Transfer in Time-Series Data Streams

Jesse Read

Time series and transfer learning workshop @ Huawei Paris
19th of October 2023

Agenda for This Talk

Transfer in Time-Series Data Streams

Learning from data streams.

Modern data streams often arise in the context of time series data, including reinforcement learning. Learning methodologies, concept drift, continual learning, transfer learning: we[1] aim to provide a more holistic view.

We[2] will take a look at model-agnostic transfer in stream-based learning paradigms: supervised, unsupervised, reinforcement learning.

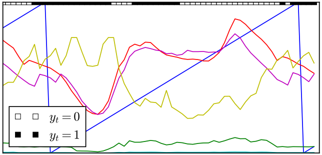
[1] Read and Žliobaitė, *Learning from Data Streams: An Overview and Update*, 2023; Žliobaitė and Read, *A Historical Context for Data Streams*, 2023

[2] Read, "From Multi-label Learning to Cross-Domain Transfer: A Model-Agnostic Approach", 2023; Chehboune, Kaddah, and Read, "Transferable Deep Metric Learning for Clustering", 2023

Outline

Learning from Data Streams: Revisiting Assumptions

Data Streams are Everywhere



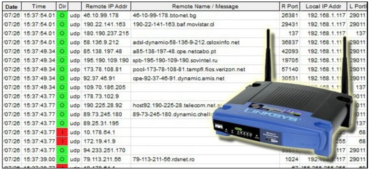


Image Source: [1]

Applications: IoT, energy and traffic systems, and demand prediction, monitoring and tracking, event and fraud detection, click/web logs, finance, health, news, social networks, forecasting, reinforcement learning, ...

Data Streams

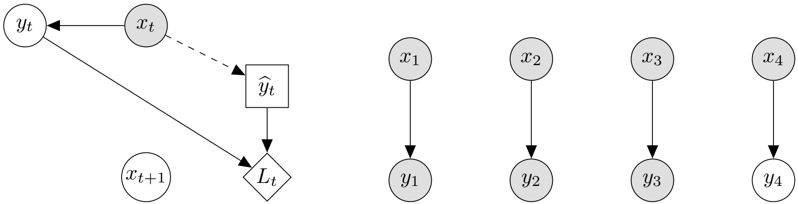
A data stream,

$$x_1, x_2, \ldots, x_t, \ldots, x_\infty$$

where, at real time t we observe x_t, generated from concept (generating distribution)

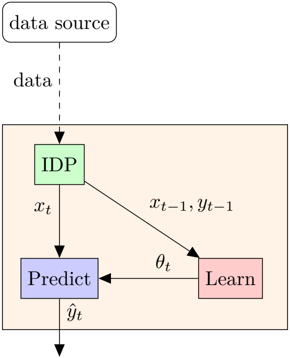
$$x_t \sim P(X)$$

Or, $(x_t, y_t) \sim P(X, Y)$ (supervised case); e.g., (iid assumption):



But sometimes we *don't get* x_t at time t; and data is not iid!

Data Streams: an Implementation Issue



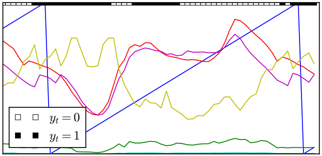
IDP = Instance Delivery Process; a software/hardware issue (requires buffer, or – what to if data doesn't arrive on time? – delay, missing values, concept shift, etc.)

Typical models (θ_t): k-nearest neighbours, incremental decision trees, neural networks, . . . , model agnostic.

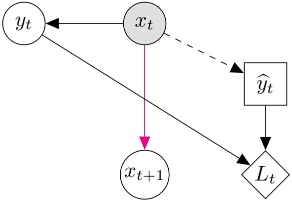
Multiple streams: x_t, y_t, stream θ_t, stream $\widehat{y}_t$, etc (ϵ_t, . . .).

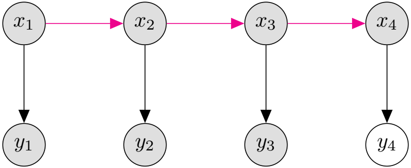
Time Series Data Streams

Nearly all realistic applications of online learning applied to data streams involve time series forecasting.

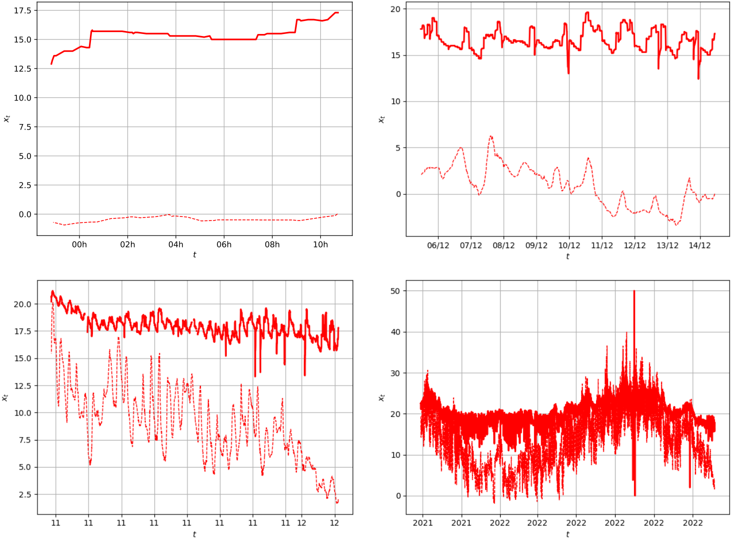


But online learning is only one option (batch-learning is allowed!). For example (non-iid data),





Example: Temperature

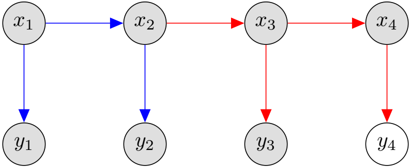


Concept Drift

Everyone who talks about '*data stream learning*' usually must talk about concept drift. This means:

$$P_{t-1}(X, Y) \neq P_t(X, Y)$$

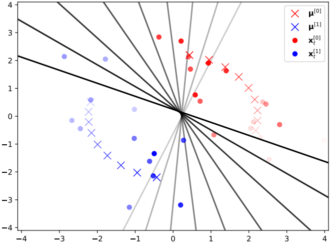
for at least some t (note time indices on P!).

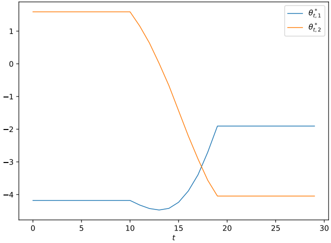


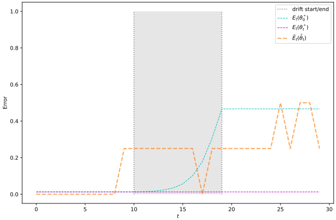
Also known as *domain shift*, *co-variate shift* (if specific to X), *change point*, etc.

What is the difference between concept drift in data streams and regular (stationary) time series?
All plots in previous slide *refer to the same data*!

Detecting Concept Drift

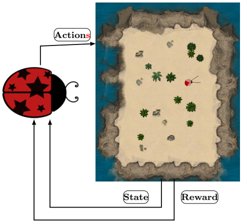


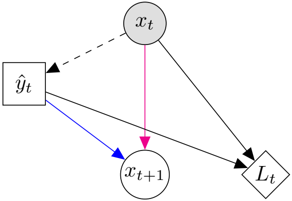


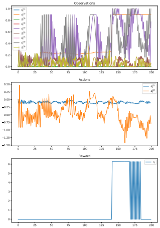


Concept drift is typically be detected in the error signal (stream), *in the case of fully-supervised stream learning*.

Reinforcement Learning = Learning from Time Series







In reinforcement learning, we observe

- stream of actions – *outputs which may affect the future*!
- stream of state-observations
- stream of rewards

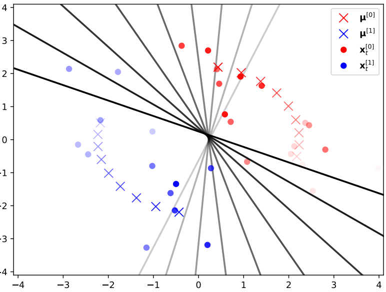
Output of learning: function, $\pi : a_t \mapsto s_t$
just like classifier or regressor (but called a policy).

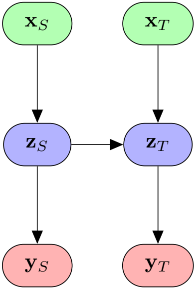
Almost no mention of 'data stream learning', 'concept drift', etc[3] in the literature.

[3] Chaouki, Read, and Bifet, "Online Decision Tree Construction with Deep Reinforcement Learning", 2023

Model Agnostic Cross-Domain Transfer

Adaptation to Concept Drift = Transfer Learning



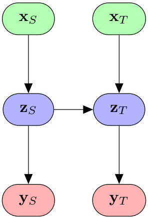


Transfer Learning

Quick guide to transfer learning:

1. Find related source task
2. Use it to improve the model you deploy on target task

For example, deep transfer learning (forward transfer, from Source task to Target):



In data stream learning:

- source task = prior concept
- target task = current concept.

Model-Agnostic Model-based Transfer

Traditional assumption: target task similar to source task.

Task similarity $=1$ implies optimal results (no transfer required).
What is similarity $=0$?
But also, 'variance (*dissimilarity*) in training domains is good' (for generalisation/transfer) – in contradiction with previous point!

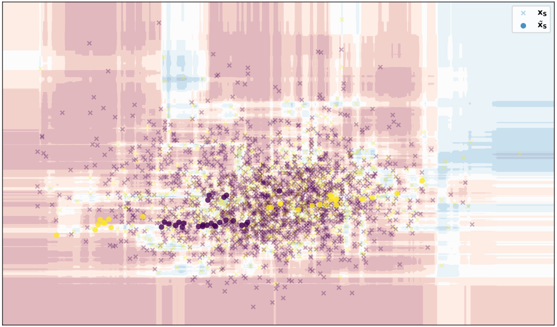
$$\begin{tikzcd}
\tilde{\mathbf{x}}_S \arrow[d, "h_S"] & \mathbf{x}_T \arrow[l, "f"'] \arrow[d, "h_T"] \\
\widehat{\mathbf{y}}_S \arrow[r, "h_T"] & \mathbf{y}_T
\end{tikzcd}$$

Main idea: use predictions from old models as features,

- No fine-tuning required (old model $=$ 'frozen' layer)
- Model agnostic (neural network not required)
- Can be cascaded across time (in the context of streams)
- Works 'without any relatedness'!

Insomniac Fungi

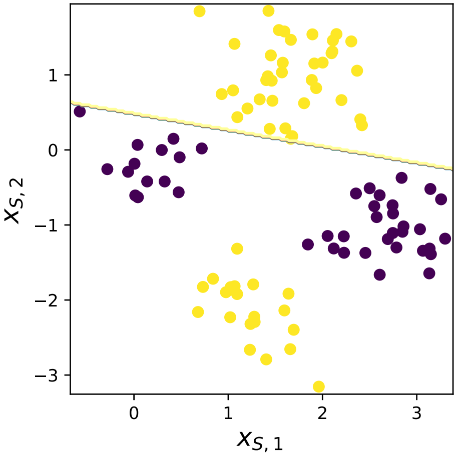
Model (random forest) for classifying (human) insomniac patients according to their answers to a psychological questionnaire ($\boldsymbol{x}$):

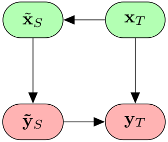


Features from a yeast genome ($\boldsymbol{x}'$; different problem) is cast into $\boldsymbol{x}$ instead, and given an insomnia profile $\widehat{\boldsymbol{y}}'$; result: $+2\%$ accuracy when used as a feature for *predicting genome function*.

XOR Example

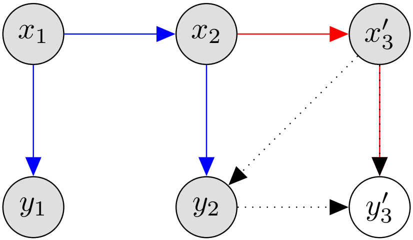
A target task XOR (data shown, some noise added) is solved via predictions from AND-function (decision boundary shown) as an additional feature:





In Application to Concept-Drift Adaptation (In Streams)

In data streams, it means we can continue using the old model, prior to drift, *even if drift was drastic*.



Data is generated (we don't see this) as

$$x, y \sim P(X, Y)$$
$$x', y' \sim P'(X, Y)$$

Prediction as

$$\widehat{y}' = h'(x', h(x'))$$

Model-based Transfer; Discussion Points

- All models are related (somehow)
- New take on old recipe: predictive power + regularization; intersection with ideas such as pretrained-, frozen layers ('parameter isolation'), random basis functions, 'extreme learning machines', ...
- The more multi-label the better (richer $\widehat{\boldsymbol{y}}$)
- Transfer learning vs reduction/reuse/recycling of models?
- Model-driven learning vs data-driven learning
- Implication: The old 'irrelevant' models are not!

Model-based Transfer; Discussion Points

- All models are related (somehow)
- New take on old recipe: predictive power + regularization; intersection with ideas such as pretrained-, frozen layers ('parameter isolation'), random basis functions, 'extreme learning machines', . . .
- The more multi-label the better (richer $\widehat{\boldsymbol{y}}$)
- Transfer learning vs reduction/reuse/recycling of models?
- Model-driven learning vs data-driven learning
- Implication: The old 'irrelevant' models are not!

Main limitations:

- It doesn't always work; savings (by using old models) eventually outweighed by cost (of using old models).
- Reliance on the richness and utility of the predictions/features ($\widehat{\boldsymbol{y}}$).

Metric-based Transfer

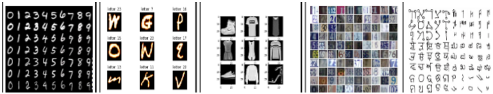
Metric-based Transfer (for Clustering)

A good reward r_ϕ (metric; equivalently loss) can facilitate rapid learning. Clustering:

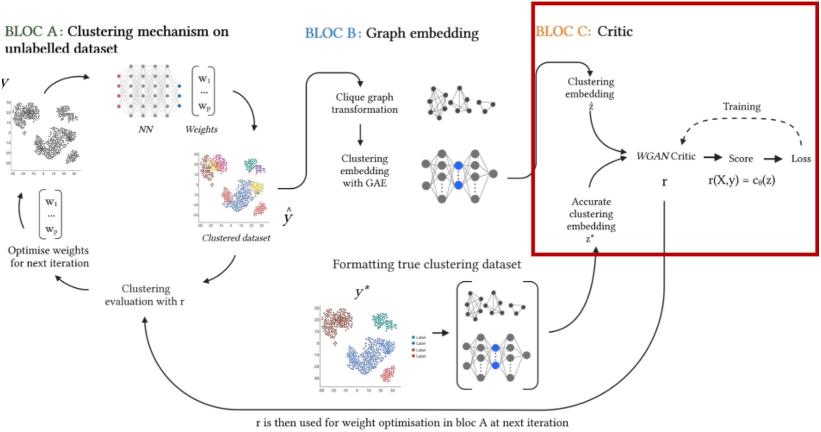
$$\widehat{\mathbf{y}} = \underset{\mathbf{y}}{\operatorname{argmax}}\, r_\phi(\mathbf{y}; \mathbf{X})$$

where $r_\phi \in \mathbb{R}$ is big when clustering is good. But clustering is unsupervised – what is 'good'?

Answer: we have many examples of $\mathbf{X}, \mathbf{y}_*$ to learn from (existing labeled datasets)!



Inside the framework of a GAN (generate and critique a 'clustering'),



we learn ϕ. Aim: walk away with r_ϕ *suited to any* data $\mathbf{X}$; with an off-the-shelf optimizer:

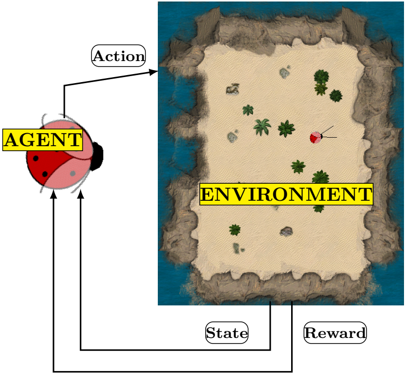
$$\widehat{\mathbf{y}} = \operatorname*{argmax}_{\mathbf{y}} r_\phi(\mathbf{X}, \mathbf{y})$$

Chehboune, Kaddah, and Read, "Transferable Deep Metric Learning for Clustering", 2023

Reward-based Transfer in Reinforcement Learning

Transfer in Reinforcement Learning

Model-driven (frozen/chain-based) transfer even less likely to work in reinforcement learning: causality, highly-dynamic environment, more significant and more frequent concept drift/shift (including action and state dimensionality/observability), etc. And exploration is important – but optimal policy may be deterministic.



Reward-based Transfer in Reinforcement Learning

Typical suggestions for storage of knowledge in RL:

- Policy π
- Q-function
- Environment model $p(s' \mid a, s)$
- Features/latent layers

We choose:

$$r_\phi$$

Meta-learning (learning to learn); a kind of transfer learning, very close to multi-task learning.

Our vector of transfer is ϕ!

Indeed, we want a metric (reward) for multiple tasks! In reinforcement learning, the reward function is really a big deal!

Inverse Reinforcement Learning; or: From where to get r_ϕ

Imitation learning may (if lucky) provide $\pi \approx \pi_*$, i.e., the *expert's behaviour*; but we need inverse reinforcement learning for the discovery of r (what the expert is maximizing).

- Usual purpose: learning about human preferences in complex systems (energy networks, games, autonomous vehicles, ...);
- Our primary purpose: Transfer to new environments.

In reinforcement learning: $(p, r) \rightsquigarrow \pi_*$
(from [unknown] environment to policy, via interaction)

In inverse reinforcement learning (IRL): $(p, \pi_*) \rightsquigarrow r$
(from policy and environment to reward via observation).

$$r_\phi(s_t, a_t)$$

Ongoing work; Simo Alami Chehbourne thesis – under review!

Summary and Conclusions

Summary

Some general views (and specific approaches), providing:

- Revision of some problematic assumptions in data stream learning – and stronger links to time series
- Reinforcement learning as time-series data-stream learning
- Connection between adaptation to concept drift and transfer learning (including, in unsupervised, supervised, and reinforcement learning)
- A model-driven cross-domain transfer; implications for adaptation to drift: transfer via prediction vector from old models.
- And metric-based transfer (for clustering, reinforcement learning – when there's no label vector), via inverse reinforcement learning; connection between GANs and critics.

Transfer in Time-Series Data Streams

Jesse Read

Time series and transfer learning workshop @ Huawei Paris

19th of October 2023

Thank you!

http://www.lix.polytechnique.fr/~jread/
jesse.read@polytechnique.edu

References I

Acknowledgements to co-authors (and references to related work within):

- Chaouki, Ayman, Jesse Read, and Albert Bifet. "Online Decision Tree Construction with Deep Reinforcement Learning". In: *EWRL '23: Sixteenth European Workshop on Reinforcement Learning*. 2023.
- Chehboune, Mohamed Alami, Rim Kaddah, and Jesse Read. "Transferable Deep Metric Learning for Clustering". In: *IDA 2023: Advances in Intelligent Data Analysis XXI, 21st International Symposium*. 2023, pp. 15–28. URL: https://link.springer.com/chapter/10.1007/978-3-031-30047-9_2.
- Read, Jesse. "From Multi-label Learning to Cross-Domain Transfer: A Model-Agnostic Approach". In: *Applied Intelligence* 08.2023 (2023), pp. 1537–7497. URL: http://arxiv.org/abs/2207.11742.

References II

Read, Jesse and Indrė Žliobaitė. *Learning from Data Streams: An Overview and Update*. Tech. rep. 2212.14720. ArXiv. ArXiv.org, 2023. URL: https://arxiv.org/pdf/2212.14720.pdf.

Žliobaitė, Indrė and Jesse Read. *A Historical Context for Data Streams*. Tech. rep. 'in press'. ArXiv. ArXiv.org, 2023. URL: TBA.