ELLA: An Efficient Lifelong Learning Algorithm

Paul Ruvolo
Bryn Mawr College

Eric Eaton
Bryn Mawr College

Abstract

The problem of learning multiple tasks that arrive sequentially, known as *lifelong learning*, is of great importance to the creation of intelligent, general-purpose, and flexible machines. This paper develops a method for online multitask learning in the lifelong learning setting. The proposed Efficient Lifelong Learning Algorithm (ELLA) maintains a sparsely shared basis for all task models, transfers knowledge from the basis to learn each new task, and refines the basis over time to maximize performance across all learned tasks. The proposed method has strong connections to both online dictionary learning for sparse coding and current batch multi-task learning methods, and provides robust theoretical performance guarantees. Empirically, ELLA yields nearly identical performance to batch multi-task learning while learning tasks sequentially in over 1,000x less time.

Introduction

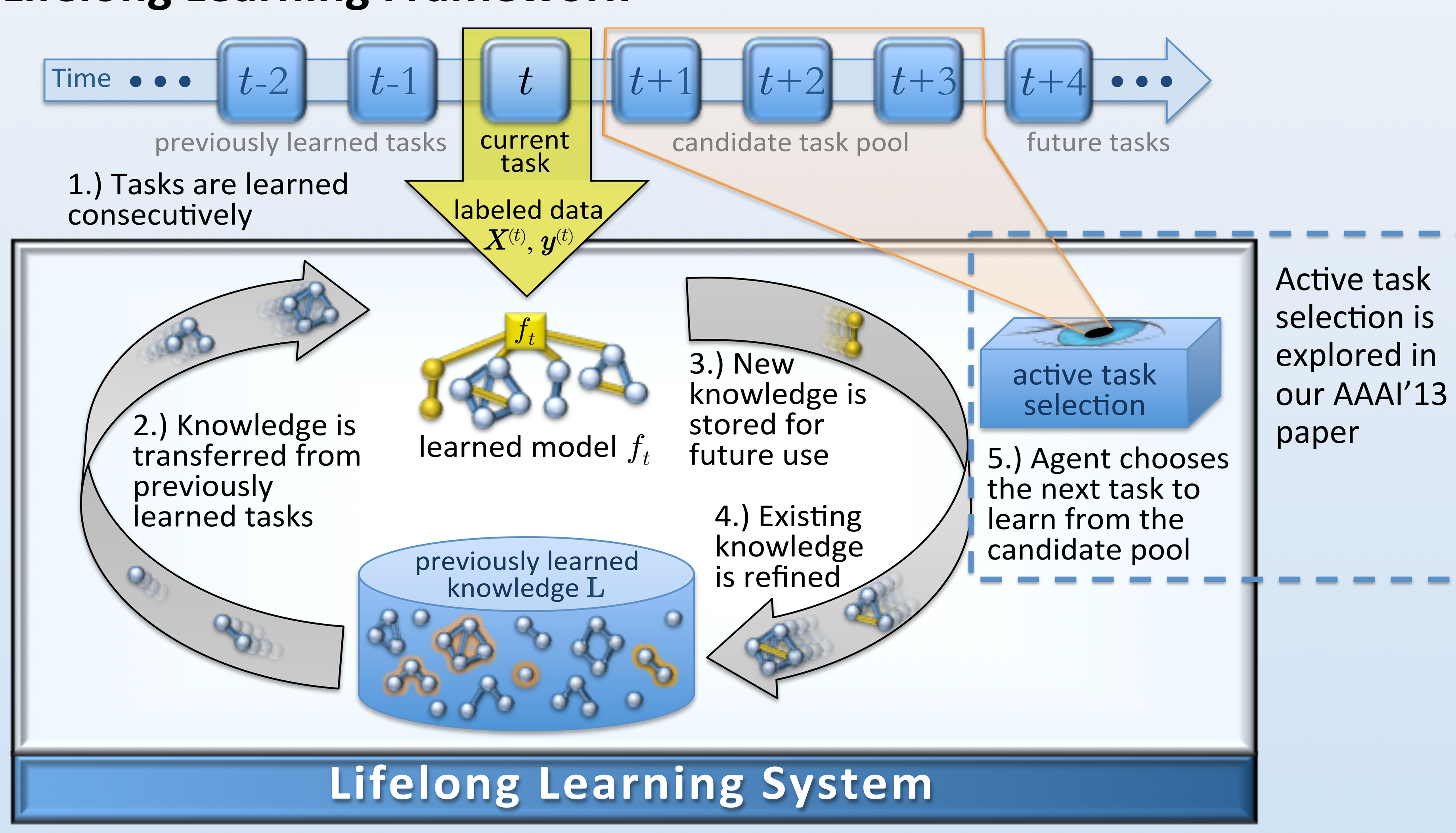
Goal: Develop intelligent agents that
1. Quickly learn new tasks
2. Learn continually with experience
3. Exhibit versatility over multiple tasks

	Transfer Learning	Batch Multi-Task Learning
Optimizes performance over	Target task	All tasks
Learns tasks consecutively	Yes, efficiently	Very inefficiently
Computational cost	Low	High

Lifelong learning includes elements of both transfer and multi-task learning

ELLA's Capabilities:
1. Optimized performance over all tasks
2. Efficient learning of each new consecutive task via transfer
3. Equivalent performance to batch MTL with over 1,000x speedup

Lifelong Learning Framework



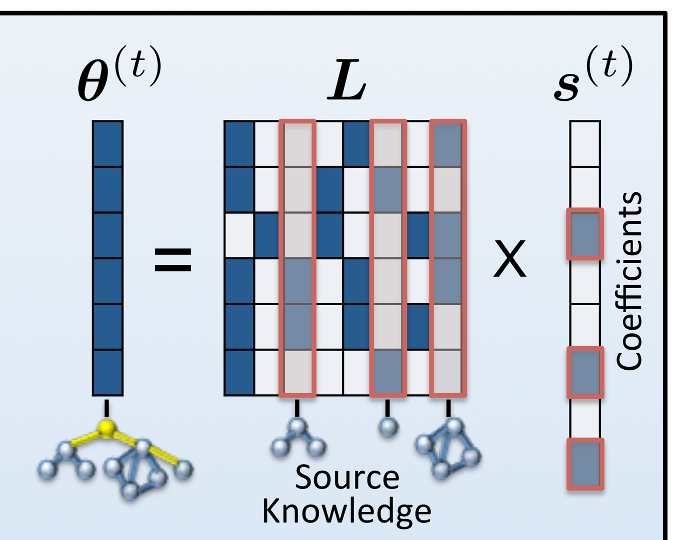
Task Structure Model

ELLA's goal is to fit a parametric model for each task t

$$f^{(t)}(\mathbf{x}) = f(\mathbf{x}; \boldsymbol{\theta}^{(t)}) \qquad \boldsymbol{\theta}^{(t)} \in \mathbb{R}^d$$

The parameter vectors for each function are assumed to be linear combinations of a shared latent basis $\mathbf{L}$

$$\boldsymbol{\theta}^{(t)} = \mathbf{L}\boldsymbol{s}^{(t)} \qquad \mathbf{L} \in \mathbb{R}^{d \times k},\ \boldsymbol{s}^{(t)} \in \mathbb{R}^k$$



We minimize the following objective function to encourage models to utilize few latent basis vectors:

$$e_T(\mathbf{L}) = \underbrace{\frac{1}{T}}_{\text{\#tasks seen so far}} \sum_{t=1}^{T} \min_{\boldsymbol{s}^{(t)}} \left\{ \underbrace{\frac{1}{n_t} \sum_{i=1}^{n_t} \mathcal{L}\left(f\left(\boldsymbol{x}_i^{(t)}; \mathbf{L}\boldsymbol{s}^{(t)}\right), y_i^{(t)}\right)}_{\text{model fit to data}} + \underbrace{\mu \|\boldsymbol{s}^{(t)}\|_1}_{\text{sparsity}} \right\} + \underbrace{\lambda \|\mathbf{L}\|_{\mathsf{F}}^2}_{\text{complexity}}$$

Efficient Lifelong Learning

Minimizing e_T is computationally expensive for two reasons:
1. Evaluating the objective function scales with the number of training instances n_t
2. The number of optimization problems grows linearly with the number of tasks T

To address (1) we replace the inner summation with the 2nd-order Taylor expansion around the optimal task-specific model: $\boldsymbol{\theta}^{(t)} = \arg\min_{\boldsymbol{\theta}} \frac{1}{n_t} \sum_{i=1}^{n_t} \mathcal{L}\left(f\left(\boldsymbol{x}_i^{(t)}; \boldsymbol{\theta}\right), y_i^{(t)}\right)$

To address (2) we optimize $\boldsymbol{s}^{(t)}$ only when training on task t and not on other tasks

These simplifications yield the following update equations to learn given $(\boldsymbol{X}^{(t)}, \boldsymbol{y}^{(t)})$:

$$\boldsymbol{s}^{(t)} \leftarrow \arg\min_{\boldsymbol{s}^{(t)}} \ell(\mathbf{L}_m, \boldsymbol{s}^{(t)}, \boldsymbol{\theta}^{(t)}, \boldsymbol{D}^{(t)})$$

$$\mathbf{L}_{m+1} \leftarrow \arg\min_{\mathbf{L}} \lambda \|\mathbf{L}\|_{\mathsf{F}}^2 + \frac{1}{T} \sum_{t=1}^{T} \ell\left(\mathbf{L}, \boldsymbol{s}^{(t)}, \boldsymbol{\theta}^{(t)}, \boldsymbol{D}^{(t)}\right)$$

where

$$\ell(\mathbf{L}, \mathbf{s}, \boldsymbol{\theta}, \mathbf{D}) = \mu \|\mathbf{s}\|_1 + \|\boldsymbol{\theta} - \mathbf{L}\mathbf{s}\|_{\mathbf{D}}^2$$

$\boldsymbol{D}^{(t)}$ is ½ the Hessian of the single-task loss evaluated at $\boldsymbol{\theta}^{(t)}$

Base Learning Algorithms

ELLA can support any base learner with a twice-differentiable loss function

Linear Regression: ($\mathbf{y}^{(t)} \in \mathbb{R}^{n_t}$, $f(\mathbf{x}; \boldsymbol{\theta}) = \boldsymbol{\theta}^\top \mathbf{x}$, and $\mathcal{L}(\cdot)$ is squared loss)

$$\boldsymbol{\theta}^{(t)} = \left(\boldsymbol{X}^{(t)} \boldsymbol{X}^{(t)\top}\right)^{-1} \boldsymbol{X}^{(t)} \boldsymbol{y}^{(t)}$$

$$\boldsymbol{D}^{(t)} = \frac{1}{2n_t} \boldsymbol{X}^{(t)} \boldsymbol{X}^{(t)\top}$$

Logistic Regression: ($\mathbf{y}^{(t)} \in \{-1,+1\}^{n_t}$, $f(\mathbf{x}; \boldsymbol{\theta}) = \left(1+e^{-\boldsymbol{\theta}^\top \mathbf{x}}\right)^{-1}$, and $\mathcal{L}(\cdot)$ is log-loss)

$\boldsymbol{\theta}^{(t)}$ is the logistic regression fit to $\boldsymbol{X}^{(t)}, \boldsymbol{y}^{(t)}$ using a standard solver

$$\boldsymbol{D}^{(t)} = \frac{1}{2n_t} \sum_{i=1}^{n_t} f(\boldsymbol{x}_i^{(t)}, \boldsymbol{\theta}^{(t)}) (1 - f(\boldsymbol{x}_i^{(t)}, \boldsymbol{\theta}^{(t)})) \boldsymbol{x}_i^{(t)} \boldsymbol{x}_i^{(t)\top}$$

Theory

Assumptions:
1. Tuples $(\mathbf{D}^{(t)}, \boldsymbol{\theta}^{(t)})$ are drawn i.i.d. from a distribution with compact support
2. The sparse coding solution is unique and is sensitive to changes in $s_\gamma^{(t)}$ (non-zero entries of $\boldsymbol{s}^{(t)}$): $\forall\, \mathbf{L}, \mathbf{D}^{(t)}$, and $\boldsymbol{\theta}^{(t)}$ the smallest eigenvalue of $\mathbf{L}_\gamma^\top \mathbf{D}^{(t)} \mathbf{L}_\gamma \geq \kappa > 0$

Theorems:
1. The basis $\mathbf{L}$ becomes more stable over time: $\mathbf{L}_{T+1} - \mathbf{L}_T = O\left(\frac{1}{T}\right)$
2. The penalty for not re-optimizing the $\boldsymbol{s}^{(t)}$'s vanishes as T gets large:

$$\hat{g}_T(\mathbf{L}) = \lambda \|\mathbf{L}\|_{\mathsf{F}}^2 + \tfrac{1}{T} \textstyle\sum_{t=1}^{T} \ell\left(\mathbf{L}, \boldsymbol{s}^{(t)}, \boldsymbol{\theta}^{(t)}, \boldsymbol{D}^{(t)}\right)$$

$$g_T(\mathbf{L}) = \lambda \|\mathbf{L}\|_{\mathsf{F}}^2 + \tfrac{1}{T} \textstyle\sum_{t=1}^{T} \min_{\boldsymbol{s}} \ell\left(\mathbf{L}, \boldsymbol{s}, \boldsymbol{\theta}^{(t)}, \boldsymbol{D}^{(t)}\right)$$

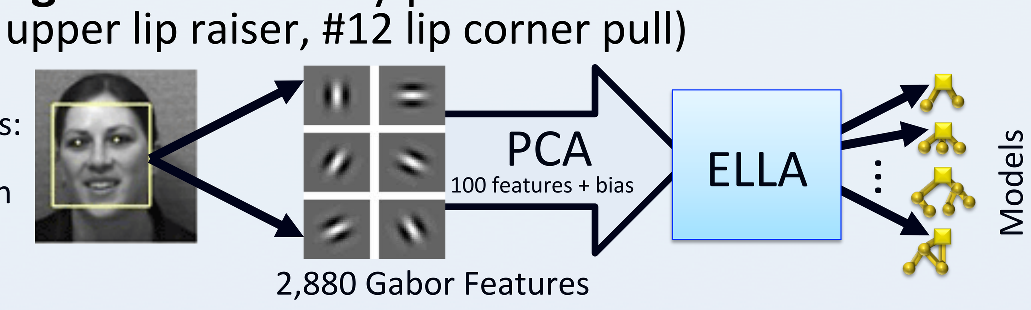
$$\text{as } T \to \infty,\ \hat{g}_T(\mathbf{L}_T) - g_T(\mathbf{L}_T) \text{ converges a.s. to } 0$$

3. The basis $\mathbf{L}$ converges to a fixed point of the expected loss e_T

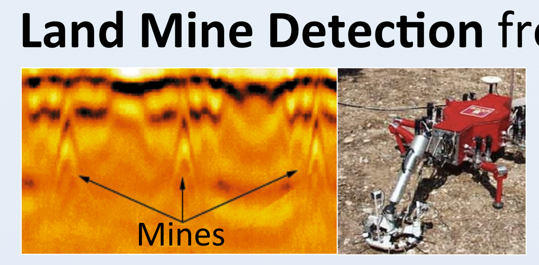
Connections to Dictionary Learning for Sparse Coding:
Online dictionary learning for sparse coding (Mairal et al., 2009) is a special case of ELLA where the $\boldsymbol{\theta}^{(t)}$'s are given instead of learned and the $\mathbf{D}^{(t)}$'s are identity matrices

Results

Facial Expression Recognition: identify presence of facial action units (#5 upper lid raiser, #10 upper lip raiser, #12 lip corner pull)

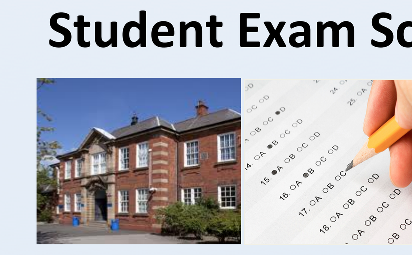


Land Mine Detection from radar images



29 Classification Tasks:
•29 regions
•2 terrain types
•14,820 instances total

Student Exam Score Prediction



139 Regression Tasks:
•139 schools
•15,362 students total
•4 school-specific features
•3 student-specific features

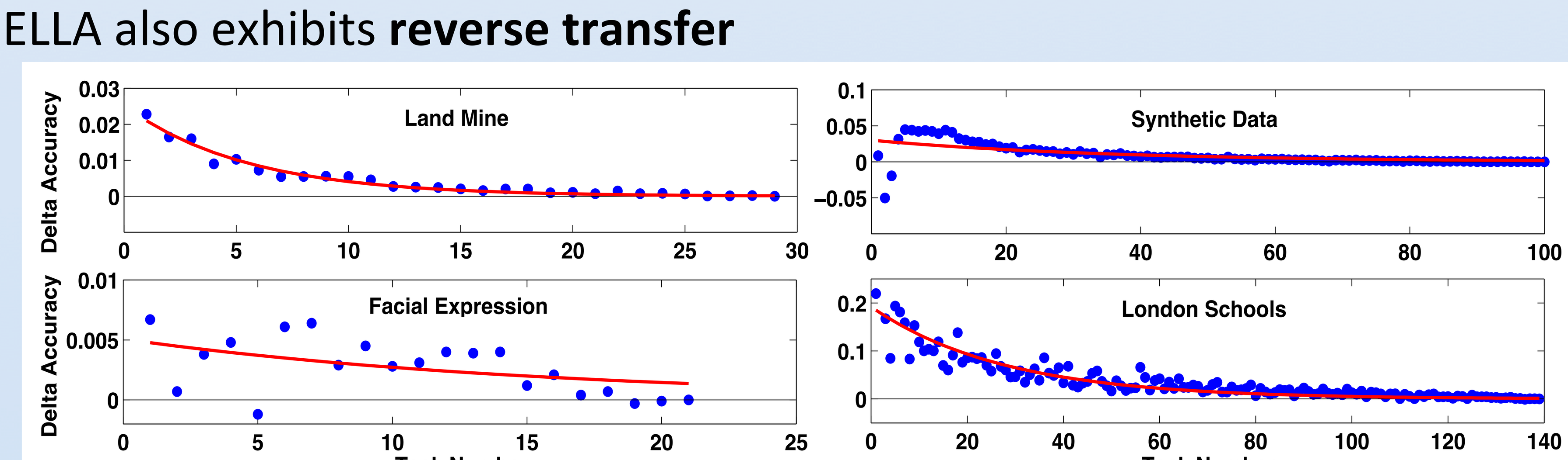
ELLA achieves nearly identical accuracy to batch MTL,

Dataset	Problem Type	Batch MTL Accuracy	ELLA Relative Accuracy	OMTL Relative Accuracy	STL Relative Accuracy
Land Mine	Classification	0.7802 ± 0.013 (AUC)	$99.73 \pm 0.7\%$	$82.2 \pm 3.0\%$	$97.97 \pm 1.5\%$
Facial Expr.	Classification	0.6577 ± 0.021 (AUC)	$99.37 \pm 3.1\%$	$97.58 \pm 3.8\%$	$97.34 \pm 3.9\%$
Syn. Data	Regression	-1.084 ± 0.006 (-rMSE)	$97.74 \pm 2.7\%$	N/A	$92.91 \pm 1.5\%$
London Sch.	Regression	-10.10 ± 0.066 (-rMSE)	$98.90 \pm 1.5\%$	N/A	$97.20 \pm 0.4\%$

while learning over 1,000 times faster

Dataset	Batch Runtime (seconds)	ELLA All Tasks (speedup)	ELLA New Task (speedup)	OMTL All Tasks (speedup)	OMTL New Task (speedup)	STL All Tasks (speedup)	STL New Task (speedup)
Land Mine	231±6.2	1,350±58	39,150±1,682	22±0.88	638±25	3,342±409	96,918±11,861
Facial Expr.	2,200±92	1,828±100	38,400±2,100	948±65	19,900±1,360	8,511±1,107	178,719±23,239
Syn. Data	1,300±141	5,026±685	502,600±68,500	N/A	N/A	156,489±17,564	1.6E6±1.8E5
London Sch.	715±36	2,721±225	378,219±31,275	N/A	N/A	36,000±4,800	5.0E6±6.7E5

ELLA also exhibits **reverse transfer**



Reverse transfer occurs when earlier tasks improve from later learning without retraining on the earlier tasks

Acknowledgement: This research was supported by ONR grant #N00014-11-1-0139

ELLA has equivalent accuracy to batch multi-task learning, but is 1,000x faster and can learn online