Towards Large Scale Transfer Learning[*]

Alexandru Niculescu-Mizil
IBM T.J. Watson Research Center
Yorktown Heights, NY 10598

Transfer learning has been the focus of much research recently, and has been proven over and over again to lead to significant improvements in the performance, especially when training data is scarce. So far, however, transfer learning has been mostly considered in off-line learning settings, and has only been applied on a limited scale. Most problems it has been applied to have at most a few dozen tasks that are usually carefully chosen in advance to be related to each other.

Large scale problems, with hundreds, thousands, or even more tasks, such as personalized web search, personalized handwritten and speech recognition, or robot navigation, offer great opportunities for transfer learning for three reasons. First, with lots of tasks, there is a high chance that some tasks are closely related to each other, thus greatly benefiting from transfer. Second, having a large number of tasks makes it likely that the amount of training data available for each tasks is limited, which is where transfer learning provides the most benefit. Third, the cumulative benefits are greater when improvements are made on lots of tasks.

A major challenge in transfer learning is the development of methods that are robust to the presence of unrelated tasks. This is especially important when dealing with large number of tasks, since chances are that, for each task, there are vastly more dissimilar tasks than similar ones. Unfortunately, most current transfer learning methods operate under the assumption that all tasks are related, and their performance degrades rapidly in the presence of unrelated tasks.

Here, we present an on-line transfer learning algorithm with built in task selection capabilities. The main feature of the algorithm is its ability to filter out unrelated tasks with very little data and performance overhead, even when there are orders of magnitude more unrelated tasks than related ones. We achieve this by implementing a trust-based system that controls transfer among tasks: if trust is high between two tasks, then there will be a lot of transfer among them, while, if trust is low, almost no transfer happens. Trusts are dynamically adjusted throughout learning via a multiplicative updates scheme that ensures that trusts between unrelated tasks decrease exponentially fast.

To test the task selection capabilities of our method we ran a preliminary experiment on an 100 dimensional synthetic data set where we controlled the number of related and unrelated (decoy) tasks. The experiment was ran in an on-line setting: at each step, an example from a randomly selected task is presented to the learner, the learner makes a prediction, then it receives the true label and performs the necessary updates. Figure 1 shows the cumulative number of mistakes per task as the training progresses. Figure 2 shows the number of mistakes per task after 100 instances from each task are presented to the learner, when there are 20 related tasks and varying numbers of decoy tasks. As expected, without task selection, the transfer learning performance decreases rapidly as the number of unrelated tasks grows. Our algorithm, however, handles the decoy tasks well making, on average, a single extra mistake every time the number of decoy tasks is doubled. It only costs the algorithm five mistakes per task to filter out 620 unrelated tasks!

We also tested our method in an off-line setting on a, not so large scale, land mine detection problem with 19 tasks.[1] Figure 3 shows the area under the ROC curve vs. the number of passes through the training set. Without task selection, transfer learning does not provide any benefit. By filtering out unhelpful tasks, our method yields a significant performance boost.

[*] Topic: Transfer Learning; Preference: Oral

[1] Thanks to Lawrence Carin for making the data available

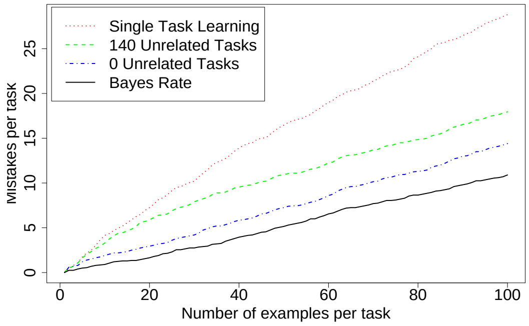

Figure 1: Number of mistakes as learning progresses for single task and multitask learning when there are 20 related tasks.

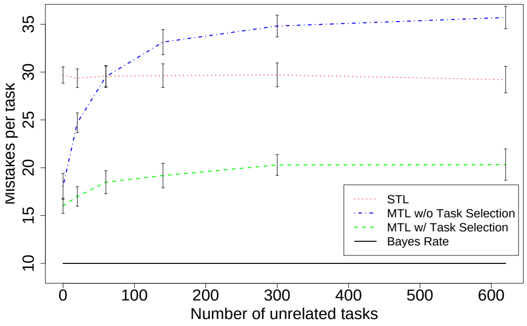

Figure 2: Number of mistakes after 100 example presentations for each task as a function of the number of unrelated tasks. Errorbars represent one standard deviation.

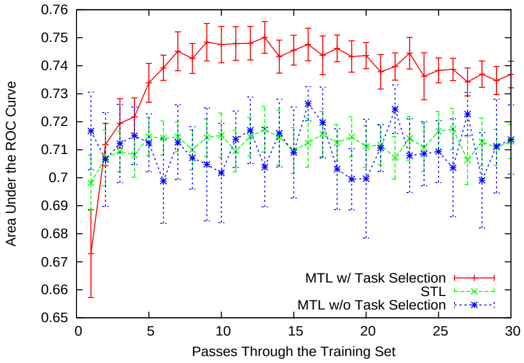

Figure 3: Average AUC performance for land mine detection.