Multi-Modal Ensembles of Regressor Chains for Multi-Output Prediction

Symposium on Intelligent Data Analysis 2022
April 20-22, 2022, Rennes, France

Ekaterina Antonenko and Jesse Read

April 21, 2022

Laboratoire d'informatique, École Polytechnique, IP Paris
DigitalentLab, MIA, Moteurs d'Intelligence Artificielle

Definition of a multi-output problem

Given: Dataset $\mathcal{D} = \{(\boldsymbol{x}^i, \boldsymbol{y}^i)\}_{i=1}^N$ of N samples:

- features $\boldsymbol{x}^i = [x_1^i, ..., x_M^i]$
- outputs $\boldsymbol{y}^i = [y_1^i, ..., y_L^i]$

Goal: Model $f(X) = \boldsymbol{y}$ which outputs predictions $\hat{\boldsymbol{y}}^i = [\hat{y}_1^i, ..., \hat{y}_L^i]$ having $\mathcal{D}$ observed.

Definition of a multi-output problem

Given: Dataset $\mathcal{D} = \{(\boldsymbol{x}^i, \boldsymbol{y}^i)\}_{i=1}^N$ of N samples:

- features $\boldsymbol{x}^i = [x_1^i, ..., x_M^i]$
- outputs $\boldsymbol{y}^i = [y_1^i, ..., y_L^i]$

Goal: Model $f(X) = \boldsymbol{y}$ which outputs predictions $\hat{\boldsymbol{y}}^i = [\hat{y}_1^i, ..., \hat{y}_L^i]$ having $\mathcal{D}$ observed.

Example:

	X	Age	Height	Body length	Weight
A	...	12,44	127,4	151	294,5
B	...	9,44	137,6	156	328
C	...	10,44	128,6	157	377
D	...	6,13	125,6	150	305,5
E	...	6,15	139	156	325
F	...	?	?	?	?

Definition of a multi-output problem

Given: Dataset $\mathcal{D} = \{(\boldsymbol{x}^i, \boldsymbol{y}^i)\}_{i=1}^N$ of N samples:

- features $\boldsymbol{x}^i = [x_1^i, ..., x_M^i]$
- outputs $\boldsymbol{y}^i = [y_1^i, ..., y_L^i]$

Goal: Model $f(X) = \boldsymbol{y}$ which outputs predictions $\hat{\boldsymbol{y}}^i = [\hat{y}_1^i, ..., \hat{y}_L^i]$ having $\mathcal{D}$ observed.

Example:

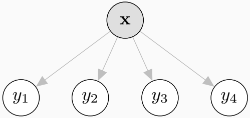
	X	Age	Height	Body length	Weight
A	· · ·	12,44	127,4	151	294,5
B	· · ·	9,44	137,6	156	328
C	· · ·	10,44	128,6	157	377
D	· · ·	6,13	125,6	150	305,5
E	· · ·	6,15	139	156	325
F	· · ·	?	?	?	?

Idea: to model these labels together in order to get better prediction performance

Some approaches to multi-output problems

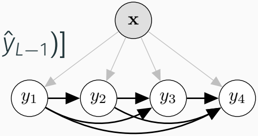
- Independent models (= *binary relevance* for classification):

$$\hat{y} = [\hat{y}_1, ..., \hat{y}_L] = [h_1(x), ..., h_L(x)]$$



- Fully-cascaded chain[1]:

$$\hat{y} = [\hat{y}_1, ..., \hat{y}_L] = [h_1(x), h_2(x, \hat{y}_1), ..., h_L(x, \hat{y}_1, ..., \hat{y}_{L-1})]$$



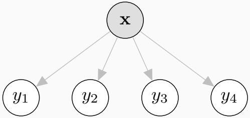
$h_1, h_2, ..., h_L$ = *Base Estimators* (i.e. any single-output models)

[1]*Classifier chains for multi-label classification*, Read et al., 2011, *Multi-Label Classification Methods for Multi-Target Regression*, Spyromitros-Xioufis et al., 2016.

Some approaches to multi-output problems

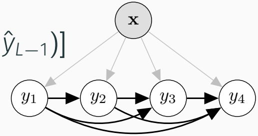
- **Independent models** (= *binary relevance* for classification):

$$\hat{y} = [\hat{y}_1, ..., \hat{y}_L] = [h_1(x), ..., h_L(x)]$$



- **Fully-cascaded chain**[1]:

$$\hat{y} = [\hat{y}_1, ..., \hat{y}_L] = [h_1(x), h_2(x, \hat{y}_1), ..., h_L(x, \hat{y}_1, ..., \hat{y}_{L-1})]$$



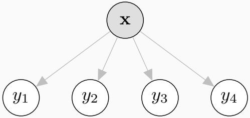
$h_1, h_2, ..., h_L$ = *Base Estimators* (i.e. any single-output models)

	x	Age	Height	Body length	Weight
A	...	?	?	?	?
B	...	?	?	?	?
C	...	?	?	?	?
D	...	?	?	?	?
E	...	?	?	?	?

[1] *Classifier chains for multi-label classification*, Read et al., 2011, *Multi-Label Classification Methods for Multi-Target Regression*, Spyromitros-Xioufis et al., 2016.

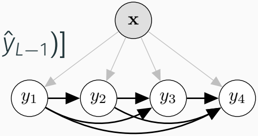
Some approaches to multi-output problems

- **Independent models** (= *binary relevance* for classification):



$$\hat{y} = [\hat{y}_1, ..., \hat{y}_L] = [h_1(x), ..., h_L(x)]$$

- **Fully-cascaded chain**[1]:



$$\hat{y} = [\hat{y}_1, ..., \hat{y}_L] = [h_1(x), h_2(x, \hat{y}_1), ..., h_L(x, \hat{y}_1, ..., \hat{y}_{L-1})]$$

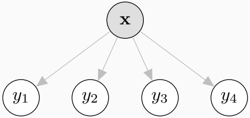
$h_1, h_2, ..., h_L$ = *Base Estimators* (i.e. any single-output models)

	x	Age	Height	Body length	Weight
A	. . .	12,44	?	?	?
B	. . .	9,44	?	?	?
C	. . .	10,44	?	?	?
D	. . .	6,13	?	?	?
E	. . .	6,15	?	?	?

[1] *Classifier chains for multi-label classification*, Read et al., 2011, *Multi-Label Classification Methods for Multi-Target Regression*, Spyromitros-Xioufis et al., 2016.

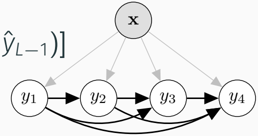
Some approaches to multi-output problems

- **Independent models** (= *binary relevance* for classification):



$$\hat{y} = [\hat{y}_1, ..., \hat{y}_L] = [h_1(x), ..., h_L(x)]$$

- **Fully-cascaded chain**[1]:



$$\hat{y} = [\hat{y}_1, ..., \hat{y}_L] = [h_1(x), h_2(x, \hat{y}_1), ..., h_L(x, \hat{y}_1, ..., \hat{y}_{L-1})]$$

$h_1, h_2, ..., h_L$ = *Base Estimators* (i.e. any single-output models)

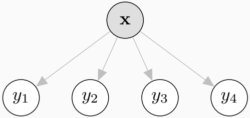
	x	Age	Height	Body length	Weight
A	· · ·	12,44	127,4	?	?
B	· · ·	9,44	137,6	?	?
C	· · ·	10,44	128,6	?	?
D	· · ·	6,13	125,6	?	?
E	· · ·	6,15	139	?	?

[1] *Classifier chains for multi-label classification*, Read et al., 2011, *Multi-Label Classification Methods for Multi-Target Regression*, Spyromitros-Xioufis et al., 2016.

Some approaches to multi-output problems

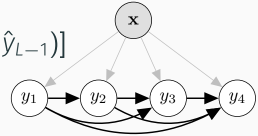
- **Independent models** (= *binary relevance* for classification):

$$\hat{y} = [\hat{y}_1, ..., \hat{y}_L] = [h_1(x), ..., h_L(x)]$$



- **Fully-cascaded chain**[1]:

$$\hat{y} = [\hat{y}_1, ..., \hat{y}_L] = [h_1(x), h_2(x, \hat{y}_1), ..., h_L(x, \hat{y}_1, ..., \hat{y}_{L-1})]$$



$h_1, h_2, ..., h_L$ = *Base Estimators* (i.e. any single-output models)

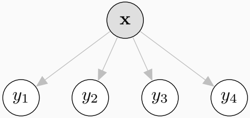
	x	Age	Height	Body length	Weight
A	· · ·	12,44	127,4	151	?
B	· · ·	9,44	137,6	156	?
C	· · ·	10,44	128,6	157	?
D	· · ·	6,13	125,6	150	?
E	· · ·	6,15	139	156	?

[1] *Classifier chains for multi-label classification*, Read et al., 2011, *Multi-Label Classification Methods for Multi-Target Regression*, Spyromitros-Xioufis et al., 2016.

Some approaches to multi-output problems

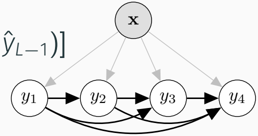
- **Independent models** (= *binary relevance* for classification):

$$\hat{y} = [\hat{y}_1, ..., \hat{y}_L] = [h_1(x), ..., h_L(x)]$$



- **Fully-cascaded chain**[1]:

$$\hat{y} = [\hat{y}_1, ..., \hat{y}_L] = [h_1(x), h_2(x, \hat{y}_1), ..., h_L(x, \hat{y}_1, ..., \hat{y}_{L-1})]$$

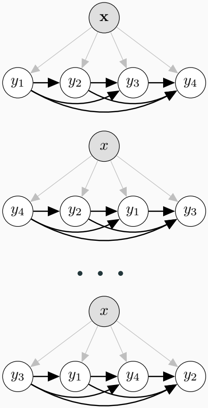


$h_1, h_2, ..., h_L$ = *Base Estimators* (i.e. any single-output models)

	x	Age	Height	Body length	Weight
A	. . .	12,44	127,4	151	294,5
B	. . .	9,44	137,6	156	328
C	. . .	10,44	128,6	157	377
D	. . .	6,13	125,6	150	305,5
E	. . .	6,15	139	156	325

[1] *Classifier chains for multi-label classification*, Read et al., 2011, *Multi-Label Classification Methods for Multi-Target Regression*, Spyromitros-Xioufis et al., 2016.

Ensembles of chains[2]



Mean average

[2] *Multi-Label Classification Methods for Multi-Target Regression*, Spyromitros-Xioufis et al., 2016.

Does the chaining approach work?

Classification

Classifier Chains have proved to be **flexible and effective** and have achieved **state-of-the-art** empirical performance

- *Classifier Chains: A Review and Perspectives*, Read et al., 2021

Regression

Regressor Chains show **relatively few advantages** compared to individual regression models. State-of-the-art methods:

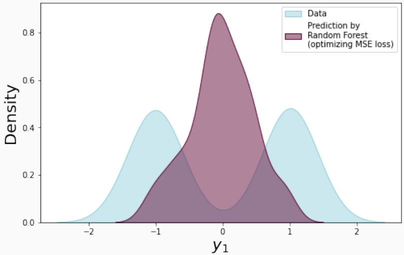
- Multi-output Decision Trees (DT)
- Multi-output Random Forests (RF)
- Independent Regressors (IR)

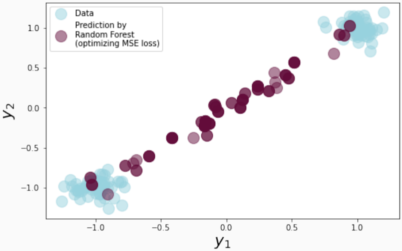
Regressor chains: why don't they work?

One possible reason: inadequate choice of the loss function

Most models optimize $\text{MSE} = \frac{1}{N}\sum_{j=1}^{N}(y_j - \hat{y}_j)^2$.

Example: Multi-modal distribution $\implies$ standard models may be inappropriate.





Optimizing MSE does not help to exploit the dependencies between the targets.

Multi-Modal Ensembles of Regressor Chains (mmERC)

Uniform Cost Function (UCF)[3] is an analogue of 0/1 loss for regression.

$$\mathrm{UCF}(\delta) = \frac{1}{N} \sum_{i=1}^{N} \begin{cases} 0 \text{ if } \|\boldsymbol{y}^i - \hat{\boldsymbol{y}}^i\|_2 < \frac{\delta}{2}, \\ 1 \text{ otherwise.} \end{cases}$$

Goal = challenge: optimize UCF.

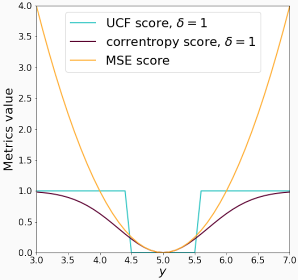
[3] *Maximum a posteriori estimates in linear inverse problems with log-concave priors are proper bayes estimators*, Burger and Lucka, 2014.

Multi-Modal Ensembles of Regressor Chains (mmERC)

Uniform Cost Function (UCF)[3] is an analogue of 0/1 loss for regression.

$$\mathrm{UCF}(\delta) = \frac{1}{N}\sum_{i=1}^{N}\begin{cases}0 \text{ if } \|\boldsymbol{y}^i - \hat{\boldsymbol{y}}^i\|_2 < \frac{\delta}{2},\\ 1 \text{ otherwise.}\end{cases}$$

Goal = challenge: optimize UCF.



[3] *Maximum a posteriori estimates in linear inverse problems with log-concave priors are proper bayes estimators*, Burger and Lucka, 2014.

Base Estimator level

ERC

- Single round of training

mmERC

- Train on all dataset
- Choose portion of data giving best predictions
- Retrain on this part

Algorithm 1 mmERC: Training h_j for target y_j (done for $j = 1, \ldots, L$)

```
procedure FIT(h_j, {x, y_j})            ▷ Train b.e. h_j for target y_j on {x, y_j}
    h̃_j ← clone of h_j
    fit h̃_j on {(x, y_j)}               ▷ First training phase (full training set)
    y_pred ← h̃_j(x)                     ▷ Prediction of h̃_j on x
    corr ← 1 − e^{−(y_j − y_pred)^2}    ▷ Correntropy
    {x′, y′_j} ⊂ {x, y_j}               ▷ Top s-instances wrt (lowest) corr, 0 < s < 1
    fit h_j on {(x′, y′_j)}             ▷ Second training phase
    return h_j                          ▷ Return the trained model
```

Multi-Modal Ensembles of Regressor Chains (mmERC)

Base Estimator level

ERC

- Single round of training

mmERC

- Train on all dataset
- Choose portion of data giving best predictions
- Retrain on this part

Algorithm 1 mmERC: Training h_j for target y_j (done for $j = 1, \ldots, L$)

procedure FIT(h_j, $\{\boldsymbol{x}, y_j\}$) ▷ Train b.e. h_j for target y_j on $\{\boldsymbol{x}, y_j\}$
    $\widetilde{h}_j \leftarrow$ clone of h_j
    fit $\widetilde{h}_j$ on $\{(\boldsymbol{x}, y_j)\}$ ▷ First training phase (full training set)
    $y_{pred} \leftarrow \widetilde{h}_j(\boldsymbol{x})$ ▷ Prediction of $\widetilde{h}_j$ on $\boldsymbol{x}$
    $corr \leftarrow 1 - e^{-(y_j - y_{pred})^2}$ ▷ Correntropy
    $\{\boldsymbol{x}', y'_j\} \subset \{\boldsymbol{x}, y_j\}$ ▷ Top s-instances wrt (lowest) $corr$, $0 < s < 1$
    fit h_j on $\{(\boldsymbol{x}', y'_j)\}$ ▷ Second training phase
    return h_j ▷ Return the trained model

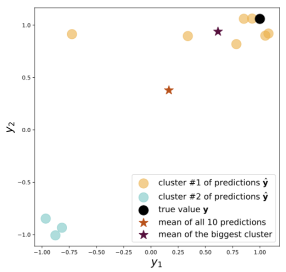
Ensemble level

ERC

- Mean for all predictions

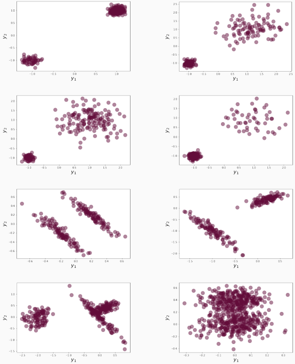
mmERC

- Choose the largest cluster of predictions
- Take mean for this cluster only



mmERC: results on 40 synthetic datasets

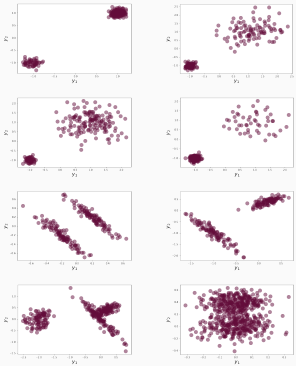
Datasets (x, y_1, y_2):



For each y distribution, there are five x distributions	
A:	$\sim U(0,1)$ where U stands for uniform distribution
B:	$\sim \{0,1\}$ (according to the cluster)
C:	$\{\sim U(0,1), \sim U(1,2)\}$ (according to the cluster)
D:	$\{\sim \mathcal{N}(0,1), \sim \mathcal{N}(1,1)\}$ (according to the cluster)
E:	$\sim \mathcal{N}(0,1)$

mmERC: results on 40 synthetic datasets

Datasets (x, y_1, y_2):



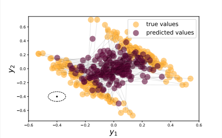
For each y distribution, there are five x distributions

A:	$\sim U(0, 1)$ where U stands for uniform distribution
B:	$\sim \{0, 1\}$ (according to the cluster)
C:	$\{\sim U(0, 1), \sim U(1, 2)\}$ (according to the cluster)
D:	$\{\sim \mathcal{N}(0, 1), \sim \mathcal{N}(1, 1)\}$ (according to the cluster)
E:	$\sim \mathcal{N}(0, 1)$

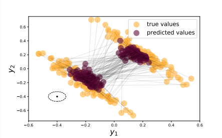
Results (UCF):

Regressor	A	B	C	D	E	Average	AvgRank
DT	0.71	0.50	0.50	0.70	0.73	0.63 ± 0.01	7.9
RF	0.84	0.47	0.45	0.78	0.84	0.67 ± 0.04	10.2
IR (dt)	0.79	0.50	0.52	0.74	0.78	0.66 ± 0.02	11.1
IR (rf)	0.86	0.47	0.47	0.79	0.87	0.69 ± 0.04	11.0
IR (svr)	0.72	**0.40**	0.52	0.70	0.72	0.61 ± 0.02	6.0
RC (dt)	0.74	0.50	0.51	0.70	0.72	0.63 ± 0.01	8.6
RC (rf)	0.81	0.45	0.45	0.75	0.82	0.66 ± 0.03	8.8
RC (svr)	0.70	**0.40**	0.51	0.67	0.71	0.60 ± 0.02	4.2
ERC (dt)	0.78	0.50	0.49	0.72	0.76	0.65 ± 0.02	8.6
ERC (rf)	0.83	0.44	**0.44**	0.76	0.83	0.66 ± 0.04	8.6
ERC (svr)	0.71	**0.40**	0.50	0.67	0.72	0.60 ± 0.02	5.0
mmERC (dt)	0.72	0.50	0.51	0.69	0.71	0.63 ± 0.01	8.2
mmERC (rf)	0.69	0.43	**0.44**	**0.63**	**0.67**	$\mathbf{0.57 \pm 0.02}$	**2.2**
mmERC (svr)	**0.69**	**0.40**	0.52	0.67	0.68	0.59 ± 0.02	4.6

(grouped by x distribution)



Random Forest



mmERC (RF)

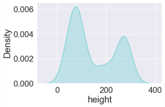
mmERC: results on *yacon* dataset

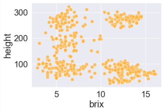
Yacon dataset:

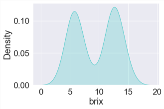
mmERC: results on *yacon* dataset

Yacon dataset:



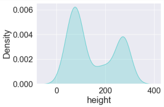


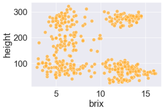


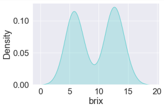


mmERC: results on *yacon* dataset

Yacon dataset:

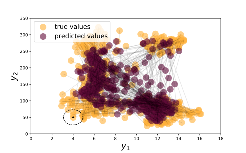




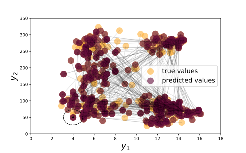


Results:

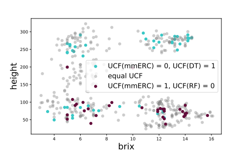
DT	RF	IR (dt)	IR (rf)	IR (svr)	RC (dt)	RC (rf)	RC (svr)	ERC (dt)	ERC (rf)	ERC (svr)	mmERC (dt)	mmERC (rf)	mmERC (svr)
0.92	0.94	0.92	0.94	0.91	0.92	0.94	0.92	0.92	0.94	0.94	0.89	0.89	**0.83**



mmERC (rf)



DT



mmERC (rf) vs. DT

Discussion

- **mmERC** on average **outperforms the independent regressors** and standard Regressor Chains
- **mmERC improves ERC** for all base estimators in most of the scenarios
- Decision Trees: recognize cluster shape, but may assign clusters randomly, not smooth
  Random Forests: "smooth", but put the predictions between the real clusters
  mmERC: improves the performance of Random Forests and outputs a **smooth function** at the same time

Thank you!