unified task (unifying all tasks at that level) is chosen. This hypothesis is then used to define the inductive bias for the next level in the hierarchy. The complexity of each learning stage is determined by the size of the hypothesis equivalence class defined by applying the shared transformations in each level to the chosen hypothesis of the previous level. We state sufficient conditions for the optimality of our approach and provide generalization guarantees.

Acknowledgements

This work was supported in part by grants from the Israel Science Foundation and the Intel Collaborative Research Institute for Computational Intelligence (ICRI-CI).

References

J. Baxter. A model of inductive bias learning. *J. Artif. Intell. Res. (JAIR)*, 12:149–198, 2000.

S. Ben-David and R.S. Borbely. A notion of task relatedness yielding provable multiple-task learning guarantees. *Machine Learning*, 73(3):273–287, 2008.

J.O. Berger. *Statistical decision theory and Bayesian analysis*. Springer, 1985. ISBN 0387960988.

L. Brieman, J.H. Friedman, R.A. Olshen, and C.J. Stone. Classification and regression trees. *Wadsworth Inc*, 67, 1984.

R. Caruana. Multitask learning. *Machine Learning*, 28(1):41–75, 1997. ISSN 0885-6125.

H. Daume III. Bayesian multitask learning with latent hierarchies. In *Proceedings of the Twenty-Fifth Conference on Uncertainty in Artificial Intelligence*, pages 135–142. AUAI Press, 2009.

T. Evgeniou, C.A. Micchelli, and M. Pontil. Learning multiple tasks with kernel methods. *Journal of Machine Learning Research*, 6(1):615, 2006. ISSN 1532-4435.

David Haussler. Quantifying inductive bias: Ai learning algorithms and valiant's learning framework. *Artificial intelligence*, 36(2):177–221, 1988.

C.N. Silla and A.A. Freitas. A survey of hierarchical classification across different application domains. *Data Mining and Knowledge Discovery*, pages 1–42, 2010. ISSN 1384-5810.

Leslie G. Valiant. A theory of the learnable. *Communications of the ACM*, 27(11):1134–1142, 1984.

Y. Xue, X. Liao, L. Carin, and B. Krishnapuram. Multi-task learning for classification with Dirichlet process priors. *The Journal of Machine Learning Research*, 8:35–63, 2007. ISSN 1532-4435.

Y. Zhang and D.Y. Yeung. A Convex Formulation for Learning Task Relationships in Multi-Task Learning. In *UAI*, 2010.