Online Multi-Task Learning via Sparse Dictionary Optimization

Paul Ruvolo

Olin College of Engineering

Eric Eaton

Penn Engineering

This work was supported by ONR Grant #N00014-11-1-0139

Motivation

Goal: Develop intelligent systems that

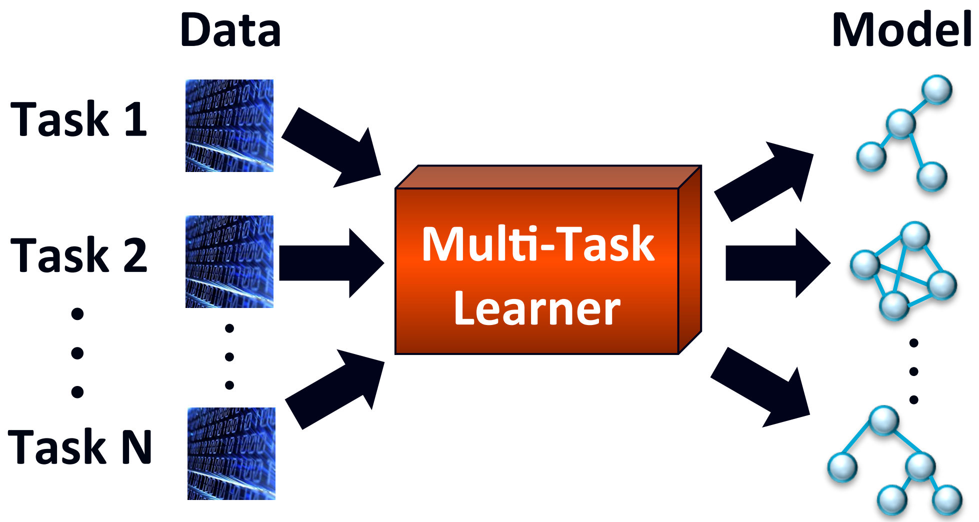
1. Quickly learn new tasks
2. Learn continually with experience
3. Exhibit versatility over multiple tasks

Accomplish these goals by sharing knowledge between tasks and with other agents

Sharing Knowledge Between Tasks

Multi-Task Learning

- Train task models simultaneously



Data

Model

Task 1

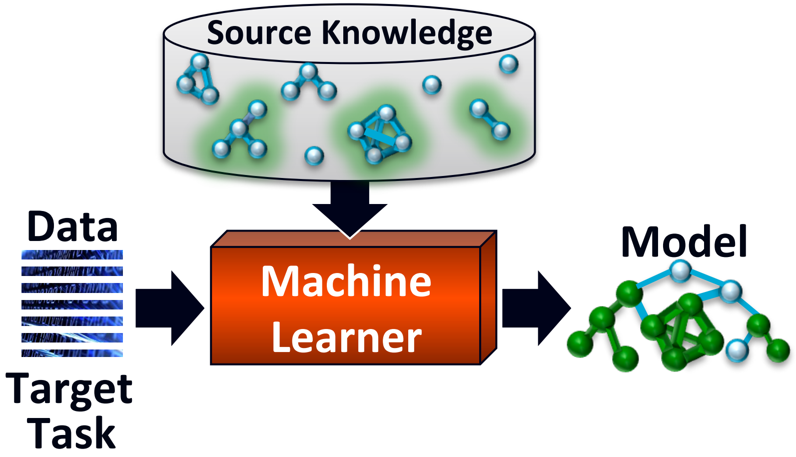
Task 2

Task N

Multi-Task Learner

Transfer Learning

- Transfer knowledge from source tasks to learn a new target task



Source Knowledge

Data

Target Task

Machine Learner

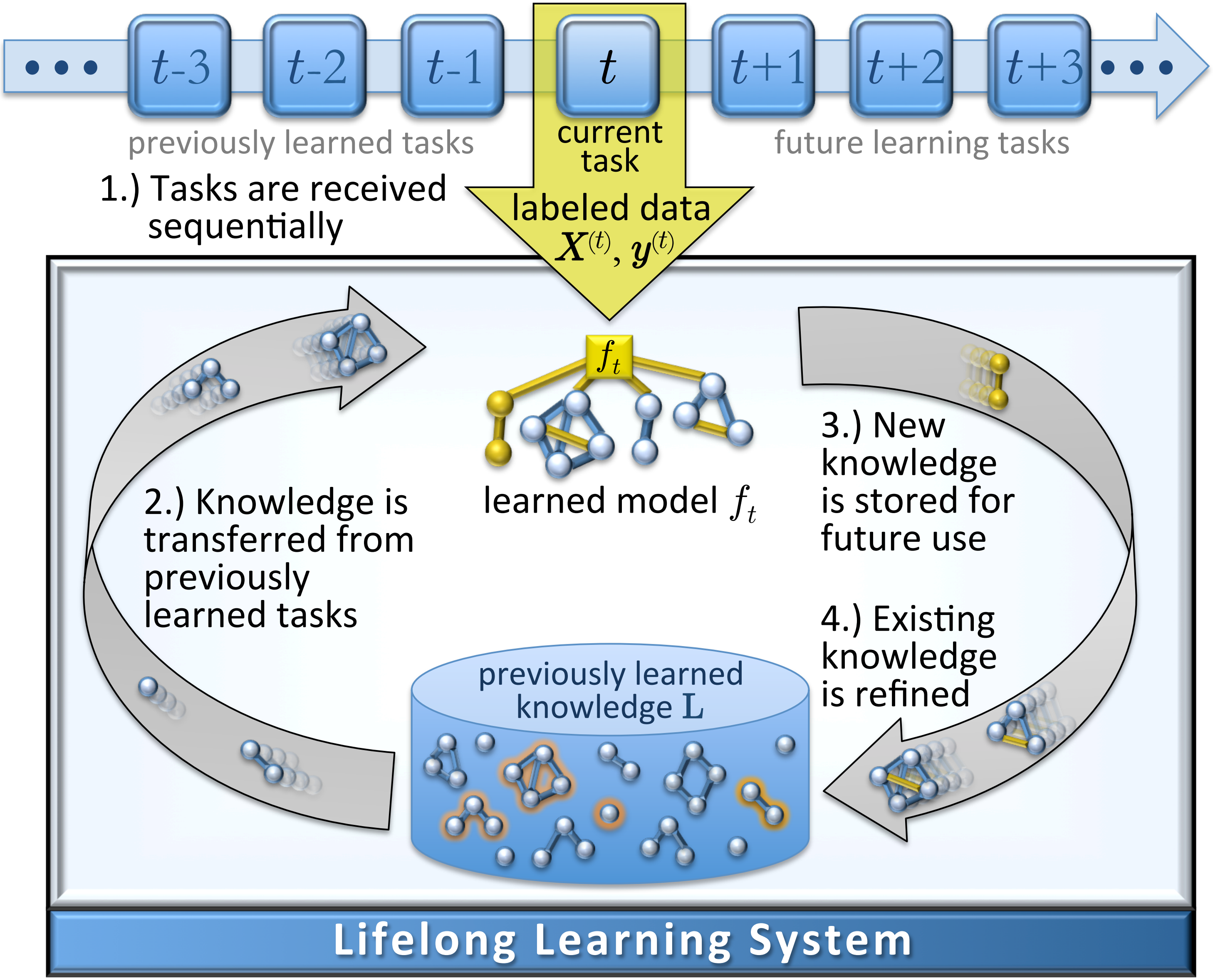
Model

Overview

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

- This work investigates online multi-task learning (MTL) based on sparse dictionary optimization
  - Evaluated in lifelong learning settings
  - Builds upon our earlier work on the Efficient Lifelong Learning Algorithm (ELLA) [Ruvolo & Eaton, ICML '13]

Online Multi-Task Learning



previously learned tasks

current task

future learning tasks

1.) Tasks are received sequentially

labeled data $\boldsymbol{X}^{(t)}, \boldsymbol{y}^{(t)}$

learned model f_t

2.) Knowledge is transferred from previously learned tasks

3.) New knowledge is stored for future use

4.) Existing knowledge is refined

previously learned knowledge $\mathbf{L}$

Lifelong Learning System

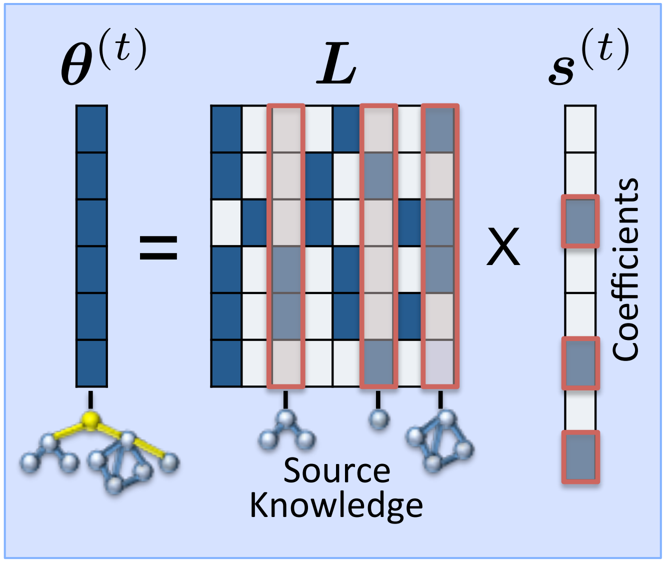
Task Structure Model

- We assume 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 parameters $\boldsymbol{\theta}^{(t)}$ are linear combinations of a shared basis $\mathbf{L}$

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



Multi-Task Learning Objective Fn:

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

Sparse Coding Connection

- We can re-write this MTL objective as a sparse coding problem [Ruvolo & Eaton, ICML '13]

$$e_T\left(\mathbf{L}\right) = \frac{1}{T}\sum_{t=1}^{T}\min_{\boldsymbol{s}^{(t)}}\left\{\frac{1}{n_t}\sum_{i=1}^{n_t}\mathcal{L}\left(f\left(\boldsymbol{x}_i^{(t)};\mathbf{L}\boldsymbol{s}^{(t)}\right),y_i^{(t)}\right)+\mu\|\boldsymbol{s}^{(t)}\|_1\right\}+\lambda\|\mathbf{L}\|_{\mathsf{F}}^2$$

$$g_T\left(\mathbf{L}\right) = \frac{1}{T}\sum_{t=1}^{T}\min_{\boldsymbol{s}^{(t)}}\left\{\|\boldsymbol{\theta}^{(t)}-\mathbf{L}\boldsymbol{s}^{(t)}\|_{\mathbf{D}^{(t)}}^2+\mu\|\boldsymbol{s}^{(t)}\|_1\right\}+\lambda\|\mathbf{L}\|_{\mathsf{F}}^2$$

where $\boldsymbol{\theta}^{(t)}$ is the optimal single-task model for task t

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

$\|\mathbf{x}\|_{\mathbf{D}}^2 = \mathbf{x}^\top\mathbf{D}\mathbf{x}$

Sparse Coding Connection

$$g_T\left(\mathbf{L}\right) = \frac{1}{T}\sum_{t=1}^{T}\min_{\boldsymbol{s}^{(t)}}\left\{\|\boldsymbol{\theta}^{(t)} - \mathbf{L}\boldsymbol{s}^{(t)}\|_{\mathbf{D}^{(t)}}^{2} + \mu\|\boldsymbol{s}^{(t)}\|_1\right\} + \lambda\|\mathbf{L}\|_{\mathsf{F}}^{2}$$

Question: Are there dictionary learning algorithms we can borrow from the sparse-coding literature to efficiently solve $g_T()$?

K-SVD [Aharon et al. 2006]

Objective Function:

$$\arg\min_{\mathbf{L}} \sum_{i=1}^{n} \min_{\mathbf{s}^{(i)}} \left\{ \|\mathbf{L}\mathbf{s}^{(i)} - \mathbf{x}_i\|_2^2 + \mu \|\mathbf{s}^{(i)}\|_0 \right\}$$

The k-SVD algorithm iterates two steps until convergence:

Step 1: update codes for each point

$$\mathbf{s}^{(i)} \leftarrow \arg\min_{\mathbf{s}} \left\{ \|\mathbf{L}\mathbf{s} - \mathbf{x}_i\|_2^2 + \mu \|\mathbf{s}\|_0 \right\}$$

Step 2: update each basis vector and the weights of the data points that utilize this basis vector

$$m \in \mathcal{A} \Leftrightarrow \mathbf{s}_j^{(m)} \neq 0$$

$$\mathbf{l}_j, \mathbf{s}_j^{(\mathcal{A})} \leftarrow \arg\min_{\mathbf{l}_j, \mathbf{s}_j^{(\mathcal{A})}} \sum_{i=1}^{n} \left(\|\mathbf{L}\mathbf{s}^{(i)} - \mathbf{x}_i\|_2^2 + \mu \|\mathbf{s}^{(i)}\|_0 \right)$$

K-SVD [Aharon et al. 2006]

Step 2 Objective Function:

$$\mathbf{l}_j, \mathbf{s}_j^{(\mathcal{A})} \leftarrow \arg\min_{\mathbf{l}_j, \mathbf{s}_j^{(\mathcal{A})}} \sum_{i=1}^{n} \left(\|\mathbf{L}\mathbf{s}^{(i)} - \mathbf{x}\|_2^2 + \mu \|\mathbf{s}^{(i)}\|_0 \right)$$

Step 2 Solution:

$$\mathbf{e}_i = \mathbf{x}_i - \sum_{r \neq j} s_r^{(i)} \mathbf{l}_r$$

$$(\mathbf{U}, \mathbf{\Sigma}, \mathbf{V}) = \mathrm{svd}\left(\mathbf{E}_{\mathcal{A}}\right) \qquad m \in \mathcal{A} \Leftrightarrow \mathbf{s}_j^{(m)} \neq 0$$

$$\mathbf{l}_j \leftarrow \mathbf{u}_1$$

$$\mathbf{s}_j^{(\mathcal{A})} \leftarrow \sigma_{1,1} \mathbf{v}_1$$

Surprisingly we can efficiently find the global minimum!

Adapting K-SVD to Multi-Task Learning

MTL Objective Function:

$$\arg\min_{\boldsymbol{L}} \sum_{t=1}^{T} \min_{\boldsymbol{s}^{(t)}} \left\{ \|\boldsymbol{\theta}^{(t)} - \boldsymbol{L}\boldsymbol{s}^{(t)}\|_{\boldsymbol{D}^{(t)}}^2 + \mu\|\boldsymbol{s}^{(t)}\|_0 \right\} + \lambda\|\boldsymbol{L}\|_{\mathsf{F}}^2$$

K-SVD Objective Function:

$$\arg\min_{\boldsymbol{L}} \sum_{t=1}^{T} \min_{\boldsymbol{s}^{(t)}} \left\{ \|\boldsymbol{\theta}^{(t)} - \boldsymbol{L}\boldsymbol{s}^{(t)}\|_2^2 + \mu\|\boldsymbol{s}^{(t)}\|_0 \right\}$$

Key Idea: Use K-SVD to efficiently solve the MTL objective

- Need to use the generalized SVD $(\mathbf{U}, \mathbf{\Sigma}, \mathbf{V}) = \mathrm{gsvd}\left(\mathbf{E}_{\mathcal{A}}, \mathbf{M}, \mathbf{W}\right)$ instead of SVD to account for 2nd order information, where

$$\mathbf{M} = \frac{1}{|\mathcal{A}_j|} \sum_{t' \in \mathcal{A}_j} \mathbf{D}^{(t')} \qquad w_t = \frac{\mathbf{1}^\top \mathbf{D}^{(t)} \mathbf{1}}{\sum_{t' \in \mathcal{A}_j} \mathbf{1}^\top \mathbf{D}^{(t')} \mathbf{1}}$$

feature relationship matrix

task relationship matrix

ELLA-SVD

MTL Objective Function: (fit via iterative optimization)

$$e_T\left(\mathbf{L}\right) = \frac{1}{T}\sum_{t=1}^{T} \min_{\boldsymbol{s}^{(t)}} \left\{ \frac{1}{n_t}\sum_{i=1}^{n_t} \mathcal{L}\left(f\left(\boldsymbol{x}_i^{(t)};\mathbf{L}\boldsymbol{s}^{(t)}\right), y_i^{(t)}\right) + \mu\|\boldsymbol{s}^{(t)}\|_1 \right\} + \lambda\|\mathbf{L}\|_{\mathsf{F}}^2$$

ELLA-SVD: Given a new task t,

1. Train a single-task model $\boldsymbol{\theta}^{(t)}$ for task t
2. Reconstruct $\boldsymbol{\theta}^{(t)}$ in the current basis (LASSO)

$$\mathbf{s}^{(t)} \leftarrow \arg\min_{\mathbf{s}} \left\{ \|\mathbf{L}\mathbf{s} - \boldsymbol{\theta}^{(t)}\|_{\mathbf{D}^{(t)}}^2 + \mu\|\mathbf{s}\|_0 \right\}$$

3. Update the basis

For each $j \in \{1 \ldots k\}$ where $\mathbf{s}_j^{(t)} \neq 0$

$$\mathbf{l}_j, \mathbf{s}_j^{(\mathcal{A})} \leftarrow \arg\min_{\mathbf{l}_j, \mathbf{s}_j^{(\mathcal{A})}} \sum_{t=1}^{T} \left(w_t \|\mathbf{L}\mathbf{s}^{(t)} - \boldsymbol{\theta}^{(t)}\|_{\mathbf{M}}^2 + \mu\|\mathbf{s}^{(t)}\|_0 \right)$$

where:

$$\mathbf{M} = \frac{1}{|\mathcal{A}_j|}\sum_{t' \in \mathcal{A}_j} \mathbf{D}^{(t')} \qquad w_t = \frac{\mathbf{1}^\top \mathbf{D}^{(t)} \mathbf{1}}{\sum_{t' \in \mathcal{A}_j} \mathbf{1}^\top \mathbf{D}^{(t')} \mathbf{1}}$$

Per-Task Computational Complexity

ELLA-SVD: O(base learner $+ d^2k + k^2d + qd^3 + qr^2d$)

q = sparsity of $s^{(t)}$

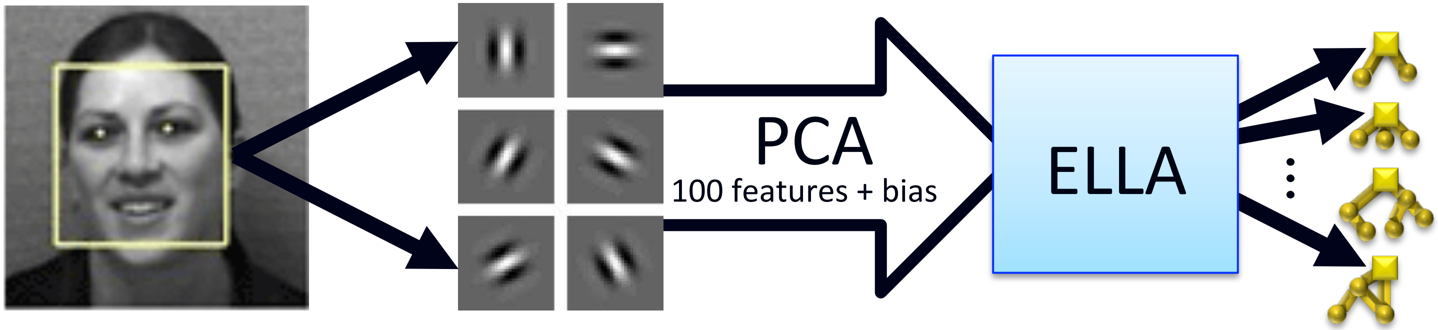
r = # tasks utilizing same basis component

ELLA: O(base learner + d^3k^2)

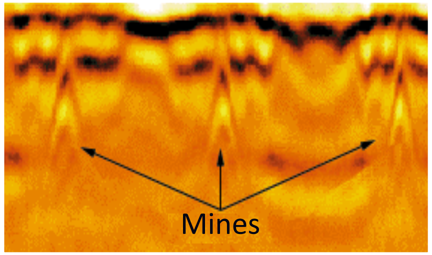
ELLA-SVD is much more efficient than the original ELLA

Applications

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

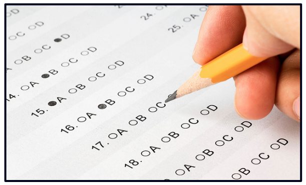


Land Mine Detection from radar images [Xue et al. 2007]



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

Exam Score Prediction for London schools [Kumar et al. 2012]



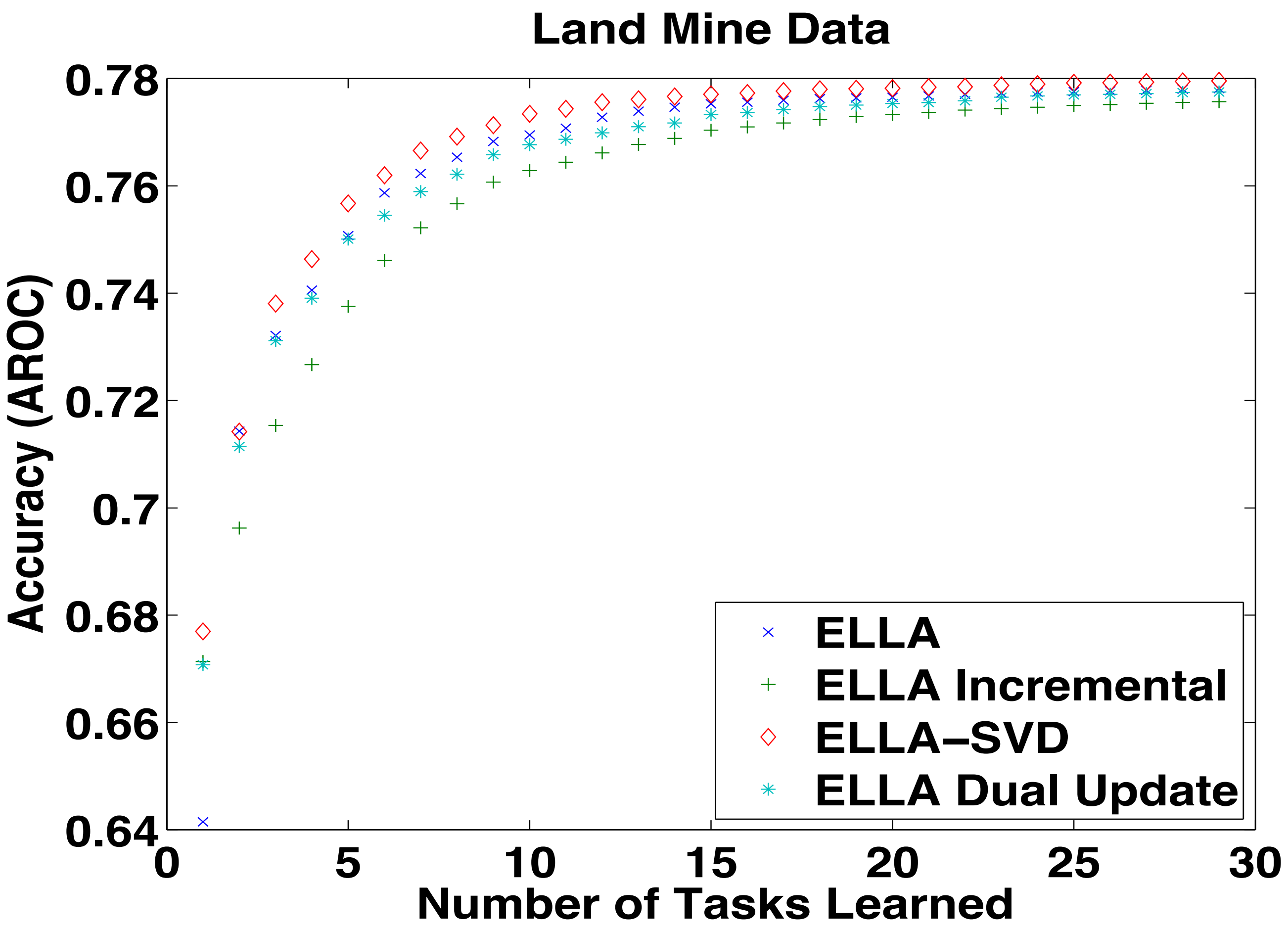
139 Regression Tasks:
- 139 schools
- 15,362 students total
- 4 school-specific features
- 3 student-specific features
- Exam year + bias term

Experiments

- We tested four methods
- Each method has the same first step of updating the weights, $\mathbf{s}^{(t)}$, for the current task
- The second step depends on the algorithm
    - **ELLA** [Ruvolo & Eaton, ICML '13]**:** update all columns of $\mathbf{L}$ jointly
    - **ELLA Incremental:** update columns of $\mathbf{L}$ one at a time (a more efficient but suboptimal version of ELLA)
    - **ELLA-SVD:** update each column of $\mathbf{L}$ and the corresponding entries of $\mathbf{S}$ jointly
    - **ELLA Dual Update:** execute ELLA-SVD update and then ELLA Incremental update (a hybrid approach)

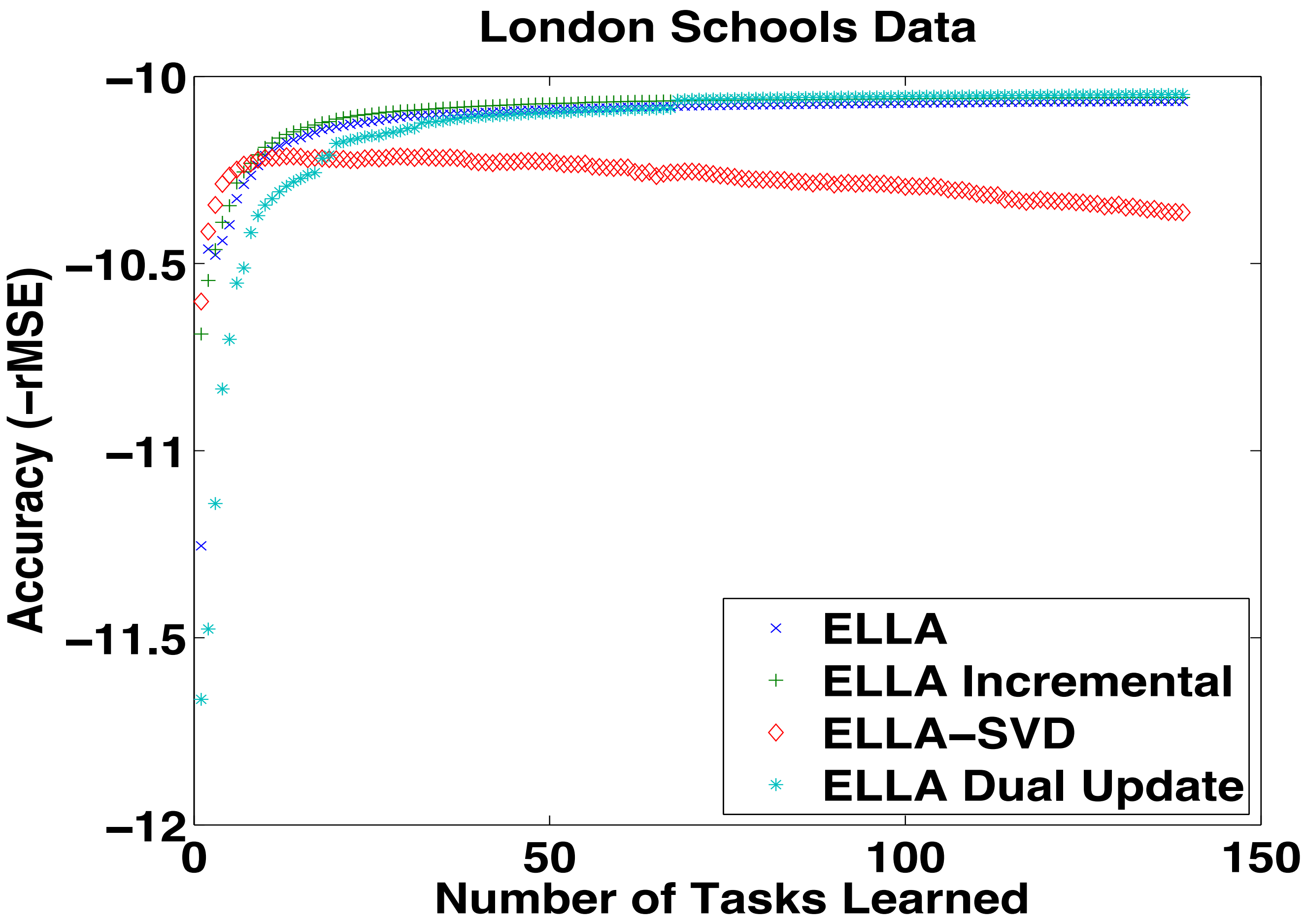
Results

In some cases ELLA-SVD works really well…

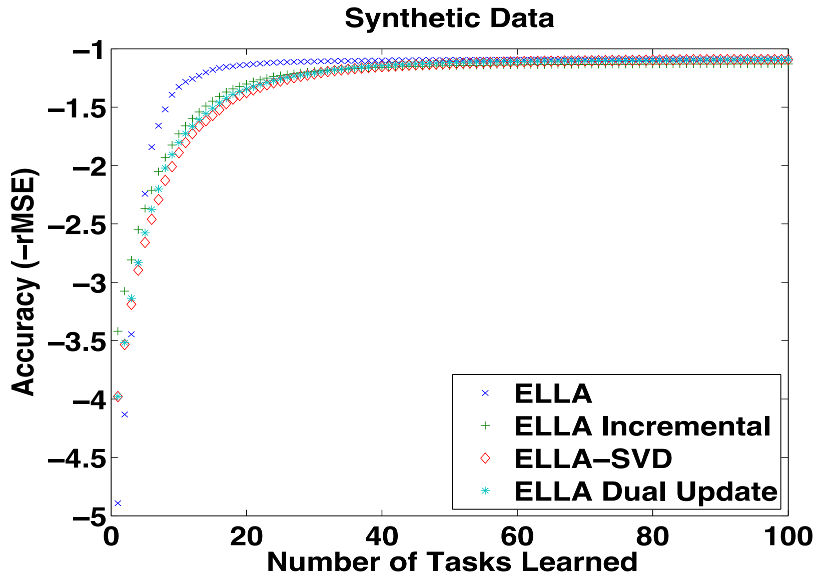


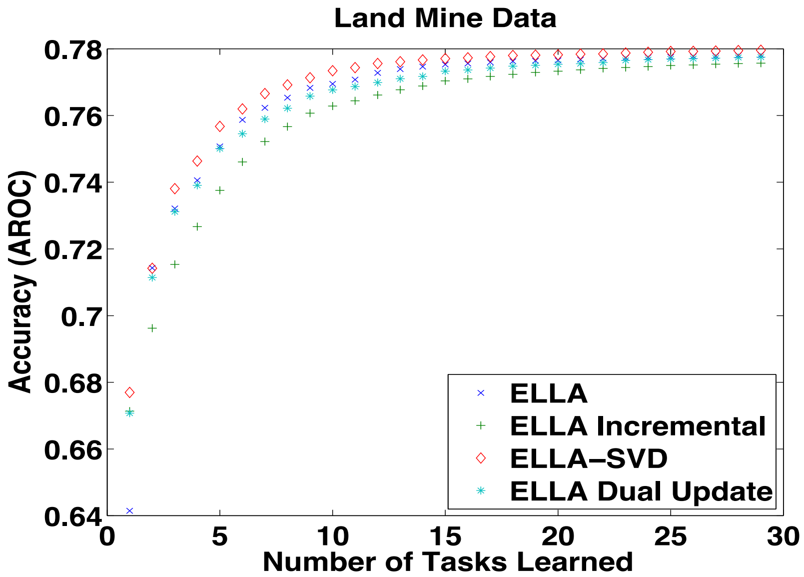
Results

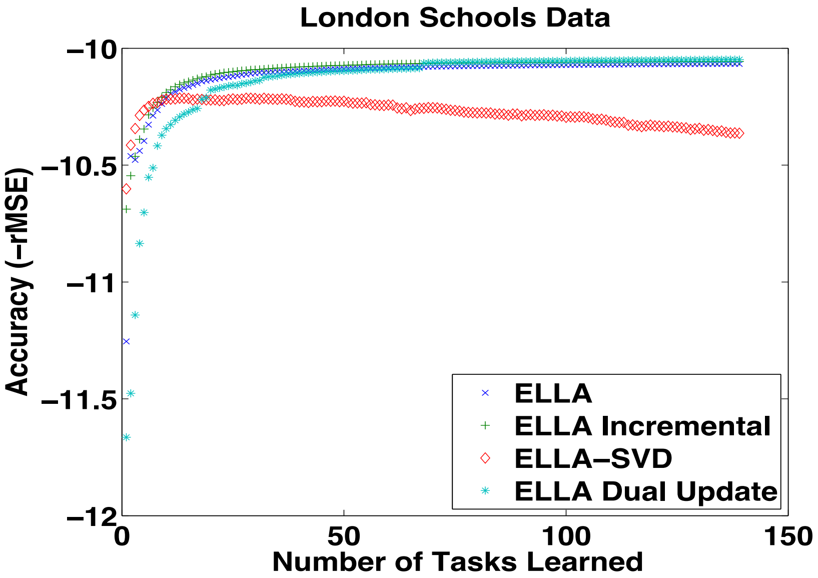
ELLA-SVD can suffer if the feature similarity matrix is set incorrectly (in this case, due to school-specific features in this data set)

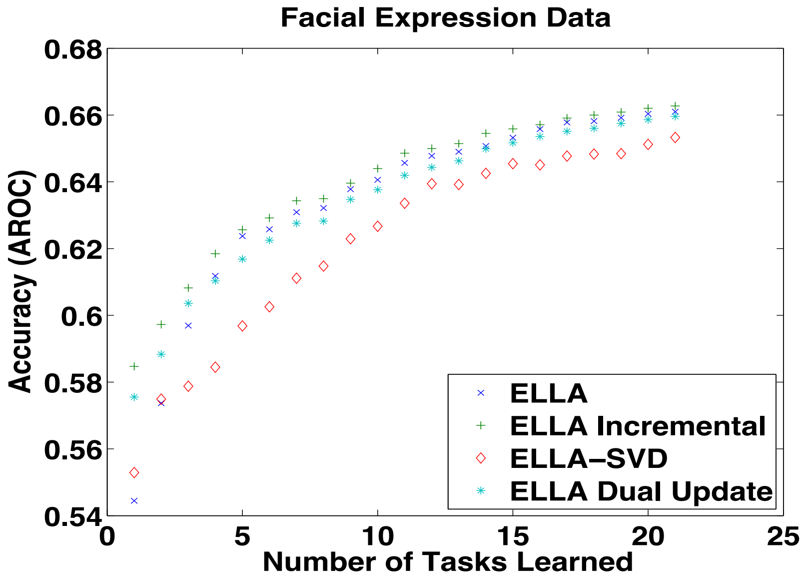


Results


Synthetic Data


Land Mine Data


London Schools Data


Facial Expression Data

Summary

- The k-SVD algorithm can be adapted to the multi-task learning setting
- Combining two update methods yields an algorithm with good computational complexity and accuracy (ELLA Dual Update)

Thank you!

Paul Ruvolo
paul.ruvolo@olin.edu

Eric Eaton
eeaton@seas.upenn.edu

This work was supported by ONR Grant #N00014-11-1-0139

Backup Slides

Adapting K-SVD to Multi-Task Learning

Multi-task Learning Objective Function:

$$g_T(\mathbf{L}) = \frac{1}{T}\sum_{t=1}^{T}\min_{\boldsymbol{s}^{(t)}}\left\{\|\boldsymbol{\theta}^{(t)} - \mathbf{L}\boldsymbol{s}^{(t)}\|_{\mathbf{D}^{(t)}}^2 + \mu\|\boldsymbol{s}^{(t)}\|_0\right\} + \lambda\|\mathbf{L}\|_{\mathsf{F}}^2$$

Two-step procedure:

Step 1 is almost identical to k-SVD

$$\mathbf{s}^{(t)} \leftarrow \arg\min_{\mathbf{s}}\left\{\|\mathbf{L}\mathbf{s} - \boldsymbol{\theta}^{(t)}\|_{\mathbf{D}^{(t)}}^2 + \mu\|\mathbf{s}\|_0\right\}$$

Adapting K-SVD to Multi-Task Learning

Multi-task Learning Objective Function:

$$g_T(\mathbf{L}) = \frac{1}{T}\sum_{t=1}^{T} \min_{\boldsymbol{s}^{(t)}} \left\{ \|\boldsymbol{\theta}^{(t)} - \mathbf{L}\boldsymbol{s}^{(t)}\|_{\mathbf{D}^{(t)}}^2 + \mu\|\boldsymbol{s}^{(t)}\|_0 \right\} + \lambda\|\mathbf{L}\|_{\mathsf{F}}^2$$

Step 2 Goal:

$$\mathbf{l}_j, \mathbf{s}_j^{(\mathcal{A})} \leftarrow \arg\min_{\mathbf{l}_j, \mathbf{s}_j^{(\mathcal{A})}} \sum_{t=1}^{T} \left(\|\mathbf{L}\mathbf{s}^{(t)} - \boldsymbol{\theta}^{(t)}\|_{\mathbf{D}^{(t)}}^2 + \mu\|\mathbf{s}^{(t)}\|_0 \right)$$

$$m \in \mathcal{A} \Leftrightarrow \mathbf{s}_j^{(m)} \neq 0$$

Problem: the SVD step in the k-SVD algorithm minimizes

$$\mathbf{l}_j, \mathbf{s}_j^{(\mathcal{A})} \leftarrow \arg\min_{\mathbf{l}_j, \mathbf{s}_j^{(\mathcal{A})}} \sum_{i=1}^{n} \left(\|\mathbf{L}\mathbf{s}^{(i)} - \mathbf{x}_i\|_2^2 + \mu\|\mathbf{s}^{(i)}\|_0 \right)$$

Generalized K-SVD

Step 2 Goal:

$$\mathbf{l}_j, \mathbf{s}_j^{(\mathcal{A})} \leftarrow \arg\min_{\mathbf{l}_j, \mathbf{s}_j^{(\mathcal{A})}} \sum_{t=1}^{T} \left(\|\mathbf{L}\mathbf{s}^{(t)} - \boldsymbol{\theta}^{(t)}\|_{\mathbf{D}^{(t)}}^2 + \mu \|\mathbf{s}^{(t)}\|_0 \right)$$

By replacing the SVD in step 2 with the generalized SVD we can efficiently minimize:

$$\mathbf{l}_j, \mathbf{s}_j^{(\mathcal{A})} \leftarrow \arg\min_{\mathbf{l}_j, \mathbf{s}_j^{(\mathcal{A})}} \sum_{t=1}^{T} \left(w_t \|\mathbf{L}\mathbf{s}^{(t)} - \boldsymbol{\theta}^{(t)}\|_{\mathbf{M}}^2 + \mu \|\mathbf{s}^{(t)}\|_0 \right)$$

Where $\mathbf{M}$ is PSD and $\mathbf{w}$ has all positive entries:

$$\mathbf{M} = \frac{1}{|\mathcal{A}_j|} \sum_{t' \in \mathcal{A}_j} \mathbf{D}^{(t')} \qquad w_t = \frac{\mathbf{1}^\top \mathbf{D}^{(t)} \mathbf{1}}{\sum_{t' \in \mathcal{A}_j} \mathbf{1}^\top \mathbf{D}^{(t')} \mathbf{1}}$$