Multi-label Learning: An Update

Jesse Read

ÉCOLE POLYTECHNIQUE
IP PARIS

June 9, 2023 @ BCAM. Bilbao.

5th BIDAS Workshop

Outline

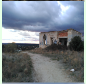
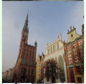
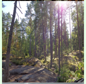
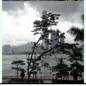
Introduction: Multi-Label Learning

Multi-label Classification

Multi-label classification: a subset/vector of labels is be assigned to each input instance.

$\boldsymbol{x} =$ 

$\boldsymbol{y} = [1, 0, 1, 0] \Leftrightarrow$ labels $\{\texttt{Beach}, \texttt{Foliage}\}$ are relevant to $\boldsymbol{x}$.

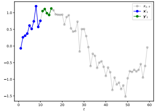
Input	Beach	Sunset	Foliage	Urban
	1	0	1	0
	0	1	0	0
	0	1	0	1
	0	1	1	0
	0	0	1	1
	?	?	?	?

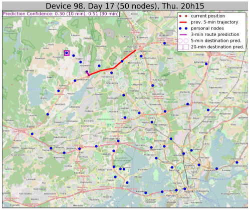
Task:

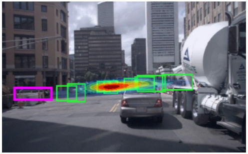
$$\widehat{\boldsymbol{y}} = [?, ?, ?, ?] = h(\boldsymbol{x}) \qquad \widehat{\boldsymbol{y}} \in \{0, 1\}^m$$

Also,

- text categorization
- missing-value imputation
- recommender systems
- time-series forecasting
- network inference
- tracking and localization
- image segmentation
- molecule design
- audio labelling
- ...







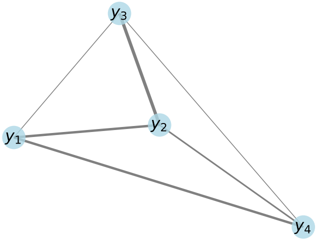
	Mol1	Mol2	Mol3	Mol4	Mol5	Mol6
	1,3	0,2	1,4	1,7	3,5	1,3
	2	1,7	1,5	7,5	8,2	7,6
	0,2	0	0,3	0,4	1,2	2,2
	3,1	1,1	1,3	1,1	1,7	5,2
	4,7	2,1	2,5	1,5	2,3	8,5
	?	?	?	?	?	?

Label Dependence: The 'Why' of Multi-Label Learning

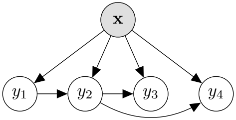
Graph of correlation among the labels of the *Music-Emotions* data

Standard 'Recipe'/Traditional Approach

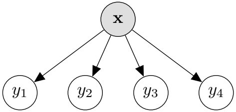
1. 'We measure label dependence using <xxx>'



2. 'We construct a model called <yyy>'
3. 'We show <zzz>%-improvement vs independent models'



vs

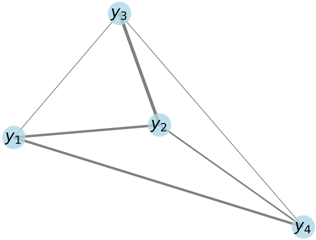


vs . . . ?

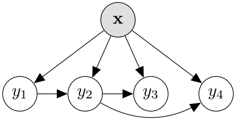
Implication: Predictive performance ⇔ **label dependence**.

Standard 'Recipe'/Traditional Approach

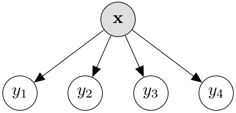
1. 'We measure label dependence using <xxx>'



2. 'We construct a model called <yyy>'
3. 'We show <zzz>%-improvement vs independent models'



vs



vs . . . ?

Implication: Predictive performance $\Leftrightarrow$ **label dependence**.

This talk: A fresh investigation, and an updated view.

A Timeline of Multi-label Learning in Academia

$<$ 2000s Just use independent models.

. . . 2010 Model labels together; label dependence/co-occurrences.

. . . 2015 Using label dependence in a more sophisticated/efficient way.

. . . 2015 Multi-label learning for image, text, forecasting, recommendation, audio, health applications, distilling wine . . .

2020 [. . . and for covid19].

. . . 2020 Just use independent models

. . . 2020 Must use deep [convolution / recurrent] neural networks.

2020 . . . . . . deep [graph-embedding / residual / generative adversarial / transformer/. . .] neural networks with [missing / weak / incremental / evolving / imbalanced / millions of/. . .] labels.

2023 Still persistent in the literature[1]

[1] Mylonas et al., "On the Persistence of Multilabel Learning, Its Recent Trends, and Its Open Issues", 2023

A Timeline of Multi-label Learning in Academia

$<$ 2000s Just use independent models.

. . . 2010 Model labels together; label dependence/co-occurrences.

. . . 2015 Using label dependence in a more sophisticated/efficient way.

. . . 2015 Multi-label learning for image, text, forecasting, recommendation, audio, health applications, distilling wine . . .

2020 [. . . and for covid19].

. . . 2020 Just use independent models

. . . 2020 Must use deep [convolution / recurrent] neural networks.

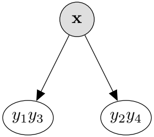
2020 . . . . . . deep [graph-embedding / residual / generative adversarial / transformer/. . .] neural networks with [missing / weak / incremental / evolving / imbalanced / millions of/. . .] labels.

2023 Still persistent in the literature[1]

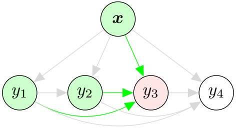
Multi-target regression? $< 1/10$-th volume of literature.

[1] Mylonas et al., "On the Persistence of Multilabel Learning, Its Recent Trends, and Its Open Issues", 2023

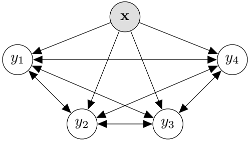
Multi-label Classifiers: Examples



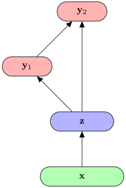
Random k-Label Sets and Meta Labels



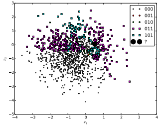
Classifier Chains and Bayesian Networks



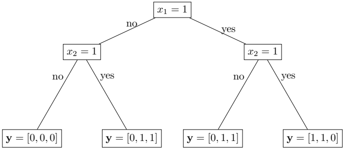
Conditional Dependency Networks



Neural Networks



k-Nearest Neighbours



Decision Trees and Random Forests

Algorithm Adaptation vs Task Adaptation / Problem Transformation

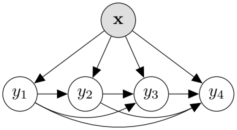
---

Refs. in Bogatinovski et al., "Comprehensive comparative study of multi-label classification methods", 2022; plus Cisse, Al-Shedivat, and Bengio, "ADIOS: Architectures Deep In Output Space", 2016

Classifier Chains: An Example of ‘Problem Transformation’

A chain (structure) over the output variables;

- Cascaded prediction across a chain/graph
- Motivation: Model label dependence



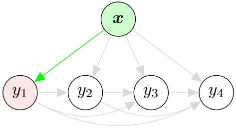
$\boldsymbol{X}$	Y_1	Y_2	Y_3	Y_4
$\boldsymbol{x}^{(1)}$	0	1	1	1
$\boldsymbol{x}^{(2)}$	1	0	0	0
$\boldsymbol{x}^{(3)}$	0	1	0	1
$\boldsymbol{x}^{(4)}$	1	0	0	0
$\boldsymbol{x}^{(5)}$	0	0	0	0
$\tilde{\boldsymbol{x}}$	$\widehat{y_1}$	$\widehat{y_2}$	$\widehat{y_3}$	$\widehat{y_4}$

Read et al., ECML-PKDD 2009 and Read et al., “Classifier Chains: A Review and Perspectives”, 2021

Classifier Chains: An Example of ‘Problem Transformation’

A chain (structure) over the output variables;

- Cascaded prediction across a chain/graph
- Motivation: Model label dependence



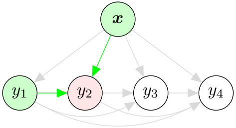
$\boldsymbol{X}$	Y_1	Y_2	Y_3	Y_4
$\boldsymbol{x}^{(1)}$	0	1	1	1
$\boldsymbol{x}^{(2)}$	1	0	0	0
$\boldsymbol{x}^{(3)}$	0	1	0	1
$\boldsymbol{x}^{(4)}$	1	0	0	0
$\boldsymbol{x}^{(5)}$	0	0	0	0
$\tilde{\boldsymbol{x}}$	$\widehat{y_1}$			

Read et al., ECML-PKDD 2009 and Read et al., “Classifier Chains: A Review and Perspectives”, 2021

Classifier Chains: An Example of 'Problem Transformation'

A chain (structure) over the output variables;

- Cascaded prediction across a chain/graph
- Motivation: Model label dependence



$\boldsymbol{X}$	Y_1	Y_2	Y_3	Y_4
$\boldsymbol{x}^{(1)}$	0	1	1	1
$\boldsymbol{x}^{(2)}$	1	0	0	0
$\boldsymbol{x}^{(3)}$	0	1	0	1
$\boldsymbol{x}^{(4)}$	1	0	0	0
$\boldsymbol{x}^{(5)}$	0	0	0	0

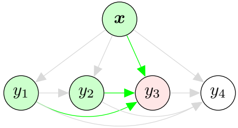
$\tilde{\boldsymbol{x}}$	$\widehat{y_1}$	$\widehat{y_2}$		

Read et al., ECML-PKDD 2009 and Read et al., "Classifier Chains: A Review and Perspectives", 2021

Classifier Chains: An Example of ‘Problem Transformation’

A chain (structure) over the output variables;

- Cascaded prediction across a chain/graph
- Motivation: Model label dependence



$\boldsymbol{X}$	Y_1	Y_2	Y_3	Y_4
$\boldsymbol{x}^{(1)}$	0	1	1	1
$\boldsymbol{x}^{(2)}$	1	0	0	0
$\boldsymbol{x}^{(3)}$	0	1	0	1
$\boldsymbol{x}^{(4)}$	1	0	0	0
$\boldsymbol{x}^{(5)}$	0	0	0	0
$\tilde{\boldsymbol{x}}$	$\widehat{y}_1$	$\widehat{y}_2$	$\widehat{y}_3$	

For example, $\widehat{y}_3 = h_3(\boldsymbol{x}, \widehat{y}_1, \widehat{y}_2)$ with base classifier (or regressor) h_3 (e.g., decision tree, logistic regression, ...).

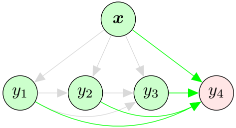
Typical example of a “problem transformation" (or model agnostic) meta method that works well vs independent models

Read et al., ECML-PKDD 2009 and Read et al., “Classifier Chains: A Review and Perspectives”, 2021

Classifier Chains: An Example of 'Problem Transformation'

A chain (structure) over the output variables;

- Cascaded prediction across a chain/graph
- Motivation: Model label dependence



$\boldsymbol{X}$	Y_1	Y_2	Y_3	Y_4
$\boldsymbol{x}^{(1)}$	0	1	1	1
$\boldsymbol{x}^{(2)}$	1	0	0	0
$\boldsymbol{x}^{(3)}$	0	1	0	1
$\boldsymbol{x}^{(4)}$	1	0	0	0
$\boldsymbol{x}^{(5)}$	0	0	0	0
$\tilde{\boldsymbol{x}}$	$\widehat{y}_1$	$\widehat{y}_2$	$\widehat{y}_3$	$\widehat{y}_4$

For example, $\widehat{y}_3 = h_3(\boldsymbol{x}, \widehat{y}_1, \widehat{y}_2)$ with base classifier (or regressor) h_3 (e.g., decision tree, logistic regression, ...).

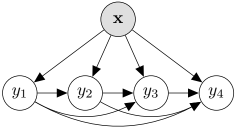
Typical example of a "problem transformation" (or model agnostic) meta method that works well vs independent models – but why?

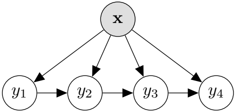
Read et al., ECML-PKDD 2009 and Read et al., "Classifier Chains: A Review and Perspectives", 2021

Issues with the Standard Approach/Assumptions

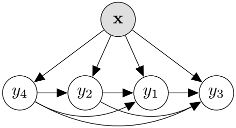
War Story 1 (Initial Intuition Fails)

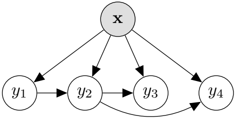
These models perform well:





These ones perform not so well:



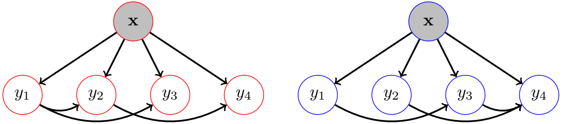


Yet it is difficult to associate accuracy to a particular [type of] structure based on dependence, time order, or ‘inherent’ hierarchy.

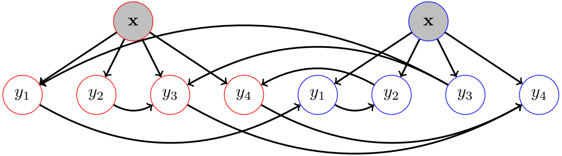
War Story 2 (Sanity Check Fails)

Take two totally unrelated datasets; stick them together; search for inherent structure.

Hypothesis: Find something like this,



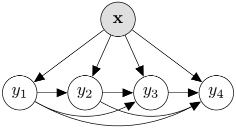
Outcome: Found something like this,



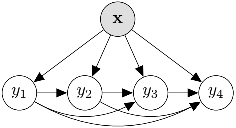
Remark: It's not [only] a small data problem!

Credits to Laurence Park here

War Story 3 (More Weirdness)



outperforms



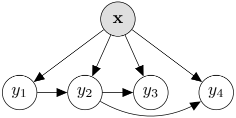
Average accuracy over 100 random train/test splits:

$$\text{(Left) } 0.47 > 0.41 \text{ (Right)}$$

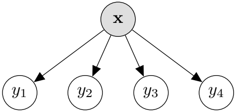
and the left wins 100/100 times! Yet, it's the

- **same model** (classifier chains)!
- same base classifier (SGD)
- same structure!
- same data (Scene dataset; same splits)
  except: $\mathbf{Y}_{\text{Right}} = \mathbf{1} - \mathbf{Y}_{\text{Left}}$ (all bits are flipped).

War Story 4 (Theory != Practice?)



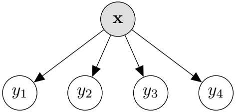
outperforms



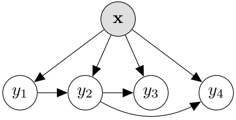
(significantly) *under* *Hamming loss* metric which does not require joint modelling to optimize[2]!

[2]Neither do ranking-based metrics, by the way; Dembczyński et al., "On Label Dependence and Loss Minimization in Multi-label Classification", 2012

War Story 5 (Back to Square One?)



equals performance of



under 0/1-*Loss*/exact-match metric which requires joint modelling to optimize, and even though we know there is label dependence. (By the way: especially common in multi-target regression[3]).

[3] Borchani et al., "A Survey on Multi-output Regression", 2015

Label Dependence: A Fresh Investigation and Update

Suggestion 1: Because Label Dependence

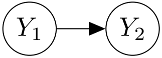
Argument: If label variables are correlated/interdependent, we should model/predict them together; accuracy will better.

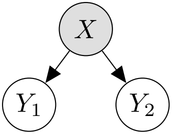
Label dependence:

$$P(Y_1, Y_2) \neq P(Y_1)P(Y_2)$$

Actually, we should be interested in conditional dependence:

$$P(Y_1, Y_2|\boldsymbol{x}) \neq P(Y_1|\boldsymbol{x})P(Y_2|\boldsymbol{x})$$

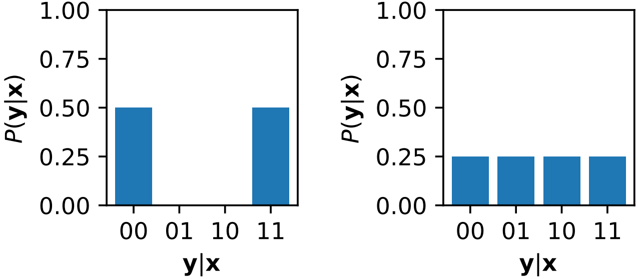
e.g., 

vs 

Dembczyński et al., "On Label Dependence and Loss Minimization in Multi-label Classification", 2012

Label Dependence and Loss Metrics

Posterior of two multi-label classifiers (2 labels, test instance **x**):



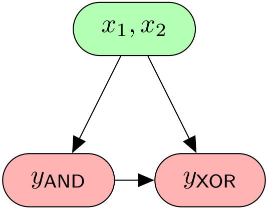
$\mathbb{E}$[Hamming loss] the same; $\mathbb{E}$[0/1-loss]: twice as large!

Not only a question of dependence, but of loss metrics and uncertainty; modelling together $\neq$ predicting together.

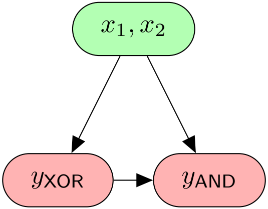
Figures from work with Ekaterina Antonenko and Ander Carreño

A ‘Logical’ Problem: The ‘Wrong’ Dependence

X_1	X_2	Y_{XOR}	Y_{AND}
0	0	0	0
0	1	1	1
1	0	1	1
1	1	0	1



outperforms
(sometimes†)



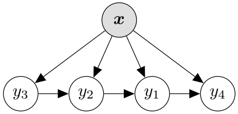
†when

$$P_\star(y_1, y_2 \mid \boldsymbol{x}) \neq \widehat{P}(y_1, y_2 \mid \boldsymbol{x})$$

where $\widehat{P}$ depends on base classifier, inference, etc. We measured the ‘wrong’ dependence; but got extra capacity from it!

Suggestion 2: Put Easy Labels First

Argument: There may be error propagation across the structure, so we should, e.g., put easy labels first.



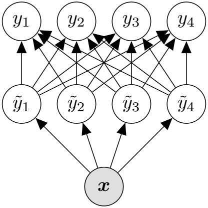
But

- Empirically: *Incorrect* label predictions may also *increase* the accuracy of *other* label predictions!
- Observation $\boldsymbol{x}$ is available at each step; error should *not* propagate![4]

[4] A more complete discussion in Senge, Coz, and Hüllermeier, "On the Problem of Error Propagation in Classifier Chains for Multi-label Classification", 2014 (probabilistic/probability-tree view)

Suggestion 3: Error Correction

Argument: We can ‘correct’ errors at prediction time, e.g., via stacking.



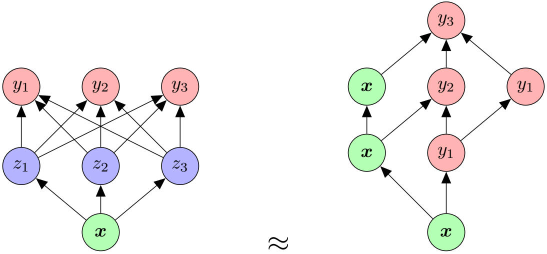
Maybe we can[5]; but

- $P(y_1|\tilde{y}_1, \tilde{y}_2, \tilde{y}_3, \tilde{y}_4) \neq P(y_1, y_2, y_3, y_4|\boldsymbol{x})$, i.e., this is not label dependence modelling, we only correct bias of individual models;
- Empirical results: Not much improvement (esp. in 0/1 loss)
- involves a separate training mechanism for each layer.

[5] e.g., (among many others) Loza Mencía and Janssen, “Learning rules for multi-label classification: a stacking and a separate-and-conquer approach”, 2016

Suggestion 4: Build (Deep) Neural Networks

Argument: Structure among labels ⇒ 'deep' 'neural' network.
Classifiers as activation/transfer functions, labels as hidden nodes.
A bit like ResNets.

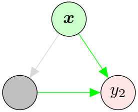


OK (depth works). But, note in 'deep chains' (right):

- No back propagation (this is deep *prediction*, but not deep learning);
- the hidden nodes are not 'hidden'.

Consider prediction task

$$\tilde{x} \mapsto \widehat{y}_2$$



and the data available at training time (left) vs test time (right):

	X_1		Y_2
Basis expansion	x	ϕ_1	y_2
Stacking	x	$\tilde{y}_2$	y_2
Classifier chain	x	y_1	y_2
Neural network	x		y_2

X_1		Y_2
$\tilde{x}$	ϕ_2	$\widehat{y}_2$
$\tilde{x}$	$\tilde{y}_2$	$\widehat{y}_2$
$\tilde{x}$	$\widehat{y}_1$	$\widehat{y}_2$
$\tilde{x}$	z	$\widehat{y}_2$

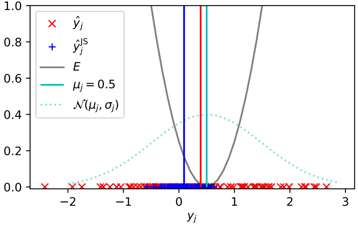
Suggestion 5: Structure Provides Regularisation

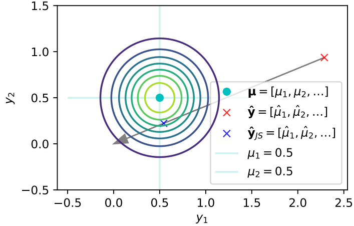
Argument: Modelling labels together provides better results even if they are independent, because of regularization.

For example the James Stein estimator

$$\widehat{\boldsymbol{y}}_{JS} = \frac{1-(m-2)\hat{\sigma}^2}{\|\widehat{\boldsymbol{y}}\|^2}\widehat{\boldsymbol{y}} = \lambda \cdot \widehat{\boldsymbol{y}}$$

where λ *shrinks* (regularises) the max.-likelihood estimate $\widehat{\boldsymbol{y}}$.

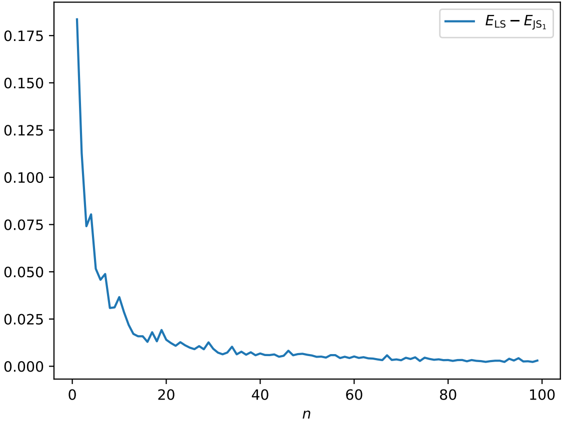




This helps explain the bit-inversion story!

Good discussion by Waegeman, Dembczyński, and Hüllermeier, "Multi-target prediction: a unifying view on problems and methods", 2019

Benefit from modelling non-existent label dependence (shown where blue > 0):



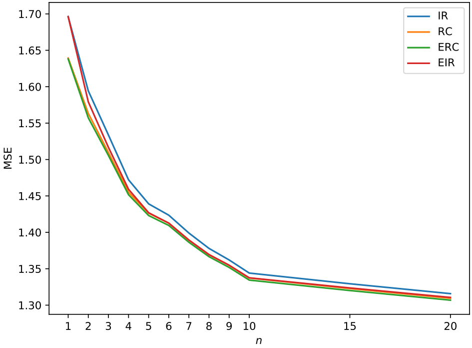
But: gains are minimal when $n \gg m$ (many examples, few labels).

Suggestion 6: The 'Ensemble Effect'

Argument: Modelling labels *appears* to provide better results but actually the ensemble deserves the credit, by providing

- More predictive power
- More regularisation

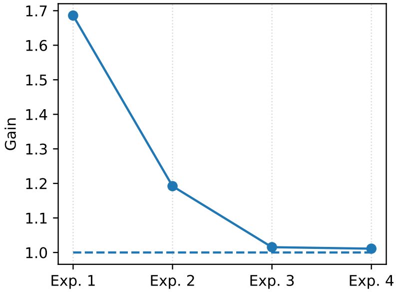
Here, reg. ***only*** (task is fully linear/independent models/concept):



E = Ensemble, I = Independent, C = Chain

So Which Is It Then?

Classifier chains vs independent models (Music-Emotions data):



- Experiment 1: All effects confounded (logistic reg., 0/1 loss)
- Experiment 2: Remove motiv. of label dependence ($\rightarrow$h.loss)
- Experiment 3: Remove influence of capacity (+ depth)
- Experiment 4: Remove influence of regularisation (+ reg.)

interesting result: 20% higher accuracy by modelling label dependence, even when theoretically pointless!

Read, *From Multi-label Learning to Cross-Domain Transfer*. In Press/Accepted 2023

A First Update

We can offer a minor rephrasing:

> *We should model and predict labels together mainly because of label dependence (i.e., if our loss metric suggests that we need to learn it), but we can also get benefits from additional capacity and regularisation brought by additional structure inherent to modelling labels together.*
>
> *With enough data/computational power, regularised deep neural network architectures likely to overpower traditional methods of multi-label learning.*
>
> *But (to say it again): this is not because of label dependence modelling!*

Reasons to Retain Interest

Modelling labels together with model-agnostic/base-classifier approaches (and other algorithm-adaptations):

- still work well especially on fewer training examples (important for, e.g., small data and recovery from concept drift in data streams)
- require no hidden units; depth/non-linearity comes 'for free'
- requires no back propagation
- more choice (decision trees, including a mixture of different models, . . .) – for reasons of interpretability or reliability; and

And we have seen **improvement from modelling totally unrelated tasks together**

Reasons to Retain Interest

Modelling labels together with model-agnostic/base-classifier approaches (and other algorithm-adaptations):

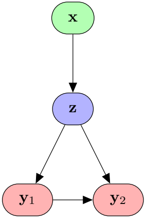
- still work well especially on fewer training examples (important for, e.g., small data and recovery from concept drift in data streams)
- require no hidden units; depth/non-linearity comes 'for free'
- requires no back propagation
- more choice (decision trees, including a mixture of different models, . . .) – for reasons of interpretability or reliability; and

And we have seen **improvement from modelling totally unrelated tasks together** $\leftarrow$ **very interesting!?**

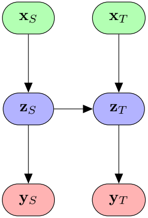
Cross-Domain Transfer Learning: Lessons from MLL

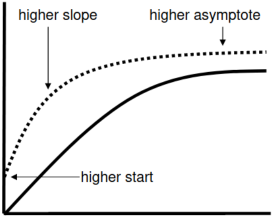
Transfer Learning: A Quick Intro

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



vs

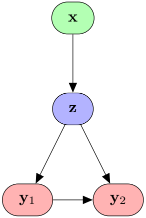




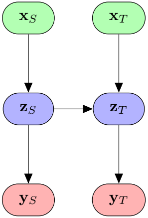
Plot (right) from Torrey and Shavlik, "Transfer learning", 2010.

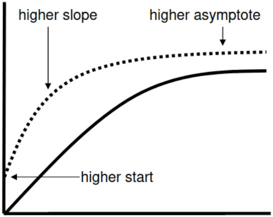
Transfer Learning: A Quick Intro

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



vs





Plot (right) from Torrey and Shavlik, "Transfer learning", 2010.

> A key word was: *related*. But what if relatedness is not a requirement?

In other words: we go back to where our sanity check failed (multi-label learning with no label dependence).

Thoughts on That

Transfer learning by connecting the model from an unrelated source task; This is similar to connecting the first layer of a neural network randomly.

> *"[C]onnecting the first layer randomly is just about the stupidest thing you could do" – Yann LeCun*

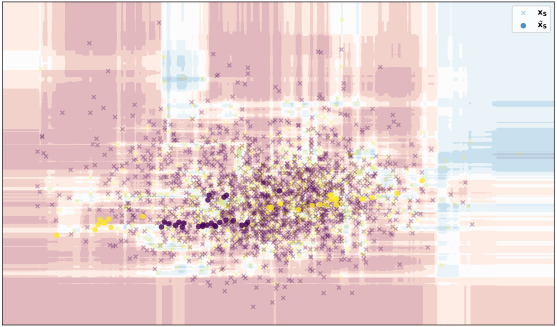
Remarks:

- He said "just about"
- He didn't say it didn't work
- Theres a minor difference: We mean, not randomly drawn from all possible models, rather randomly drawn from all [a collection of] existing *trained* models

So let's try it anyway. . .

Insomniac Fungi

A model (random forest) for classifying patients[6]: suffering insomnia (red) or not (blue), based on sleep measurements $\boldsymbol{x}_S = [x_1, x_2]$:

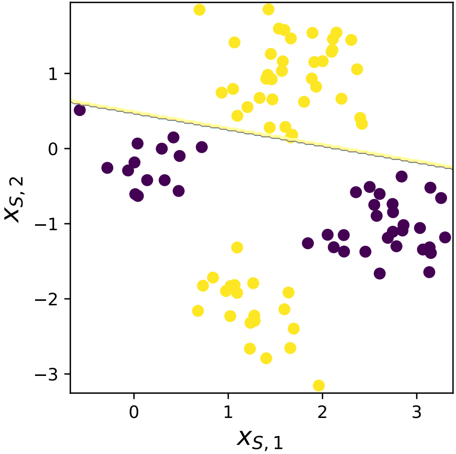


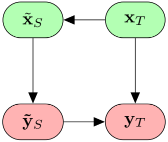
A yeast genome vector is cast into $\tilde{\boldsymbol{x}}_S$, and given an insomnia diagnosis $\widehat{y}_S$; which, when used as new feature, boosts $+2\%$ accuracy when predicting genome function.

[6] Medical data thanks to Olivier Pallanca

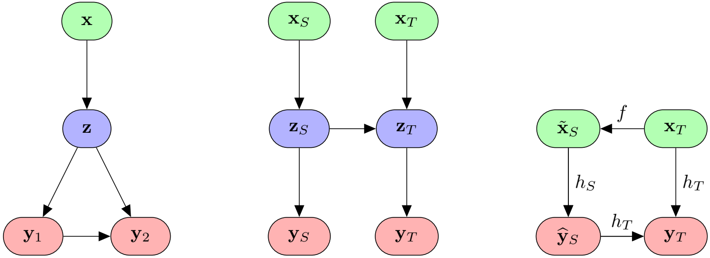
Replicating on Synthetic Data

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



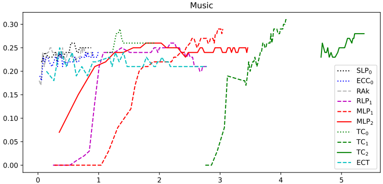


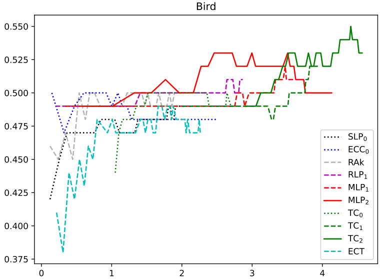
Multi-Label Chain vs Deep Transfer vs Chain Transfer



Main difference from standard Deep Transfer: A model agnostic approach; require only outputs (as per chained multi-label models).

Results (Did it Work?)





Well, it, ‘shows promise’ (green line gets higher).

- Not a state-of-the-art method.
- But it works!
- Reminder: extremely difficult transfer setting: no model introspection, no source data, no label/task dependence
- Advantages: depth without the deep learning; use ‘any’ source model

Discussion Points

- Intersection with *multi-task learning*, *deep transfer learning*, *lifelong learning*, *concept drift adaptation*; ‘pretrained’, ‘frozen’ layers, ‘parameter isolation’, ‘universal computation’, ...
- Interest of multi-label models: larger $\widehat{\boldsymbol{y}}$ (more information)
- What does it mean for a label/task to be related to another?
- Transfer learning vs reduction/reuse/recycling of models?
- Model-driven learning rather than [raw] data-driven learning
- Example implication: When adapting to concept shift in data streams; keep using the ‘irrelevant’ models!

Summary (and Recent Advances and Open Questions)

Multi-label Learning: An Update

Take-away points (so far):

- Many methods, many applications; still relevant!
- But not just '*model label dependence* $\Rightarrow$ good accuracy'
- You may model labels/tasks together even if there is 'none'!
  Label dependence, **model capacity**, **regularization**
- You may ignore label dependence and still perform optimally
- Big data + deep learning $\Rightarrow$ good accuracy
- But extreme multi-label classification can imply extreme cost

And let's not forget about

- Interpretability: *how* methods work; *why* labels together; different contexts of uncertainty
- Ever larger/more complex problems via data-driven learning 'from scratch' – increasingly challenging/wasteful!
  (sparse learning is interesting) Reduce/Reuse/Recycle models!

But many current trends do not need to be considered separately.

Current Work: Missing Value Imputatation

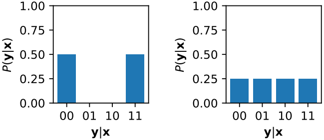
- Missing values are common
- Problem: Often, too many to ignore
  e.g., in high-d data (many attributes, few instances)
- Missing inputs $\approx$ noisy labels
- 'Structured missing-ness' and predicting modes
- Proposed: Auto-replicative Random Forests: $f : \mathbf{X} \mapsto \mathbf{X}$
- Connection to Denoising Auto-Encoder

X_1	X_2	X_3	X_4
0	1	1	0
1	?	0	?
0	1	0	0
1	?	0	1
0	0	?	?

$\Rightarrow$

X_1	X_2	X_3	X_4	Y_1	Y_2	Y_3	Y_4
0	1	1	0	0	1	1	0
1	1	0	0	1	-	0	-
0	1	0	0	0	1	0	0
1	0	0	1	1	-	0	1
0	0	0	0	0	0	-	-

& repeat (impute). We produce multi-mode estimates $\mathbf{x} \sim P(\mathbf{x})$.



Work with Ekaterina Antonenko and Ander Carreño

Current Work: Multi-label ECG Classification and Transfer

- Multi-label classification of ECG signals (multiple heart issues)
- A pre-trained deep neural network works well, but
- Problem: Multiple datasets, partial[ly-overlapping] labels
- Problem: Poor domain transfer

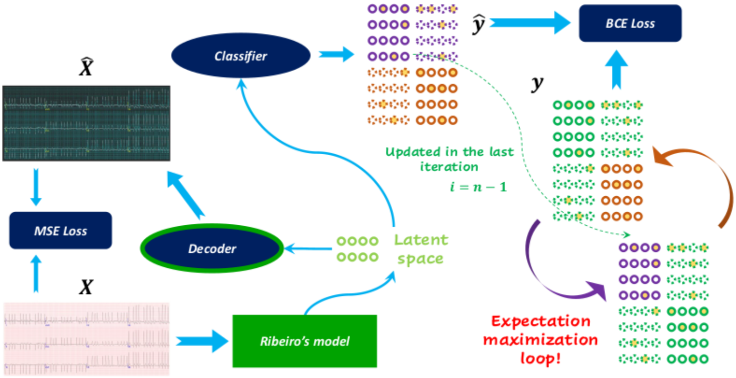


Image credit: Eran Zvuloni ,

Work with Eran Zvuloni, Duy Nhat Vo, Joachim A. Behar, . . .

Multi-label Learning: An Update

Jesse Read

Thank you!

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

References I

This talk is based on (many more references within): *From Multi-label Learning to Cross-Domain Transfer: A Model-Agnostic Approach*, J. Read, 2022.
`https://arxiv.org/pdf/2011.11197.pdf`
Accepted/In Press; *Applied Intelligence*.

- Bogatinovski, Jasmin et al. "Comprehensive comparative study of multi-label classification methods". In: *Expert Systems with Applications* 203 (2022), p. 117215.
- Borchani, Hanen et al. "A Survey on Multi-output Regression". In: *Wiley Int. Rev. Data Min. and Knowl. Disc.* 5.5 (Sept. 2015), pp. 216–233. ISSN: 1942-4787. DOI: 10.1002/widm.1157. URL: http://dx.doi.org/10.1002/widm.1157.

References II

Cisse, Moustapha, Maruan Al-Shedivat, and Samy Bengio. "ADIOS: Architectures Deep In Output Space". In: *Proceedings of The 33rd International Conference on Machine Learning*. Vol. 48. New York, New York, USA: PMLR, 2016, pp. 2770–2779.

Dembczyński, Krzysztof et al. "On Label Dependence and Loss Minimization in Multi-label Classification". In: *Mach. Learn.* 88.1-2 (July 2012), pp. 5–45. ISSN: 0885-6125. DOI: 10.1007/s10994-012-5285-8.

Loza Mencía, Eneldo and Frederik Janssen. "Learning rules for multi-label classification: a stacking and a separate-and-conquer approach". In: *Machine Learning* 105.1 (2016), pp. 77–126. ISSN: 1573-0565. DOI: 10.1007/s10994-016-5552-1. URL: https://doi.org/10.1007/s10994-016-5552-1.

Mylonas, Nikolaos et al. "On the Persistence of Multilabel Learning, Its Recent Trends, and Its Open Issues". In: *IEEE Intelligent Systems* 38.2 (2023), pp. 28–31.

References III

Read, Jesse. *From Multi-label Learning to Cross-Domain Transfer: A Model-Agnostic Approach*. Tech. rep. 2207.11742. ArXiv. ArXiv.org, 2023. URL: http://arxiv.org/abs/2207.11742.

Read, Jesse et al. "Classifier Chains: A Review and Perspectives". In: *Journal of Artificial Intelligence Research (JAIR)* 70 (2021). https://jair.org/index.php/jair/article/view/12376/26658, pp. 683–718. URL: https://jair.org/index.php/jair/article/view/12376.

— ."Classifier Chains for Multi-label Classification". In: *ECML-PKDD 2009: 20th European Conference on Machine Learning*. Bled, Slovenia: Springer, 2009, pp. 254–269. URL: http://link.springer.com/chapter/10.1007%2F978-3-642-04174-7_17.

References IV

- Senge, Robin, Juan José del Coz, and Eyke Hüllermeier. "On the Problem of Error Propagation in Classifier Chains for Multi-label Classification". In: *Data Analysis, Machine Learning and Knowledge Discovery*. Ed. by Myra Spiliopoulou, Lars Schmidt-Thieme, and Ruth Janning. Cham: Springer International Publishing, 2014, pp. 163–170. ISBN: 978-3-319-01595-8.
- Torrey, Lisa and Jude Shavlik. "Transfer learning". In: *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques*. IGI global, 2010, pp. 242–264.
- Waegeman, Willem, Krzysztof Dembczyński, and Eyke Hüllermeier. "Multi-target prediction: a unifying view on problems and methods". In: *Data Mining and Knowledge Discovery* 33.2 (2019), pp. 293–324. ISSN: 1573-756X. DOI: 10.1007/s10618-018-0595-5. URL: https://doi.org/10.1007/s10618-018-0595-5.