Multi-label Learning

Jesse Read

ÉCOLE POLYTECHNIQUE
IP PARIS

July 4, 2023

Hi! PARIS Summer School

Multi-label Learning, Part I (Lecture, 90 min)

1. Introduction and Motivation (10 mins)
2. Formalization: Loss Metrics and Label Dependence (10 mins)
3. Adaptation of Classic ML Methods (5 mins)
4. Model-Agnostic Methods and Graphical Models (20 mins)
5. Deep Multi-label Learning (20 mins)
6. Modern Applications, Trends, and Open Areas (20 mins)
7. Summary and Questions (5 mins)

Introduction and Motivation (10 mins)

Multi-label Classification

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

$\boldsymbol{x} =$ 

$$\boldsymbol{y} = [1, 0, 1, 0] \Leftrightarrow \{\texttt{Beach}, \texttt{Foliage}\}$$

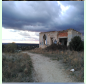
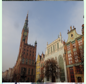
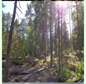
Multi-label Classification

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

$\boldsymbol{x} =$ 

$$\boldsymbol{y} = [1, 0, 1, 0] \Leftrightarrow \{\texttt{Beach}, \texttt{Foliage}\}$$

And Multi-label Learning: Learn the model $h : \boldsymbol{x} \mapsto \boldsymbol{y}$ for any $\boldsymbol{x}$.

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
	?	?	?	?

The task (of the model) is to make predictions:

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

Multi-Label Text (and Media) Classification

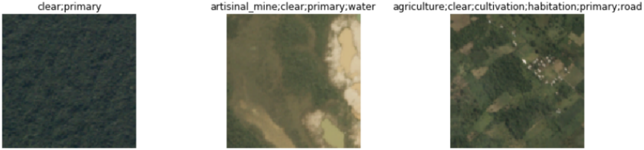
Image Source: [1]

The set of all possible labels (*genres*, in this case) is usually predefined.

Labels as Keywords

Image Source: [2]

Another Image-Classification Example

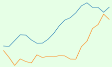
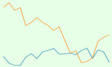
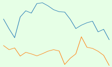
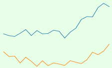
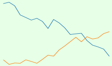
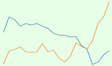


Missing-Value Imputation and Recommender Systems

	0	0	1	1	0
	1	1	?	0	?
	0	0	1	0	0
	1	1	?	0	1
	0	0	0	?	?
	1	0	?	1	?

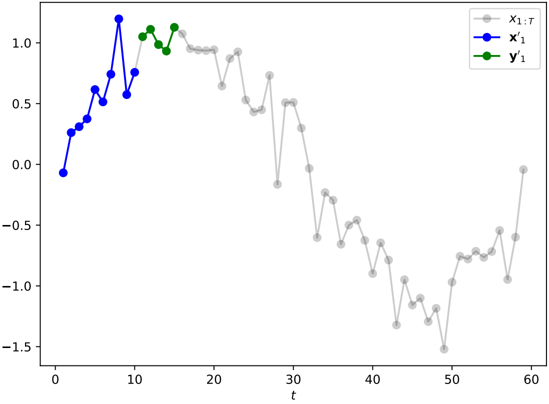
i.e., assign item-labels to users (or user-labels to items).

Time Series Classification

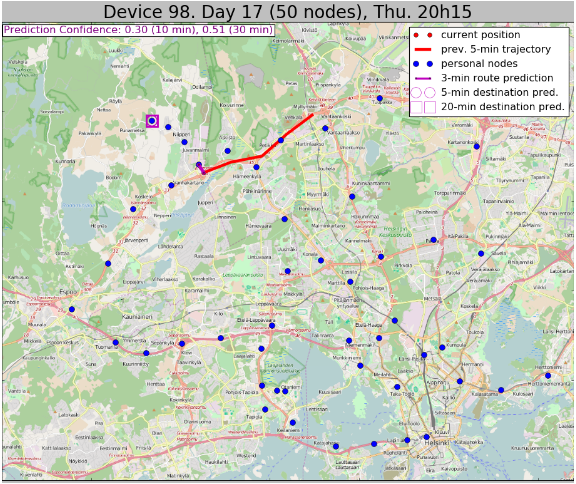
Input	Forecast 1	Forecast 2	Prescribe A	Prescribe B
	-1.01	-0.03	1	0
	0.47	-0.15	0	0
	-0.33	-0.70	1	0
	-1.39	1.57	0	1
	-0.96	1.82	0	1
	?	?	?	?

For example, ECG, EEG, signals. How will a patient's state evolve? Which diagnoses? Which treatments?

Time Series Forecasting

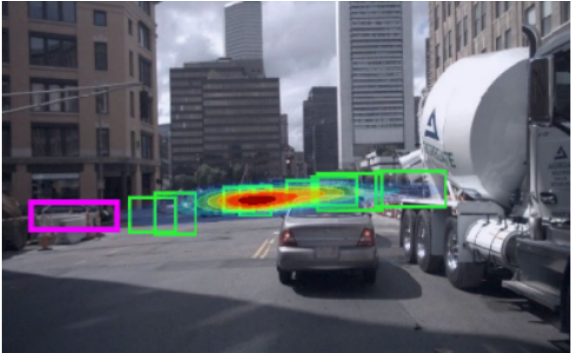


Trajectory Prediction



Trajectory prediction in urban environment using mobile phone data

Structured Output Prediction



Object prediction [3]

Drug Design

	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
	?	?	?	?	?	?

Molecule design prediction (binding affinities (Y) of molecules (X) to new proteins): [4]

Formalization: Loss Metrics and Label Dependence (10 mins)

A Standard Machine Learning Setup

We are given data set $\boldsymbol{X}, \boldsymbol{Y}$. We want to build model h in order to obtain predictions

$$\widehat{\boldsymbol{y}} = h(\boldsymbol{x})$$

That minimize expected loss where the loss metric

$$L(\boldsymbol{y}, \widehat{\boldsymbol{y}})$$

i.e., our model h should produce

$$\min_{\widehat{\boldsymbol{y}} \in \{0,1\}^m} \mathbb{E}_{\boldsymbol{y} \sim p(\boldsymbol{y}|\boldsymbol{x})}[L(\boldsymbol{y}, \widehat{\boldsymbol{y}})]$$

We might also be interested in estimating distribution $p(\boldsymbol{y} \mid \boldsymbol{x})$.

Multi-label Specificities

$$\boxed{\min_{\widehat{\boldsymbol{y}} \in \{0,1\}^m} \mathbb{E}_{\boldsymbol{y} \sim p(\boldsymbol{y}|\boldsymbol{x})}[L(\boldsymbol{y}, \widehat{\boldsymbol{y}})]}$$

- Exponential complexity, wrt m labels!
- Label dependence (joint distribution)

Important Background: Label Dependence

Often one considers marginal dependence:

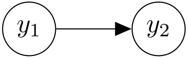
$$P(y_1, y_2) \neq P(y_1)P(y_2)$$

Actually, we should be interested in conditional dependence:

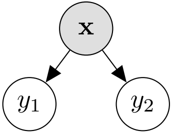
$$P(y_1, y_2 \mid \boldsymbol{x}) \neq P(y_1 \mid \boldsymbol{x})P(y_2 \mid \boldsymbol{x})$$

which is more difficult to measure (requires building models). It's not the same, e.g.,

e.g.,



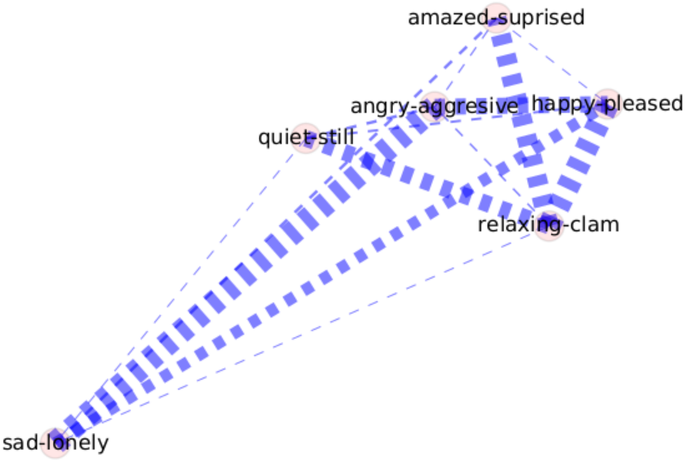
and



may be equivalent!

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

Example Representation of Label Dependence



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

Loss Metrics (L): How Bad is a Prediction $\hat{\boldsymbol{y}}$

Example (Music/Emotions Dataset): We predict `sad-lonely` and `angry-aggressive`, but true label set is *only* `sad-lonely`. How bad is this prediction? In other words: what is the loss?

- Hamming loss (decomposable; **average**):

$$L_H([1,0,0,0,0,0],[1,0,1,0,0,0]) = 1/6$$

  (not too bad)

- 0/1 loss (non-decomposable; **exact match**):

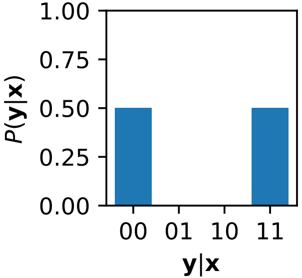
$$L_{0/1}([1,0,0,0,0,0],[1,0,1,0,0,0]) = 1$$

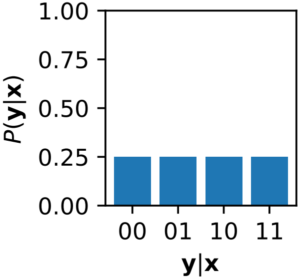
  (worst case)

The minimizer is not (necessarily) the same! If 0/1 loss, then we need to consider the joint (predictive posterior) distribution $p(\boldsymbol{y} \mid \boldsymbol{x})$.

Examples of $p(\boldsymbol{y} \mid \boldsymbol{x})$ (Predictive Posterior)

where $\boldsymbol{y} \in \{0,1\}^2$ $(m = 2)$, given input instance $\mathbf{x}$:





$P(y_1, y_2 \mid \boldsymbol{x})$	$y_1 = 0$	$y_1 = 1$
$y_2 = 0$	0.00	0.50
$y_2 = 1$	0.50	0.00

$P(y_1, y_2 \mid \boldsymbol{x})$	$y_1 = 0$	$y_1 = 1$
$y_2 = 0$	0.25	0.25
$y_2 = 1$	0.25	0.25

The marginal probabilities $p(y_j \mid \boldsymbol{x})$ are the same.

A Closer Look: Hamming Loss

Hamming loss is the averaged sum of errors,

$$L_{HL} = \frac{1}{m}\sum_{j=1}^{m} L(y_j, \widehat{y}_j)$$

where, for a given label, e.g., y_2,

$$L(y_2, \widehat{y}_2) = \begin{cases} 1 & y_2 \neq \widehat{y}_2, \\ 0 & y_2 = \widehat{y}_2 \end{cases}$$

i.e., it is decomposable across labels;

$$P(y_2|\boldsymbol{x}) = \sum_{y_1 \in \{0,1\}} P(y_1|\boldsymbol{x})P(y_2|\boldsymbol{x}, y_1)$$

To minimize this loss[1]: $P(\boldsymbol{y} \mid \boldsymbol{x})$ is not required! $P(y_j \mid \boldsymbol{x})$ is sufficient;

$$\widehat{y}_j = h_j(\boldsymbol{x}) = \operatorname*{argmax}_{y \in \{0,1\}} p(y_j \mid \boldsymbol{x})$$

[1] And others based upon in, like ranking loss

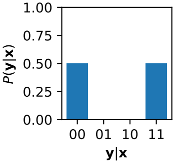
A Closer Look: 0/1 Loss

Subset 0/1 loss, is an exact match,

$$L_{0/1}(\boldsymbol{y}, \widehat{\boldsymbol{y}}) = \begin{cases} 1 & \boldsymbol{y} \neq \widehat{\boldsymbol{y}}, \\ 0 & \boldsymbol{y} = \widehat{\boldsymbol{y}} \quad \text{(exactly, i.e., } L_{HL}(\boldsymbol{y}, \widehat{\boldsymbol{y}}) = 0) \end{cases}$$

We need to model label dependence! We need to know $p(\boldsymbol{y} \mid \boldsymbol{x})$.
To minimize this loss:

$$\widehat{\boldsymbol{y}} = h(\boldsymbol{x}) = \underset{\boldsymbol{y} \in \{0,1\}^m}{\operatorname{argmax}}\, p(\boldsymbol{y} \mid \boldsymbol{x})$$



$P(y_1, y_2 \mid \boldsymbol{x})$	$y_1 = 0$	$y_1 = 1$
$y_2 = 0$	0.00	0.50
$y_2 = 1$	0.50	0.00

$P(y_2 = 1 \mid \boldsymbol{x}) = 0.5$, but $P(y_2 = 1, y_1 = 0 \mid \boldsymbol{x}) = 0$! Best case (without joint model): $\mathbb{E}[L_{0/1}] = 0.75$ loss. Best case (with joint model): $\mathbb{E}[L_{0/1}] = 0.5$ loss.

Adaptation of Classic ML Methods (5 mins)

A Typical Offering

For example, **algorithm adapted** methods in SCIKITLEARN:

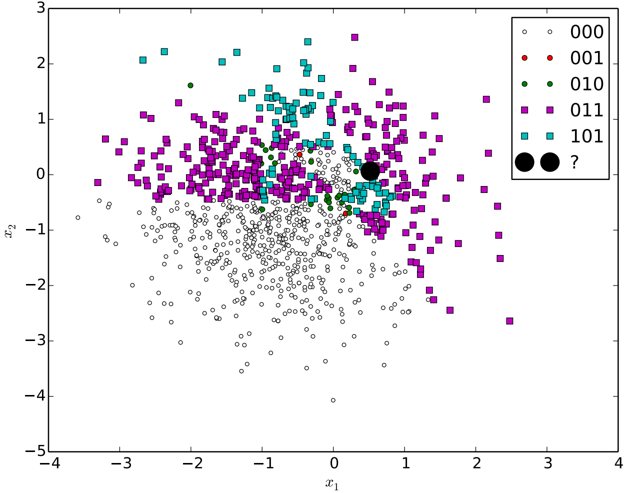
- `sklearn.tree.DecisionTreeClassifier`
- `sklearn.tree.ExtraTreeClassifier`
- `sklearn.ensemble.ExtraTreesClassifier`
- `sklearn.neighbors.KNeighborsClassifier`
- `sklearn.neural_network.MLPClassifier`
- `sklearn.ensemble.RandomForestClassifier`
- `sklearn.linear_model.RidgeClassifierCV`
- `sklearn.multiclass.OneVsRestClassifier`
- `sklearn.multioutput.ClassifierChain`

i.e., Decision Trees, Nearest-Neighbours, Neural Networks.

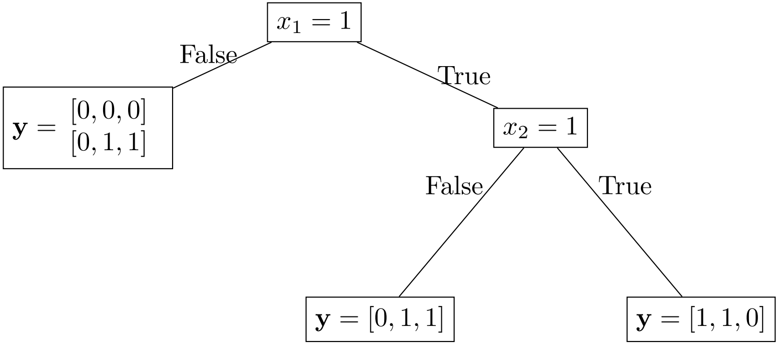
. . . and some **task adaptation** / problem transformation / model agnostic methods – we come back to these soon!

Refs. in Bogatinovski et al., "Comprehensive comparative study of multi-label classification methods", 2022

k-Nearest Neighbours

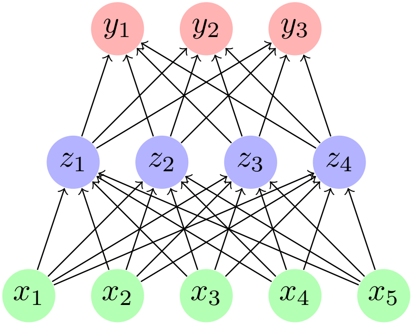


Decision Tree Methods



Multi-labelled examples at the leaves; summation over (labels) wrt impurity criteria when inducing the tree.

Neural Networks (Multi-Layer Perceptrons)



Limitations of Algorithm-Adaptations

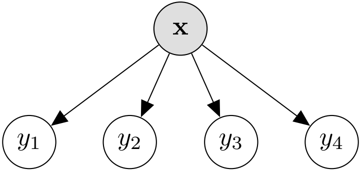
Much of the multi-label literature (and industry application) is dominated by these methods. However,

- you get stuck with a particular class of model (inflexibile)
- in many cases, a reliable probabilistic interpretation is missing
- a bit 'old fashioned'; not well adapted to image or text input

Model-Agnostic Methods and Graphical Models (20 mins)

Transformation to Independent Binary Classification

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



The binary relevance method (BR transformation) $=$ *one binary classifier trained for each label*, i.e., independent models.

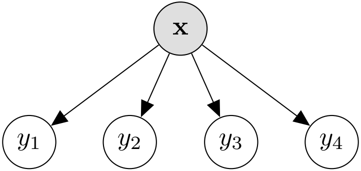
Transformation to Independent Binary Classification

$\boldsymbol{X}$	Y_1
$\boldsymbol{x}^{(1)}$	0
$\boldsymbol{x}^{(2)}$	1
$\boldsymbol{x}^{(3)}$	0
$\boldsymbol{x}^{(4)}$	1
$\boldsymbol{x}^{(5)}$	0
$\tilde{\boldsymbol{x}}$	?

$\boldsymbol{X}$	Y_2
$\boldsymbol{x}^{(1)}$	1
$\boldsymbol{x}^{(2)}$	0
$\boldsymbol{x}^{(3)}$	1
$\boldsymbol{x}^{(4)}$	0
$\boldsymbol{x}^{(5)}$	0
$\tilde{\boldsymbol{x}}$	?

$\boldsymbol{X}$	Y_3
$\boldsymbol{x}^{(1)}$	0
$\boldsymbol{x}^{(2)}$	1
$\boldsymbol{x}^{(3)}$	0
$\boldsymbol{x}^{(4)}$	1
$\boldsymbol{x}^{(5)}$	0
$\tilde{\boldsymbol{x}}$	?

$\boldsymbol{X}$	Y_4
$\boldsymbol{x}^{(1)}$	1
$\boldsymbol{x}^{(2)}$	0
$\boldsymbol{x}^{(3)}$	1
$\boldsymbol{x}^{(4)}$	0
$\boldsymbol{x}^{(5)}$	0
$\tilde{\boldsymbol{x}}$	?

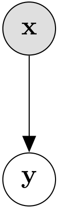


The binary relevance method (BR transformation) = *one binary classifier trained for each label*, i.e., independent models.

Transformation to Multi-Class (Meta-Labels)

e.g., `beach+sunset` considered a single label.

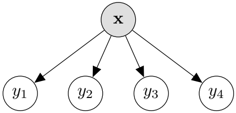
$\boldsymbol{X}$	Y
$\boldsymbol{x}^{(1)}$	0 1 1 0
$\boldsymbol{x}^{(2)}$	1 0 0 0
$\boldsymbol{x}^{(3)}$	0 1 0 0
$\boldsymbol{x}^{(4)}$	1 0 0 1
$\boldsymbol{x}^{(5)}$	0 0 0 1
$\tilde{\boldsymbol{x}}$	?



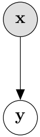
The label powerset method (or meta-label classifier) = *a single target multi-class classifier*. Labels are modeled together, but ($\boldsymbol{y} \in \{0,1\}^m$) ...

- Overfitting
- Complexity.

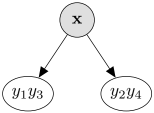
Probabilistic Graphical Models



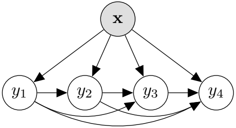
Binary Relevance



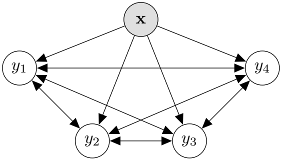
Label Powerset



Random k-Label Sets and Meta Labels



Prob. Classifier Chains and Bayesian Networks



Conditional Dependency Networks

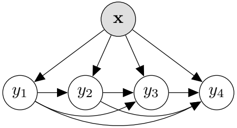
Arrows represent $P(\text{child} \mid \text{parents})$ and more generally (bending the rules a bit) a prediction $\widehat{\text{output}} = h(\text{input})$ where h is any base classifier.

References herein: Read et al., "Classifier Chains: A Review and Perspectives", 2021

Classifier Chains: An Example of ‘Problem Transformation’

A chain (structure, graph) over the output variables;

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



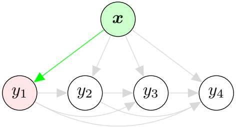
$\boldsymbol{X}$	Y_1	Y_2	Y_3	Y_4
$\boldsymbol{x}^{(1)}$	0	1	1	1
$\boldsymbol{x}^{(2)}$	1	0	0	0
$\boldsymbol{x}^{(3)}$	0	1	0	1
$\boldsymbol{x}^{(4)}$	1	0	0	0
$\boldsymbol{x}^{(5)}$	0	0	0	0
$\tilde{\boldsymbol{x}}$	$\widehat{y_1}$	$\widehat{y_2}$	$\widehat{y_3}$	$\widehat{y_4}$

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, graph) over the output variables;

- Cascaded prediction across the 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

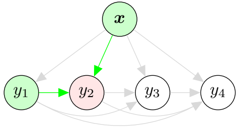
$\tilde{\boldsymbol{x}}$	$\widehat{y_1}$			

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, graph) over the output variables;

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



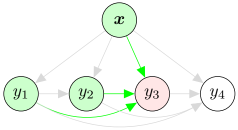
$\boldsymbol{X}$	Y_1	Y_2	Y_3	Y_4
$\boldsymbol{x}^{(1)}$	0	1	1	1
$\boldsymbol{x}^{(2)}$	1	0	0	0
$\boldsymbol{x}^{(3)}$	0	1	0	1
$\boldsymbol{x}^{(4)}$	1	0	0	0
$\boldsymbol{x}^{(5)}$	0	0	0	0
$\tilde{\boldsymbol{x}}$	$\widehat{y_1}$	$\widehat{y_2}$		

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, graph) over the output variables;

- Cascaded prediction across the 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, ...).

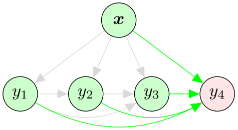
This is a greedy *approximation* of $\operatorname{argmax} p(\boldsymbol{y} \mid \boldsymbol{x})$.

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, graph) over the output variables;

- Cascaded prediction across the 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
$\tilde{\boldsymbol{x}}$	$\widehat{y}_1$	$\widehat{y}_2$	$\widehat{y}_3$	$\widehat{y}_4$

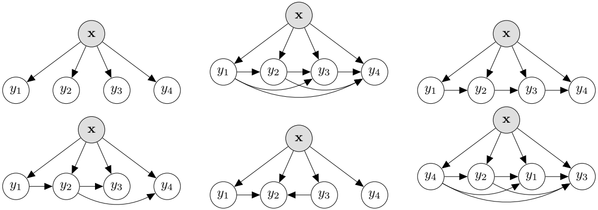
For example, $\widehat{y}_3 = h_3(\boldsymbol{x}, \widehat{y}_1, \widehat{y}_2)$ with base classifier (or regressor) h_3 (e.g., decision tree, logistic regression, ...).

This is a greedy *approximation* of argmax $p(\boldsymbol{y} \mid \boldsymbol{x})$.

FAQ. "Why this order in particular, could another one work better?"

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

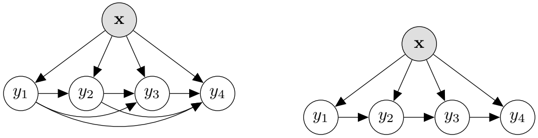
Structure Search: Some Options



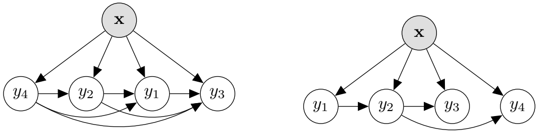
1. Random structure (often in an ensemble).
2. Use an existing hierarchy (expert knowledge)
3. Impose a full/complete structure
4. Search for a structure, based on (heuristic)
   - marginal label dependence;
   - conditional label dependence,
   - accuracy of individual models
   - accuracy of overall structure

Structure Search is Difficult

These models perform well:



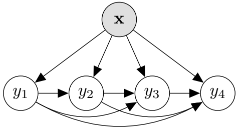
These ones perform not so well:



- Difficult to associate accuracy to a particular structure
- Considerations of measurements of dependence, time order, or 'inherent' hierarchy, are *at best* a rough guide
- A super-exponential number of possible structures
- Can never know (without uncertainty) which is the 'ground truth'

Probabilistic Inference: Also difficult

Even with a single chosen structure,



Recall (to minimize 0/1-loss), we want:

$$\widehat{\boldsymbol{y}} = \underset{\boldsymbol{y}\in\{0,1\}^m}{\text{argmax}}\ P(\boldsymbol{y} \mid \boldsymbol{x}) = h(\boldsymbol{x})$$

$$= \underset{\boldsymbol{y}\in\{0,1\}^m}{\text{argmax}}\ P(y_1 \mid \boldsymbol{x}) \prod_{j=1}^{m} P(y_2 \mid \boldsymbol{x}, y_1, \ldots, y_{j-1}) \quad \triangleright \text{ from the graph}$$

e.g., (when 4 labels)

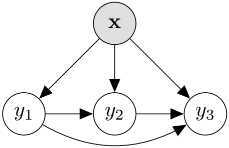
$$\boldsymbol{y} \in \{[0,0,0,0],[0,0,0,1],\ldots,[1,1,1,1]\}$$

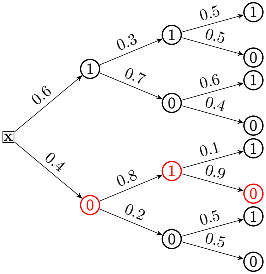
and, in general, $\boldsymbol{y} \in \{0,1\}^m$ for m labels; exponential complexity!

Probabilistic Classifier Chains: Inference as Tree-Search

$$\widehat{y}_j = h_j(\boldsymbol{x}) = \underset{y_j \in \{0,1\}}{\text{argmax}}\ P(y_j|\boldsymbol{x}, y_1, \ldots, y_{j-1})$$

e.g., logistic regression, then:





$$\widehat{\boldsymbol{y}} = \underset{\boldsymbol{y} \in \{0,1\}^m}{\text{argmax}}\ P(y_1|\boldsymbol{x}) \prod_{j=2}^{m} P(y_j|\boldsymbol{x}, y_1, \ldots, y_{j-1})$$

This is not the same as $\widehat{y}_1, \widehat{y}_2, \widehat{y}_3$ obtained greedly. We now have $p(\boldsymbol{y} \mid \boldsymbol{x})$. Expensive, but many approximations via tree search.

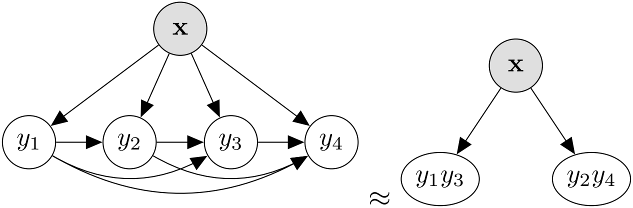
.

Dembczyński, Cheng, and Hüllermeier, "Bayes optimal multilabel classification via probabilistic classifier chains", 2010; Mena et al., "An Overview of Inference Methods in Probabilistic Classifier Chains for Multilabel Classification", 2016

Meta Labels (e.g., RakEL) vs Probabilistic Classifier Chains

e.g., (recall) `beach+sunset` considered a meta label (transformation to multi-class as a special case).

$$\underset{\boldsymbol{y}\in\{0,1\}^L}{\operatorname{argmax}}\, P(y_1|\boldsymbol{x}) \prod_{j=2}^{L} P(y_j|\boldsymbol{x}, y_1, \ldots, y_{j-1}) \approx \underset{\boldsymbol{y}\in\mathcal{S}_L\times\mathcal{S}_R}{\operatorname{argmax}}\, P(\boldsymbol{y}|\boldsymbol{x})$$



(more efficient search vs smaller space(s) to search through)

Summary of Problem-Transformation Methods

We have a

- Principled way to minimize 0/1 loss (exact match);
- A flexible and interpretble (and probabilistic) structure; and
- Can use our favourite off-the-shelf classifiers (model agnostic)

But:

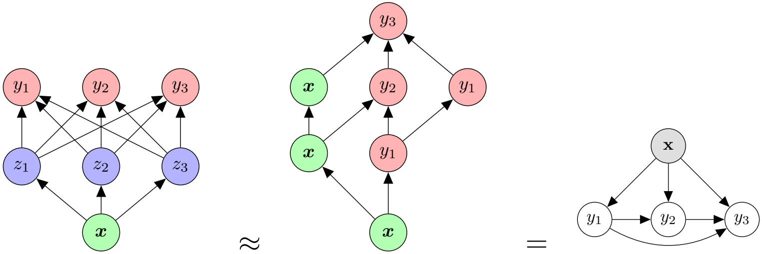
- (to make a long story short) sometimes the gain results from 'black magic' rather than owing to the principled approach
- Methodology tends to be scale poorly
- Still a bit old fashioned, perhaps?

What next? Deep learning provides black magic, scalability and is fashionable!

Deep Multi-label Learning (20 mins)

Graphical Models = Deep Neural Networks

We already have this (from graphical models): Structure among labels ⇒ 'deep'; base classifiers as transfer functions ⇒ 'neural'.



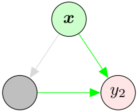
('$\approx$' in terms of capacity; '$=$' in terms of greedy inference)

But previously, we didn't have deep *learning*:

- No back propagation
- The hidden nodes are not 'hidden'.

Consider prediction task

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



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

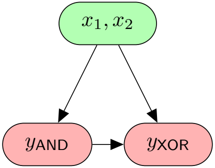
	X_1		Y_2
Basis expansion	x	ϕ	y_2
Stacking	x	$\tilde{y}_2$	y_2
Classifier chain	x	y_1	y_2
Neural network	x		y_2

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

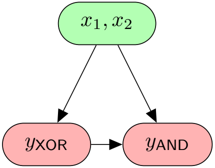
We're talking about capacity more than dependency here!

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



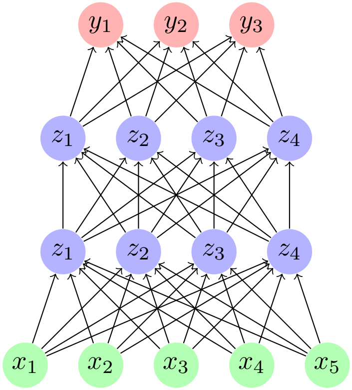
(sometimes†)

†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!

Deep Multi-Label Learning



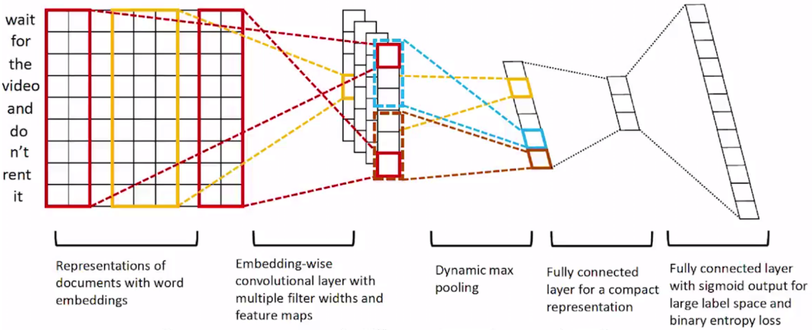
'Off-the-shelf' deep multi-label learning.

Basic idea: Powerful embeddings/capacity; go nuts with your favourite deep-learning framework (easy to add CNN, etc. layers).

Nam et al., "Large-Scale Multi-label Text Classification - Revisiting Neural Networks", 2014; Read and Perez-Cruz, *Deep Learning for Multi-label Classification*, 2013; Wang et al., "CNN-RNN: A unified framework for multi-label image classification", 2016

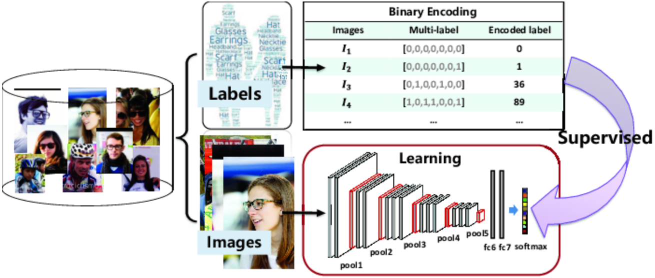
Extreme Multi-label Classification (XMC)

An example[2]

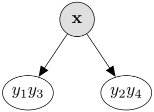


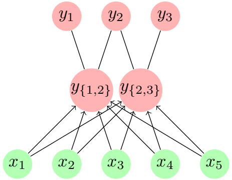
[2] e.g., Jasinska-Kobus et al., "Probabilistic Label Trees for Extreme Multi-label Classification", 2020

Deep Multi-Label via Multi-Class Transformation



Basic idea: Transform multi-labels to single labels[3];
i.e., 'deep' version of meta labels[4]:

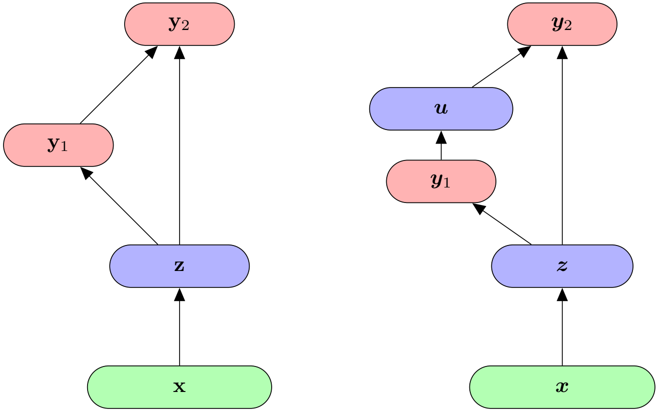




[3] Chenghua Li et al. "DeepBE: Learning deep binary encoding for multi-label classification". In: *Proceedings of the IEEE conference on computer vision and pattern recognition workshops.* 2016, pp. 39–46

[4] Tsoumakas, Katakis, and Vlahavas, "Random k-Labelsets for Multi-Label Classification", 2011; Read, Puurula, and Bifet, "Multi-label Classification with Meta Labels", 2014

Deep in the Label Space

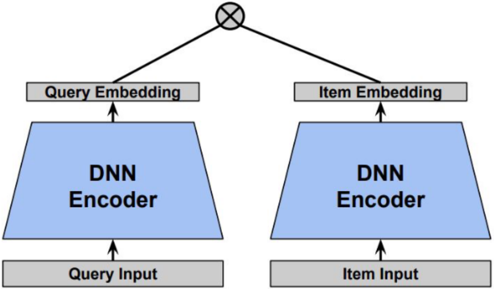


Basic idea: Embeddings for the output space as well.

We can also import the 'probabilistic chains' into this context.

Cisse, Al-Shedivat, and Bengio, "ADIOS: Architectures Deep In Output Space", 2016; Read and Hollmén, *Multi-label Classification using Labels as Hidden Nodes*, 2017

Two-Tower Networks

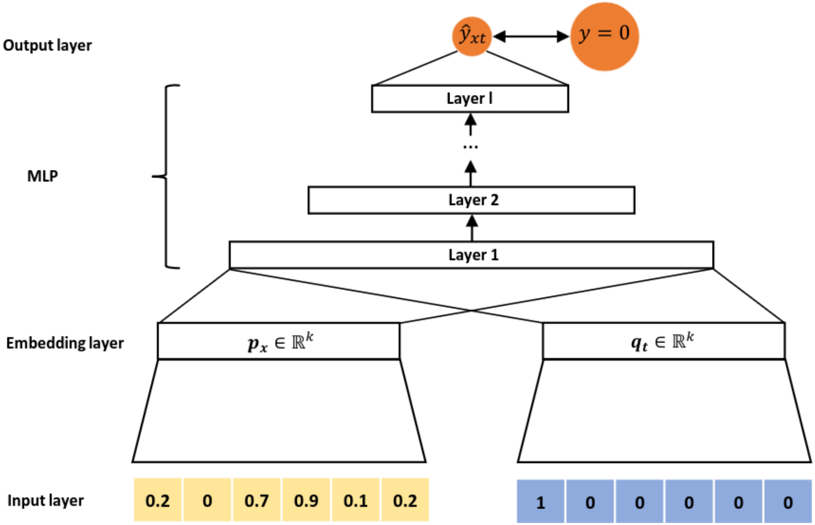


The two tower networks[5] have been generalized to multi-label learning[6].

Basic idea: Embed the instance ($\boldsymbol{x}$; left); embed the item (j; right); provide score $y_j(\boldsymbol{x}) \in \{0, 1\}$ (at the top).

[5] e.g., Yang et al., "Mixed negative sampling for learning two-tower neural networks in recommendations", 2020; He et al., "Neural collaborative filtering", 2017

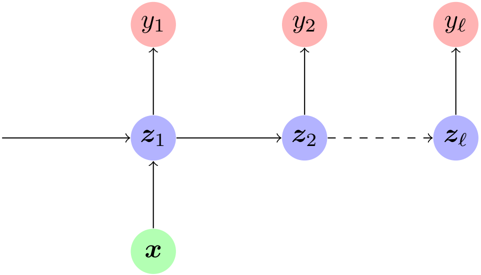
[6] Iliadis, De Baets, and Waegeman, "Multi-target prediction for dummies using two-branch neural networks", 2022 (in the general sense of multi-target prediction)

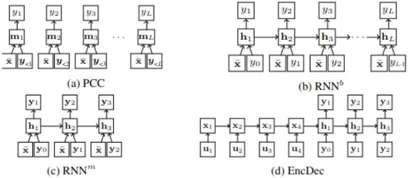


Related to the 'independent models' transformation, but more efficient, and flexible.

Iliadis, De Baets, and Waegeman, "Multi-target prediction for dummies using two-branch neural networks", 2022

Recurrent Neural Networks





Main idea: only predict positve labels $y_1, y_2, \ldots, y_\ell \subset \{1, 2, \ldots, m\}$; more efficient use of architecture.

Nam et al., "Maximizing Subset Accuracy with Recurrent Neural Networks in Multi-label Classification", 2017

Modern Applications, Trends, and Open Areas (20 mins)

Missing Value Imputatation

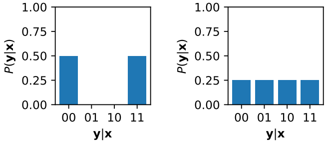
- Missing values – what to do?
- Connection to recommender systems, multi-label learning
- Missing inputs ≈ noisy labels
- Where to start: Two-Tower Networks, Denoising Auto-Encoders, Expectation Maximization

X_1	X_2	X_3	X_4	Y
0	1	1	0	0
1	?	0	?	2
0	1	0	0	2
1	?	0	1	1
0	0	?	?	2

⇒

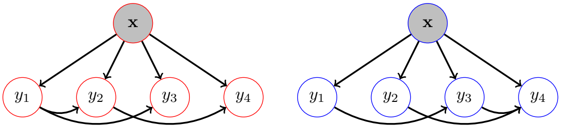
X_1	X_2	X_3	X_4	Y
0	1	1	0	0
1	1	0	0	2
0	1	0	0	2
1	0	0	1	1
0	0	0	0	2

Should keep information about uncertainty.

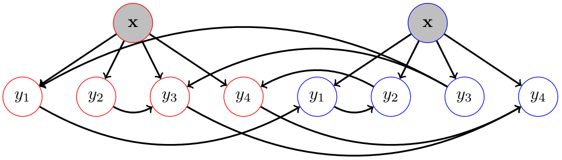


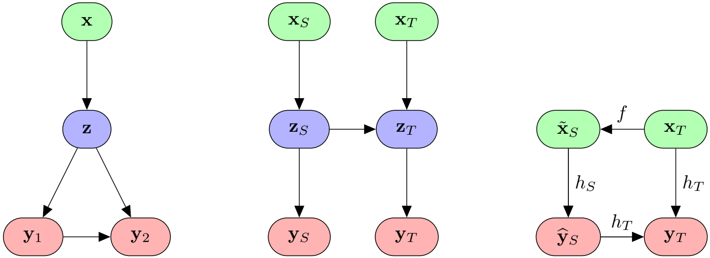
Multi-Task and Transfer Learning

Common opinion: label dependence is fundamental. So, if we take two totally unrelated datasets; and stick them together; search for inherent structure, we should find something like this,



In reality: we can find something like this,

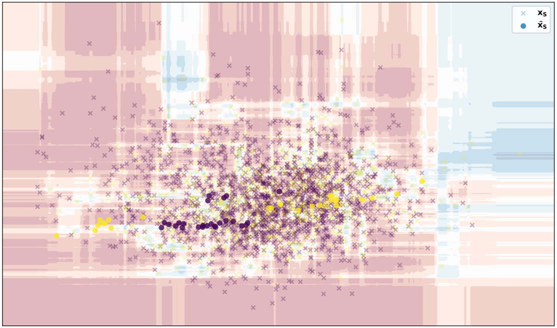




Multi-label (Chain) vs Deep Transfer vs Chain Transfer.

A case study (toy example):

Souce dataset: function of yeast genomes; Target dataset: insomia diagnosis among human patients.



The predictions of genome functionality are useful features for insomnia prediction (+2% accuracy).

A hint towards Foundation Models without back-propagation; reduce, re-use (modularize), recycle models.

e.g., Read, *From Multi-label Learning to Cross-Domain Transfer: A Model-Agnostic Approach*, 2023

Partial and Weak Labels

We get partial labels from noisy annotators[7]:

And weak labels[8] from lazy annotators (unknown missing labels):

training image	GroundTruth	Tagged Labels
	people clothing cloud sky water sea nature	people clothing sky

Other scenarios: ambiguous/imprecise labels (multiple annotators).

[7] e.g., Xie and Huang, "Partial multi-label learning", 2018

[8] e.g., Sun, Zhang, and Zhou, "Multi-label learning with weak label", 2010

Multiple Problems in MLL: A Case Study

- Multi-labelled ECG signals (heart multi-diagnostic)
- A pre-trained deep neural network works well, but
- poor domain transfer (multiple collections); and
- different label sets; missing labels when combined.

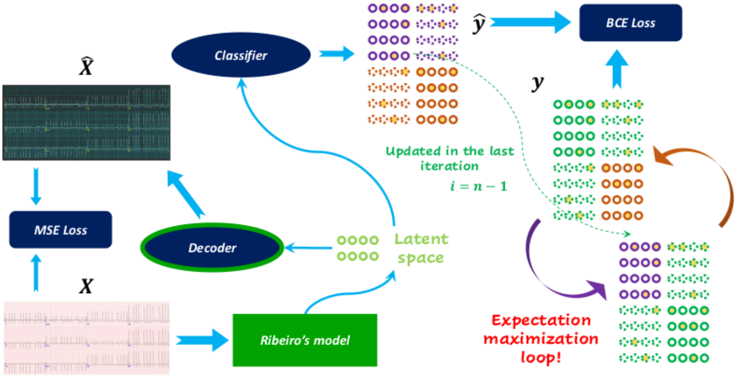


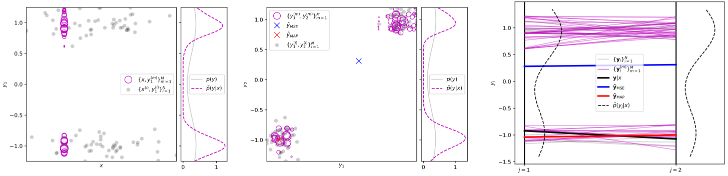
Image credit: Eran Zvuloni

Multi-Target Regression

So far our focus was multi-label classification. But modelling continuous targets is essential for many tasks: e.g., forecasting, structured output.

Key points:

- Several methods (e.g., greedy chains, decision trees, neural networks) can be applied off the shelf as in multi-label classification.
- Expect relatively less improvement from modelling labels together (Why? Think: **loss metric**; **non-linearities**)
- Difficulty to model $p(\boldsymbol{y} \mid \boldsymbol{x})$: tree search not possible (unless **discretization**; Monte Carlo tree search).



Other Issues and Open Questions

- Data streams and concept drift (in the label space)
- Dynamic structures
- Interepretation and explainability: which graph/structure makes sense?

Summary and Questions (5 mins)

Questions? Comments?

References I

Bogatinovski, Jasmin et al. "Comprehensive comparative study of multi-label classification methods". In: *Expert Systems with Applications* 203 (2022), p. 117215.

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, Weiwei Cheng, and Eyke Hüllermeier. "Bayes optimal multilabel classification via probabilistic classifier chains". In: *ICML '10: 27th International Conference on Machine Learning*. Haifa, Israel: Omnipress, 2010, pp. 279–286.

References II

- 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.
- He, Xiangnan et al. "Neural collaborative filtering". In: *Proceedings of the 26th international conference on world wide web*. 2017, pp. 173–182.
- Iliadis, Dimitrios, Bernard De Baets, and Willem Waegeman. "Multi-target prediction for dummies using two-branch neural networks". In: *Machine Learning* (2022), pp. 1–34.
- Jasinska-Kobus, Kalina et al. "Probabilistic Label Trees for Extreme Multi-label Classification". In: (2020). URL: https://arxiv.org/pdf/2009.11218v1.pdf.
- Li, Chenghua et al. "DeepBE: Learning deep binary encoding for multi-label classification". In: *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. 2016, pp. 39–46.

References III

- Mena, Deiner et al. "An Overview of Inference Methods in Probabilistic Classifier Chains for Multilabel Classification". In: *Wiley Int. Rev. Data Min. and Knowl. Disc.* 6.6 (Nov. 2016). https://digibuo.uniovi.es/dspace/bitstream/handle/10651/39325/surveypcc-gracc.pdf, pp. 215–230. ISSN: 1942-4787. DOI: 10.1002/widm.1185. URL: https://doi.org/10.1002/widm.1185.
- Nam, Jinseok et al. "Large-Scale Multi-label Text Classification - Revisiting Neural Networks". In: *ECML-PKDD '14: 25th European Conference on Machine Learning and Knowledge Discovery in Databases*. 2014, pp. 437–452.
- Nam, Jinseok et al. "Maximizing Subset Accuracy with Recurrent Neural Networks in Multi-label Classification". In: *Advances in Neural Information Processing Systems 30*. 2017, pp. 5413–5423. URL: http://papers.nips.cc/paper/7125-maximizing-subset-accuracy-with-recurrent-neural-networks-in-multi-label-classification.pdf.

References IV

- 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 and Jaakko Hollmén. *Multi-label Classification using Labels as Hidden Nodes*. Tech. rep. 1503.09022v3. ArXiv. ArXiv.org, 2017. URL: http://arxiv.org/abs/1503.09022v3.
- Read, Jesse and Fernando Perez-Cruz. *Deep Learning for Multi-label Classification*. Tech. rep. ArXiv. ArXiv, 2013. URL: http://arxiv.org/abs/1502.05988.
- Read, Jesse, Antti Puurula, and Albert Bifet. "Multi-label Classification with Meta Labels". eng. In: *ICDM'14: IEEE International Conference on Data Mining*. Shenzen, China: IEEE, 2014, pp. 941–946.

References V

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.

Sun, Yu-Yin, Yin Zhang, and Zhi-Hua Zhou. "Multi-label learning with weak label". In: *Proceedings of the twenty-fourth AAAI conference on artificial intelligence*. 2010, pp. 593–598.

References VI

- Tsoumakas, Grigorios, Ioannis Katakis, and Ioannis Vlahavas. "Random k-Labelsets for Multi-Label Classification". In: *IEEE Transactions on Knowledge and Data Engineering* 23.7 (2011). Ed. by IEEE, pp. 1079–1089.
- Wang, Jiang et al. "CNN-RNN: A unified framework for multi-label image classification". In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016, pp. 2285–2294.
- Xie, Ming-Kun and Sheng-Jun Huang. "Partial multi-label learning". In: *Proceedings of the AAAI conference on artificial intelligence*. Vol. 32. 1. 2018.
- Yang, Ji et al. "Mixed negative sampling for learning two-tower neural networks in recommendations". In: *Companion Proceedings of the Web Conference 2020*. 2020, pp. 441–447.