Regret Bounds for Lifelong Learning

Pierre Alquier

ENSAE ParisTech
École nationale de la statistique et de l'administration économique
université PARIS-SACLAY

May 24, 2018
Workshop on Multi-Armed Bandits & Learning Algorithms
Rotterdam School of Management, Erasmus University

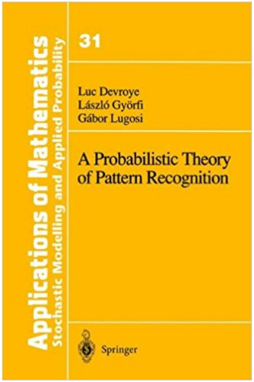
1. Transfer learning, multitask learning, lifelong learning...
2. A strategy for lifelong learning, with regret analysis
3. Open questions

Batch learning

Predict label Y from object X based on some data,

- data often assumed i.i.d from P,
- build $\hat{f}$ based on the whole dataset,
- minimize $R(\hat{f})$ where

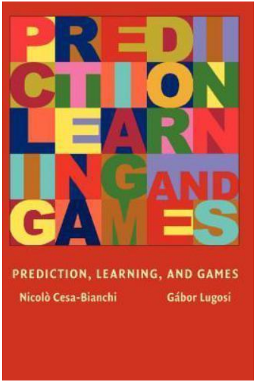
$$R(f) = \mathbb{E}_{(X,Y)\sim P}[\ell(Y, f(X))]$$



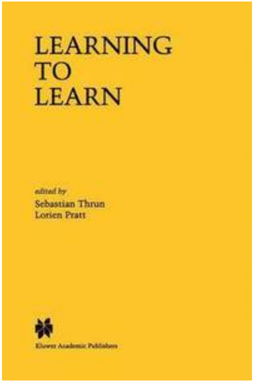
Online learning

- no probabilistic assumption,
- data revealed sequentially, at time t build $\hat{f}_t$ based on data seen so far
- minimize

$$\sum_{t=1}^{T} \ell(Y_t, \hat{f}_t(X_t))$$



Tentative definition - from Thrun and Pratt

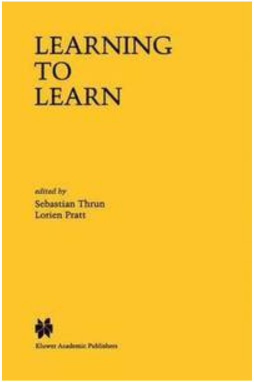


Given

- a task,
- a training experience, and
- a performance measure,

a program is said to learn if its performance at the task improves with experience.

Tentative definition - from Thrun and Pratt



Given

- a **family of** tasks,
- training experience for each of these tasks, and
- a family of performance measures,

an algorithm is said to **learn to learn** if its performance at each task improve with experience **and with the number of tasks.**

Multitask learning

Multitask learning

Given M tasks t, with M risks $R_t(\cdot)$ and M datasets

$$\mathcal{S}_t := \Big((X_{t,1}, Y_{t,1}), \dots, (X_{t,n_M}, Y_{t,n_M})\Big)$$

propose M predictors

$$\hat{f}_t(\cdot) = \hat{f}_t(\mathcal{S}_1, \dots, \mathcal{S}_M; \cdot)$$

that aims at minimizing (for example)

$$R_1(\hat{f}_1) + \dots + R_M(\hat{f}_M).$$

Multitask learning

Multitask learning

Given M tasks t, with M risks $R_t(\cdot)$ and M datasets

$$\mathcal{S}_t := \Big((X_{t,1}, Y_{t,1}), \dots, (X_{t,n_M}, Y_{t,n_M})\Big)$$

propose M predictors

$$\hat{f}_t(\cdot) = \hat{f}_t(\mathcal{S}_1, \dots, \mathcal{S}_M; \cdot)$$

that aims at minimizing (for example)

$$R_1(\hat{f}_1) + \dots + R_M(\hat{f}_M).$$

Nice, but what if yet another new task appears ?

Learning-to-learn

Learning-to-learn (LTL)

Given M tasks t with risk $R_t(\cdot)$, and M datasets

$$\mathcal{S}_t := \Big((X_{t,1}, Y_{t,1}), \dots, (X_{t,n_M}, Y_{t,n_M})\Big)$$

learn information $\mathcal{I} = \mathcal{I}(\mathcal{S}_1, \dots, \mathcal{S}_M)$ such that, when a **new** task with risk $R(\cdot)$ and a new dataset

$$\mathcal{S} := \Big((X_1, Y_1), \dots, (X_n, Y_n)\Big)$$

arrives, I can build a predictor

$$\hat{f}_t(\cdot) = \hat{f}_t(\mathcal{S}, \mathcal{I}; \cdot) \text{ such that } R(\hat{f}) \text{ is small.}$$

Probabilistic setting for LTL

Possible probabilistic setting :

Probabilistic setting for LTL

Possible probabilistic setting :

- $P_1, \dots, P_M$ i.i.d from $\mathcal{P}$,

Probabilistic setting for LTL

Possible probabilistic setting :

- $P_1, \ldots, P_M$ i.i.d from $\mathcal{P}$,
- $(X_{t,1}, Y_{t,1}), \ldots, (X_{t,n_M}, Y_{t,n_M})$ i.i.d from P_t,

Probabilistic setting for LTL

Possible probabilistic setting :

- $P_1, \dots, P_M$ i.i.d from $\mathcal{P}$,
- $(X_{t,1}, Y_{t,1}), \dots, (X_{t,n_M}, Y_{t,n_M})$ i.i.d from P_t,
- $R_t(f) = \mathbb{E}_{(X,Y)\sim P_t}[\ell(Y, f(X))]$,

Probabilistic setting for LTL

Possible probabilistic setting :

- $P_1, \ldots, P_M$ i.i.d from $\mathcal{P}$,
- $(X_{t,1}, Y_{t,1}), \ldots, (X_{t,n_M}, Y_{t,n_M})$ i.i.d from P_t,
- $R_t(f) = \mathbb{E}_{(X,Y)\sim P_t}[\ell(Y, f(X))]$,
- quantitative criterion to minimize w.r.t $\mathcal{I}$

$$\mathcal{R}_{\mathrm{LTL}}(\mathcal{I}) = \mathbb{E}_{P\sim\mathcal{P}}\left\{\min_{f\in\mathcal{C}} \mathbb{E}_{(X,Y)\sim P}\left[\ell(Y, f(\mathcal{I}, X))\right]\right\}.$$

Probabilistic setting for LTL

Possible probabilistic setting :

- $P_1, \dots, P_M$ i.i.d from $\mathcal{P}$,
- $(X_{t,1}, Y_{t,1}), \dots, (X_{t,n_M}, Y_{t,n_M})$ i.i.d from P_t,
- $R_t(f) = \mathbb{E}_{(X,Y)\sim P_t}[\ell(Y, f(X))]$,
- quantitative criterion to minimize w.r.t $\mathcal{I}$

$$\mathcal{R}_{\mathrm{LTL}}(\mathcal{I}) = \mathbb{E}_{P\sim\mathcal{P}} \left\{ \min_{f\in\mathcal{C}} \mathbb{E}_{(X,Y)\sim P} \left[\ell(Y, f(\mathcal{I}, X))\right] \right\}.$$

Note the strong Bayesian flavor...

Example of LTL : dictionary learning

The $X_{t,i} \in \mathbb{R}^K$, but all the relevant information is in $DX_{t,i} \in \mathbb{R}^k$, $k \ll K$. The matrix D is unknown.

Example of LTL : dictionary learning

The $X_{t,i} \in \mathbb{R}^K$, but all the relevant information is in $DX_{t,i} \in \mathbb{R}^k$, $k \ll K$. The matrix D is unknown.

- $\beta_1, \dots, \beta_M$ i.i.d from $\mathcal{P}$,

Example of LTL : dictionary learning

The $X_{t,i} \in \mathbb{R}^K$, but all the relevant information is in $DX_{t,i} \in \mathbb{R}^k$, $k \ll K$. The matrix D is unknown.

- $\beta_1, \ldots, \beta_M$ i.i.d from $\mathcal{P}$,
- $(X_{t,1}, Y_{t,1}), \ldots, (X_{t,n}, Y_{t,n})$ i.i.d from P_{β_t} :

$$Y = \beta_t^T DX + \varepsilon,$$

Example of LTL : dictionary learning

The $X_{t,i} \in \mathbb{R}^K$, but all the relevant information is in $DX_{t,i} \in \mathbb{R}^k$, $k \ll K$. The matrix D is unknown.

- $\beta_1, \dots, \beta_M$ i.i.d from $\mathcal{P}$,
- $(X_{t,1}, Y_{t,1}), \dots, (X_{t,n}, Y_{t,n})$ i.i.d from P_{β_t} :

$$Y = \beta_t^T DX + \varepsilon,$$

- $R_t(\beta, \Delta) = \mathbb{E}_{(X,Y)\sim P_{\beta_t}}[\ell(Y, \beta^T \Delta X)]$,

Example of LTL : dictionary learning

The $X_{t,i} \in \mathbb{R}^K$, but all the relevant information is in $DX_{t,i} \in \mathbb{R}^k$, $k \ll K$. The matrix D is unknown.

- $\beta_1, \dots, \beta_M$ i.i.d from $\mathcal{P}$,
- $(X_{t,1}, Y_{t,1}), \dots, (X_{t,n}, Y_{t,n})$ i.i.d from P_{β_t} :

$$Y = \beta_t^T DX + \varepsilon,$$

- $R_t(\beta, \Delta) = \mathbb{E}_{(X,Y)\sim P_{\beta_t}}[\ell(Y, \beta^T \Delta X)]$,
- quantitative criterion to minimize w.r.t M

$$\mathcal{R}_{\mathrm{LTL}}(\Delta) = \mathbb{E}_{\beta\sim\mathcal{P}} \left\{ \mathbb{E}_{(X,Y)\sim P_\beta} \left[\ell(Y, \beta^T \Delta X) \right] \right\}.$$

Example of LTL : dictionary learning

The $X_{t,i} \in \mathbb{R}^K$, but all the relevant information is in $DX_{t,i} \in \mathbb{R}^k$, $k \ll K$. The matrix D is unknown.

- $\beta_1, \dots, \beta_M$ i.i.d from $\mathcal{P}$,
- $(X_{t,1}, Y_{t,1}), \dots, (X_{t,n}, Y_{t,n})$ i.i.d from P_{β_t} :

$$Y = \beta_t^T DX + \varepsilon,$$

- $R_t(\beta, \Delta) = \mathbb{E}_{(X,Y)\sim P_{\beta_t}}[\ell(Y, \beta^T \Delta X)]$,
- quantitative criterion to minimize w.r.t M

$$\mathcal{R}_{\rm LTL}(\Delta) = \mathbb{E}_{\beta\sim\mathcal{P}} \left\{ \mathbb{E}_{(X,Y)\sim P_\beta} \left[\ell(Y, \beta^T \Delta X) \right] \right\}.$$

Maurer, Pontil and Romera-Paredes studied the estimator

$$\hat{D} = \arg\min_{\Delta} \sum_{t=1}^{M} \arg\min_{\|\beta_t\|_1 \leq \alpha} \sum_{i=1}^{n} \ell(Y_{t,i}, \beta_t^T \Delta X_{t,i})$$

Going online : lifelong learning

Lifelong learning (LL)

Online version of learning-to-learn ?

Recent work with The Tien Mai and Massimiliano Pontil.

Objectives :

Going online : lifelong learning

Lifelong learning (LL)

Online version of learning-to-learn ?

Recent work with The Tien Mai and Massimiliano Pontil.
Objectives :

- consider that tasks can be revealed sequentially. Use the tools of online learning theory : avoid probabilistic assumptions.

Going online : lifelong learning

Lifelong learning (LL)

Online version of learning-to-learn ?

Recent work with The Tien Mai and Massimiliano Pontil.
Objectives :

- consider that tasks can be revealed sequentially. Use the tools of online learning theory : avoid probabilistic assumptions.
- if possible, define a general strategy that does not depend on the learning algorithm used within each task.

1. Transfer learning, multitask learning, lifelong learning...
2. A strategy for lifelong learning, with regret analysis
3. Open questions

Massimiliano Pontil
(UCL, IIT)

The Tien Mai
(U. of Oslo)

PMLR Proceedings of Machine Learning Research

Volume 54 All Volumes JMLR MLOSS FAQ Submission Format RSS

Regret Bounds for Lifelong Learning

[edit]

Pierre Alquier, The Tien Mai, Massimiliano Pontil ; Proceedings of the 20th International Conference on Artificial Intelligence and Statistics, PMLR 54:261-269, 2017.

Abstract

We consider the problem of transfer learning in an online setting. Different tasks are presented sequentially and processed by a within-task algorithm. We propose a lifelong learning strategy which refines the underlying data representation used by the within-task algorithm, thereby transferring information from one task to the next. We show that when the within-task algorithm comes with some regret bound, our strategy inherits this good property. Our bounds are in expectation for a general loss function, and uniform for a convex loss. We discuss applications to dictionary learning and finite set of

Setting

- objects in $\mathcal{X}$, labels in $\mathcal{Y}$,

Setting

- objects in $\mathcal{X}$, labels in $\mathcal{Y}$,
- set of functions $\mathcal{G} : \mathcal{X} \to \mathcal{Z}$ and $\mathcal{H} : \mathcal{Z} \to \mathcal{Y}$,

Setting

- objects in $\mathcal{X}$, labels in $\mathcal{Y}$,
- set of functions $\mathcal{G} : \mathcal{X} \to \mathcal{Z}$ and $\mathcal{H} : \mathcal{Z} \to \mathcal{Y}$,
- loss function ℓ.

Lifelong-learning problem (LL)

Propose initial g.

Setting

- objects in $\mathcal{X}$, labels in $\mathcal{Y}$,
- set of functions $\mathcal{G} : \mathcal{X} \to \mathcal{Z}$ and $\mathcal{H} : \mathcal{Z} \to \mathcal{Y}$,
- loss function ℓ.

Lifelong-learning problem (LL)

Propose initial g.

For $t = 1, 2, \dots,$

Setting

- objects in $\mathcal{X}$, labels in $\mathcal{Y}$,
- set of functions $\mathcal{G} : \mathcal{X} \to \mathcal{Z}$ and $\mathcal{H} : \mathcal{Z} \to \mathcal{Y}$,
- loss function ℓ.

Lifelong-learning problem (LL)

Propose initial g.

For $t = 1, 2, \dots,$

1. propose initial h_t.
   For $i = 1, \dots, n_t$

Setting

- objects in $\mathcal{X}$, labels in $\mathcal{Y}$,
- set of functions $\mathcal{G} : \mathcal{X} \to \mathcal{Z}$ and $\mathcal{H} : \mathcal{Z} \to \mathcal{Y}$,
- loss function ℓ.

Lifelong-learning problem (LL)

Propose initial g.

For $t = 1, 2, \dots,$

1. propose initial h_t.
   For $i = 1, \dots, n_t$
   1. $x_{t,i}$ revealed,

Setting

- objects in $\mathcal{X}$, labels in $\mathcal{Y}$,
- set of functions $\mathcal{G} : \mathcal{X} \to \mathcal{Z}$ and $\mathcal{H} : \mathcal{Z} \to \mathcal{Y}$,
- loss function ℓ.

Lifelong-learning problem (LL)

Propose initial g.

For $t = 1, 2, \dots,$

1. propose initial h_t.
   For $i = 1, \dots, n_t$
   1. $x_{t,i}$ revealed,
   2. predict $\hat{y}_{t,i} = h_t \circ g(x_{t,i})$,

Setting

- objects in $\mathcal{X}$, labels in $\mathcal{Y}$,
- set of functions $\mathcal{G} : \mathcal{X} \to \mathcal{Z}$ and $\mathcal{H} : \mathcal{Z} \to \mathcal{Y}$,
- loss function ℓ.

Lifelong-learning problem (LL)

Propose initial g.

For $t = 1, 2, \dots,$

1. propose initial h_t.
   For $i = 1, \dots, n_t$
   1. $x_{t,i}$ revealed,
   2. predict $\hat{y}_{t,i} = h_t \circ g(x_{t,i})$,
   3. $y_{t,i}$ revealed, suffer loss $\hat{\ell}_{t,i} := \ell(y_{t,i}, \hat{y}_{t,i})$,

Setting

- objects in $\mathcal{X}$, labels in $\mathcal{Y}$,
- set of functions $\mathcal{G} : \mathcal{X} \to \mathcal{Z}$ and $\mathcal{H} : \mathcal{Z} \to \mathcal{Y}$,
- loss function ℓ.

Lifelong-learning problem (LL)

Propose initial g.

For $t = 1, 2, \dots,$

1. propose initial h_t.
   For $i = 1, \dots, n_t$
   1. $x_{t,i}$ revealed,
   2. predict $\hat{y}_{t,i} = h_t \circ g(x_{t,i})$,
   3. $y_{t,i}$ revealed, suffer loss $\hat{\ell}_{t,i} := \ell(y_{t,i}, \hat{y}_{t,i})$,
   4. update h_t.

Setting

- objects in $\mathcal{X}$, labels in $\mathcal{Y}$,
- set of functions $\mathcal{G} : \mathcal{X} \to \mathcal{Z}$ and $\mathcal{H} : \mathcal{Z} \to \mathcal{Y}$,
- loss function ℓ.

Lifelong-learning problem (LL)

Propose initial g.

For $t = 1, 2, \dots,$

1. propose initial h_t.
   For $i = 1, \dots, n_t$
   1. $x_{t,i}$ revealed,
   2. predict $\hat{y}_{t,i} = h_t \circ g(x_{t,i})$,
   3. $y_{t,i}$ revealed, suffer loss $\hat{\ell}_{t,i} := \ell(y_{t,i}, \hat{y}_{t,i})$,
   4. update h_t.
2. udpate g.

Within-task algorithm

For $t = 1, 2, \ldots,$

1. Solve a usual online task, input $z_{t,i} = g(x_{t,i})$, output $y_{t,i}$.
2. udpate g.

Within-task algorithm

For $t = 1, 2, \ldots,$

1. Solve a usual online task, input $z_{t,i} = g(x_{t,i})$, output $y_{t,i}$.
2. udpate g.

We can do it using any online algorithm. Will be refered to as "within-task algorithm".

For many algorithms, bounds are known on the (normalized)-regret :

$$\mathcal{R}_t(g) = \underbrace{\frac{1}{n_t} \sum_{i=1}^{n_t} \ell(y_{t,i}, \hat{y}_{t,i})}_{= \frac{1}{n_t} \sum_{i=1}^{n_t} \hat{\ell}_{t,i} = \hat{L}_t(g)} - \frac{1}{n_t} \inf_{h \in \mathcal{H}} \sum_{i=1}^{n_t} \ell(y_{t,i}, h(z_{t,i})).$$

Examples of within-task algorithms

Online gradient for convex ℓ

Initialize $h = 0$.
Update $h \leftarrow h - \eta \nabla_{f=h} \ell(y_{t,i}, f(z_{t,i}))$.

Examples of within-task algorithms

Online gradient for convex ℓ

Initialize $h = 0$.
Update $h \leftarrow h - \eta \nabla_{f=h} \ell(y_{t,i}, f(z_{t,i}))$.

Many variants and improvements (projected gradient, online Newton-step, ...).
$\mathcal{R}_t(g)$ in $1/\sqrt{n_t}$ or $1/n_t$ depending on assumptions on ℓ.

Examples of within-task algorithms

Online gradient for convex ℓ

Initialize $h = 0$.
Update $h \leftarrow h - \eta \nabla_{f=h} \ell(y_{t,i}, f(z_{t,i}))$.

Many variants and improvements (projected gradient, online Newton-step, ...).
$\mathcal{R}_t(g)$ in $1/\sqrt{n_t}$ or $1/n_t$ depending on assumptions on ℓ.

EWA (Exponentially Weighted Aggregation)

Prior $\rho_1 = \pi$, initialize $h \sim \rho_1$.
Update $\rho_{i+1}(\mathrm{d}f) \propto \exp[-\eta \ell(y_{t,i}, f(z_{t,i}))] \rho_i(\mathrm{d}f)$, $h \sim \rho_{i+1}$.

Examples of within-task algorithms

Online gradient for convex ℓ

Initialize $h = 0$.
Update $h \leftarrow h - \eta \nabla_{f=h} \ell(y_{t,i}, f(z_{t,i}))$.

Many variants and improvements (projected gradient, online Newton-step, ...).
$\mathcal{R}_t(g)$ in $1/\sqrt{n_t}$ or $1/n_t$ depending on assumptions on ℓ.

EWA (Exponentially Weighted Aggregation)

Prior $\rho_1 = \pi$, initialize $h \sim \rho_1$.
Update $\rho_{i+1}(\mathrm{d}f) \propto \exp[-\eta \ell(y_{t,i}, f(z_{t,i}))] \rho_i(\mathrm{d}f)$, $h \sim \rho_{i+1}$.

$\mathbb{E}[\mathcal{R}_t(g)]$ in $1/\sqrt{n_t}$ under boundedness assumption.
Integrated variant : $\mathcal{R}_t(g)$ in $1/n_t$ if ℓ is exp-concave.

EWA for lifelong learning

EWA-LL

Prior $\pi = \rho_1$ on $\mathcal{G}$. Draw $g \sim \pi$.

For $t = 1, 2, \ldots$

1. run the within-task algorithm on task t. Suffer $\hat{L}_t(g)$.
2. update $\rho_{t+1}(\mathrm{d}f) \propto \exp[-\eta \hat{L}_t(f)]\rho_t(\mathrm{d}f)$.
3. draw $g \sim \rho_{t+1}$.

EWA for lifelong learning

EWA-LL

Prior $\pi = \rho_1$ on $\mathcal{G}$. Draw $g \sim \pi$.
For $t = 1, 2, \ldots$

1. run the within-task algorithm on task t. Suffer $\hat{L}_t(g)$.
2. update $\rho_{t+1}(\mathrm{d}f) \propto \exp[-\eta \hat{L}_t(f)] \rho_t(\mathrm{d}f)$.
3. draw $g \sim \rho_{t+1}$.

Next : we provide two examples that are corollaries of a general result (stated later).

Example 1 : dictionary learning

$$\begin{array}{ccccc} \mathcal{X} = \mathbb{R}^K & \rightarrow & \mathcal{Z} = \mathbb{R}^k & \rightarrow & \mathcal{Y} = \mathbb{R} \\ x & \mapsto & Dx & \mapsto & \langle h, Dx \rangle = h^T Dx. \end{array}$$

Example 1 : dictionary learning

$$\begin{array}{ccccc} \mathcal{X} = \mathbb{R}^K & \to & \mathcal{Z} = \mathbb{R}^k & \to & \mathcal{Y} = \mathbb{R} \\ x & \mapsto & Dx & \mapsto & \langle h, Dx \rangle = h^T Dx. \end{array}$$

- within-task algorithm : online gradient descent on h.

Example 1 : dictionary learning

$$\begin{array}{ccccc} \mathcal{X} = \mathbb{R}^K & \rightarrow & \mathcal{Z} = \mathbb{R}^k & \rightarrow & \mathcal{Y} = \mathbb{R} \\ x & \mapsto & Dx & \mapsto & \langle h, Dx \rangle = h^T Dx. \end{array}$$

- within-task algorithm : online gradient descent on h.
- EWA-LL, prior : columns of D i.i.d uniform on unit sphere.

Example 1 : dictionary learning

$$\begin{array}{ccccc} \mathcal{X} = \mathbb{R}^K & \to & \mathcal{Z} = \mathbb{R}^k & \to & \mathcal{Y} = \mathbb{R} \\ x & \mapsto & Dx & \mapsto & \langle h, Dx \rangle = h^T Dx. \end{array}$$

- within-task algorithm : online gradient descent on h.
- EWA-LL, prior : columns of D i.i.d uniform on unit sphere.

Theorem (Corollary 4.4) - ℓ is bounded by B & L-Lipschitz

$$\mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T}\frac{1}{n_t}\sum_{i=1}^{n_t}\hat{\ell}_{t,i}\right] \leq \inf_{D}\frac{1}{T}\sum_{t=1}^{T}\inf_{\|h_t\|\leq C}\frac{1}{n_t}\sum_{i=1}^{n_t}\ell(y_{t,i}, h_t^T D x_{t,i}) + \frac{C}{4}\sqrt{\frac{Kk}{T}}(\log(T)+7) + \frac{BL}{\sqrt{T}} + \frac{1}{T}\sum_{t=1}^{T}\frac{BL\sqrt{2k}}{\sqrt{n_t}}.$$

Example 1 : dictionary learning

$$\begin{array}{ccccc} \mathcal{X} = \mathbb{R}^K & \to & \mathcal{Z} = \mathbb{R}^k & \to & \mathcal{Y} = \mathbb{R} \\ x & \mapsto & Dx & \mapsto & \langle h, Dx \rangle = h^T Dx. \end{array}$$

- within-task algorithm : online gradient descent on h.
- EWA-LL, prior : columns of D i.i.d uniform on unit sphere.

Theorem (Corollary 4.4) - ℓ is bounded by B & L-Lipschitz

$$\mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T}\frac{1}{n_t}\sum_{i=1}^{n_t}\hat{\ell}_{t,i}\right] \leq \inf_{D}\frac{1}{T}\sum_{t=1}^{T}\inf_{\|h_t\|\leq C}\frac{1}{n_t}\sum_{i=1}^{n_t}\ell(y_{t,i}, h_t^T D x_{t,i}) + \frac{C}{4}\sqrt{\frac{Kk}{T}}(\log(T)+7) + \frac{BL}{\sqrt{T}} + \frac{BL\sqrt{2k}}{\sqrt{\bar{n}}}.$$

Example 1 (dictionary learning) : simulations

- simulations $\mathcal{X} = \mathbb{R}^5 \to \mathcal{Z} = \mathbb{R}^2 \to \mathcal{Y} = \mathbb{R}$ with ℓ the quadratic loss, $T = 150$, each $n_t = 100$.

Example 1 (dictionary learning) : simulations

- simulations $\mathcal{X} = \mathbb{R}^5 \to \mathcal{Z} = \mathbb{R}^2 \to \mathcal{Y} = \mathbb{R}$ with ℓ the quadratic loss, $T = 150$, each $n_t = 100$.
- implementation of EWA-LL, at each step, D is updated using N iterations of Metropolis-Hastings.

Example 1 (dictionary learning) : simulations

- simulations $\mathcal{X} = \mathbb{R}^5 \to \mathcal{Z} = \mathbb{R}^2 \to \mathcal{Y} = \mathbb{R}$ with ℓ the quadratic loss, $T = 150$, each $n_t = 100$.
- implementation of EWA-LL, at each step, D is updated using N iterations of Metropolis-Hastings.

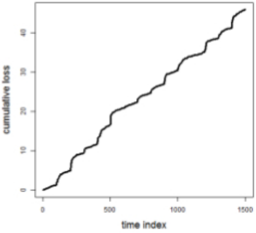


Figure 1: The cumulative loss of the oracle for the first 15 tasks.

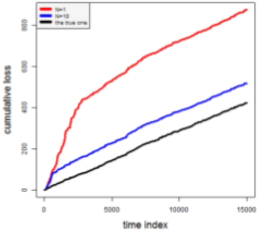


Figure 2: Cumulative loss of EWA-LL ($N = 1$ in red and $N = 10$ in blue) and cumulative loss of the oracle.

Example 2 : finite set of predictors

$$x \overset{g\in\mathcal{G}}{\mapsto} g(x) \overset{h\in\mathcal{H}}{\mapsto} h(g(x)).$$

$$\mathrm{card}(\mathcal{G}) = G < +\infty, \ \mathrm{card}(\mathcal{H}) = H < +\infty$$

Example 2 : finite set of predictors

$$x \overset{g\in\mathcal{G}}{\mapsto} g(x) \overset{h\in\mathcal{H}}{\mapsto} h(g(x)).$$

$$\mathrm{card}(\mathcal{G}) = G < +\infty, \ \mathrm{card}(\mathcal{H}) = H < +\infty$$

- within-task algorithm : EWA, uniform prior.

Example 2 : finite set of predictors

$$x \overset{g\in\mathcal{G}}{\mapsto} g(x) \overset{h\in\mathcal{H}}{\mapsto} h(g(x)).$$

$$\mathrm{card}(\mathcal{G}) = G < +\infty, \ \mathrm{card}(\mathcal{H}) = H < +\infty$$

- within-task algorithm : EWA, uniform prior.
- EWA-LL, uniform prior.

Example 2 : finite set of predictors

$$x \overset{g\in\mathcal{G}}{\longmapsto} g(x) \overset{h\in\mathcal{H}}{\longmapsto} h(g(x)).$$
$$\mathrm{card}(\mathcal{G}) = G < +\infty,\ \mathrm{card}(\mathcal{H}) = H < +\infty$$

- within-task algorithm : EWA, uniform prior.
- EWA-LL, uniform prior.

Theorem (Corollary 4.2) - ℓ bounded by C & α-exp-concave

$$\mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T}\frac{1}{m}\sum_{i=1}^{m}\hat{\ell}_{t,i}\right] \leq \inf_{g\in\mathcal{G}}\frac{1}{T}\sum_{t=1}^{T}\inf_{h_t\in\mathcal{H}}\frac{1}{m}\sum_{i=1}^{m}\ell(y_{t,i}, h_t\circ g(x_{t,i})) + C\sqrt{\frac{\log G}{2T}} + \frac{\alpha \log H}{\bar{n}}.$$

Example 2 : improvement on existing results

The "online-to-batch" trick allows to deduce from our online method a statistical estimator with a controled LTL risk in

$$\mathcal{O}\left(\sqrt{\frac{\log G}{T}} + \frac{\log H}{n}\right).$$

Example 2 : improvement on existing results

The "online-to-batch" trick allows to deduce from our online method a statistical estimator with a controled LTL risk in

$$\mathcal{O}\left(\sqrt{\frac{\log G}{T}} + \frac{\log H}{n}\right).$$

In this case, a previous bound by Pentina and Lampert was in

$$\mathcal{O}\left(\sqrt{\frac{\log G}{T}} + \sqrt{\frac{\log H}{n}}\right).$$

A PAC-Bayesian Bound for Lifelong Learning

Anastasia Pentina APENTINA@IST.AC.AT
IST Austria (Institute of Science and Technology Austria), 3400 Am Campus 1, Klosterneuburg, Austria
Christoph H. Lampert CHL@IST.AC.AT
IST Austria (Institute of Science and Technology Austria), 3400 Am Campus 1, Klosterneuburg, Austria

General regret bound

Theorem (Theorem 3.1) - ℓ bounded by C

If for any $g \in \mathcal{G}$, the within-task algorithm has a regret bound $\mathcal{R}_t(g) \leq \beta(g, n_t)$, then

$$\mathbb{E}\left[\frac{1}{T}\sum_{t=1}^{T}\frac{1}{n_t}\sum_{i=1}^{n_t}\hat{\ell}_{t,i}\right] \leq \inf_{\rho}\left\{\int\left[\frac{1}{T}\sum_{t=1}^{T}\inf_{h_t\in\mathcal{H}}\frac{1}{n_t}\sum_{i=1}^{n_t}\ell(y_{t,i}, h_t\circ g(x_{t,i})) + \frac{1}{T}\sum_{t=1}^{T}\beta(g, n_t)\right]\rho(\mathrm{d}g) + \frac{\eta C^2}{8} + \frac{\mathcal{K}(\rho,\pi)}{\eta T}\right\}.$$

Efficient algorithms ?

Our online analysis allows to avoid explicit probabilistic assumptions on the data, and allows a free choice of the within-task algorithm.

Efficient algorithms ?

Our online analysis allows to avoid explicit probabilistic assumptions on the data, and allows a free choice of the within-task algorithm.

However, EWA-LL is not "truly online" as its computation requires to store all the data seen so far.

Efficient Lifelong Learning Algorithm : ELLA

ELLA: An Efficient Lifelong Learning Algorithm

Paul Ruvolo PRUVOLO@CS.BRYNMAWR.EDU
Eric Eaton EEATON@CS.BRYNMAWR.EDU
Bryn Mawr College, Computer Science Department, 101 North Merion Avenue, Bryn Mawr, PA 19010 USA

Abstract

The problem of learning multiple consecutive tasks, known as *lifelong learning*, is of great importance to the creation of intelligent, general-purpose, and flexible machines. In this paper, we develop a method for online multi-task 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 tasks. We show that ELLA has strong connections to both online dictionary learning for sparse coding and state-of-the-art batch multi-task learning methods, and provide robust theoretical performance guarantees. We show empirically that ELLA yields nearly identical performance to batch multi-task learning while learning tasks sequentially in three orders of magnitude (over 1,000x) less time.

1. Introduction

Versatile learning systems must be capable of efficiently and continually acquiring knowledge over a series of prediction tasks. In such a lifelong learning setting, the agent receives tasks sequentially. At any time, the agent may be asked to solve a problem from any previous task, and so must maximize its performance across all learned tasks at each step. When the solutions to these tasks are related through some underlying structure, the agent may share knowledge between tasks to improve learning performance, as explored in both transfer and multi-task learning.

Despite this commonality, current algorithms for transfer and multi-task learning are insufficient for lifelong learning. Transfer learning focuses on efficiently modeling a new target task by leveraging solutions to previously learned source tasks, without considering potential improvements to the source task models. In contrast, multi-task learning (MTL) focuses on maximizing performance across *all* tasks through shared knowledge, at potentially high computational cost. Lifelong learning includes elements of both paradigms, focusing on efficiently learning each consecutive task by building upon previous knowledge while optimizing performance across all tasks. In particular, lifelong learning incorporates the notion of *reverse transfer*, in which learning subsequent tasks can improve the performance of previously learned task models. Lifelong learning could also be considered as online MTL.

In this paper, we develop an Efficient Lifelong Learning Algorithm (ELLA) that incorporates aspects of both transfer and multi-task learning. ELLA learns and maintains a library of latent model components as a shared basis for all task models, supporting soft task grouping and overlap (Kumar & Daumé III, 2012). As each new task arrives, ELLA transfers knowledge through the shared basis to learn the new model, and refines the basis with knowledge from the new task. By refining the basis over time, newly acquired knowledge is integrated into existing basis vectors, thereby improving previously learned task models. This process is computationally efficient, and we provide robust theoretical guarantees on ELLA's performance and convergence. We evaluate ELLA on three challenging multi-task data sets: land mine detection, facial expression recognition, and student exam score prediction. Our results show that ELLA achieves nearly identical performance to batch MTL with three orders of magnitude (over 1,000x) speedup in learning time. We also compare ELLA to a current method for online MTL (Saha et al., 2011), and find that ELLA has both lower computational cost and higher performance.

2. Related Work

Early work on lifelong learning focused on sharing distance metrics using task clustering (Thrun & O'Sullivan, 1996), and transferring invariances in neu-

Proceedings of the 30th International Conference on Machine Learning, Atlanta, Georgia, USA, 2013. JMLR: W&CP volume 28. Copyright 2013 by the author(s).

- dictionary learning,

Efficient Lifelong Learning Algorithm : ELLA

ELLA: An Efficient Lifelong Learning Algorithm

Paul Ruvolo PRUVOLO@CS.BRYNMAWR.EDU
Eric Eaton EEATON@CS.BRYNMAWR.EDU
Bryn Mawr College, Computer Science Department, 101 North Merion Avenue, Bryn Mawr, PA 19010 USA

Abstract

The problem of learning multiple consecutive tasks, known as *lifelong learning*, is of great importance to the creation of intelligent, general-purpose, and flexible machines. In this paper, we develop a method for online multi-task 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 tasks. We show that ELLA has strong connections to both online dictionary learning for sparse coding and state-of-the-art batch multi-task learning methods, and provide robust theoretical performance guarantees. We show empirically that ELLA yields nearly identical performance to batch multi-task learning while learning tasks sequentially in three orders of magnitude (over 1,000x) less time.

1. Introduction

Versatile learning systems must be capable of efficiently and continually acquiring knowledge over a series of prediction tasks. In such a lifelong learning setting, the agent receives tasks sequentially. At any time, the agent may be asked to solve a problem from any previous task, and so must maximize its performance across all learned tasks at each step. When the solutions to these tasks are related through some underlying structure, the agent may share knowledge between tasks to improve learning performance, as explored in both transfer and multi-task learning.

Despite this commonality, current algorithms for transfer and multi-task learning are insufficient for lifelong learning. Transfer learning focuses on efficiently modeling a new target task by leveraging solutions to previously learned source tasks, without considering potential improvements to the source task models. In contrast, multi-task learning (MTL) focuses on maximizing performance across *all* tasks through shared knowledge, at potentially high computational cost. Lifelong learning includes elements of both paradigms, focusing on efficiently learning each consecutive task by building upon previous knowledge while optimizing performance across all tasks. In particular, lifelong learning incorporates the notion of *reverse transfer*, in which learning subsequent tasks can improve the performance of previously learned task models. Lifelong learning could also be considered as online MTL.

In this paper, we develop an Efficient Lifelong Learning Algorithm (ELLA) that incorporates aspects of both transfer and multi-task learning. ELLA learns and maintains a library of latent model components as a shared basis for all task models, supporting soft task grouping and overlap (Kumar & Daumé III, 2012). As each new task arrives, ELLA transfers knowledge through the shared basis to learn the new model, and refines the basis with knowledge from the new task. By refining the basis over time, newly acquired knowledge is integrated into existing basis vectors, thereby improving previously learned task models. This process is computationally efficient, and we provide robust theoretical guarantees on ELLA's performance and convergence. We evaluate ELLA on three challenging multi-task data sets: land mine detection, facial expression recognition, and student exam score prediction. Our results show that ELLA achieves nearly identical performance to batch MTL with three orders of magnitude (over 1,000x) speedup in learning time. We also compare ELLA to a current method for online MTL (Saha et al., 2011), and find that ELLA has both lower computational cost and higher performance.

2. Related Work

Early work on lifelong learning focused on sharing distance metrics using task clustering (Thrun & O'Sullivan, 1996), and transferring invariances in neu-

Proceedings of the 30th International Conference on Machine Learning, Atlanta, Georgia, USA, 2013. JMLR: W&CP volume 28. Copyright 2013 by the author(s).

- dictionary learning,
- fast update of D and β at each step, truly online : no need to store the data,

Efficient Lifelong Learning Algorithm : ELLA

ELLA: An Efficient Lifelong Learning Algorithm

Paul Ruvolo PRUVOLO@CS.BRYNMAWR.EDU
Eric Eaton EEATON@CS.BRYNMAWR.EDU
Bryn Mawr College, Computer Science Department, 101 North Merion Avenue, Bryn Mawr, PA 19010 USA

Abstract

The problem of learning multiple consecutive tasks, known as *lifelong learning*, is of great importance to the creation of intelligent, general-purpose, and flexible machines. In this paper, we develop a method for online multi-task 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 tasks. We show that ELLA has strong connections to both online dictionary learning for sparse coding and state-of-the-art batch multi-task learning methods, and provide robust theoretical performance guarantees. We show empirically that ELLA yields nearly identical performance to batch multi-task learning while learning tasks sequentially in three orders of magnitude (over 1,000x) less time.

1. Introduction

Versatile learning systems must be capable of efficiently and continually acquiring knowledge over a series of prediction tasks. In such a lifelong learning setting, the agent receives tasks sequentially. At any time, the agent may be asked to solve a problem from any previous task, and so must maximize its performance across all learned tasks at each step. When the solutions to these tasks are related through some underlying structure, the agent may share knowledge between tasks to improve learning performance, as explored in both transfer and multi-task learning.

Despite this commonality, current algorithms for transfer and multi-task learning are insufficient for lifelong learning. Transfer learning focuses on efficiently modeling a new target task by leveraging solutions to previously learned source tasks, without considering potential improvements to the source task models. In contrast, multi-task learning (MTL) focuses on maximizing performance across *all* tasks through shared knowledge, at potentially high computational cost. Lifelong learning includes elements of both paradigms, focusing on efficiently learning each consecutive task by building upon previous knowledge while optimizing performance across all tasks. In particular, lifelong learning incorporates the notion of *reverse transfer*, in which learning subsequent tasks can improve the performance of previously learned task models. Lifelong learning could also be considered as online MTL.

In this paper, we develop an Efficient Lifelong Learning Algorithm (ELLA) that incorporates aspects of both transfer and multi-task learning. ELLA learns and maintains a library of latent model components as a shared basis for all task models, supporting soft task grouping and overlap (Kumar & Daumé III, 2012). As each new task arrives, ELLA transfers knowledge through the shared basis to learn the new model, and refines the basis with knowledge from the new task. By refining the basis over time, newly acquired knowledge is integrated into existing basis vectors, thereby improving previously learned task models. This process is computationally efficient, and we provide robust theoretical guarantees on ELLA's performance and convergence. We evaluate ELLA on three challenging multi-task data sets: land mine detection, facial expression recognition, and student exam score prediction. Our results show that ELLA achieves nearly identical performance to batch MTL with three orders of magnitude (over 1,000x) speedup in learning time. We also compare ELLA to a current method for online MTL (Saha et al., 2011), and find that ELLA has both lower computational cost and higher performance.

2. Related Work

Early work on lifelong learning focused on sharing distance metrics using task clustering (Thrun & O'Sullivan, 1996), and transferring invariances in neu-

Proceedings of the 30th International Conference on Machine Learning, Atlanta, Georgia, USA, 2013. JMLR: W&CP volume 28. Copyright 2013 by the author(s).

- dictionary learning,
- fast update of D and β at each step, truly online : no need to store the data,
- very good empirical performances,

Efficient Lifelong Learning Algorithm : ELLA

ELLA: An Efficient Lifelong Learning Algorithm

Paul Ruvolo PRUVOLO@CS.BRYNMAWR.EDU
Eric Eaton EEATON@CS.BRYNMAWR.EDU
Bryn Mawr College, Computer Science Department, 101 North Merion Avenue, Bryn Mawr, PA 19010 USA

Abstract

The problem of learning multiple consecutive tasks, known as *lifelong learning*, is of great importance to the creation of intelligent, general-purpose, and flexible machines. In this paper, we develop a method for online multi-task 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 tasks. We show that ELLA has strong connections to both online dictionary learning for sparse coding and state-of-the-art batch multi-task learning methods, and provide robust theoretical performance guarantees. We show empirically that ELLA yields nearly identical performance to batch multi-task learning while learning tasks sequentially in three orders of magnitude (over 1,000x) less time.

1. Introduction

Versatile learning systems must be capable of efficiently and continually acquiring knowledge over a series of prediction tasks. In such a lifelong learning setting, the agent receives tasks sequentially. At any time, the agent may be asked to solve a problem from any previous task, and so must maximize its performance across all learned tasks at each step. When the solutions to these tasks are related through some underlying structure, the agent may share knowledge between tasks to improve learning performance, as explored in both transfer and multi-task learning.

Despite this commonality, current algorithms for transfer and multi-task learning are insufficient for lifelong learning. Transfer learning focuses on efficiently modeling a new target task by leveraging solutions to previously learned source tasks, without considering potential improvements to the source task models. In contrast, multi-task learning (MTL) focuses on maximizing performance across *all* tasks through shared knowledge, at potentially high computational cost. Lifelong learning includes elements of both paradigms, focusing on efficiently learning each consecutive task by building upon previous knowledge while optimizing performance across all tasks. In particular, lifelong learning incorporates the notion of *reverse transfer*, in which learning subsequent tasks can improve the performance of previously learned task models. Lifelong learning could also be considered as online MTL.

In this paper, we develop an Efficient Lifelong Learning Algorithm (ELLA) that incorporates aspects of both transfer and multi-task learning. ELLA learns and maintains a library of latent model components as a shared basis for all task models, supporting soft task grouping and overlap (Kumar & Daumé III, 2012). As each new task arrives, ELLA transfers knowledge through the shared basis to learn the new model, and refines the basis with knowledge from the new task. By refining the basis over time, newly acquired knowledge is integrated into existing basis vectors, thereby improving previously learned task models. This process is computationally efficient, and we provide robust theoretical guarantees on ELLA's performance and convergence. We evaluate ELLA on three challenging multi-task data sets: land mine detection, facial expression recognition, and student exam score prediction. Our results show that ELLA achieves nearly identical performance to batch MTL with three orders of magnitude (over 1,000x) speedup in learning time. We also compare ELLA to a current method for online MTL (Saha et al., 2011), and find that ELLA has both lower computational cost and higher performance.

2. Related Work

Early work on lifelong learning focused on sharing distance metrics using task clustering (Thrun & O'Sullivan, 1996), and transferring invariances in neu-

Proceedings of the 30th International Conference on Machine Learning, Atlanta, Georgia, USA, 2013. JMLR: W&CP volume 28. Copyright 2013 by the author(s).

- dictionary learning,
- fast update of D and β at each step, truly online : no need to store the data,
- very good empirical performances,
- no regret bound.

More progress on dictionary learning

- dictionary learning,

Incremental Learning-to-Learn with Statistical Guarantees

Giulia Denevi[1,2] giulia.denevi@iit.it
Carlo Ciliberto[3] c.ciliberto@ucl.ac.uk
Dimitris Stamos[3] d.stamos.12@ucl.ac.uk
Massimiliano Pontil [1,3] massimiliano.pontil@iit.it

March 23, 2018

Abstract

In learning-to-learn the goal is to infer a learning algorithm that works well on a class of tasks sampled from an unknown meta distribution. In contrast to previous work on batch learning-to-learn, we consider a scenario where tasks are presented sequentially and the algorithm needs to adapt incrementally to improve its performance on future tasks. Key to this setting is for the algorithm to rapidly incorporate new observations into the model as they arrive, without keeping them in memory. We focus on the case where the underlying algorithm is Ridge Regression parameterized by a positive semidefinite matrix. We propose to learn this matrix by applying a stochastic strategy to minimize the empirical error incurred by Ridge Regression on future tasks sampled from the meta distribution. We study the statistical properties of the proposed algorithm and prove non-asymptotic bounds on its excess transfer risk, that is, the generalization performance on new tasks from the same meta distribution. We compare our online learning-to-learn approach with a state of the art batch method, both theoretically and empirically.

1 INTRODUCTION

Learning-to-learn (LTL) or meta learning aims at finding an algorithm that is best suited to address a class of learning problems (tasks). These tasks are sampled from an unknown meta distribution and are only partially observed via a finite collection of training examples, see [6, 19, 36] and references therein. This problem plays a large role in artificial intelligence in that it can improve the efficiency of learning from human supervision. In particular, substantial improvement over "learning in isolation" (also known as independent task learning) is to be expected when the sample size per task is small, a setting which naturally arises in many applications [10, 30, 32, 37].

LTL is particularly appealing when considered from an online or incremental perspective. In this setting, which is sometimes referred to as lifelong learning (see, e.g. [33]), the tasks are observed sequentially – via corresponding sets of training examples – from a common environment and we aim to improve the learning ability of the underlying algorithm on future yet-to-be-seen tasks from the same environment. Practical scenarios of lifelong learning are wide ranging, including computer vision [30], robotics [10], user modelling and many more.

[1] Computational Statistics and Machine Learning, Istituto Italiano di Tecnologia, 16163 Genova, Italy
[2] Department of Mathematics, University of Genova, 16146 Genova, Italy
[3] Department of Computer Science, University College London, WC1E 6BT, London, UK

1

More progress on dictionary learning

Incremental Learning-to-Learn with Statistical Guarantees

Giulia Denevi[1,2] giulia.denevi@iit.it
Carlo Ciliberto[3] c.ciliberto@ucl.ac.uk
Dimitris Stamos[3] d.stamos.12@ucl.ac.uk
Massimiliano Pontil [1,3] massimiliano.pontil@iit.it

March 23, 2018

arXiv:1803.08089v1 [stat.ML] 21 Mar 2018

Abstract

In learning-to-learn the goal is to infer a learning algorithm that works well on a class of tasks sampled from an unknown meta distribution. In contrast to previous work on batch learning-to-learn, we consider a scenario where tasks are presented sequentially and the algorithm needs to adapt incrementally to improve its performance on future tasks. Key to this setting is for the algorithm to rapidly incorporate new observations into the model as they arrive, without keeping them in memory. We focus on the case where the underlying algorithm is Ridge Regression parameterized by a positive semidefinite matrix. We propose to learn this matrix by applying a stochastic strategy to minimize the empirical error incurred by Ridge Regression on future tasks sampled from the meta distribution. We study the statistical properties of the proposed algorithm and prove non-asymptotic bounds on its excess transfer risk, that is, the generalization performance on new tasks from the same meta distribution. We compare our online learning-to-learn approach with a state of the art batch method, both theoretically and empirically.

1 INTRODUCTION

Learning-to-learn (LTL) or meta learning aims at finding an algorithm that is best suited to address a class of learning problems (tasks). These tasks are sampled from an unknown meta distribution and are only partially observed via a finite collection of training examples, see [6, 19, 36] and references therein. This problem plays a large role in artificial intelligence in that it can improve the efficiency of learning from human supervision. In particular, substantial improvement over "learning in isolation" (also known as independent task learning) is to be expected when the sample size per task is small, a setting which naturally arises in many applications [10, 30, 32, 37].
LTL is particularly appealing when considered from an online or incremental perspective. In this setting, which is sometimes referred to as lifelong learning (see, e.g. [33]), the tasks are observed sequentially – via corresponding sets of training examples – from a common environment and we aim to improve the learning ability of the underlying algorithm on future yet-to-be-seen tasks from the same environment. Practical scenarios of lifelong learning are wide ranging, including computer vision [30], robotics [10], user modelling and many more.

[1]Computational Statistics and Machine Learning, Istituto Italiano di Tecnologia, 16163 Genova, Italy
[2]Department of Mathematics, University of Genova, 16146 Genova, Italy
[3]Department of Computer Science, University College London, WC1E 6BT, London, UK

1

- dictionary learning,
- fast update of β at each step, fast update of D at the end of each task, truly online,

More progress on dictionary learning

Incremental Learning-to-Learn with Statistical Guarantees

Giulia Denevi[1,2] giulia.denevi@iit.it
Carlo Ciliberto[3] c.ciliberto@ucl.ac.uk
Dimitris Stamos[3] d.stamos.12@ucl.ac.uk
Massimiliano Pontil [1,3] massimiliano.pontil@iit.it

March 23, 2018

Abstract

In learning-to-learn the goal is to infer a learning algorithm that works well on a class of tasks sampled from an unknown meta distribution. In contrast to previous work on batch learning-to-learn, we consider a scenario where tasks are presented sequentially and the algorithm needs to adapt incrementally to improve its performance on future tasks. Key to this setting is for the algorithm to rapidly incorporate new observations into the model as they arrive, without keeping them in memory. We focus on the case where the underlying algorithm is Ridge Regression parameterized by a positive semidefinite matrix. We propose to learn this matrix by applying a stochastic strategy to minimize the empirical error incurred by Ridge Regression on future tasks sampled from the meta distribution. We study the statistical properties of the proposed algorithm and prove non-asymptotic bounds on its excess transfer risk, that is, the generalization performance on new tasks from the same meta distribution. We compare our online learning-to-learn approach with a state of the art batch method, both theoretically and empirically.

1 INTRODUCTION

Learning-to-learn (LTL) or meta learning aims at finding an algorithm that is best suited to address a class of learning problems (tasks). These tasks are sampled from an unknown meta distribution and are only partially observed via a finite collection of training examples, see [6, 19, 36] and references therein. This problem plays a large role in artificial intelligence in that it can improve the efficiency of learning from human supervision. In particular, substantial improvement over “learning in isolation” (also known as independent task learning) is to be expected when the sample size per task is small, a setting which naturally arises in many applications [10, 30, 32, 37].
LTL is particularly appealing when considered from an online or incremental perspective. In this setting, which is sometimes referred to as lifelong learning (see, e.g. [33]), the tasks are observed sequentially – via corresponding sets of training examples – from a common environment and we aim to improve the learning ability of the underlying algorithm on future yet-to-be-seen tasks from the same environment. Practical scenarios of lifelong learning are wide ranging, including computer vision [30], robotics [10], user modelling and many more.

[1]Computational Statistics and Machine Learning, Istituto Italiano di Tecnologia, 16163 Genova, Italy
[2]Department of Mathematics, University of Genova, 16146 Genova, Italy
[3]Department of Computer Science, University College London, WC1E 6BT, London, UK

1

- dictionary learning,
- fast update of β at each step, fast update of D at the end of each task, truly online,
- very good empirical performances,

More progress on dictionary learning

Incremental Learning-to-Learn with Statistical Guarantees

Giulia Denevi[1,2] giulia.denevi@iit.it
Carlo Ciliberto[3] c.ciliberto@ucl.ac.uk
Dimitris Stamos[3] d.stamos.12@ucl.ac.uk
Massimiliano Pontil[1,3] massimiliano.pontil@iit.it

March 23, 2018

arXiv:1803.08089v1 [stat.ML] 21 Mar 2018

Abstract

In learning-to-learn the goal is to infer a learning algorithm that works well on a class of tasks sampled from an unknown meta distribution. In contrast to previous work on batch learning-to-learn, we consider a scenario where tasks are presented sequentially and the algorithm needs to adapt incrementally to improve its performance on future tasks. Key to this setting is for the algorithm to rapidly incorporate new observations into the model as they arrive, without keeping them in memory. We focus on the case where the underlying algorithm is Ridge Regression parameterized by a positive semidefinite matrix. We propose to learn this matrix by applying a stochastic strategy to minimize the empirical error incurred by Ridge Regression on future tasks sampled from the meta distribution. We study the statistical properties of the proposed algorithm and prove non-asymptotic bounds on its excess transfer risk, that is, the generalization performance on new tasks from the same meta distribution. We compare our online learning-to-learn approach with a state of the art batch method, both theoretically and empirically.

1 INTRODUCTION

Learning-to-learn (LTL) or meta learning aims at finding an algorithm that is best suited to address a class of learning problems (tasks). These tasks are sampled from an unknown meta distribution and are only partially observed via a finite collection of training examples, see [6, 19, 36] and references therein. This problem plays a large role in artificial intelligence in that it can improve the efficiency of learning from human supervision. In particular, substantial improvement over "learning in isolation" (also known as independent task learning) is to be expected when the sample size per task is small, a setting which naturally arises in many applications [10, 30, 32, 37].

LTL is particularly appealing when considered from an online or incremental perspective. In this setting, which is sometimes referred to as lifelong learning (see, e.g. [33]), the tasks are observed sequentially – via corresponding sets of training examples – from a common environment and we aim to improve the learning ability of the underlying algorithm on future yet-to-be-seen tasks from the same environment. Practical scenarios of lifelong learning are wide ranging, including computer vision [30], robotics [10], user modelling and many more.

[1] Computational Statistics and Machine Learning, Istituto Italiano di Tecnologia, 16163 Genova, Italy
[2] Department of Mathematics, University of Genova, 16146 Genova, Italy
[3] Department of Computer Science, University College London, WC1E 6BT, London, UK

1

- dictionary learning,
- fast update of β at each step, fast update of D at the end of each task, truly online,
- very good empirical performances,
- LTL bound in

$$\mathcal{O}\left(\sqrt{\frac{1}{T}}+\sqrt{\frac{1}{n}}\right).$$

Algorithms : open questions

Open question 1

An efficient algorithm with theoretical guarantees (if possible beyond dictionary learning).

Algorithms : open questions

Open question 1

An efficient algorithm with theoretical guarantees (if possible beyond dictionary learning).

- theoretical analysis of ELLA ?

Algorithms : open questions

Open question 1

An efficient algorithm with theoretical guarantees (if possible beyond dictionary learning).

- theoretical analysis of ELLA ?
- can we justify to update D at each step ? this leads to the next big open problem...

Optimality of the bounds

- ELLA : updates D at each step. Doing so, after T tasks with n steps in each task, we would expect a bound in

$$\mathcal{O}\left(\sqrt{\frac{1}{nT}} + \sqrt{\frac{1}{n}}\right).$$

Optimality of the bounds

- ELLA : updates D at each step. Doing so, after T tasks with n steps in each task, we would expect a bound in

$$\mathcal{O}\left(\sqrt{\frac{1}{nT}} + \sqrt{\frac{1}{n}}\right).$$

- Denevi *et al* : bound in

$$\mathcal{O}\left(\sqrt{\frac{1}{T}} + \sqrt{\frac{1}{n}}\right).$$

Optimality of the bounds

- ELLA : updates D at each step. Doing so, after T tasks with n steps in each task, we would expect a bound in

$$\mathcal{O}\left(\sqrt{\frac{1}{nT}} + \sqrt{\frac{1}{n}}\right).$$

- Denevi *et al* : bound in

$$\mathcal{O}\left(\sqrt{\frac{1}{T}} + \sqrt{\frac{1}{n}}\right).$$

So, what are the optimal rates in LL & LTL ?

Insights from a toy model

- θ_1 fixed once and for all,
- task t : $\theta_{2,t}$ fixed for the task
- for $i = 1, \dots, n$, $y_{t,i} = (\theta_1 + \varepsilon_{1,i,t}, \theta_{2,t} + \varepsilon_{2,i,t})$ with $\varepsilon_{j,i,t} \sim \mathcal{N}(0, 1)$.

Insights from a toy model

- θ_1 fixed once and for all,
- task t : $\theta_{2,t}$ fixed for the task
- for $i = 1, \dots, n$, $y_{t,i} = (\theta_1 + \varepsilon_{1,i,t}, \theta_{2,t} + \varepsilon_{2,i,t})$ with $\varepsilon_{j,i,t} \sim \mathcal{N}(0,1)$.

$\hat{\theta}_1 = \frac{1}{nT} \sum_{t=1}^{T} \sum_{i=1}^{n} (y_{t,i})_1$ can be computed in the online setting and one has

$$\mathbb{E}\left(|\hat{\theta}_1 - \theta_1|\right) = \mathcal{O}\left(\sqrt{\frac{1}{nT}}\right).$$

Insights from a toy model

- θ_1 fixed once and for all,
- task t : $\theta_{2,t}$ fixed for the task
- for $i = 1, \dots, n$, $y_{t,i} = (\theta_1 + \varepsilon_{1,i,t}, \theta_{2,t} + \varepsilon_{2,i,t})$ with $\varepsilon_{j,i,t} \sim \mathcal{N}(0,1)$.

$\hat{\theta}_1 = \frac{1}{nT} \sum_{t=1}^{T} \sum_{i=1}^{n} (y_{t,i})_1$ can be computed in the online setting and one has

$$\mathbb{E}\left(|\hat{\theta}_1 - \theta_1|\right) = \mathcal{O}\left(\sqrt{\frac{1}{nT}}\right).$$

Fits our setting with $x = \emptyset$, $g_{\theta_1}(x) = \theta_1$, $h_{\theta_2}(z) = (z, \theta_2)$.

Insights from a toy model

- θ_1 fixed once and for all,
- task t : $\theta_{2,t}$ and $\varepsilon_{1,t} \sim \mathcal{N}(0,1)$ fixed for the task.
- for $i = 1, \dots, n$, $y_{t,i} = (\theta_1 + \varepsilon_{1,t}, \theta_{2,t} + \varepsilon_{2,i,t})$ with $\varepsilon_{2,i,t} \sim \mathcal{N}(0,1)$.

$\hat{\theta}_1 = \frac{1}{T}\sum_{t=1}^{T}(y_{t,i})_1$ can be computed in the online setting and one has

$$\mathbb{E}\left(|\hat{\theta}_1 - \theta_1|\right) = \mathcal{O}\left(\sqrt{\frac{1}{T}}\right).$$

Still fits our setting and LTL !

Optimal rates : open questions

Open question 2

What are the optimal rates in lifelong learning and in LTL ?

Optimal rates : open questions

Open question 2

What are the optimal rates in lifelong learning and in LTL ?

- requires to define properly class of predictors,

Optimal rates : open questions

Open question 2

What are the optimal rates in lifelong learning and in LTL ?

- requires to define properly class of predictors,
- the optimal rate will also depend on the setting. This leads to the next question...

Are our definitions even right ?

- Note that the terminology is not exen fixed : for example, Pentina and Lampert call lifelong learning what we call learning to learn (we don't claim we are right !).

Are our definitions even right ?

- Note that the terminology is not exen fixed : for example, Pentina and Lampert call lifelong learning what we call learning to learn (we don't claim we are right !).
- We used :
  1. LTL : samples from all the tasks presented at once.
  2. LL : tasks presented sequentially, within each task, pairs presented sequentially.
  3. why not tasks presented sequentially, but within each task, samples presented all at once ?

Are our definitions even right ?

- Note that the terminology is not exen fixed : for example, Pentina and Lampert call lifelong learning what we call learning to learn (we don't claim we are right !).
- We used :
  1. LTL : "Batch-within-batch"
  2. LL : "Online-within-online"
  3. "Batch-within-online", see our paper and Denivi *et al.*

Towards more models ?

One can imagine even more settings :

- observations not ordered by tasks ?

Towards more models ?

One can imagine even more settings :

- observations not ordered by tasks ?
- for some tasks, the information is complete, for other tasks, this is not the case. For example some tasks are sequential predictions, others are bandit problems.

Towards more models ?

One can imagine even more settings :

- observations not ordered by tasks ?
- for some tasks, the information is complete, for other tasks, this is not the case. For example some tasks are sequential predictions, others are bandit problems.
- more complicated : we use within tasks an algorithm for which we don't have a regret bound, for example deep neural network for image classification in self-driving cars. We have a partial feedback that is not the missclassification rate but depends on it : number of accidents, user feedback...

Towards more models ?

One can imagine even more settings :

- observations not ordered by tasks ?
- for some tasks, the information is complete, for other tasks, this is not the case. For example some tasks are sequential predictions, others are bandit problems.
- more complicated : we use within tasks an algorithm for which we don't have a regret bound, for example deep neural network for image classification in self-driving cars. We have a partial feedback that is not the missclassification rate but depends on it : number of accidents, user feedback...

Do we really need a paper for each possible variant ?...

Setting : open questions

Open question 3

Which settings are relevant ? Which settings are not ? To what extent is a general theory possible ?