Tasks Without Borders: A New Approach to Online Multi-Task Learning

Alexander Zimin [1] **Christoph H. Lampert** [1]

Abstract

We introduce MTLAB, a new algorithm for learning multiple related tasks with strong theoretical guarantees. Its key idea is to perform learning sequentially over the data of all tasks, without interruptions or restarts at task boundaries. Predictors for individual tasks are derived from this process by an additional online-to-batch conversion step.

By learning across task boundaries, MTLAB achieves a sublinear regret of true risks in the number of tasks. In the lifelong learning setting, this leads to an improved generalization bound that converges with the total number of samples across all observed tasks, instead of the number of examples per tasks or the number of tasks independently. At the same time, it is widely applicable: it can handle finite sets of tasks, as common in multi-task learning, as well as stochastic task sequences, as studied in lifelong learning.

1. Introduction

In recent years, machine learning has become a core technology in many commercially relevant applications. One observation in this context was that real-world learning tasks often do not occur in isolation, but rather as collections or temporal sequences of many, often highly related tasks. Examples include click-through rate prediction for online ads, personalized voice recognition for smart devices, or handwriting recognition of different languages.

Multi-task learning (Caruana, 1997) has been developed exactly to handle such situations. It is based on an intuitive idea that sharing information between tasks should help the learning process and therefore lead to improved prediction quality. In practice, however, this is not guaranteed and multi-task learning can even lead to a reduction of prediction quality, so called *negative transfer*. The question when negative transfer occurs and how it can be avoided has triggered a surge of research interest to better understanding the theoretical properties of multi-task learning, as well as related research areas, such as *lifelong learning* (Baxter, 2000; Pentina & Lampert, 2014), where more and more tasks occur sequentially, and *task curriculum learning* (Pentina et al., 2015), where the order in which to learn tasks needs to be determined.

In this work, we describe a new approach to multi-task learning that has strong theoretical guarantees, in particular improving the rate of convergence over some previous work. Our core idea is to decouple the process of predictor learning from the task structure. This is also the main difference of our approach to previous work, which typically learned one predictor for each task. We treat the available data for all tasks as parts of a single large online-learning problem, in which individual tasks simply correspond to subsets of the data stream that is processed. To obtain predictors for the individual tasks, we make use of online-to-batch conversion methods. We name the method *MTLAB (multi-task learning across boundaries)*.

Our main contribution is a sublinear bound on the task regret of MTLAB with true risks. As a corollary, we show that MTLAB improves the existing convergence rates in the case of lifelong learning. From the regret-type bounds, we derive high probability bounds on the expected risk of each task, which constitutes a second main contribution of our work.

For real-world problems, not all tasks might be related to all previous ones. Our third contribution is a theoretically well-founded, yet practical, mechanism to avoid negative transfer in this case: we show that by splitting the set of tasks into homogeneous groups and using MTLAB to learn individual predictors on each of the resulting subsequences of samples, one obtains the same strong guarantees for each of the learned predictors while avoiding negative transfer.

2. Multi-task learning of sequential tasks

In this section we present the main notation and introduce the MTLAB approach to information transfer between tasks.

[1]Institute of Science and Technology, Austria. Correspondence to: Alexander Zimin <azimin@ist.ac.at>, Christoph H. Lampert <chl@ist.ac.at>.

Proceedings of the 1st Adaptive & Multitask Learning Workshop, Long Beach, California, 2019.

We face a sequence of tasks $k_1, \ldots, k_n, \ldots$, where each k_t from a task environment $\mathbb{K}$, and the sequence is a random realization of a stochastic process over $\mathbb{K}$. Note that this general formulation includes the situations most commonly studied in the literature: the case of finitely many fixed tasks (in which case the distribution over the tasks sequence is a delta peak) and the lifelong learning setting with i.i.d. (Baxter, 2000; Pentina & Lampert, 2014) or non-i.i.d. tasks (Pentina & Lampert, 2015).

All tasks share the same *input set* $\mathcal{X}$, *output set* $\mathcal{Y}$, and *hypothesis set* $\mathcal{H}$. Each task k_t, however, has its own associated *joint probability distribution*, D_t, over $\mathcal{X} \times \mathcal{Y}$, conditioned on k_t. Whenever we *observe* a task k_t, we receive a set $S_t = \{(x_{t,i}, y_{t,i})\}_{i=1}^{m_t}$ sampled i.i.d. from the task distribution D_t, and we are given a loss function, $\ell_t : \mathcal{H} \times \mathcal{X} \times \mathcal{Y} \to [0,1]$ that measures the quality of predictions. Alternatively, one can assume that all tasks share the same, a priori known, loss function. *Learning* a task k_t means to identify a hypothesis $h \in \mathcal{H}$ with as small as possible *per-task risk* $\mathrm{er}_t(h)$, which is defined as $\mathbb{E}_{(x,y)\sim D_t}\left[\ell_t(h,x,y)\right]$.

The PAC-Bayes framework, originated in (McAllester, 1999b; Shawe-Taylor & Williamson, 1997), studies the performance of stochastic (Gibbs) predictors. A stochastic predictor is defined by a probability distribution Q over the hypotheses set. For any Gibbs predictor with a distribution Q we define the corresponding true risk of a predictor as

$$\mathrm{er}_t(Q) = \mathbb{E}_{h\sim Q}\left[\mathrm{er}_t(h)\right]. \tag{1}$$

As described in the introduction, we do not require that data for all tasks is available at the same time. Instead, we adopt an online learning protocol for tasks: at step t we observe the dataset S_t for task k_t, and we output the distribution $\hat{Q}_t$.

2.1. Learning across task boundaries

Our first goal is, at any step n, to bound the *regret* of a learned sequence of predictors $\hat{Q}_1, \ldots, \hat{Q}_n$ with respect to any fixed reference distribution Q from some set, Δ, of distributions, i.e.

$$\mathcal{R}_n(Q) = \sum_{t=1}^{n} \mathrm{er}_t(\hat{Q}_t) - \sum_{t=1}^{n} \mathrm{er}_t(Q). \tag{2}$$

Note that the regret is defined using *true risks*, that we do not observe, in contrast to empirical risks. This makes the problem setting very different from the traditional online learning where the empirical performance is considered.

The main idea of MTLAB is to run an online learning algorithm on the samples from all tasks, essentially ignoring the task structure of the problem, and then use a properly defined online-to-batch conversion to obtain predictors for the individual tasks. In this paper, we work with a *Proximal Point Algorithm* (Martinet, 1970) run on the level of samples. Let P be some prior distribution over $\mathcal{H}$. We set $Q_{1,0} = P$ and, once we receive a dataset $S_t = \{(x_{t,1}, y_{t,1}), \ldots, (x_{t,m_t}, y_{t,m_t})\}$ on step t, we compute a sequence of predictors $Q_{t,i}$ each being a solution to

$$\min_{\tilde{Q}\in\Delta}\Big\{\frac{\eta}{m_t}\mathbb{E}_{h\sim\tilde{Q}}\left[\ell_t(h, x_{t,i}, y_{t,i})\right] + \mathrm{KL}(\tilde{Q}|Q_{t,i-1})\Big\}, \tag{3}$$

for all $i = 1, \ldots, m_t$ with $\eta > 0$. Afterwards, the algorithm outputs a predictor $\hat{Q}_t = \frac{1}{m_t}\sum_{i=1}^{m_t} Q_{t,i}$ for task t, and sets $Q_{t+1,0} = Q_{t,m_t}$, to be used as a starting distribution for the next task.

Input: decision set Δ, initial distribution P, learning rate η
Initialization: $Q_{1,0} = P$
At any time point $t = 1, 2, \ldots$**:**
- **receive** dataset S_t of size m_t
- **compute** for $i = 1, \ldots, m_t$
  $Q_{t,i} = \operatorname{argmin}_{\tilde{Q}\in\Delta}\{\frac{\eta}{m_t}\mathbb{E}_{h\sim\tilde{Q}}\left[\ell_t(h, z_{t,i})\right] + \mathrm{KL}(\tilde{Q}|Q_{t,i-1})\}$
- **output** the batch solution: $\hat{Q}_t \leftarrow \frac{1}{m_t}\sum_{i=1}^{m_t} Q_{t,i}$
- **set** prior of the next task: $Q_{t+1,0} \leftarrow Q_{t,m_t}$

Figure 1. MTLAB algorithm

We call the above procedure MTLAB (multi-task learning across task boundaries) and summarize it in Figure 1. Our first main result is a regret bound for the true risks of the sequence of distributions that it produces.

Theorem 1. *Let $\bar{m} = n/(\sum_{t=1}^{n} 1/m_t)$ be the harmonic mean of $m_1, \ldots, m_n$ and let P be a fixed prior distribution that is chosen independently of the data. The predictors produced by MTLAB satisfy with probability $1-\delta$ (over the random training sets) uniformly over $Q \in \Delta$*

$$\mathcal{R}_n(Q) \le \frac{\eta n}{4\bar{m}} + \frac{2KL(Q|P) + \log\frac{2}{\delta}}{\eta}. \tag{4}$$

Corollary. *Set $\eta = \sqrt{\frac{\bar{m}}{n}}$. Then, with probability $1-\delta$, it holds uniformly over $Q \in \Delta$*

$$\frac{1}{n}\mathcal{R}_n(Q) \le \frac{1}{\sqrt{n\bar{m}}}\left(\frac{1}{4} + 2KL(Q|P) + \log\frac{2}{\delta}\right). \tag{5}$$

To put this result into perspective, we compare it to the average regret bounds given in (Alquier et al., 2017), where the goal is to find the best possible data representation for tasks. Even though the settings are a bit different, it gives a good idea of the qualitative nature of our result. (Alquier et al., 2017) provides $\mathcal{O}(\frac{1}{\sqrt{n}} + \frac{1}{\sqrt{m}})$ bound (if all tasks are of the same size m) that can be sometimes improved to $\mathcal{O}(\frac{1}{\sqrt{n}} + \frac{1}{m})$. In either case, convergence happen only in

the regime when the number of tasks *and* the amount of data for each task both tend to infinity. In contrast to this, the right hand side of inequality (5) converges to zero even if only one of the two quantities grows, so in particular for the most common case that the number of tasks grows to infinity, but the amount of data per task remains bounded.

The examples of real-world implementations of MTLAB are provided in the supplementary material.

2.2. Connection to traditional PAC-Bayes bounds

We obtain further insight into the behavior of MTLAB by comparing it to the situation in which each task is learned independently. A more traditional PAC-Bayes bound (e.g. (McAllester, 1999a)) states that with probability $1-\delta$ the following inequality holds for all Q

$$\mathrm{er}_t(Q) \leq \frac{1}{m_t}\sum_{i=1}^{m_t}\mathbb{E}_Q\left[\ell_t(h, x_{t,i}, y_{t,i})\right] + \frac{\mathrm{KL}(Q|P) + \log\frac{1}{\delta}}{\sqrt{m_t}}. \tag{6}$$

This inequality suggests a learning algorithm, namely to minimize the upper bound with respect to Q. In principle, MTLAB is based on a similar objective, but it acts on the sample level and it automatically provides relevant prior distributions for each task. Thereby it is able to achieve better guarantee than one could get by combining separate bounds of the form (6) for multiple tasks.

2.3. MTLAB for lifelong learning

The bound of Theorem 1 holds for any stochastic process over the tasks. In particular, it holds in special case where tasks are sampled independently from a hyper distribution over the task environment, which is usually called *lifelong learning* (Baxter, 2000; Pentina & Lampert, 2014). In this setting, we have a fixed distribution $\mathcal{T}$ over $\mathbb{K}$, and the sequence $k_1, \ldots, k_n$ is an i.i.d. sample from this distribution. One can then define the *lifelong risk* as

$$\mathcal{E}(h) = \mathbb{E}_{k\sim\mathcal{T}}\left[\mathbb{E}_{(x,y)\sim D_k}\left[\ell_k(h(x), y)\right]\right], \tag{7}$$

where D_k and ℓ_k are the distribution and loss function for a task k, respectively. The risk of the Gibbs predictor is then $\mathcal{E}(Q) = \mathbb{E}_{h\sim Q}\left[\mathcal{E}(h)\right]$. Let $\hat{Q}_1, \ldots, \hat{Q}_n$ be the output of MTLAB, then we define the corresponding batch solution as $\bar{Q}_n = \frac{1}{n}\sum_{t=1}^n \hat{Q}_t$ and observe

$$\mathcal{E}(\bar{Q}_n) = \frac{1}{n}\sum_{t=1}^n \mathcal{E}(\hat{Q}_t) = \mathbb{E}\left[\frac{1}{n}\sum_{t=1}^n \mathrm{er}_t(\hat{Q}_t)\right]. \tag{8}$$

Using Theorem 1 we obtain the following guarantee.

Theorem 2. *In the lifelong learning setting, if we run MTLAB with $\eta = \frac{\sqrt{\bar{m}}}{\sqrt{n}}$, for any fixed prior distribution P that is chosen independently from the data, with probability $1-\delta$ uniformly over $Q \in \Delta$*

$$\mathcal{E}(\bar{Q}_n) - \mathcal{E}(Q) \leq \frac{1}{\sqrt{n\bar{m}}}\left(\frac{1}{4} + 2KL(Q|P) + \log\frac{2}{\delta}\right). \tag{9}$$

Typical results for this setting, such as shown in (Pentina & Lampert, 2014; Maurer et al., 2016; Alquier et al., 2017), show the additive convergence rate $\mathcal{O}(\frac{1}{\sqrt{n}} + \frac{1}{\sqrt{\bar{m}}})$, which goes to zero only in the case of infinite data *and* infinite tasks. In contrast, the generalization error for MTLAB converges in the most realistic scenario of finite data per task and increasing number of tasks.

3. Per-task bounds

The results of the previous section provide guarantees on MTLAB's multi-task regret. In this section we compliment those results by presenting a modification that provides guarantees for individual risks of each task. The detailed proofs of all statements can be found in the supplementary material.

As a start, let us consider a bound that can be obtained immediately from Theorem 1. We make use of the following notion of relatedness between tasks that is commonly used in the field of domain adaptation (Ben-David et al., 2007).

Definition 1. *For a fixed hypothesis class $\mathcal{H}$, the* discrepancy *between tasks k_i and k_j is defined as*

$$disc(k_i, k_j) = \sup_{h\in\mathcal{H}} \left|er_i(h) - er_j(h)\right|. \tag{10}$$

The following theorem is an immediate corollary of Theorem 1.

Theorem 3. *Let P be a fixed prior distribution that is chosen independently of the data. Let $\hat{Q}_t$ be a sequence of predictors produced by MTLAB run with $\eta = \sqrt{\frac{\bar{m}}{n}}$ and let $\bar{Q}_n = \frac{1}{n}\sum_{t=1}^n \hat{Q}_t$. Then the following inequality holds with probability $1-\delta$, uniformly over $Q \in \Delta$*

$$er_n(\bar{Q}_n) \leq er_n(Q) + \frac{2}{n}\sum_{i=1}^n disc(k_i, k_n) + \frac{1}{\sqrt{n\bar{m}}}\left(\frac{1}{4} + 2KL(Q|P) + \log\frac{2}{\delta}\right). \tag{11}$$

This bound resembles the guarantees typical in the setting of *learning from drifting distributions* (Mohri & Medina, 2012). It converges if $\frac{1}{n}\sum_{i=1}^n \mathrm{disc}(k_i, k_n) \to 0$ with n, so if either tasks are identical to each other, or if tasks get suitably more similar on average with growing n. This is a good example of possible negative transfer: when the previous tasks are not related to the current one as measured by the discrepancies, the average discrepancy term will prevent the bound from convergence.

The main question is if we can avoid the negative transfer and improve upon the bound of Theorem 3 in the case when $\frac{1}{n}\sum_{i=1}^{n} \text{disc}(k_i, k_n)$ does not vanish over time. Consider, for example, a simple case of two alternating tasks, i.e. $\frac{1}{n}\sum_{i=1}^{n} \text{disc}(k_i, k_n) \to \frac{1}{2}$ for $n \to \infty$. If we split the sequence of tasks into two subsequences, one for tasks with even and one for tasks with odd indices, and then run MTLAB separately for each sequence, we could nevertheless guarantee the convergence of the error rate for the resulting procedure. Unfortunately, it is rather easy to construct examples in which convergence to zero is not achievable, even with the best possible split of the sequence of tasks into subsequences. Consequently, we redefine our goal to prove error rates that converge below a given threshold ε.

We present an online algorithm, MTLAB.MS (for *MTLAB with Multiple Sequences*), that splits the tasks into subsequences on the fly given some distance $\text{dist}(k_i, k_j)$ between tasks. MTLAB.MS keeps a representative task for each subsequence, and we use the distances to the representatives to decide which subsequence to extend with the new task, or if a new subsequence needs to be initialized.

Pseudo-code for MTLAB.MS is provided in Algorithm 2. The notation $\tilde{Q}, P' = \text{MTLAB}(S, P)$ denotes a single run of MTLAB that takes a dataset S, runs its learning procedure starting from distribution P and outputs two distributions: the final distribution P' to be used in the subsequent runs and the aggregate distribution $\tilde{Q}$ that is a final predictor for the task. Further notation used are: $\mathcal{I}_n$ are the indices of the tasks in the subsequence chosen at step n, $s_n = |\mathcal{I}_n|$ is the size of this subsequence, $\bar{m}_n$ is the harmonic average of the sizes of tasks in the chosen subsequence and η_n is the learning rate of MTLAB associated with the chosen subsequence.

The following theorem shows that if MTLAB.MS could be run with the task discrepancies as distances, it would, for any given threshold ε, yield subsequences with generalization error below ε.

Theorem 4. *Let P be a fixed prior distribution that is chosen independently of the data. If we run MTLAB.MS with $\text{dist}(k_i, k_j) = \text{disc}(k_i, k_j)$, we get with probability $1 - \delta$, uniformly over $Q \in \Delta$*

$$er_n(\bar{Q}_n) \leq er_n(Q) + 2\varepsilon + \frac{2\eta_n}{\bar{m}_n} + \frac{2KL(Q|P) + \log\frac{n}{\delta}}{\eta_n s_n}.$$

This theorem works when the transfer algorithm uses a fixed learning rate η for each subsequence. It is possible to prove a similar statement for the case when the parameters are optimized for the length of each subsequence using the machinery developed in (Zimin & Lampert, 2017). However, the final statement gets more complicated and adds little to the discussions in the current paper. Therefore, we leave this extension for future work.

Input: task distance dist, prior distribution P, threshold ε
Initialization:
set of representative tasks $R = \emptyset$
set of priors $\mathcal{P} = \emptyset$
At any time point $t = 1, 2, \ldots$**:**

- **receive** dataset S_t.
- **set** $\mathcal{I} = \{r \in R : \text{dist}(k_r, k_t) \leq \varepsilon\}$
- **if** $\mathcal{I} = \emptyset$ **then**
  – add t to the set of representatives R
  – set $\mathcal{P}(t) = P$
- **choose** the closest representatives
$$r^\star = \operatorname*{argmin}_{r \in \mathcal{I}} \text{dist}(k_r, k_t) \tag{12}$$
- **run** the transfer algorithm:
$$\bar{Q}_t, P' = \text{MTLAB}(S_t, \mathcal{P}(r^\star)) \tag{13}$$
- **set** $\mathcal{P}(r^\star) = P'$
- **output** $\bar{Q}_t$

Figure 2. MTLAB.MS algorithm

Theorem 4 confirms that it is possible to avoid effects of negative transfer by carefully choosing the tasks we transfer knowledge from at each step. MTLAB.MS is a computationally efficient way of doing this.

In practice, however, the true discrepancy values are unknown. The most direct method to determine the right subsequence for each task is to estimate the discrepancies from the data and use the estimates in the MTLAB.MS algorithm. In the supplementary material we detail two approaches for discrepancy estimation: a) using a part of the labelled training data and b) using separate unlabelled datasets. In both cases it is possible to prove the statements similar to Theorem 4.

4. Conclusion

We introduced a new and widely applicable algorithm for sequentially learning of multiple tasks. By performing learning across tasks boundaries it is able to achieve a sublinear regret bound and improves the convergence rates in the lifelong learning scenario. MTLAB's way of not interrupting or restarting the learning process at task boundaries results in faster convergence rates than what can be achieved by learning individual predictors for each task: in particular, the generalization error decreases with the product of the number of tasks and the number of samples per task, instead of separately in each of these quantities. We also introduced a mechanism for the situation when the tasks to be learned are not all related to each other. We show that by constructing suitable subsequences of task, the convergence properties can hold even in this case.

References

Alquier, P., Mai, T. T., and Pontil, M. Regret bounds for lifelong learning. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2017.

Baxter, J. A model of inductive bias learning. *Journal of Artificial Intelligence Research (JAIR)*, 12:149–198, 2000.

Ben-David, S., Blitzer, J., Crammer, K., and Pereira, F. Analysis of representations for domain adaptation. In *Conference on Neural Information Processing Systems (NIPS)*, 2007.

Caruana, R. Multitask learning. *Machine Learning (ML)*, 28(1):41–75, 1997.

Martinet, B. Régularisation d'inéquations variationnelles par approximations successives. *Rev. Française Informat. Recherche Opérationnelle*, pp. 154–158, 1970.

Maurer, A., Pontil, M., and Romera-Paredes, B. The benefit of multitask representation learning. *Journal of Machine Learning Research (JMLR)*, 17(1):2853–2884, 2016.

McAllester, D. A. PAC-Bayesian model averaging. In *Workshop on Computational Learning Theory (COLT)*, 1999a.

McAllester, D. A. Some PAC-Bayesian theorems. *Machine Learning (ML)*, 37(3):355–363, 1999b.

Mohri, M. and Medina, A. M. New analysis and algorithm for learning with drifting distributions. In *Algorithmic Learning Theory (ALT)*, 2012.

Pentina, A. and Lampert, C. H. A PAC-Bayesian bound for lifelong learning. *International Conference on Machine Learing (ICML)*, 2014.

Pentina, A. and Lampert, C. H. Lifelong learning with non-iid tasks. In *Conference on Neural Information Processing Systems (NIPS)*, 2015.

Pentina, A., Sharmanska, V., and Lampert, C. H. Curriculum learning of multiple tasks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

Shawe-Taylor, J. and Williamson, R. C. A pac analysis of a bayesian estimator. In *Workshop on Computational Learning Theory (COLT)*, 1997.

Zimin, A. and Lampert, C. H. Learning theory for conditional risk minimization. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2017.