LEARNING ALGORITHMS FOR ACTIVE LEARNING

Philip Bachman*, Alessandro Sordoni*, Adam Trischler
Maluuba Research
first.last@maluuba.com

ABSTRACT

We present a model that learns active learning algorithms via metalearning. For each metatask, our model jointly learns: a data representation, an item selection heuristic, and a one-shot classifier. Our model uses the item selection heuristic to construct a labeled support set for the one-shot classifier. Using metatasks based on the Omniglot and MovieLens datasets, we show that our model performs well in synthetic and practical settings.

1 INTRODUCTION

In active learning, a model selects the items for which it will observe labels, to maximize prediction performance and minimize labeling cost. Active learning is motivated by the observation that a model may perform better while training on less labeled data if it can choose the data on which it trains (Cohn et al., 1996). Previous work has proposed various heuristics for selecting items to label, such as choosing the item whose label the model is most uncertain about, or the item whose label is expected to maximally reduce the model’s uncertainty about labels for other items (Settles, 2010; Gilad-Bachrach et al., 2005; Houlsby et al., 2011).

We propose moving away from engineered selection heuristics towards learning active learning algorithms end-to-end via metalearning. Our model interacts with labeled items for many related tasks in order to learn an active learning strategy for the task at hand. In recommendation systems, for example, ratings data for existing users can inform a strategy that efficiently elicits preferences for new users who lack prior rating data, thus bootstrapping the system quickly out of the cold-start setting (Golbandi et al., 2010; 2011; Sun et al., 2013; Kawale et al., 2015). A learned strategy could outperform task-agnostic heuristics by sharing experience across related tasks. In particular, the model’s (i) data representation, (ii) strategy for selecting items to label, and (iii) prediction function could all co-adapt. Moving from pipelines of independently-engineered components to end-to-end learning has lead to rapid improvements in, e.g., computer vision, speech recognition, and machine translation (Krizhevsky et al., 2012; He et al., 2016; Hannun et al., 2014; Wu et al., 2016).

We build on the Matching Networks (MN) introduced by Vinyals et al. (2016). We extend the MN’s one-shot learning ability to settings where labels are not available a priori. We cast active learning as a sequential decision problem: at each step the model requests the label for a particular item, then adds this item to a labeled support set, which is used for MN-style prediction. We train our model end-to-end via reinforcement learning and backpropagation. We expedite the training process by allowing our model to observe and mimic a strong selection policy with oracle knowledge of the labels. We demonstrate empirically that our model learns effective active learning algorithms for both image classification and bootstrapping a movie recommendation system from a cold-start.

2 TASK AND MODEL DESCRIPTION

We summarize our model’s architecture and objectives in Figure 1 and Equation 2. Succinctly, our model iteratively picks items to label from a pool of unlabeled items, thereby constructing the labeled support set used by a Matching Network to classify test items. We train the underlying data representation, iterative item selection function, and MN end-to-end via metalearning.

Our model refines its behaviour over many training episodes, in order to maximize performance during test episodes not encountered in training. In each episode, our model interacts with a support set \( S \equiv \{(x, y)\} \) comprising items \( x \) for which the model can request labels \( y \), and a similarly-defined...
At each step, our model repeatedly samples an episode to optimize Equation 2. During training, our model optimizes the following objective:

$$D = \{(x, y), S_t = \{(x, \cdot), \cdot\} \rightarrow \mathcal{E} = \{(x, y), \cdot\}$$

To stabilize the advantage estimates used in GAE, we rely on a clear review of optimization techniques for general stochastic computation graphs, see Schulman et al. (2016). To expedite the training process, we construct our model’s selection policy as a mixture of its learned active policy and an “oracle” policy, which selects items to label in proportion to their current (exponentiated) prediction error. We anneal this mixture as training progresses until it contains only the learned policy. Our model thus learns (heuristic) algorithms by (noisy) demonstration. We stabilize the advantage estimates used in GAE by relying on improvements in prediction rewards, which effectively treats the previous step’s reward as a baseline.

Algorithm 1: End-to-end active learning loop.

1. For $t = 1, T$ do
   - Select next instance
   - Read labeled instance and update controller
   - Update control state
   - Perform fast prediction
   - Perform slow prediction

2. End
We run our first experiments on the Omniglot dataset \cite{Lake2015}, following the one-shot learning setup of \cite{Vinyals2016}. We test on 20-way classification: each episode has a support set $S$ with 5 items from each of 20 classes, and an evaluation set with 1 item per class. Character classes in training and testing are disjoint. We train our model to query 50 labels (of 100 in $S$). We first train the model to query 20 labels (i.e., 1-shot setting), and then fine-tune the model for 50 queries. While fine-tuning, we add an auxiliary reward which encourages a class-balanced selection policy. In the $k$-shot setting, we evaluate accuracy after $20 \times k$ items. We compare three baselines: our model with a random selection policy, a MN whose support set is a labeled random subsample of $S$ (MN-random, a lower bound on performance), and a MN whose support set is a class-balanced subsample from $S$ (MN-balanced, a near optimal policy). In Fig. 2(a), our model falls short of the class-balanced baseline in the one-shot setting, but comes closer in the two-shot setting. In all cases our model significantly outperforms the random selection baselines. In Fig. 2(b), we examine the generalization performance of our model when testing on problems with a different number of classes than were present in training. We train active learning policies for 5-way, 10-way, 15-way, and 20-way classification. We test all policies on 20-way classification, and measure how well they approximate the (near) optimal policy which collects a class-balanced set of labels. Aside from the 5-way-trained policy, our model’s algorithms generalize well and outperform the random policy baseline.

We run additional experiments using the MovieLens-20M dataset\footnote{Available at \url{http://grouplens.org/datasets/movielens/}} which contains real movie ratings from real users. Each episode samples a user $u$, then 50 movies rated by $u$ to include in the support set and 10 movies rated by $u$ for the evaluation set. We minimize mean-squared rating prediction error. We pretrained movie embeddings for all models using a standard matrix factorization approach on the training set. We compare against several baselines in Figure 2(c) (results are averaged over 3 different random seeds). The Regression baseline performs regularized linear regression on movies from the support set whose ratings have been observed up to time $t$, where labels are observed in random order. The Gaussian Process baseline selects the next movie to label in proportion to the variance of the predictive posterior distribution over its rating. This gives an idea of the impact of using MN one-shot capabilities rather than standard regression techniques. The Popular-Entropy, Min-Max-Cos, Entropy Sampling baselines train our model end-to-end, but using a fixed selection policy. This gives an idea of the importance of learning the selection policy. Popular-Entropy \cite{Elahi2016} aims to collect ratings for movies that are popular and have been rated differently by different users. Min-Max-Cos selects the unrated movie which has minimum maximum cosine similarity to any of the rated movies. Roughly, this selects the unrated movie which differs most from the rated movies. Entropy Sampling selects movies in proportion to rating prediction entropy. Our model outperforms the baselines in terms of RMSE, particularly after requesting only a few ratings. After 10 ratings, our model achieves an improvement of 2.5% in RMSE against the best baseline. Our model can be interpreted as learning a compact parametric representation of a decision tree for bootstrapping the movie recommendation system from a cold-start setting, as proposed by \cite{Sun2013}.

Figure 2: Results for our model on Omniglot (a, b), and MovieLens (c). (a) shows how prediction accuracy improves with the number of labels requested in a challenging 20-way setting. (b) shows the number of unique labels with respect to the number of requested labels, in a 20-way setting, for models trained on problems with different numbers of classes. The algorithms learned by our model generalize to different number of classes. (c) compares our active learning policy with several baselines on a cold-start recommendation task.

3 RESULTS
REFERENCES


