Deep Transfer Learning in Ecosystems

Philippe Desjardins-Proulx[0,1,2]

[0]philippe.d.proulx@gmail.com
[1]Université du Québec, Quebec, Canada.
[2]Quebec Center for Biodiversity Science, McGill U., Quebec, Canada.

September 11, 2013

Contents

"Top-down" approaches

How Machine Learning (a branch of A.I.) gives birth to a model:

- Starts with a list of assumptions about inference.
- ...but not about how the universe behaves ("Nature in a box" [2])
- Feed data (the training set) to an algorithm (Machine Learning).
- The mathematical model is built by the algorithm.
- Highly effective for complex phenomena.
- "The unreasonable effectiveness of data" [5].

Surprisingly resilient to changes in assumptions: the main machine learning algorithms tend to achieve similar success rates (support vector machines, neural networks, decision trees) [3].

Artificial Intelligence and Statistics

Statistics and artificial intelligence have much in common. Both disciplines are concerned with planning, with combining evidence, and with making decisions. Neither is an empirical science. Each aspires to be a general science of practical reasoning.

Glenn Shafer, 1990.

Supervised Learning

Data-set $\mathcal{D}$ with n data points,
each defined with an input vector $\mathbf{x}$ and an output y:

$$\mathcal{D} = \{(\mathbf{x}, y)_i\}_{i=1}^{n}.$$

$\mathbf{x}$	y
$\{N_t, N_{t+1}, N_{t+2}, N_{t+3}, \text{humidity} \ldots\}$	N_{t+4}
{elevation, rain, vegetation type ...}	presense/absence
{species 0 group, species 1 group, phylo dist ...}	interactions
{vertices, edges, connectance, ...}	stable/unstable
{gene type, position, μ, ...}	polymorphic

Three Paradigms of Machine Learning

Supervised learning: $\{(\mathbf{x}, y)_i\}_{i=1}^{n} \mapsto (\hat{f}(\mathbf{x}) \mapsto y)$.	$\mathbf{x}$ = input/independent var.
	y = output/dependent var.
	n = number of data points
Unsupervised learning: $\{(\mathbf{x})_i\}_{i=1}^{n} \mapsto (\hat{f}(\mathbf{x}))$.	$\hat{f}$ = estimated function
	S = states (environments)
	A = actions
Reinforcement learning: $\langle S, A, T, R \rangle$.	T = transitions
	R = rewards

Supervised Learning ≈ Classification, Regression.
Unsupervised Learning ≈ Clustering [6].

Thesis Objective

Problems of Machine learning:

- "Not theory": just a bunch of specific models.
- Not very useful when faced with new problems with little date.
- Not very useful to understand biodiversity (many aspects).
- Solution: **build algorithms capable of transferring knowledge**.
- The inability to transfer is one of the biggest issue in ML [9, 13, 7, 1].

Goal of the thesis:

Learn a better model for $\mathcal{D}_{target}$ by exploiting knowledge built from $\mathbf{D} = \{\mathcal{D}_0, \mathcal{D}_1, \mathcal{D}_2, \ldots\}_{sources}$.

$$\{\text{walk}, \text{run}, \text{read}, \text{climb} \ldots\}$$

$$\{\text{food webs}, \text{pop dynamics}, \text{predator-prey systems}, \text{biogeography}, \ldots\}$$

TAMAR algorithm

One deep transfer algorithm:
TAMAR (Transfer via Automatic Mapping and Revision) [8].

$$D_{source} = \text{Food web}, D_{target} = \text{Host-Parasite network}.$$

$$\forall s_0 s_1 s_2, eat(s_0, s_2) \wedge eat(s_1, s_2) \wedge mass(s_0, s_1) \Rightarrow compete(s_0, s_1),$$

$$\forall s_0 s_1 s_2, parasite(s_0, s_2) \wedge parasite(s_1, s_2) \Rightarrow compete(s_0, s_1).$$

- Based on Markov Logic Networks [11, 4, 10].
- Step 1. Looks for mappings between the source and the target.
- Step 2. Generalize-Simplify the logical equations.
- Step 3. Search for new rules.

TAMAR: Problems

- Need to manually select the source.
- Only one source, one target...
- ...but the search space is too big with many sources.
- Do not learn to perform transfer (fixed method).
- How to get from TAMAR to multi-source deep transfer:

Strategy 1: We need smarter searches
i.e.: aggressive for promising mappings

Strategy 2: We need to *learn* how to perform effective transfers

Breaking the Redskins rule

x_0	x_1	y	x_0	x_1	y
4	0	1	8	0	0
8	0	0	4	1	1
4	1	1	8	0	0
8	0	0	4	1	1
4	1	1	20	0	0
12	0	0	16	1	1
8	1	1	12	1	1
4	1	1	8	1	1
4	0	0	4	1	1
8	0	0	12	1	0
4	1	1			

Breaking the Redskins rule

- y = US presidential election result.
- x_0 = Years in power for the incumbent.
- x_1 = Last home game of the Washington Redskins.

US presidential election = [presidential, election, president, vote, electoral, state, states, electors, candidates, candidate, elections, popular, who, us, presidents, political, votes, party, their, internet]
Incumbent = [incumbent, for, incumbents, office, an, election, voters, challenger, often, win, term, over, elections, factor, may, democratic, open, because, who, united]
Washington Redskins = [redskins, season, game, nfl, team, coach, washington, games, against, record, yards, home, head, draft, cowboys, dallas, defensive, quarterback, playoffs, gibbs]

Breaking the Redskins rule

$$sim(\text{“US presidential election”}, \text{“Washington Redskins”}) = 0.73,$$

$$sim(\text{“US presidential election”}, \text{“Incumbent”}) = 2.29.$$

Natural Language Processing (NLP)
can help guide the algorithm toward more relevant data.

Google's WORD2VEC (ANSI C99): https://code.google.com/p/word2vec/

$$\mathcal{D} = \{(\mathbf{x}, y)_i\}_{i=1}^{n} \qquad \mathbf{D} = \{\mathcal{D}_1, \mathcal{D}_2, \ldots\}$$

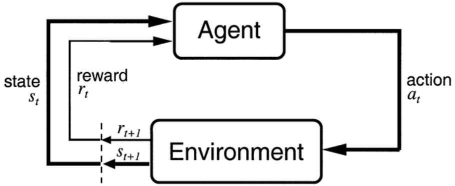
Supervised ML Algorithm

Agent

$$p\left(f|y, \mathbf{x}\right) = \frac{p(y|\mathbf{x}, f)p(f)}{p(y|\mathbf{x})}$$

$$\hat{f} : \forall f' \in f, p(\hat{f}|y, \mathbf{x}) \geq p(f'|y, \mathbf{x})$$

MLN-based Reinforcement Learning Agent $\langle S, A, T, R \rangle$:



- S: Data-sets (sources and the target).
- A: Searching for mappings & setting hyperparameters.
- No transitions $T : S \times A \times S \mapsto \mathbb{R}$.
- $R : A \times S \mapsto \mathbb{R}$, tested against "plain" SL.
- Internal state is a Markov Logic Networks [14] or neural network [12].

The case for separation of concerns:

- Learn to perform transfer, no fixed method.
- Set of actions A can be defined to fit any supervised learning (SL).
- Different agents can try to learn with different SL algorithms.

“Plan B”

- Where it could go wrong: not enough information in the labels.
- Require more semantic info for the labels. . .
- . . . e.g.: $x_1 = \{\text{ecological interaction}, \text{competition}\}$.
- Tackle the single-source problem (adaptive learning).

Implementation:

- High performance Reinforcement Agent written in C and C++.
- I do *not* code SL (e.g.: Alchemy, libsvm) or NLP (e.g.: word2vec).
- Web user-interface.

Testing:

- Test early, test often.
- Simple: compare to plain SL algorithms.
- Agents can compete with each others.
- Apply to species distributions (different species).
- Apply to interaction networks (different networks/area).
- Apply to $\{X\}\ldots$

Y Bengio.
AAAI tutorial: Deep learning of representations.
2013.

L Breiman.
Statistical modeling: the two cultures.
Statistical Science, 16(3):199–231, 2001.

J Cheng, AN Tegge, and P Baldi.
Machine learning methods for protein structure prediction.
IEEE Reviews in Biomedical Engineering, 1:41–49, 2008.

P Domingos and D Lowd.
Markov Logic: An Interface Layer for Artificial Intelligence.
Morgan & Claypool Publishers, 2009.

A Halevy, P Norvig, and F Pereira.
The unreasonable effectiveness of data.
IEEE Intelligent Systems, 24:8–12, 2009.

T Hastie, R Tibshirani, and J Friedman.

The Elements of Statistical Learning: Data Mining, Inference, and Prediction.
Springer, 3rd edition, 2009.

A Lazaric.
Transfer in reinforcement learning: A framework and a survey.
In M Wiering and M van Otterlo, editors, *Reinforcement Learning: State-of-the-Art*, volume 12 of *Adaptation, Learning, and Optimization*, pages 143–173. Springer, 2012.

L Mihalkova, T Huynh, and RJ Mooney.
Mapping and revising markov logic networks for transfer learning.
In *Proceedings of the 22nd Conference on Artificial Intelligence*, pages 608–614, 2007.

TM Mitchell.
The discipline of machine learning.
2006.

KP Murphy.
Machine Learning: A Probabilistic Perspective.

The MIT Press, 2012.

M Richardson and P Domingos.
Markov logic networks.
Machine Learning, 62(1-2):107–136, 2006.

RS Sutton and AG Barto.
Reinforcement learning: An introduction.
Second edition (in progress), 2012.

L Torrey and J Shavlik.
Transfer learning.
In E Soria, J Martin, R Magdalena-Benedito, M Martinez, and A Serrano Lopez, editors, *Handbook of Research on Machine Learning Applications and Trends: Algorithms, Methods, and Techniques*, pages 242–264. IGI Global, 2010.

L Torrey, J Shavlik, S Natarajan, P Kuppili, and T Walker.
Transfer in reinforcement learning via Markov Logic Networks.
In *AAAI Workshop on Transfer Learning for Complex Tasks*, 2008.