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

Anonymous Authors[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]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

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 $\mathbb{E}_{(x,y)\sim D_t}[\ell_t(h,x,y)]$.

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

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

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

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

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 $i = 1, \ldots, m_t$
  $Q_{t,i} = \operatorname*{argmin}_{\tilde{Q}\in\Delta}\{\frac{\eta}{m_t}\mathbb{E}_{h\sim\tilde{Q}}[\ell_t(h, z_{t,i})] + \mathrm{KL}(\tilde{Q}|Q_{t,i-1})\}$
- **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 $Q \in \Delta$*

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

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.

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

Anonymous Authors[1]

Supplementary material

Related work

There are different mullti-task learning scenarios that have been considered in the literature. The standard MTL considers a fixed number of tasks observed simultaneously, i.e. batch setting, with the goal of finding the best hypotheses for each task (Caruana, 1997; Ando & Zhang, 2005; Pontil & Maurer, 2013; Crammer & Mansour, 2012). One direction of extensions is to consider a fixed number of tasks, but allow the size of the dataset to grow (Murugesan et al., 2016; Yang et al., 2017) or even let each task to be a separate online learning problem (Lugosi et al., 2009; Dekel et al., 2007). A setting when the data points from tasks arrive in different order (for a fixed number of tasks) was also considered by (Cavallanti et al., 2010; Saha et al., 2011).

A different learning problem arises when the number of tasks increases with time. For example, in lifelong learning setting, where the tasks are i.i.d. samples from a fixed hyper-distribution, the goal is to control the generalization error that results from applying the hypothesis learned from the observed tasks on the next tasks sampled from the same distribution, e.g. (Baxter, 2000; Pentina & Lampert, 2014; Denevi et al., 2018; Ruvolo & Eaton, 2013; Ammar et al., 2014; Clingerman & Eaton, 2017). The relaxation of i.i.d. assumption on lifelong learning can be done in different ways. (Abernethy et al., 2007; Alquier et al., 2017) propose algorithms that minimize different notions of empirical regret over the tasks. (Kumagai & Iwata, 2016) models the dynamics of the change of the task data distributions and the optimal hypotheses in Bayesian setting. The general case of non-i.i.d. task generation process was studied in (Pentina & Lampert, 2015).

The idea of training a small set of predictors to be applied for future tasks was explored before in different forms. On the experimental side, (Li et al., 2017) proposes to train a set of exemplar hypotheses and determine the optimal ensemble of them for the next tasks based on different criteria. Similarly, (Ditzler, 2016) suggests to train a classifier from each arriving tasks and re-weight them for the next one. A related idea appeared in (Saha et al., 2016), where the authors proposed to weight the data from different tasks differently depending on the target task. On the theoretical side, (Crammer et al., 2008) uses a notion of disparity between tasks to determine the ones that can share the data among each other and presents the generalization bound for the case of i.i.d. dataset sampled from independently sampled tasks.

Examples

Stochastic Neural Networks We first illustrate an implementation of MTLAB in a deep learning context. Following the presentation in (Amit & Meir, 2017), we make use of stochastic neural networks. Let $\mathcal{H} = \{h_\omega, \omega \in \mathbb{R}^d\}$, and assume that the loss function is differentiable with respect to ω. We think of h_ω as a neural network with d weights and we take Δ as the set of distributions over ω of the form

$$Q(\omega) = \prod_{j=1}^{d} \mathcal{N}(\omega_j|\mu_j, \sigma_j^2), \tag{1}$$

where $\mathcal{N}(\omega|\mu, \sigma^2)$ is a Gaussian distribution with mean μ and variance σ^2. The main question is how to perform the optimization step (**??**). First, we note that the KL-divergence has a closed form expression in this case: for two distributions $Q_1(\omega) = \prod_{j=1}^{d} \mathcal{N}(\omega_j|\mu_{1,j}, \sigma_{1,j}^2)$ and $Q_2(\omega) = \prod_{j=1}^{d} \mathcal{N}(\omega_j|\mu_{2,j}, \sigma_{2,j}^2)$, we can compute $\mathrm{KL}(Q_1|Q_2)$ as

$$\frac{1}{2}\sum_{j=1}^{d}(\log\frac{\sigma_{2,j}^2}{\sigma_{1,j}^2} + \frac{\sigma_{1,j}^2 + (\mu_{1,j} - \mu_{2,j})^2}{\sigma_{2,j}^2} - 1). \tag{2}$$

Second, to be able to differentiate $\mathbb{E}_{\omega\sim Q}\left[\ell_t(h_\omega, x_{t,i}, y_{t,i})\right]$ with respect to the parameters of Q, we employ the reparameterization trick (Kingma et al., 2015), that converts a function with stochastic behavior into a deterministic function of an additional stochastic input. MTLAB tells us to perform the optimization for all data points within one task and to then average the resulting distributions. While the

[1]Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the International Conference on Machine Learning (ICML). Do not distribute.

resulting mixture distribution does not have a simple parametric form, it is easy use as a Gibbs predictor: to sample from it one first samples a random index from 1 to m_t and then samples network parameters from the corresponding Gaussian distribution.

Linear predictors A second practical setting for MTLAB is classification with linear predictors. Let $\mathcal{X} = \mathbb{R}^d$, $\mathcal{Y} = \{-1, 1\}$ and $\mathcal{H} = \{h(x) = \text{sign}\langle\omega, x\rangle, \omega \in \mathbb{R}^d\}$ with $\ell_t(a, b) = \mathbb{I}[a \neq b]$. Following (Germain et al., 2009), we restrict Δ to Gaussian distributions with unit variance and we use $P = \mathcal{N}(0, \mathbf{I}_d)$ and $Q_{t,i} = \mathcal{N}(\mu_{t,i}, \mathbf{I}_d)$. Then $\text{KL}(Q_{t,i}|Q_{t,i-1}) = \frac{\|\mu_{t,i}-\mu_{t,i-1}\|_2^2}{2}$ and $\mathbb{E}_Q[\ell_t(h, x_{t,i}, y_{t,i})] = \Phi(\frac{y_{t,i}\langle\mu, x_{t,i}\rangle}{\|x_{t,i}\|_2})$ with $\Phi(a) = \frac{1}{\sqrt{2\pi}}\int_a^{+\infty}\exp(-\frac{1}{2}x^2)dx$. Therefore, MTLAB computes as each step

$$\min_{\mu\in\mathbb{R}^d} \quad \frac{\eta}{m_t}\Phi(\frac{y_{t,i}\langle\mu, x_{t,i}\rangle}{\|x_{t,i}\|_2}) + \frac{\|\mu - \mu_{t,i-1}\|_2^2}{2}. \tag{3}$$

Note that, as commonly done in algorithms inspired by PAC-Bayes theory, after learning one can convert the randomized predictors into deterministic ones, by using the distribution's mean parameters, μ, as weight vectors. This preserves the guarantees up to a constant factor, see (Langford & Shawe-Taylor, 2003) for details.

Estimation from labelled data

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. However, choosing the subsequence is equivalent to choosing a prior distribution for the next task. For the guarantees of the theory to hold, that step needs to be done independently of the labeled data used for learning that task. We overcome this problem by splitting the labeled data into two subsets: one for estimating the discrepancy (denoted by $\hat{S}_t$) and one for learning (denoted by S_t). With $\hat{m}_t = |\hat{S}_t|$ and $m_t = |S_t|$ we define the discrepancy estimates $\widehat{\text{disc}}(k_i, k_j)$ as

$$\sup_{h\in\mathcal{H}}\left|\frac{1}{\hat{m}_i}\sum_{(x,y)\in\hat{S}_i}\ell_i(h, x, y) - \frac{1}{\hat{m}_j}\sum_{(x,y)\in\hat{S}_j}\ell_j(h, x, y)\right|. \tag{4}$$

The standard uniform convergence bounds can be leveraged to guarantee the quality of this estimation. For example, using Theorem 8 of (Bartlett & Mendelson, 2002) we can show that with probability $1 - \delta$

$$\text{disc}(k_i, k_j) \leq \widehat{\text{disc}}(k_i, k_j) + B_L(\hat{S}_i, \hat{S}_j, \delta), \tag{5}$$

where the estimation error B_L is defined as

$$B_L(\hat{S}_i, \hat{S}_j, \delta) = \mathcal{R}_{\hat{S}_i}(\mathcal{L}_i(\mathcal{H})) + \mathcal{R}_{\hat{S}_j}(\mathcal{L}_j(\mathcal{H})) + 3\sqrt{\frac{8\log\frac{2}{\delta}}{|\hat{S}_i|}} + 3\sqrt{\frac{8\log\frac{2}{\delta}}{|\hat{S}_j|}} \tag{6}$$

with $\mathcal{R}_S(\mathcal{F})$ being the empirical Rademacher complexity of a function class $\mathcal{F}$ over a dataset S and $\mathcal{L}_t(\mathcal{H}) = \{(x, y) \to \ell_t(h, x, y), \forall h \in \mathcal{H}\}$. Now we can prove the following theorem for MTLAB.MS used with $\widehat{\text{disc}}(k_i, k_j)$ as task distances.

Theorem 5. *Let P be a fixed prior distribution that is chosen independently of the data. If we run MTLAB.MS with $dist(k_i, k_j) = \widehat{disc}(k_i, k_j)$, we get with probability $1 - 2\delta$ uniformly over $Q \in \Delta$*

$$er_n(\bar{Q}_n) \leq er_n(Q) + 2\varepsilon + \frac{2\eta_n}{\bar{m}_n} + \frac{2KL(Q|P) + \log\frac{n}{\delta}}{\eta_n s_n} + \frac{1}{s_n}\sum_{t\in\mathcal{I}_n} B_L(\hat{S}_t, \hat{S}_n, \frac{\delta}{n}). \tag{7}$$

Remark. Theorem 5 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.

Estimation from unlabelled data

In many situations labeled data is scarce and setting aside a part of the training set for discrepancy estimation would leave us with too little data for training. In such situations, however, it might be rather easy to obtain at least unlabelled data. In the case when learning tasks are deterministic realizable, this additional data can be used to estimate the discrepancy between tasks before we observe the labeled data for the new task. The deterministic case assumes D_t to be a distribution over $\mathcal{X}$ only, while the target is determined by a fixed labelling function $f_t : \mathcal{X} \to \mathcal{Y}$. The loss of a hypothesis then can be written as $\ell_t(h, x, y) = \ell_t(h, f_t, x)$ and we will stick to that notation for this section. The realizable scenario assumes $f_t \in \mathcal{H}$. For this setting, we use the following definition of discrepancies (Ben-David et al., 2007; Mohri & Medina, 2012).

Definition 1. *The symmetric discrepancy $e(k_i, k_j)$ between tasks k_i and k_j is defined as*

$$\sup_{h_1,h_2\in\mathcal{H}}\left|\mathbb{E}_{x\sim D_i}[\ell_i(h_1, h_2, x)] - \mathbb{E}_{x\sim D_j}[\ell_j(h_1, h_2, x)]\right|. \tag{8}$$

Let U_i and U_j be i.i.d. unlabelled sample sets from the task distributions D_i and D_j, respectively. Then we can estimate the symmetric discrepancies by $\hat{e}(k_i, k_j)$ (Kifer et al., 2004) that are defined as

$$\sup_{h_1,h_2\in\mathcal{H}} \left| \frac{1}{|U_i|} \sum_{x\in U_i} \ell_i(h_1,h_2,x) - \frac{1}{|U_j|} \sum_{x\in U_j} \ell_j(h_1,h_2,x) \right|. \tag{9}$$

We then pass these estimates as values of $\text{dist}(k_i, k_j)$ to MTLAB.MS. Similarly to the setting with labelled data, the proposed estimates can be proven to estimate $e(k_i, k_j)$ quite closely. Proposition 2 from (Mansour et al., 2009) gives us that with probability $1-\delta$:

$$e(k_i,k_j) \le \hat{e}(k_i,k_j) + B_U(U_i,U_j,\delta), \tag{10}$$

with

$$B_U(U_i,U_j,\delta) = \mathcal{R}_{U_i}(\mathcal{L}_i^s(\mathcal{H})) + \mathcal{R}_{U_j}(\mathcal{L}_j^s(\mathcal{H})) + \frac{3\log\frac{4}{\delta}}{\sqrt{2\,|U_i|}} + \frac{3\log\frac{4}{\delta}}{\sqrt{2\,|U_j|}}, \tag{11}$$

where $\mathcal{L}_t^s(\mathcal{H}) = \{x \to \ell(h,h',x), h,h' \in \mathcal{H}\}$. To compensate for the lack of label information, we additionally need the notion of an ideal joint hypothesis, i.e. the one that minimizes joint error for two tasks:

$$\lambda_{i,j} = \min_{h\in\mathcal{H}}\{\text{er}_i(h) + \text{er}_j(h)\}. \tag{12}$$

Overall, we obtain the following performance guarantee for MTLAB.MS when run in this setting.

Theorem 6. *Let P be a fixed prior distribution that is chosen independently of the data. In the deterministic realizable case, if we run MTLAB.MS with dist$(k_i, k_j) = \hat{e}(k_i, k_j)$, we get with probability $1-2\delta$, uniformly over $Q \in \Delta$:*

$$er_n(\bar{Q}_n) \le er_n(Q) + 2\varepsilon + \frac{2\eta_n}{\bar{m}_n} + \frac{2KL(Q|P) + \log\frac{n}{\delta}}{\eta_n s_n} + \frac{1}{s_n}\sum_{t\in\mathcal{I}_n}\lambda_{t,n} + \frac{1}{s_n}\sum_{t\in\mathcal{I}_n} B_U(U_t,U_n,\frac{\delta}{n}). \tag{13}$$

Proofs

Lemma S1 (e.g. (Catoni, 2004), 5.2.1)**.** *For any function $f : \mathcal{H} \to \mathbb{R}$ and any distribution P, we have*

$$\sup_Q\{\mathbb{E}_{h\sim Q}[f(h)] - KL(Q|P)\} = \log \mathbb{E}_{h\sim P}\left[e^{f(h)}\right]. \tag{14}$$

Theorem S1. *For any sequence of distributions $\{Q_{t,i}\}$ such that $Q_{t,i}$ depends only on the samples before and including $(x_{t,i}, y_{t,i})$, for any $\lambda > 0$, we have with probability $1-\delta$*

$$\sum_{t=1}^n\sum_{i=1}^{m_t}\frac{1}{m_t}er_t(Q_{t,i}) \le \sum_{t=1}^n\sum_{i=1}^{m_t}\frac{1}{m_t}\mathbb{E}_{Q_{t,i}}[\ell_t(h,x_{t,i},y_{t,i})] \tag{15}$$

$$+ \frac{1}{\lambda}\sum_{t=1}^n\sum_{i=1}^{m_t} KL(Q_{t,i}|Q_{t,i-1}) \tag{16}$$

$$+ \frac{\lambda n}{8\bar{m}} + \frac{1}{\lambda}\log\frac{1}{\delta}. \tag{17}$$

The same holds for the reverse statement.

Proof. Let

$$u_{t,i}(h) = \ell_t(h,x_{t,i},y_{t,i}) - \text{er}_t(h) \tag{18}$$

and apply Lemma S1 to $\sum_{t=1}^n\sum_{i=1}^{m_t}\frac{1}{m_t}\mathbb{E}_{Q_{t,i}}[u_{t,i}(h)]$ to get for any $\lambda > 0$:

$$\sum_{t=1}^n\sum_{i=1}^{m_t}\frac{1}{m_t}\mathbb{E}_{Q_{t,i}}[u_{t,i}(h)] \le \frac{1}{\lambda}\sum_{t=1}^n\sum_{i=1}^{m_t}\text{KL}(Q_{t,i}|Q_{t,i-1}) \tag{19}$$

$$+ \log \mathbb{E}_{\{h_{t,i}\sim Q_{t,i-1}\}_{t,i}}\exp(\sum_{t=1}^n\sum_{i=1}^{m_t}\frac{\lambda}{m_t}u_{t,i}(h_{t,i})). \tag{20}$$

Now let $v_{t,i} = e^{\frac{\lambda}{m_t}u_{t,i}(h_{t,i})}$ and we write

$$\mathbb{E}\left[\mathbb{E}_{\{h_{t,i}\sim Q_{t,i-1}\}_{t,i}}\prod_{t=1}^n\prod_{i=1}^{m_t}v_{t,i}\right] \tag{21}$$

$$= \mathbb{E}\left[\mathbb{E}\left[\mathbb{E}_{\{h_{t,i}\sim Q_{t,i-1}\}_{t,i}}\prod_{t=1}^n\prod_{i=1}^{m_t}v_{t,i}\middle| k_{1:n}, S_{1:n-1}\right]\right] \tag{22}$$

$$= \mathbb{E}\left[\mathbb{E}_{\{h_{t,i}\sim Q_{t,i-1}\}_{t<n,i}}\left[\prod_{t=1}^{n-1}\prod_{i=1}^{m_t}v_{t,i}\right]\times\right. \tag{23}$$

$$\left.\times\,\mathbb{E}\left[\mathbb{E}_{\{h_{n,i}\sim Q_{n,i-1}\}_i}\prod_{i=1}^{m_n}v_{n,i}\middle| k_{1:n}, S_{1:n-1}\right]\right]. \tag{24}$$

Conditioned on $k_{1:n}$, $u_{n,i}(h_{n,i})$ forms a martingale difference sequence when $h_{n,i}$ is sampled from $Q_{n,i-1}$. Therefore, we can bound

$$\mathbb{E}\left[\mathbb{E}_{\{h_{n,i}\sim Q_{n,i-1}\}_i}\prod_{i=1}^{m_n}v_{n,i}\middle| k_{1:n}, S_{1:n-1}\right] \le \exp(\frac{\lambda^2}{8m_n}). \tag{25}$$

Applying the same argument iteratively, we obtain

$$\mathbb{E}\left[\mathbb{E}_{\{h_{t,i}\sim Q_{t,i-1}\}_{t,i}}\prod_{t=1}^n\prod_{i=1}^{m_t}v_{t,i}\right] \tag{26}$$

$$\le \prod_{t=1}^n\exp(\frac{\lambda^2}{8m_t}) = \exp(\frac{\lambda^2 n}{8\bar{m}}). \tag{27}$$

We get the final statement of the theorem by using Markov inequality. The reverse statement can be obtained by applying the same arguments to $-u_{t,i}$. □

Theorem S 2. *Let P be a fixed prior distribution that is independent of the data. Then for any $\lambda > 0$, we have with probability $1-\delta$ uniformly over Q*

$$\sum_{t=1}^{n}\sum_{i=1}^{m_t}\frac{1}{m_t}er_t(Q) \leq \sum_{t=1}^{n}\sum_{i=1}^{m_t}\frac{1}{m_t}\mathbb{E}_Q\left[\ell_t(h,x_{t,i},y_{t,i})\right] \tag{28}$$

$$+\frac{1}{\lambda}KL(Q|P)+\frac{\lambda n}{8\bar{m}}+\frac{1}{\lambda}\log\frac{1}{\delta}. \tag{29}$$

The same holds for the reverse statement.

Proof. Observe that

$$\sum_{t=1}^{n}\sum_{i=1}^{m_t}\frac{1}{m_t}\mathrm{er}_t(Q)-\sum_{t=1}^{n}\sum_{i=1}^{m_t}\frac{1}{m_t}\mathbb{E}_Q\left[\ell_t(h,x_{t,i},y_{t,i})\right] \tag{30}$$

$$=\mathbb{E}_Q\left[\sum_{t=1}^{n}\sum_{i=1}^{m_t}\frac{1}{m_t}\mathrm{er}_t(h)-\sum_{t=1}^{n}\sum_{i=1}^{m_t}\frac{1}{m_t}\ell_t(h,x_{t,i},y_{t,i})\right]. \tag{31}$$

Now we can apply the lemma S1 to the whole sum under the expectation to get for any $\lambda > 0$

$$\mathbb{E}_Q\left[\sum_{t=1}^{n}\sum_{i=1}^{m_t}\frac{1}{m_t}\mathrm{er}_t(h)-\sum_{t=1}^{n}\sum_{i=1}^{m_t}\frac{1}{m_t}\ell_t(h,x_{t,i},y_{t,i})\right] \tag{32}$$

$$\leq\frac{1}{\lambda}\mathrm{KL}(Q|P) \tag{33}$$

$$+\frac{1}{\lambda}\log\mathbb{E}_P e^{\sum_{t=1}^{n}\sum_{i=1}^{m_t}\frac{\lambda}{m_t}\mathrm{er}_t(h)-\sum_{t=1}^{n}\sum_{i=1}^{m_t}\frac{\lambda}{m_t}\ell_t(h,x_{t,i},y_{t,i})} \tag{34}$$

and the second term can be bounded in the same way as in Theorem S1. □

Definition 2. *Given a set Δ of distributions and an arbitrary distribution P, we define*

$$\Pi_\Delta(P)=\operatorname*{argmin}_{Q\in\Delta}KL(Q|P) \tag{35}$$

as a KL-projection of P onto Δ.

Lemma S2 (Pythagorean inequality for KL divergence, e.g. (Csiszár, 1975), Theorem 2.2)**.** *Let Δ be a convex subset of distributions, then for any distribution P and $Q\in\Delta$*

$$KL(Q|P)\geq KL(Q|\Pi_\Delta(P))+KL(\Pi_\Delta(P)|P). \tag{36}$$

Theorem S 3. *The sequence of distributions produced by the algorithm satisfies for any $Q\in\Delta$:*

$$\sum_{t=1}^{n}\sum_{i=1}^{m_t}\frac{1}{m_t}\mathbb{E}_{Q_{t,i}}\left[\ell_t(h,x_{t,i},y_{t,i})\right] \tag{37}$$

$$-\sum_{t=1}^{n}\sum_{i=1}^{m_t}\frac{1}{m_t}\mathbb{E}_{Q}\left[\ell_t(h,x_{t,i},y_{t,i})\right] \tag{38}$$

$$+\frac{1}{\eta}\sum_{t=1}^{n}\sum_{i=1}^{m_t}KL(Q_{t,i}|Q_{t,i-1}) \tag{39}$$

$$\leq\frac{1}{\eta}KL(Q|P). \tag{40}$$

Proof. We start by proving that the optimization in (4) can be performed in two steps: first computing a minimizer over the set of all distributions and then taking a KL-projection onto Δ. Let

$$\tilde{Q}_{t,i}=\operatorname*{argmin}_{Q}\{\frac{\eta}{m_t}\mathbb{E}_{h\sim Q}\left[\ell_t(h,x_{t,i},y_{t,i})\right]+\mathrm{KL}(Q|Q_{t,i-1})\}, \tag{41}$$

where the minimization is over the set of all distributions. We claim that $Q_{t,i}=\Pi_\Delta(\tilde{Q}_{t,i})$. To prove this, first observe that $\tilde{Q}_{t,i}$ can be computed explicitly, see e.g. (Catoni, 2004), Chapter 5.2,

$$d\tilde{Q}_{t,i}(h)=\frac{e^{-\frac{\eta}{m_t}\ell_t(h,x_{t,i},y_{t,i})}dQ_{t,i-1}(h)}{\int e^{-\frac{\eta}{m_t}\ell_t(h',x_{t,i},y_{t,i})}dQ_{t,i-1}(h')} \tag{42}$$

$$=\frac{e^{-\frac{\eta}{m_t}\ell_t(h,x_{t,i},y_{t,i})}dQ_{t,i-1}(h)}{W_t}. \tag{43}$$

Moreover, from this can compute for any Q

$$\mathrm{KL}(Q|\tilde{Q}_{t,i})=\log W_t \tag{44}$$

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

hence, any minimizer of one side also minimizes the other and $Q_{t,i}=\Pi_\Delta(\tilde{Q}_{t,i})$. Now using again (44), we can write

$$\frac{\eta}{m_t}\mathbb{E}_{Q_{t,i}}\left[\ell_t(h,x_{t,i},y_{t,i})\right]-\frac{\eta}{m_t}\mathbb{E}_{Q}\left[\ell_t(h,x_{t,i},y_{t,i})\right] \tag{45}$$

$$=\mathrm{KL}(Q_{t,i}|\tilde{Q}_{t,i})-\mathrm{KL}(Q_{t,i}|Q_{t,i-1}) \tag{46}$$

$$-\mathrm{KL}(Q|\tilde{Q}_{t,i})+\mathrm{KL}(Q|Q_{t,i-1}). \tag{47}$$

Notice that from Lemma S1: $-\mathrm{KL}(Q|\tilde{Q}_{t,i})\leq -\mathrm{KL}(Q|Q_{t,i})-\mathrm{KL}(Q_{t,i}|\tilde{Q}_{t,i})$ and, hence, we get

$$\frac{1}{m_t}\mathbb{E}_{Q_{t,i}}\left[\ell_t(h,x_{t,i},y_{t,i})\right]-\frac{1}{m_t}\mathbb{E}_{Q}\left[\ell_t(h,x_{t,i},y_{t,i})\right] \tag{48}$$

$$\leq\frac{1}{\eta}\mathrm{KL}(Q|Q_{t,i-1})-\frac{1}{\eta}\mathrm{KL}(Q|Q_{t,i})-\frac{1}{\eta}\mathrm{KL}(Q_{t,i}|Q_{t,i-1}). \tag{49}$$

Summing both sides from 1 to n, we obtain a telescoping sum on the right-hand side and get the statement of the theorem. □

Proof of Theorem 1. The starting point is the PAC-Bayes type bound on the differences between the true and empirical risks. From Theorems S1 and S2 from the supplementary materials with $\lambda = \eta$, we get

$$\begin{aligned}\mathcal{R}_n(Q) \leq & \sum_{t=1}^{n}\sum_{i=1}^{m_t}\frac{1}{m_t}\mathbb{E}_{Q_{t,i}}\left[\ell_t(h, x_{t,i}, y_{t,i})\right] \\ & - \sum_{t=1}^{n}\sum_{i=1}^{m_t}\frac{1}{m_t}\mathbb{E}_{Q}\left[\ell_t(h, x_{t,i}, y_{t,i})\right] \\ & + \frac{1}{\eta}\sum_{t=1}^{n}\sum_{i=1}^{m_t}\mathrm{KL}(Q_{t,i}|Q_{t,i-1}) + \frac{1}{\eta}\mathrm{KL}(Q|P) \\ & + \frac{\eta n}{4m} + \frac{1}{\eta}\log\frac{2}{\delta}.\end{aligned} \tag{50}$$

What we need now is essentially a bound on the empirical regret of the algorithm, which can be shown not only to be constant (because we use the *be-the-leader* version of the proximal point algorithm), but also to control the sum of KL divergences between the subsequent steps. Both claims are given together in the Theorem S3:

$$\begin{aligned}& \sum_{t=1}^{n}\sum_{i=1}^{m_t}\frac{1}{m_t}\mathbb{E}_{Q_{t,i}}\left[\ell_t(h, x_{t,i}, y_{t,i})\right] \\ & - \sum_{t=1}^{n}\sum_{i=1}^{m_t}\frac{1}{m_t}\mathbb{E}_{Q}\left[\ell_t(h, x_{t,i}, y_{t,i})\right] \\ & \leq \frac{1}{\eta}\mathrm{KL}(Q|P) - \frac{1}{\eta}\sum_{t=1}^{n}\sum_{i=1}^{m_t}\mathrm{KL}(Q_{t,i}|Q_{t,i-1}).\end{aligned} \tag{51}$$

□

We will use the following notations for the MACRO-Task algorithm:

- $\mathcal{J}(j)$ is the set of indices assigned to j-th subsequence.
- $s_n(j) = |\mathcal{J}(j)|$.
- I_n is the index of the subsequence chosen at the step n.
- $Q_{t,i}(j)$ is the distribution produced by the transfer algorithm on the subsequence j on the i-th sample of t-th task.
- $\bar{m}(j)$ is the harmonic mean of the sizes of tasks in the j-th subsequence.
- $\hat{Q}_t(j)$ the distribution outputted by the transfer algorithm for t-th task of j-th subsequence.

Lemma S3. *For each j-th subsequence, for any $\lambda > 0$, let*

$$\Lambda(j) = \sum_{t\in\mathcal{J}(j)}\sum_{i=1}^{m_t}\frac{\lambda}{m_t}(er_t(h_{t,i}) - \ell_t(h_{t,i}, x_{t,i}, y_{t,i})) \tag{52}$$

$$- \frac{\eta^2 s_n(j)}{\bar{m}(j)}. \tag{53}$$

Then we have

$$\mathbb{E}\left[\mathbb{E}_{\{h_{t,i}\sim Q_{t,i-1}(j)\}}e^{\Lambda(j)}\right] \leq 1. \tag{54}$$

Proof. This a simpler version of Lemma 4 from (Zimin & Lampert, 2017). □

The following property is a variant of Theorem S1 for the MACRO-Task algorithm.

Theorem S4. *The sequence of distributions produced by the MACRO-Task algorithm for the chosen subsequence satisfies for any $\lambda(j) > 0, \forall j$ with probability $1 - \delta$*

$$\sum_{t\in\mathcal{J}(I_n)}\sum_{i=1}^{m_t}\frac{1}{m_t}(er_t(Q_{t,i}(I_n)) - \mathbb{E}_{Q_{t,i}(I_n)}\left[\ell_t(h, x_{t,i}, y_{t,i})\right]) \tag{55}$$

$$\leq \frac{1}{\lambda(I_n)}\sum_{t\in\mathcal{J}(I_n)}\sum_{i=1}^{m_t}KL(Q_{t,i}|Q_{t,i-1}) \tag{56}$$

$$+ \frac{\lambda(I_n)s_n(I_n)}{\bar{m}(I_n)} + \frac{1}{\lambda(I_n)}\log\frac{n}{\delta}. \tag{57}$$

The same holds for the reverse statement.

Proof. We start with the Lemma S1, which applies even if the distributions are random, to get

$$\sum_{t\in\mathcal{J}(I_n)}\sum_{i=1}^{m_t}\frac{1}{m_t}(\mathrm{er}_t(Q_{t,i}(I_n)) - \mathbb{E}_{Q_{t,i}(I_n)}\left[\ell_t(h, x_{t,i}, y_{t,i})\right]) \tag{58}$$

$$\leq \frac{1}{\lambda(I_n)}\sum_{t\in\mathcal{J}(I_n)}\sum_{i=1}^{m_t}\mathrm{KL}(Q_{t,i}|Q_{t,i-1}) \tag{59}$$

$$+ \frac{1}{\lambda(I_n)}\log\mathbb{E}_{\{h_{t,i}\sim Q_{t,i-1}(I_n)\}}e^{D(I_n)} \tag{60}$$

with

$$D(j) = \sum_{t\in\mathcal{J}(j)}\sum_{i=1}^{m_t}\frac{\lambda(j)}{m_t}(\mathrm{er}_t(h_{t,i}) - \ell_t(h_{t,i}, x_{t,i}, y_{t,i})). \tag{61}$$

Now

$$\frac{1}{\lambda(I_n)} \log \mathbb{E}_{\{h_{t,i}\sim Q_{t,i-1}(I_n)\}} e^{D(I_n)} \tag{62}$$

$$\leq \frac{1}{\lambda(I_n)} \log \mathbb{E}_{\{h_{t,i}\sim Q_{t,i-1}(I_n)\}} e^{\Lambda(I_n)} + \frac{\lambda(I_n)s_n(I_n)}{\bar{m}(I_n)} \tag{63}$$

where $\Lambda(j)$ is defined as in Lemma S3. Denote

$$\Omega(j) = \mathbb{E}_{\{h_{t,i}\sim Q_{t,i-1}(j)\}} e^{\Lambda(j)}, \tag{64}$$

so that from Lemma S3 we get $\mathbb{E}\left[\Omega(j)\right] \leq 1$. To finish the proof we need to bound $\frac{1}{\lambda(I_n)} \log \Omega(I_n)$. For this observe

$$\mathbb{P}\left[\Omega(I_n) > \alpha\right] \leq \sum_j \mathbb{P}\left[\Omega(j) > \alpha\right] \leq \sum_j \frac{\mathbb{E}\left[\Omega(j)\right]}{\alpha} \leq \frac{n}{\alpha}. \tag{65}$$

This gives us that with probability $1-\delta$:

$$\log \Omega(I_n) \leq \log \frac{n}{\delta} \tag{66}$$

and finishes the proof. □

Theorem S5. *Let P be a fixed prior distribution independent of the data. For any $\lambda(j) > 0, \forall j$ it holds with probability $1-\delta$ uniformly in Q*

$$\sum_{t\in\mathcal{J}(I_n)} \sum_{i=1}^{m_t} \frac{1}{m_t} (er_t(Q) - \mathbb{E}_Q\left[\ell_t(h, x_{t,i}, y_{t,i})\right]) \tag{67}$$

$$\leq \frac{1}{\lambda(I_n)} KL(Q|P) + \frac{\lambda(I_n)s_n(I_n)}{\bar{m}(I_n)} + \frac{1}{\lambda(I_n)} \log \frac{n}{\delta}. \tag{68}$$

The same holds for the reverse statement.

Proof. The proof is an analogue of Theorem S2, but using arguments from Theorem S4 for the last part. □

Proof of Theorem 4. Observe that from the definition of discrepancies we get

$$\mathrm{er}_n(\bar{Q}_n) - \mathrm{er}_n(Q) \leq \frac{1}{s_n} \sum_{t\in\mathcal{J}(I_n)} (\mathrm{er}_t(\hat{Q}_t(I_n)) - \mathrm{er}_t(Q)) \tag{69}$$

$$+ \frac{2}{s_n} \sum_{t\in\mathcal{J}(I_n)} \mathrm{disc}(k_t, k_n). \tag{70}$$

From the way MTLAB.MS constructs the subsequences, we have $\frac{1}{s_n}\sum_{t\in\mathcal{J}(I_n)} \mathrm{disc}(k_t, k_n) \leq \varepsilon$. Using Theorem S4 and S5 with $\lambda(I_n) = \eta_n$, we get with probability $1-\delta$

$$\sum_{t\in\mathcal{J}(I_n)} (\mathrm{er}_t(\hat{Q}_t(I_n)) - \mathrm{er}_t(Q)) \tag{71}$$

$$\leq \sum_{t\in\mathcal{J}(I_n)} \sum_{i=1}^{m_t} \frac{1}{m_t} (\mathbb{E}_{Q_{t,i}}\left[\ell(h(x_{t,i}), y_{t,i})\right] - \mathbb{E}_Q\left[\ell(h(x_{t,i}), y_{t,i})\right]) \tag{72}$$

$$+ \frac{1}{\eta_n} \sum_{t\in\mathcal{J}(I_n)} \sum_{i=1}^{m_t} \mathrm{KL}(Q_{t,i}|Q_{t,i-1}) + \frac{1}{\eta_n}\mathrm{KL}(Q|P) \tag{73}$$

$$+ 2\frac{\eta_n s_n}{\bar{m}_n} + \frac{1}{\eta_n} \log \frac{n}{\delta}. \tag{74}$$

The final statement follows from the regret bound of Theorem S3. □

Proof of Theorem 4. From the definition of the discrepancies we get

$$\mathrm{er}_n(\bar{Q}_n) - \mathrm{er}_n(Q) \leq \frac{1}{s_n} \sum_{t\in\mathcal{J}(I_n)} (\mathrm{er}_t(\hat{Q}_t(I_n)) - \mathrm{er}_t(Q)) + \frac{2}{s_n} \sum_{t\in\mathcal{J}(I_n)} \mathrm{disc}(k_t, k_n), \tag{75}$$

where $\hat{Q}_t(j)$ is the predictor outputted by the algorithm for t-th task of j-th subsequence. From the way MTLAB.MS constructs the subsequences we are guaranteed that

$$\frac{1}{s_n} \sum_{t\in\mathcal{J}(I_n)} \mathrm{disc}(k_t, k_n) \leq \varepsilon. \tag{76}$$

Now, for any fixed j, the terms

$$\frac{1}{|\mathcal{J}(j)|} \sum_{t\in\mathcal{J}(j)} (\mathrm{er}_t(\hat{Q}_t(j)) - \mathrm{er}_t(Q)) \tag{77}$$

can be bounded in a similar way to Theorem **??**. The additional $\log n$ factor appears since we need to account for the fact that there can be at most n such terms. □

Proof of Theorem 5. Observe that from the definition of discrepancies we get.

$$\mathrm{er}_n(\bar{Q}_n) - \mathrm{er}_n(Q) \leq \frac{1}{s_n} \sum_{t\in\mathcal{J}(I_n)} (\mathrm{er}_t(\hat{Q}_t(I_n)) - \mathrm{er}_t(Q)) \tag{78}$$

$$+ \frac{2}{s_n} \sum_{t\in\mathcal{J}(I_n)} \mathrm{disc}(k_t, k_n). \tag{79}$$

From (22) we get with probability $1-\delta$ that

$$\frac{2}{s_n} \sum_{t\in\mathcal{J}(I_n)} \mathrm{disc}(k_t, k_n) \leq 2\varepsilon + \frac{2}{s_n} \sum_{t\in\mathcal{J}(I_n)} B_L(t, n, \frac{\delta}{n}). \tag{80}$$

Using Theorem S4 and S5 with $\lambda(I_n) = \eta_n$, we get with probability $1-\delta$

$$\sum_{t\in\mathcal{J}(I_n)} (\mathrm{er}_t(\hat{Q}_t(I_n)) - \mathrm{er}_t(Q)) \tag{81}$$

$$\leq \sum_{t\in\mathcal{J}(I_n)} \sum_{i=1}^{m_t} \frac{1}{m_t}(\mathbb{E}_{Q_{t,i}}\left[\ell(h(x_{t,i}), y_{t,i})\right] - \mathbb{E}_Q\left[\ell(h(x_{t,i}), y_{t,i})\right]) \tag{82}$$

$$+ \frac{1}{\eta_n} \sum_{t\in\mathcal{J}(I_n)} \sum_{i=1}^{m_t} \mathrm{KL}(Q_{t,i}|Q_{t,i-1}) + \frac{1}{\eta_n}\mathrm{KL}(Q|P) \tag{83}$$

$$+ 2\frac{\eta_n s_n}{\bar{m}_n} + \frac{1}{\eta_n}\log\frac{n}{\delta}. \tag{84}$$

The final statement follows from the regret bound of Theorem S3. □

Proof of Theorem 6. We start by applying the standard arguments from the transfer learning literature, see e.g. (Ben-David et al., 2007).

$$\mathrm{er}_n(\bar{Q}_n) - \mathrm{er}_n(Q) \tag{85}$$

$$\leq \frac{1}{s_n} \sum_{t\in\mathcal{J}(I_n)} (\mathrm{er}_t(\hat{Q}_t(I_n)) - \mathrm{er}_t(Q)) \tag{86}$$

$$+ \frac{2}{s_n} \sum_{t\in\mathcal{J}(I_n)} e(k_t, k_n) \tag{87}$$

$$+ \frac{2}{s_n} \sum_{t\in\mathcal{J}(I_n)} \lambda_{t,n}. \tag{88}$$

MACRO-Task is specifically designed to control the discrepancies, so that we get with probability $1-\delta$

$$\frac{2}{s_n} \sum_{t\in\mathcal{J}(I_n)} e(k_t, k_n) \leq 2\varepsilon + \frac{2}{s_n} \sum_{t\in\mathcal{J}(I_n)} B_U(U_t, U_n, \frac{\delta}{n}). \tag{89}$$

Using Theorem S4 and S5 with $\lambda(I_n) = \eta_n$, we get with probability $1-\delta$

$$\sum_{t\in\mathcal{J}(I_n)} (\mathrm{er}_t(\hat{Q}_t(I_n)) - \mathrm{er}_t(Q)) \tag{90}$$

$$\leq \sum_{t\in\mathcal{J}(I_n)} \sum_{i=1}^{m_t} \frac{1}{m_t}(\mathbb{E}_{Q_{t,i}}\left[\ell(h(x_{t,i}), y_{t,i})\right] - \mathbb{E}_Q\left[\ell(h(x_{t,i}), y_{t,i})\right]) \tag{91}$$

$$+ \frac{1}{\eta_n} \sum_{t\in\mathcal{J}(I_n)} \sum_{i=1}^{m_t} \mathrm{KL}(Q_{t,i}|Q_{t,i-1}) + \frac{1}{\eta_n}\mathrm{KL}(Q|P) \tag{92}$$

$$+ 2\frac{\eta_n s_n}{\bar{m}_n} + \frac{1}{\eta_n}\log\frac{n}{\delta}. \tag{93}$$

The final statement follows from the regret bound of Theorem S3. □

References

Abernethy, J., Bartlett, P., and Rakhlin, A. Multitask learning with expert advice. In *Workshop on Computational Learning Theory (COLT)*, 2007.

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

Amit, R. and Meir, R. Meta-learning by adjusting priors based on extended PAC-Bayes theory. *arXiv preprint arXiv:1711.01244*, 2017.

Ammar, H. B., Eaton, E., Ruvolo, P., and Taylor, M. Online multi-task learning for policy gradient methods. In *International Conference on Machine Learing (ICML)*, 2014.

Ando, R. K. and Zhang, T. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research (JMLR)*, 6:1817–1853, 2005.

Bartlett, P. L. and Mendelson, S. Rademacher and gaussian complexities: Risk bounds and structural results. *Journal of Machine Learning Research (JMLR)*, 3:463–482, 2002.

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.

Catoni, O. *Statistical learning theory and stochastic optimization: Ecole d'Eté de Probabilités de Saint-Flour XXXI-2001*. Springer, 2004.

Cavallanti, G., Cesa-Bianchi, N., and Gentile, C. Linear algorithms for online multitask classification. *Journal of Machine Learning Research (JMLR)*, 11:2901–2934, 2010.

Clingerman, C. and Eaton, E. Lifelong learning with Gaussian processes. In *European Conference on Machine Learning (ECML)*, 2017.

Crammer, K. and Mansour, Y. Learning multiple tasks using shared hypotheses. In *Conference on Neural Information Processing Systems (NIPS)*, 2012.

Crammer, K., Kearns, M., and Wortman, J. Learning from multiple sources. *Journal of Machine Learning Research (JMLR)*, 9:1757–1774, 2008.

Csiszár, I. I-divergence geometry of probability distributions and minimization problems. *Annals of Probability*, pp. 146–158, 1975.

Dekel, O., Long, P. M., and Singer, Y. Online learning of multiple tasks with a shared loss. *Journal of Machine Learning Research (JMLR)*, 8:2233–2264, 2007.

Denevi, G., Ciliberto, C., Stamos, D., and Pontil, M. Incremental learning-to-learn with statistical guarantees. *arXiv preprint arXiv:1803.08089*, 2018.

Ditzler, G. A study of an incremental spectral meta-learner for nonstationary environments. In *International Joint Conference on Neural Networks*, pp. 38–44, 2016.

Germain, P., Lacasse, A., Laviolette, F., and Marchand, M. Pac-bayesian learning of linear classifiers. In *International Conference on Machine Learing (ICML)*, 2009.

Kifer, D., Ben-David, S., and Gehrke, J. Detecting change in data streams. In *International Conference on Very Large Data Bases (VLDB)*, volume 30, 2004.

Kingma, D. P., Salimans, T., and Welling, M. Variational dropout and the local reparameterization trick. In *Conference on Neural Information Processing Systems (NIPS)*, 2015.

Kumagai, A. and Iwata, T. Learning future classifiers without additional data. In *Conference on Artificial Intelligence (AAAI)*, 2016.

Langford, J. and Shawe-Taylor, J. PAC-Bayes & margins. In *Conference on Neural Information Processing Systems (NIPS)*, 2003.

Li, W., Xu, Z., Xu, D., Dai, D., and Van Gool, L. Domain generalization and adaptation using low rank exemplar SVMs. *IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, 40(5):1114–1127, 2017.

Lugosi, G., Papaspiliopoulos, O., and Stoltz, G. Online multi-task learning with hard constraints. *Workshop on Computational Learning Theory (COLT)*, 2009.

Mansour, Y., Mohri, M., and Rostamizadeh, A. Domain adaptation: Learning bounds and algorithms. In *Workshop on Computational Learning Theory (COLT)*, 2009.

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

Murugesan, K., Liu, H., Carbonell, J., and Yang, Y. Adaptive smoothed online multi-task learning. In *Conference on Neural Information Processing Systems (NIPS)*, 2016.

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.

Pontil, M. and Maurer, A. Excess risk bounds for multitask learning with trace norm regularization. In *Workshop on Computational Learning Theory (COLT)*, pp. 55–76, 2013.

Ruvolo, P. and Eaton, E. Ella: An efficient lifelong learning algorithm. In *International Conference on Machine Learing (ICML)*, 2013.

Saha, A., Rai, P., DaumÃ, H., Venkatasubramanian, S., et al. Online learning of multiple tasks and their relationships. In *International Conference on Artificial Intelligence and Statistics (AISTATS)*, 2011.

Saha, B., Gupta, S., Phung, D., and Venkatesh, S. Multiple task transfer learning with small sample sizes. *Knowledge and Information Systems (KAIS)*, 46(2):315–342, 2016.

Yang, P., Zhao, P., and Gao, X. Robust online multitask learning with correlative and personalized structures. *IEEE Transactions on Knowledge and Data Engineering (T-KDE)*, 29(11):2510–2521, 2017.

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