Apprentissage Multi-cibles

Applications, Défis, et Modèles

Jesse Read

December 3, 2020

Crédite Agricole, ~~Mont Rouge~~ via CiscoWebex

Outline

Multi-Label and Multi-Target Learning

Classification Multi-label

X_1	X_2	X_3	X_4	X_5	Y_1	Y_2	Y_3	Y_4
1	0.1	3	1	0	0	1	1	0
0	0.9	1	0	1	1	0	0	0
0	0.0	1	1	0	0	1	0	0
1	0.8	2	0	1	1	0	0	1
1	0.0	2	0	1	0	0	0	1
0	0.0	3	1	1	?	?	?	?

Pour une entrée $\boldsymbol{x}$ on obtien la prediction d'un vecteur

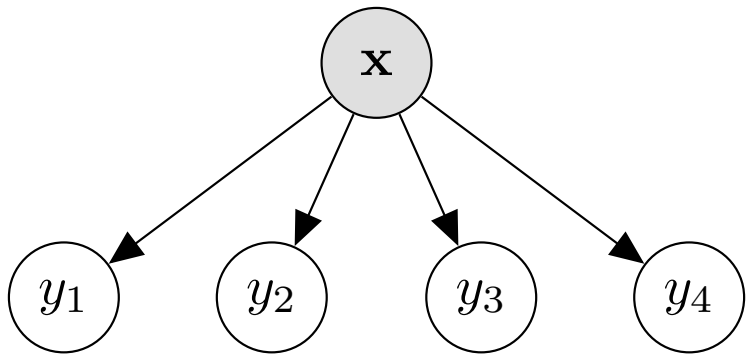
$$\widehat{\boldsymbol{y}} = \mathbf{h}(\boldsymbol{x}) = \mathbf{h}(\underbrace{[x_1, \ldots, x_d]}_{\text{inputs}}) = \underbrace{[y_1, \ldots, y_L]}_{\text{outputs}}$$

N.B. Not multi-class!

$\boldsymbol{y} = [0, 1, 1, 0] \Leftrightarrow$ labels $\{2, 3\}$ are relevant to corresponding $\boldsymbol{x}$.

Reduction #1 (to binary): Binary Relevance Method

$\boldsymbol{X}$	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.

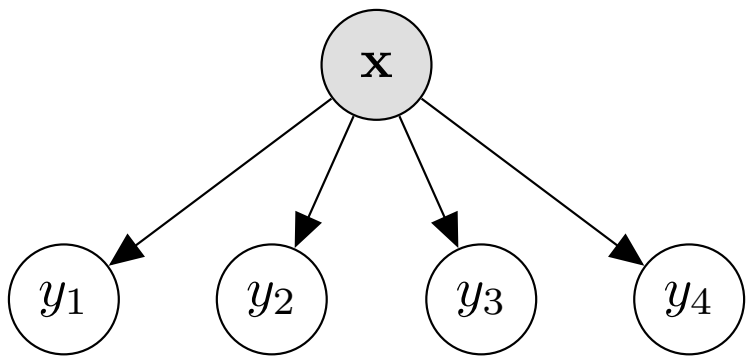
Reduction #1 (to binary): Binary Relevance Method

$\boldsymbol{X}$	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
$\tilde{\boldsymbol{x}}$	?

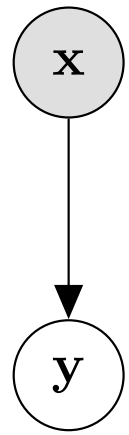
$\boldsymbol{X}$	Y_4
$\boldsymbol{x}^{(1)}$	1
$\boldsymbol{x}^{(2)}$	0
$\boldsymbol{x}^{(3)}$	1
$\boldsymbol{x}^{(4)}$	0
$\boldsymbol{x}^{(5)}$	0
$\tilde{\boldsymbol{x}}$	?



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

Reduction #2 (to multi-class): Label Powerset Method

$\boldsymbol{X}$	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
$\tilde{\boldsymbol{x}}$	?



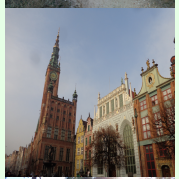
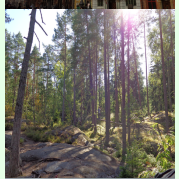
The label powerset method (LP transformation) = *a single target multi-class classifier*. Labels are modeled together, mais ...

- Overfitting
- $y \in \{0, 1\}^L$.

A Brief Timeline of Multi-label Learning in Academia

- $<$ 2000s : Use (baseline) reduction #1 (BR), or #2 (LP)
- . . . 2010 :
  - We beat BR (using label dependence)!
  - Many applications!
- . . . 2015 :
  - We beat the methods that beat BR (using label dependence in a more sophisticated way)!
  - On fait quoi exactement ? Et pourquoi ?
- . . . 2020 :
  - Models get deep, deeper, . . . ; (CNNs, LSTM, . . .)
  - Problems get big, bigger, . . .
  - Do we really need label dependence models ? (BR seems OK)
- Aujourd'hui :
  - New tasks and applications: partial labels, weak labels, label ambiguity, imprecise prediction/with abstention, . . .
  - Models: neural, graph embeddings, adversarial, attention, . . .

Example Application: Multi-Label Classification

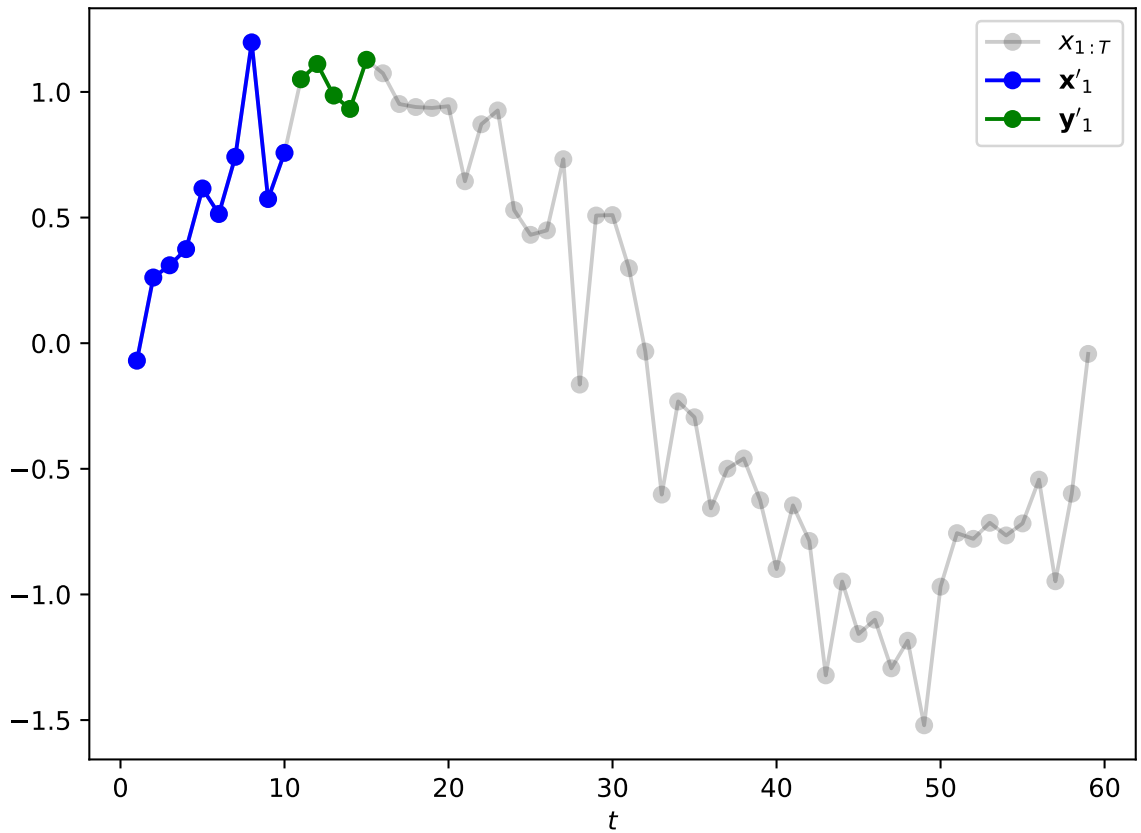
Input	Beach	Sunset	Foliage	Urban
	1	0	1	0
	0	1	0	0
	0	1	0	1
	0	1	1	0
	0	0	1	1
	?	?	?	?

Missing-data Imputation / Recommender Systems

	Film 2 X_2	Film 3 X_4	Book 1 X_1	Book 2 X_3	Song 5 X_5
	0	0	1	1	0
	1	1	?	0	?
	0	0	1	0	0
	1	1	?	0	1
	0	0	0	?	?
	1	0	?	1	?

Time Series Forecasting / Trajectory Prediction

Prévision : séries temporelles



(dont séries multidimensionelles).

Algorithm Adaptations

Support multilabel / multioutput in ScikitLearn:
Des méthodes adaptés à multi-output :

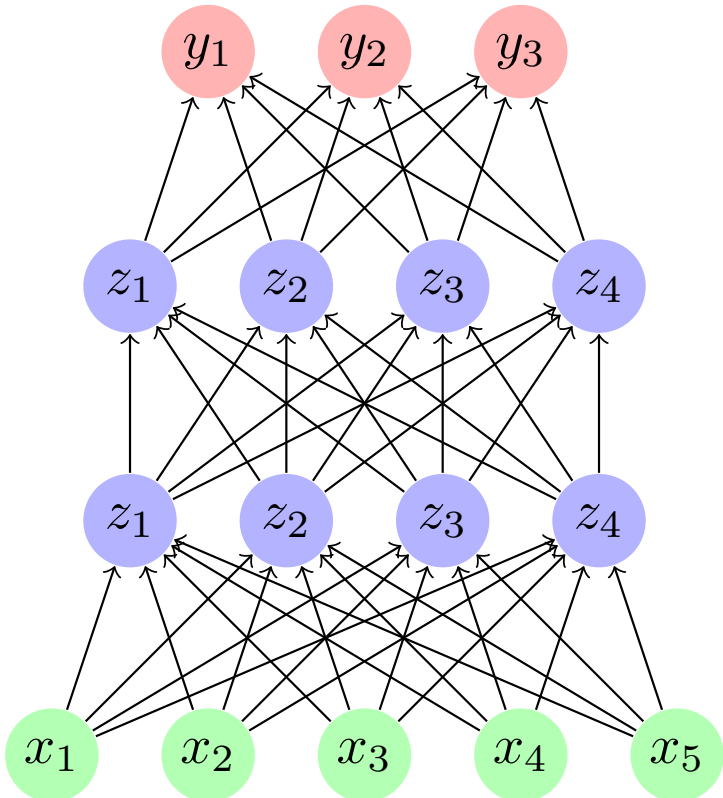
- `sklearn.tree.DecisionTreeClassifier`
- `sklearn.tree.ExtraTreeClassifier`
- `sklearn.ensemble.ExtraTreesClassifier`
- `sklearn.neighbors.KNeighborsClassifier`
- `sklearn.neural_network.MLPClassifier`
- `sklearn.neighbors.RadiusNeighborsClassifier`
- `sklearn.ensemble.RandomForestClassifier`
- `sklearn.linear_model.RidgeClassifierCV`

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

Les méthodes classifieur-agnostic :

- `sklearn.multiclass.OneVsRestClassifier` ← BR
- `sklearn.multioutput.ClassifierChain` ← Coming to this soon

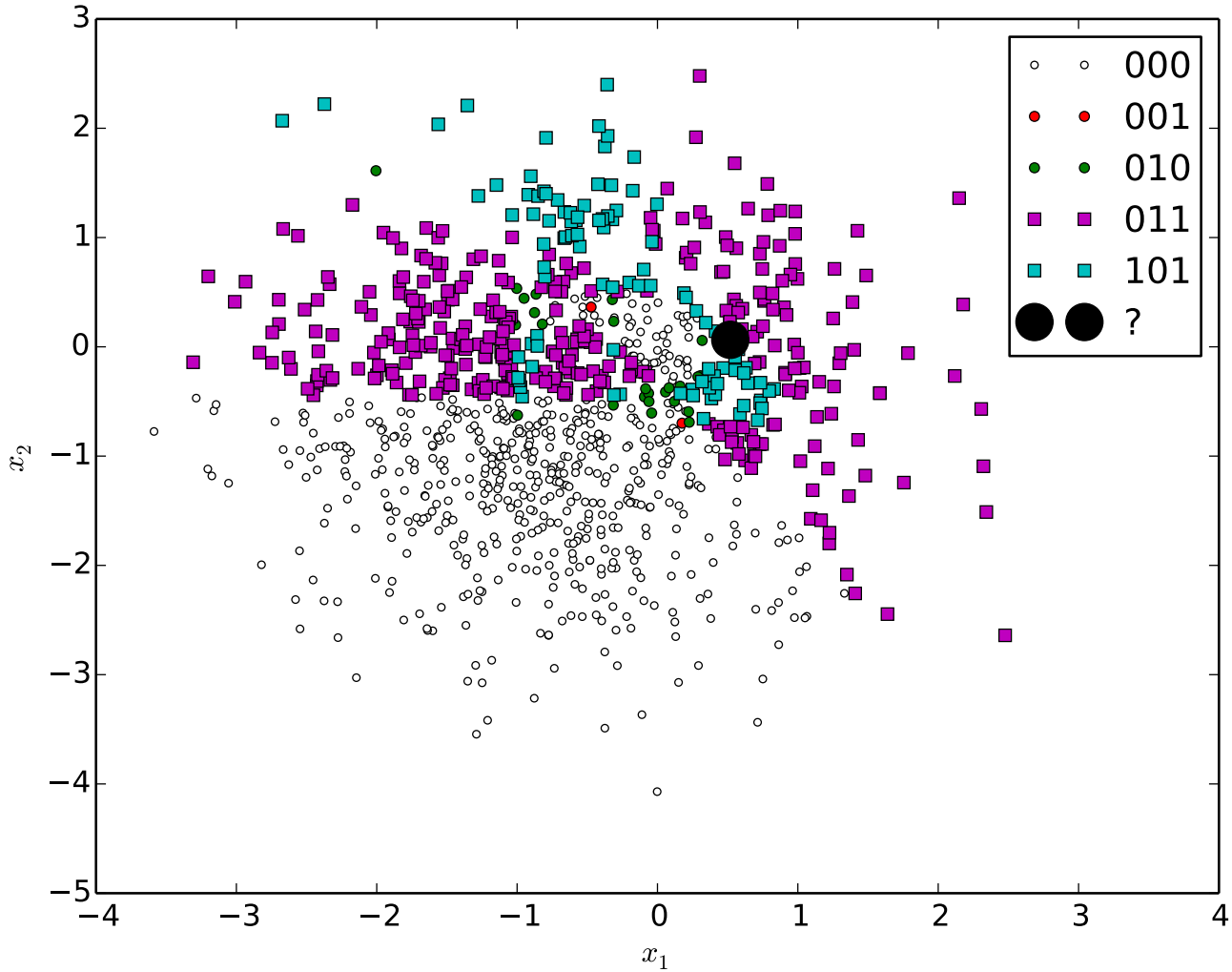
Réseaux de neurones



(on revient à ce sujet plus tard !)

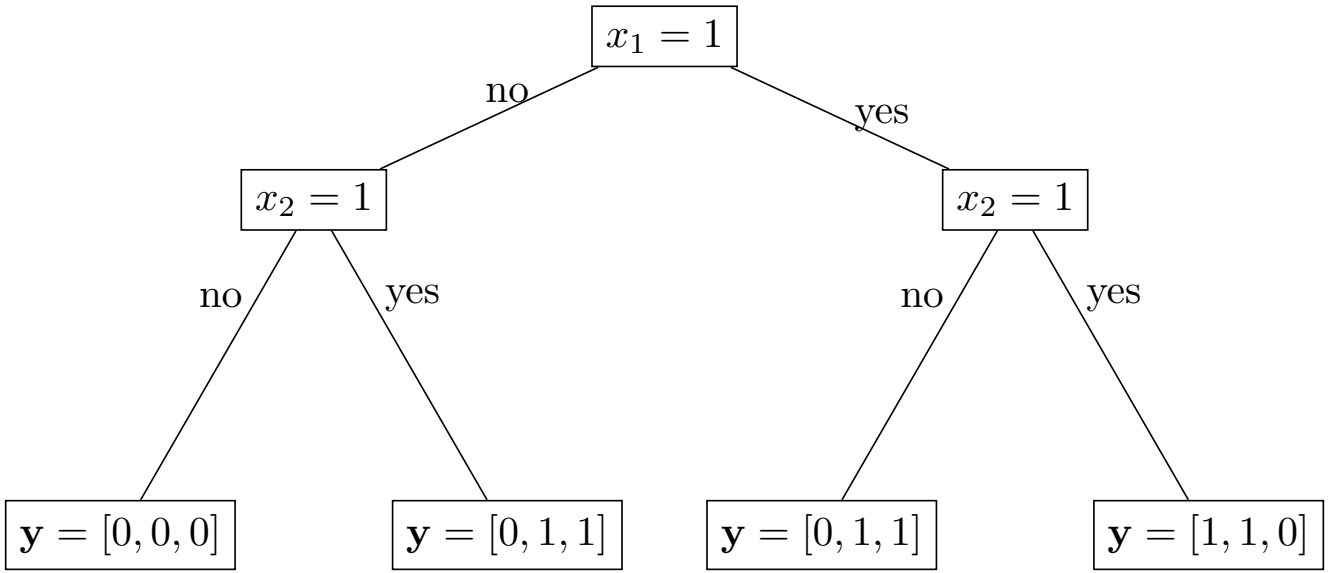
Nam et al., ECML-PKDD 2014 (application to multi-label) and dozens more!

k-Nearest Neighbours (kNN)



Zhang and Zhou, Pat. Reg. 2007 (MLkNN)

Arbres de décision – Classification



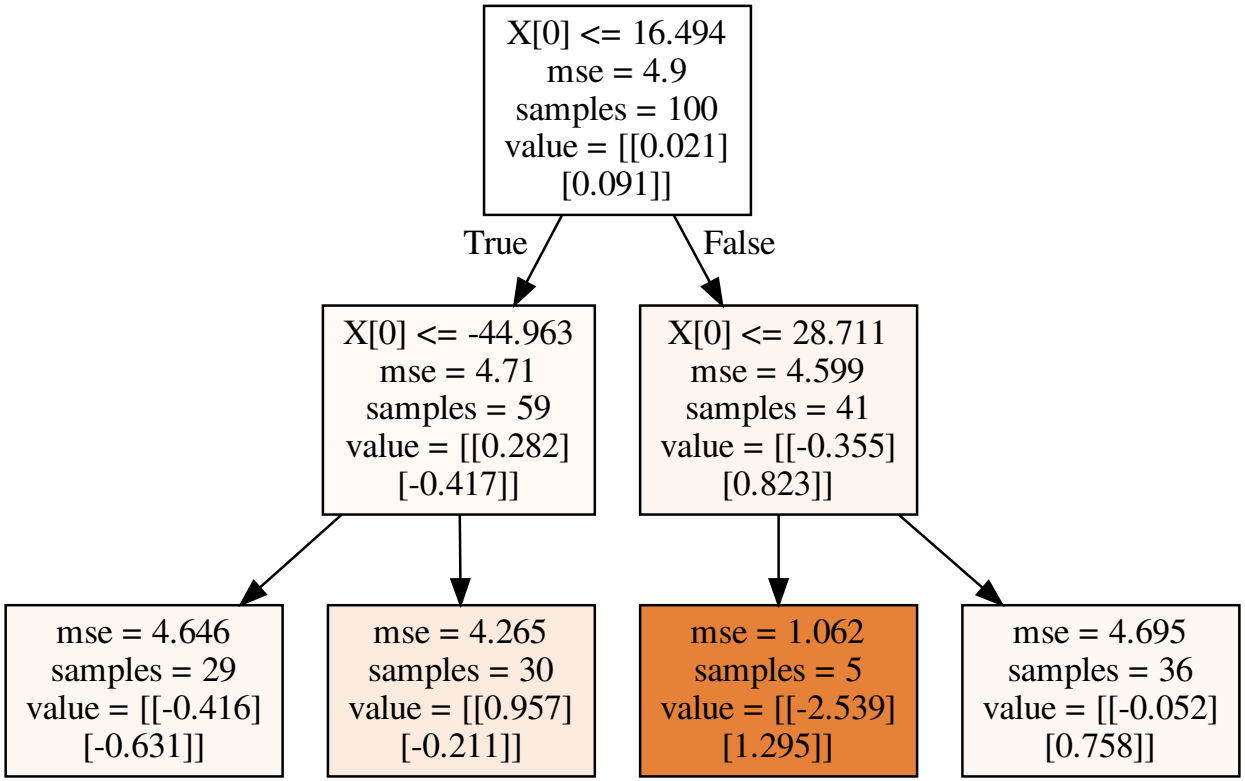
Using multi-label entropy,

$$H_{\text{ML}}(S) = -\sum_{j=1}^{L} \sum_{k \in \{0,1\}} P(y_j = k) \log_2 P(y_j = k)$$

Typical advantages/disadvantages of decision trees apply.

See, e.g., Borchani et al., "A Survey on Multi-output Regression", 2015; more recently: Stepišnik and Kocev, "Oblique Predictive Clustering Trees", 2020

Arbres de décision – Regression



Using redefined impurity measure:

$$\sum_{i=1}^{N} \sum_{j=1}^{L} (y_{ij} - \bar{y}_j)^2$$

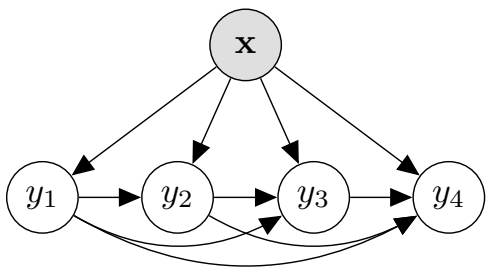
where $\bar{y}_i$ is the mean of Y_j *in the node*.

Classifier Chains

(Greedy) Classifier Chains

Une ‘chaîne’ (structure) entre les variables de sortie ;

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



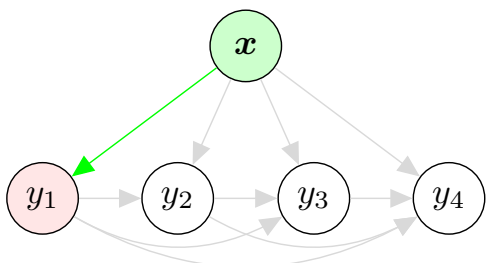
$\boldsymbol{X}$	Y_1	Y_2	Y_3	Y_4
$\boldsymbol{x}^{(1)}$	0	1	1	1
$\boldsymbol{x}^{(2)}$	1	0	0	0
$\boldsymbol{x}^{(3)}$	0	1	0	1
$\boldsymbol{x}^{(4)}$	1	0	0	0
$\boldsymbol{x}^{(5)}$	0	0	0	0
$\tilde{\boldsymbol{x}}$	$\widehat{y}_1$	$\widehat{y}_2$	$\widehat{y}_3$	$\widehat{y}_4$

Read et al., ECML-PKDD 2009 [Test of Time Award 2019]

(Greedy) Classifier Chains

Une ‘chaîne’ (structure) entre les variables de sortie ;

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



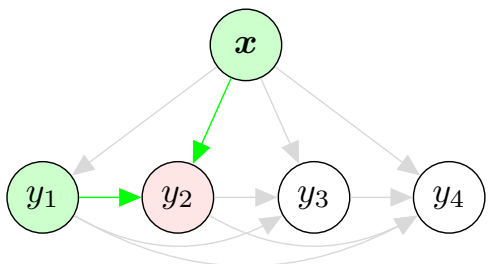
$\boldsymbol{X}$	Y_1	Y_2	Y_3	Y_4
$\boldsymbol{x}^{(1)}$	0	1	1	1
$\boldsymbol{x}^{(2)}$	1	0	0	0
$\boldsymbol{x}^{(3)}$	0	1	0	1
$\boldsymbol{x}^{(4)}$	1	0	0	0
$\boldsymbol{x}^{(5)}$	0	0	0	0
$\tilde{\boldsymbol{x}}$	$\widehat{y}_1$			

Read et al., ECML-PKDD 2009 [Test of Time Award 2019]

(Greedy) Classifier Chains

Une ‘chaîne’ (structure) entre les variables de sortie ;

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



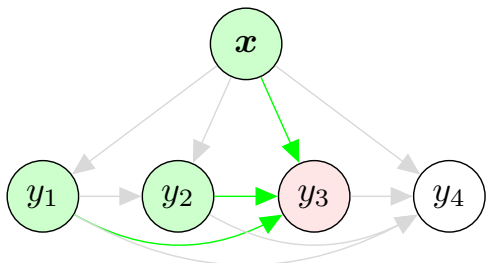
$\boldsymbol{X}$	Y_1	Y_2	Y_3	Y_4
$\boldsymbol{x}^{(1)}$	0	1	1	1
$\boldsymbol{x}^{(2)}$	1	0	0	0
$\boldsymbol{x}^{(3)}$	0	1	0	1
$\boldsymbol{x}^{(4)}$	1	0	0	0
$\boldsymbol{x}^{(5)}$	0	0	0	0
$\tilde{\boldsymbol{x}}$	$\widehat{y}_1$	$\widehat{y}_2$		

Read et al., ECML-PKDD 2009 [Test of Time Award 2019]

(Greedy) Classifier Chains

Une 'chaîne' (structure) entre les variables de sortie ;

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



$\boldsymbol{X}$	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,

$$\widehat{y}_3 = h_3(\, \boldsymbol{x}, \widehat{y}_1, \widehat{y}_2 \,)$$

Use training data to fit base classifier (or regressor) h_3 (e.g., decision tree, logistic regression, ...).

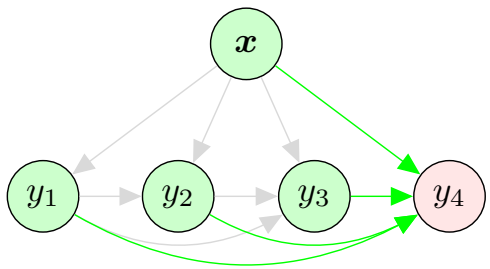
Inference: $\widehat{y}_1, \widehat{y}_2, \ldots$ are greedy *predictions* from $h_1, h_2, \ldots$.

Read et al., ECML-PKDD 2009 [Test of Time Award 2019]

(Greedy) Classifier Chains

Une 'chaîne' (structure) entre les variables de sortie ;

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



$\boldsymbol{X}$	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,

$$\widehat{y}_3 = h_3(\, \boldsymbol{x}, \widehat{y}_1, \widehat{y}_2 \,)$$

Use training data to fit base classifier (or regressor) h_3 (e.g., decision tree, logistic regression, ...).

Inference: $\widehat{y}_1, \widehat{y}_2, \ldots$ are greedy *predictions* from $h_1, h_2, \ldots$.

Read et al., ECML-PKDD 2009 [Test of Time Award 2019]

Prédictions Multi-label: On fait quoi ? Pourquoi ?

Exemples de fonctions de coût :

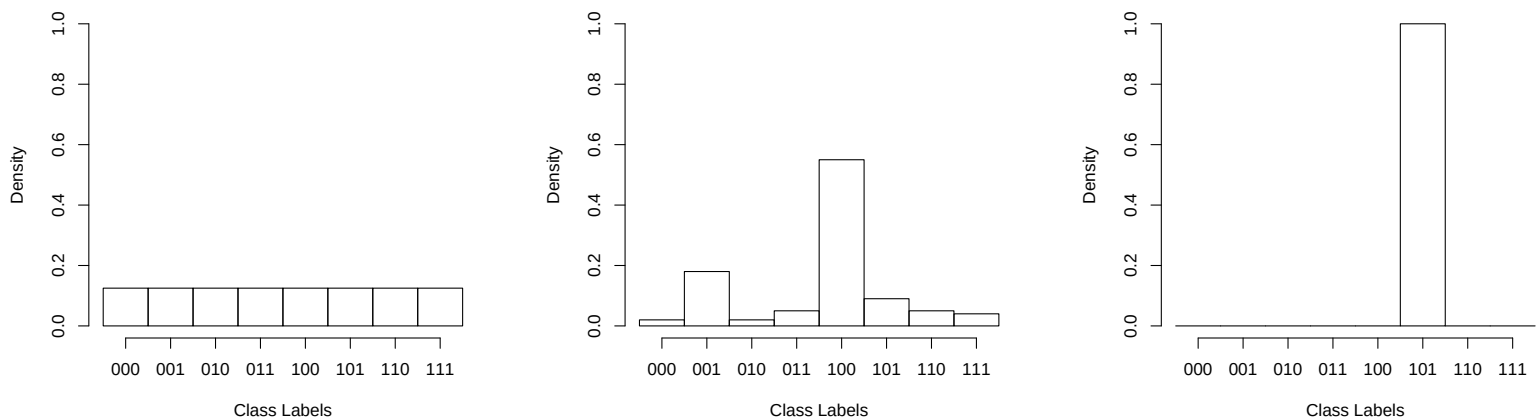
- Hamming loss: decomposable

$$\ell_H([1,0,0],[1,0,1]) = 1/3$$

- 0/1 loss: non-decomposable

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

The minimizer is not (necessarily) the same!

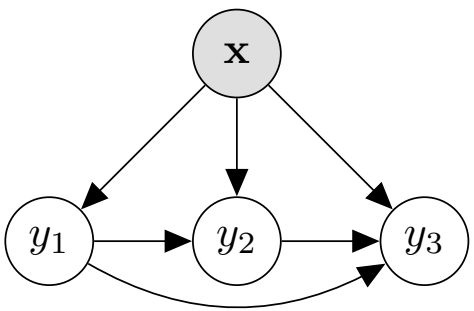


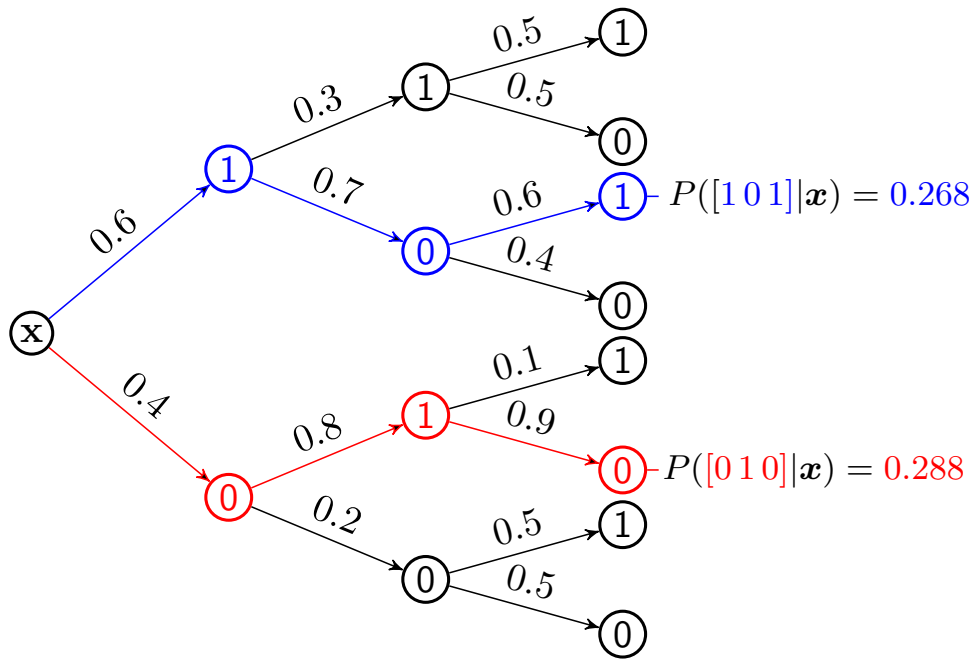
Under uncertainty, left is optimal for Hamming loss, right for 0/1-loss

Probabilistic Classifier Chains

Remarques : Au lieu d'utiliser les predictions $\widehat{y}$ comme entrées (façon *greedy*) ; on peut mettre $y_1, \ldots, y_j = \boldsymbol{y} \in \{0,1\}^j$; afin de minimiser 0/1-loss :

$$\widehat{\boldsymbol{y}} = \underset{\boldsymbol{y} \in \{0,1\}^3}{\operatorname{argmax}} P(\boldsymbol{y}|\boldsymbol{x}) \quad \text{where} \quad P(\boldsymbol{y}|\boldsymbol{x}) = \prod_{j=1}^{3} P(y_j|y_1, \ldots, y_{j-1}, \boldsymbol{x})$$



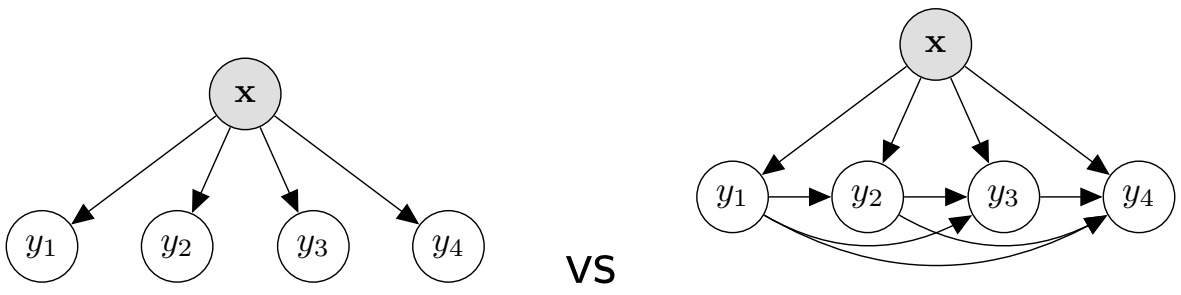


i.e., a path through the probability tree; e.g., $p([0\,1\,0]|\boldsymbol{x}) = 0.288$

Motivation for Structure in Multi-Target Learning ?

Common argument: Because label dependence! Yes for 0/1 loss.

But Hamming loss & other decomposable metrics $\Rightarrow$ classifier chains are useless? (and other structure/dependence-based models).



Risk minimization says that **yes** (chains are useless) under Hamming loss,

but empirical results show classifier chains performing well under most metrics (incl. Hamming loss)!

i.e., structure is generally effective? – then why?

Other reasons for modelling targets together (ainsi que le but de modeliser la probabilité combinée pour minimiser 0/1-loss, etc.):

- Connectivity = efficiency (sometimes)
- Connectivity = interpretation (sometimes)
- Connectivity = power (it's why deep nets or stacking works[1])
- In same cases the minimizer *is* the same (e.g., low-noise scenarios / prediction is easy) = surrogates work well.
- Multiple tasks = regularization (regularization is good)

[1] Different reason if you *train* on $y_j^{(i)}$ or $\widehat{y}_j^{(i)}$ as inputs

Waegeman, Dembczyński, and Hüllermeier, "Multi-target prediction: a unifying view on problems and methods", 2019; Read et al., "Classifier Chains: A Review and Perspectives", 2019 (accepted minor revisions)

Other reasons for modelling targets together (ainsi que le but de modeliser la probabilité combinée pour minimiser 0/1-loss, etc.):

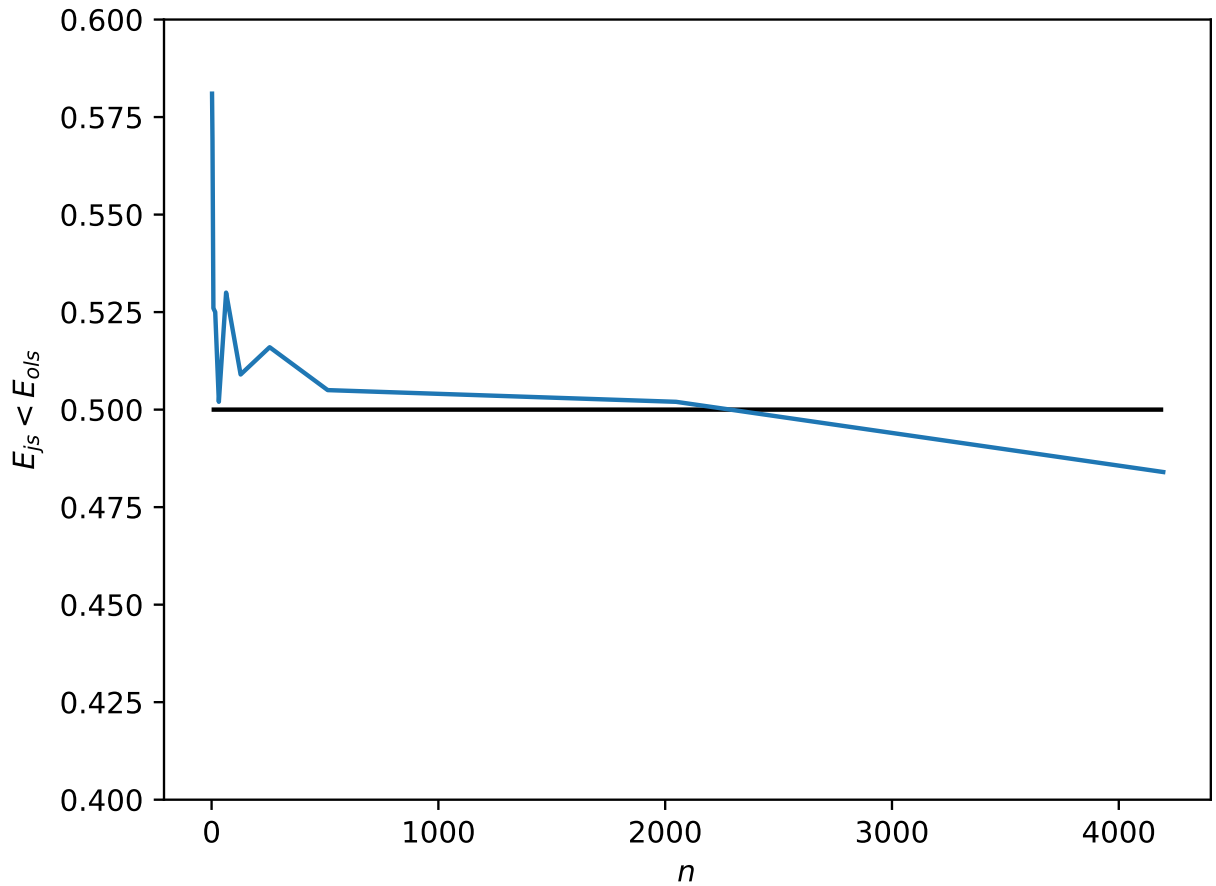
- Connectivity = efficiency (sometimes)
- Connectivity = interpretation (sometimes)
- Connectivity = power (it's why deep nets or stacking works[1])
- In same cases the minimizer *is* the same (e.g., low-noise scenarios / prediction is easy) = surrogates work well.
- Multiple tasks = regularization (regularization is good)

James-Stein Estimator

Joint-target regularization is beneficial *even if targets are intrinsically independent*.

[1] Different reason if you *train* on $y_j^{(i)}$ or $\widehat{y}_j^{(i)}$ as inputs

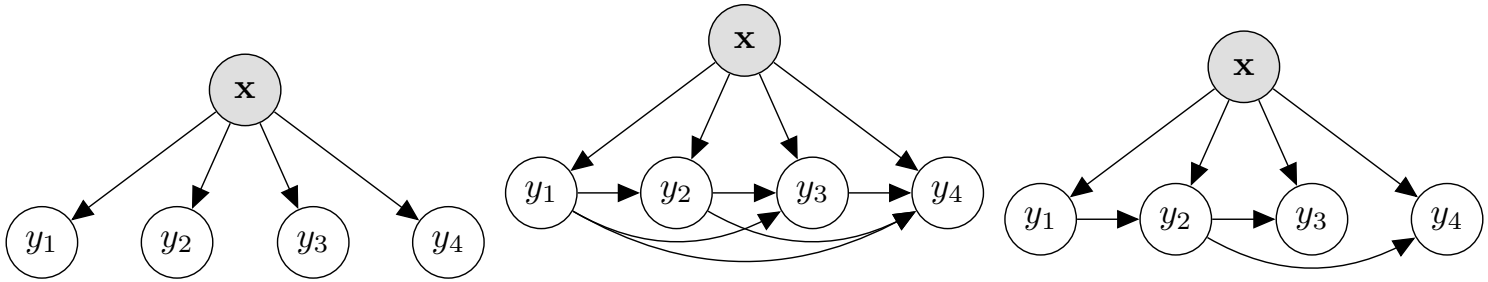
Waegeman, Dembczyński, and Hüllermeier, "Multi-target prediction: a unifying view on problems and methods", 2019; Read et al., "Classifier Chains: A Review and Perspectives", 2019 (accepted minor revisions)



Advantages quickly fade as $n \gg 0$.

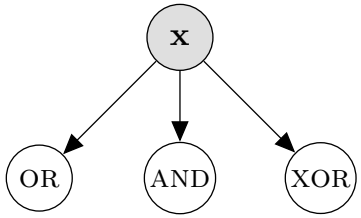
Explains reemergence of independent models vs structure debate. . .

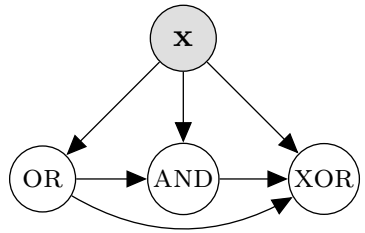
Quelle(s) Structure(s) est la Optimale

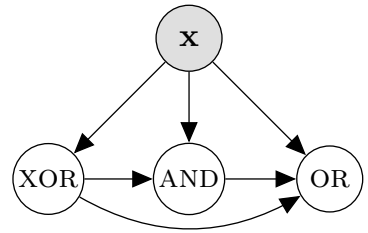


- Different chain *orders* are equivalent in theory *if* you have P
- Dependence is not the only component to consider (and your hierarchy is probably not better than a random one)
- Weaker base learner/smaller training set $\Rightarrow$ more connectivity
- Weaker (greedy) inference $=$ choose more carefully
- Best structure for loss ℓ_a, may not be the best for loss ℓ_b
- Best structure for $\boldsymbol{x}$ not the best for $\tilde{\boldsymbol{x}}$ (you can use a population; do dynamic selection)
- Search: Space is huge, but local optimum can be good

Read et al., "Classifier Chains: A Review and Perspectives", 2019 (under review)

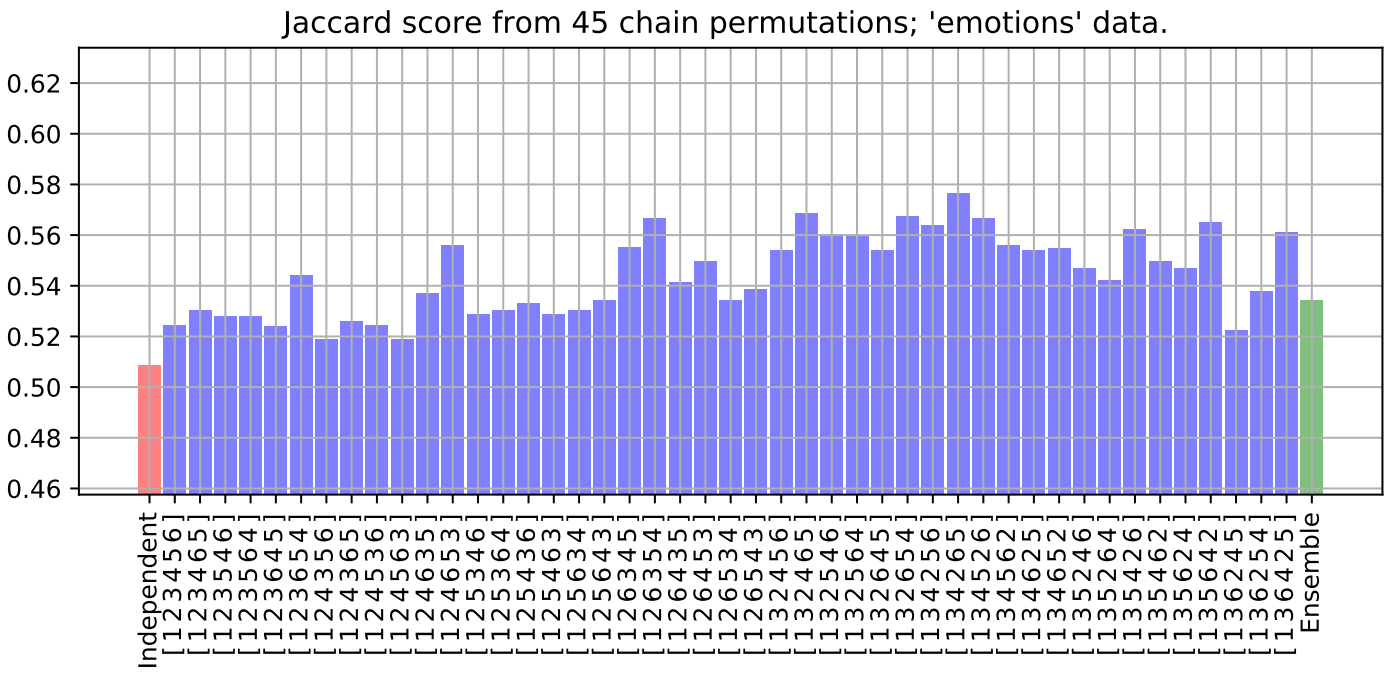






Metric	BR (left)	$\texttt{CC}_1$ (mid)	$\texttt{CC}_2^\dagger$ (right)
Hamming loss	0.17	0	0
0/1 loss	0.50	0	0

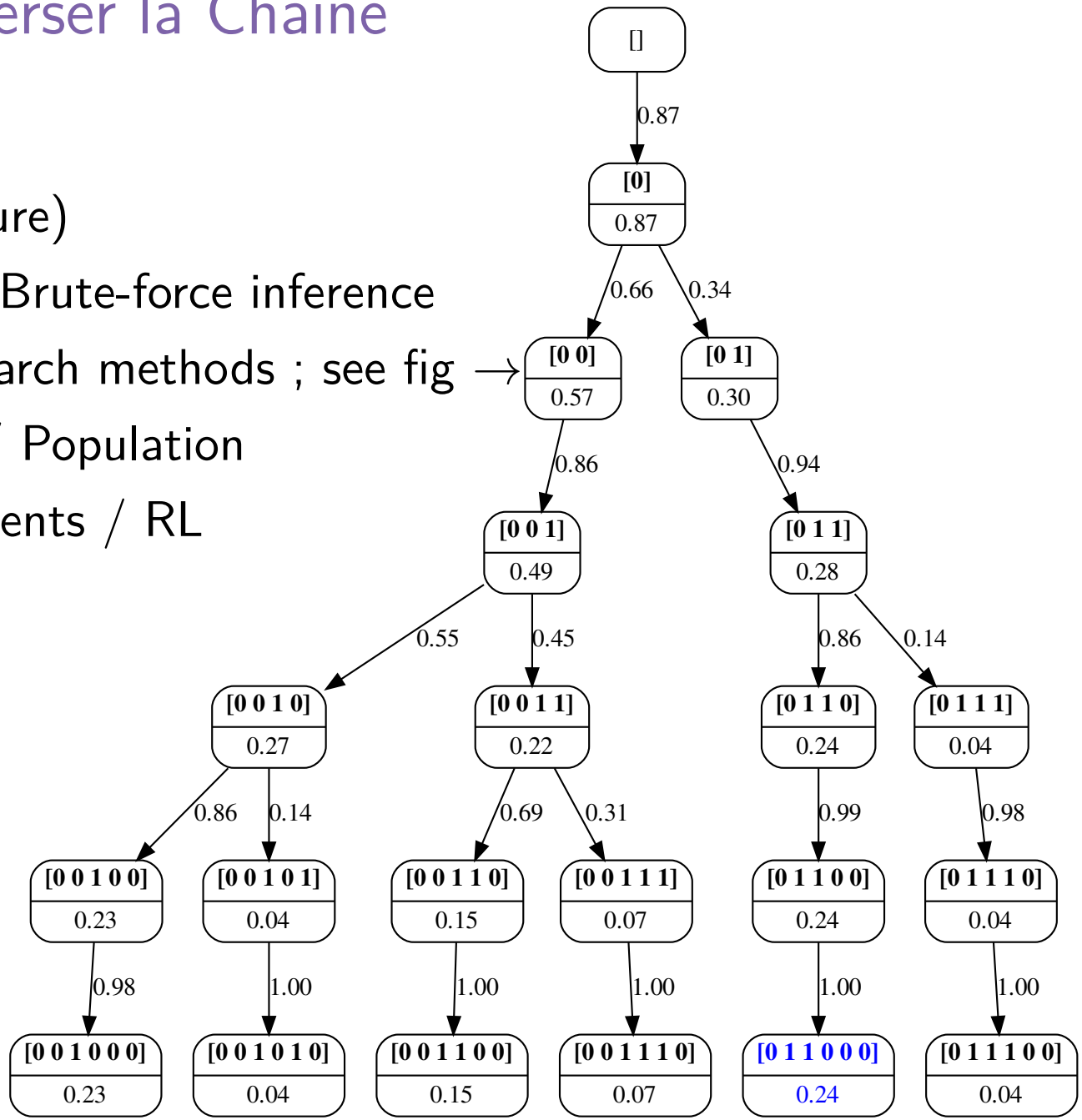
Where $\boldsymbol{x} \in \{0,1\}^2$; Base-model = Logistic regression; †But not greedy inference!



Comment Traverser la Chaîne

(given a structure)

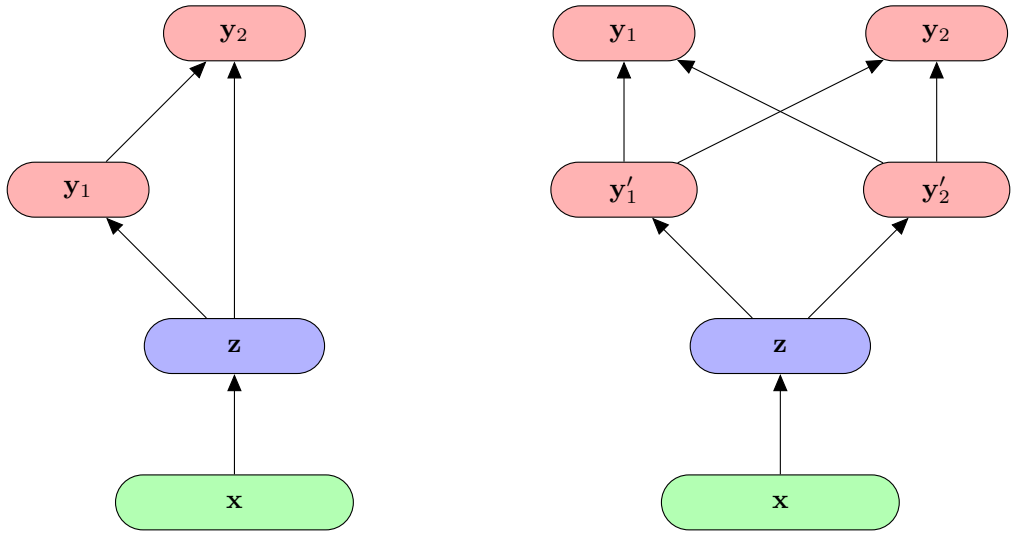
- Greedy vs Brute-force inference
- AI Tree-search methods ; see fig →
- Dynamic / Population
- Generic agents / RL



Vs Reseaux de Neurones Profonds ?

Est-ce qu'on peut faire face aux réseaux de neurones profonds ?

- par rapport à performance: oui, mais pas comme avant (parce que les bases de données sont plus grandes)
- par rapport à l'explicabilité: peut-être;
  - par exemple si on utilise les arbres de décision
  - connexions directs entre les étiquettes (pas via une couche obscure)
- pas besoin de faire face – on peux combiner :



A combination of chaining and deep-neural architectures

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

Regressor Chains

Regression Multi-Cibles

X_1	X_2	X_3	X_4	X_5	Y_1	Y_2	Y_3
2.12	1.217	-0.675	-0.451	0.342	37.00	25	0.88
-0.717	-0.826	0.064	-0.259	-0.717	-22.88	22	0.22
1.374	0.95	0.175	-0.006	-0.522	19.21	12	0.25
1.392	-0.496	-2.441	-1.012	0.268	88.23	11	0.77
1.591	0.208	0.17	-0.207	1.686	?	?	?

On peut utiliser des modèles independants (comme avant)

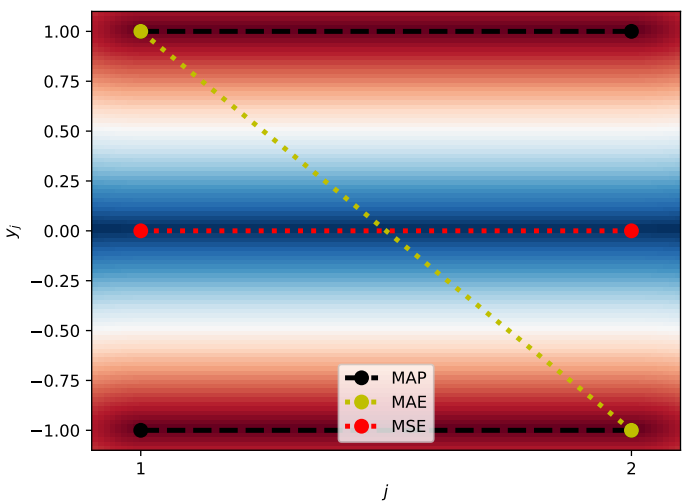
Facile de mettre les modèles dans une chaîne (comme avant; regression chains) ; mais typiquement c'est inutile de le faire (regressor chains $\approx$ independent classifiers) , parce que :

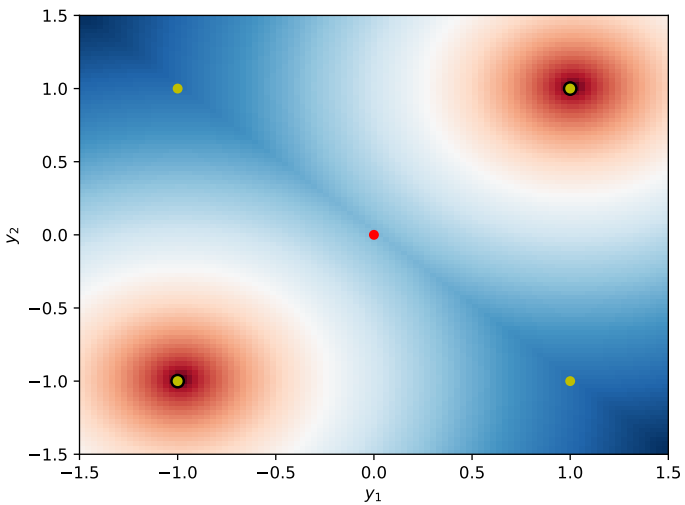
- Our loss metric has changed (probably MSE, MAE, . . .)
- We probably chose linear regression; lost our non-linearity

Regressor Chains

On peut ....

- Utiliser très bien la structure (non-linearity)
  (mais encore difficile de faire face aux arbres multi-output)
- Regarder des autres fonctions de coût – par exemple, la minimisation d'un mode (et pas forcement MSE ou MAE)

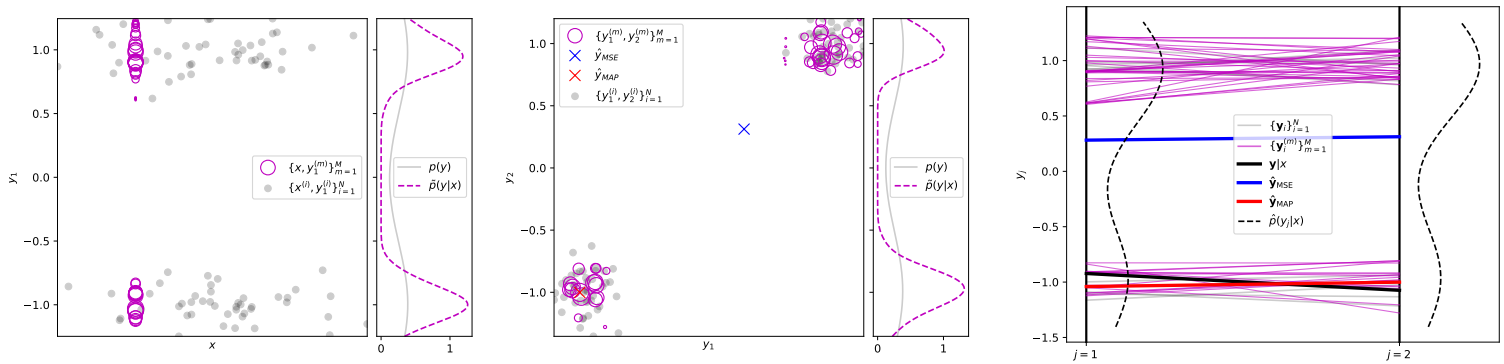




Deux trajectoires incertains (donné $\boldsymbol{x}$) sur $y_1 \in \mathbb{R}$, $y_2 \in \mathbb{R}$: MSE vs MAE vs MAP approx.

Waegeman, Dembczyński, and Hüllermeier, "Multi-target prediction: a unifying view on problems and methods", 2019; Read and Martino, "Probabilistic Regressor Chains with Monte-Carlo Methods", 2020

Sequential Monte Carlo Methods for tracking modal predictions (i.e., trajectories)[2]:



Related approach : Multi-target regression via output space quantization[3]

Other options: Multi-target decision trees[4] and ensembles; and deep learning.

---

[2] Read and Martino, Neurocomputing 2020

[3] Spyromitros-Xioufis, Sechidis, and Vlahavas, ArXiv preprint 2020

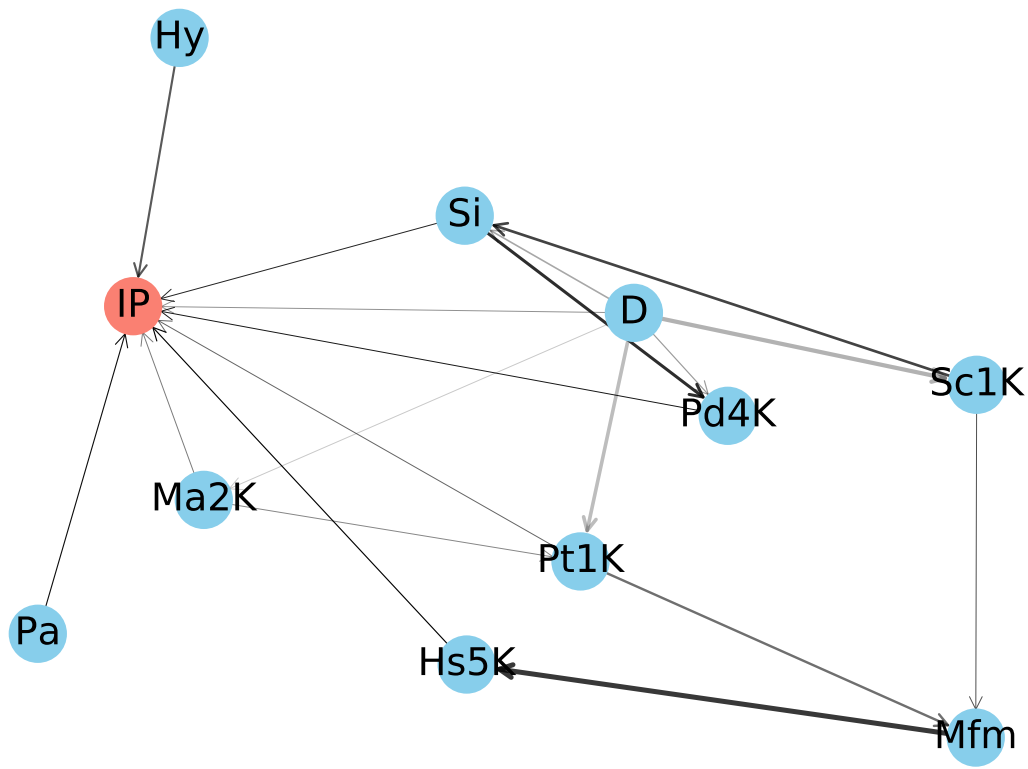
[4] Stepišnik and Kocev, ArXiv preprint 2020

Modern Multi-Output Topics

Questions Ouvertes

Loss metrics: which loss is more appropriate, and given some loss, how to minimize it in a principled way.

Interpretation/Explainability: What does label dependence mean wrt the data?



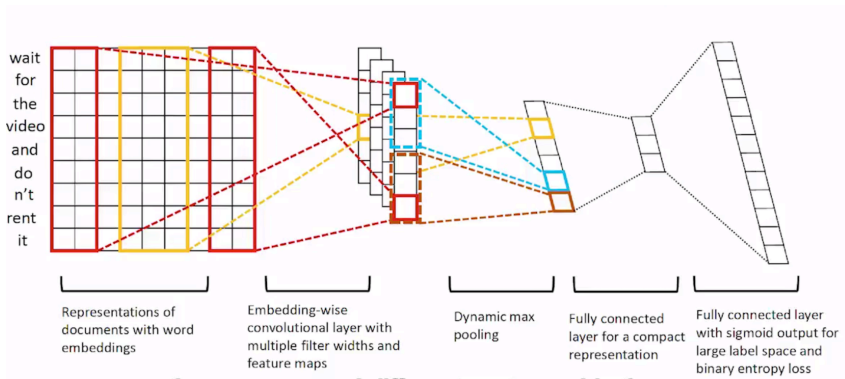
Tendances

Bigger / deeper.

Larger target spaces (4,000—3,000,000 labels), i.e., ‘extreme multi-label classification’

Wider range of applications

- tagging
- video recommendation
- . . .



Intersection with existing areas (deep learning, multi-task, transfer learning, etc.).

---

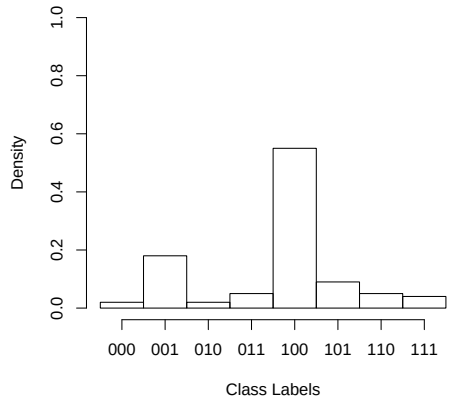
e.g., Jasinska-Kobus et al., “Probabilistic Label Trees for Extreme Multi-label Classification”, 2020 and references therein

Tendances

- Importer des problématiques ‘supplémentaires’ (utilises déjà dans des autres sous-domaines de l’apprentissage automatique) : flux de données, fouille de graphes, semi-supervised learning, time series classification, etc.
- Learning with partial labels (noisy annotators), weak labels (lazy annotators), label ambiguity and imprecise classification (messy ground truth/partial abstention).

The set of candidate labels

building window
sky street
people car
tree



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

Summary

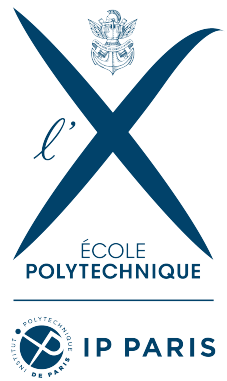
Summary: Multi-target Learning and Prediction

- Special cases: Multi-label classification; multi-target regression
- A look at methods through the lens of classifier chains and regressor chains (decision trees and neural networks as alternative/overlap)
- Main question: *If*, and *why*, and *how* to use structure
- Not answered (in detail): how to *find* that structure. But we argue that there is no single optimal structure, and there is more to multi-target learning than 'modelling label dependencies'; consider metrics, efficiency, base models, ...
- Multi-target problems getting bigger and more diverse; intersecting with other areas
- Many applications; Likely to be an important topic in the future; but more theoretical research needed

Apprentissage Multi-cibles

Applications, Défis, et Modèles

Jesse Read



Merci !

`http://jmread.github.io/`