Drawing Parallels between Multi-label Classification and Multi-target Regression

Grigorios Tsoumakas, Eleftherios Spyromitros-Xioufis, and Ioannis Vlahavas
Machine Learning and Knowledge Discovery (MLKD) group
Department of Informatics, Aristotle University of Thessaloniki, Greece

Multi-label Classification & Multi-target Regression

- Two instances of multi-target prediction

Multi-Label Classification (MLC)

X_1	X_2	…	X_n	Y_1	Y_2	…	Y_m
0.12	1	…	12	0	1	…	1
2.34	9	…	-5	0	1	…	0
1.22	3	…	40	1	0	…	0
2.18	2	…	8	?	?	…	?
1.76	7	…	23	?	?	…	?

training examples

unknown instances

n inputs

m binary targets

Multi-label Classification & Multi-target Regression

- Two instances of multi-target prediction

Multi-Target (multivariate) Regression (MTR)

X_1	X_2	…	X_n	Y_1	Y_2	…	Y_m
0.12	1	…	12	0.14	10	…	-1.3
2.34	9	…	-5	4.15	12	…	-2.0
1.22	3	…	40	1.01	28	…	-5.3
2.18	2	…	8	?	?	…	?
1.76	7	…	23	?	?	…	?

training examples

unknown instances

n inputs

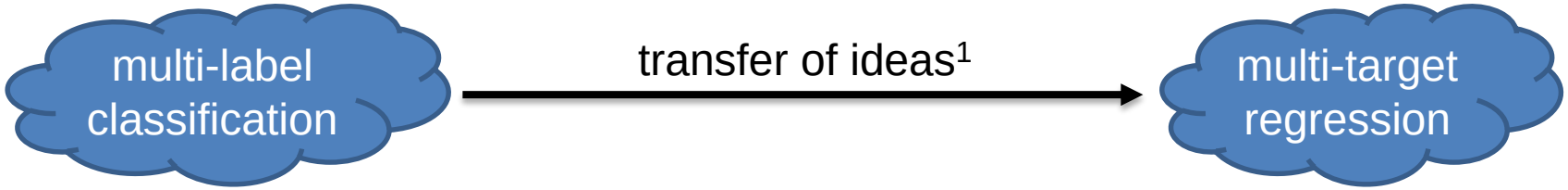
m continuous targets

MLC and MTR Applications

- MLC
  - Multimedia annotation/retrieval
  - Text categorization
  - Gene function prediction
  - …many more
- MTR
  - Ecological modeling (e.g. water quality prediction)
  - Price prediction (stocks, airline tickets, etc.)
  - Power (solar/wind) generation forecasting
  - …and many recent Kaggle competitions

Motivation

- Similar problems
  - Same baseline approach (an independent model for each target)
  - Shared challenges:
    - Scaling to large numbers of targets / Exploiting target dependencies
- MLC is a more popular research topic
  - At least 4 MLC papers in ECML/PKDD 2014 (with MLC in title)
  - A multitude of new MLC methods
- Questions:
  - Can one field benefit from the other?
  - Are there successful MLC methods that can be used in MTR?



Categorization of MLC Methods and Applicability on MTR

- Problem transformation methods
  - Modelling single-labels: multiple binary classification problems
    - E.g. Binary Relevance, Multi-label Stacking[2,3], Classifier Chains[4,5]
    - Almost directly applicable!
  - Modelling pairs: one-versus-one decomposition paradigm
    - E.g. Calibrated Label Ranking[6]
    - Approach not applicable!
  - Modelling sets: multi-class problems where distinct label subsets represent different class values
    - E.g. Label Powerset, RAkEL[7], Pruned Sets[8]
    - Approach seems not applicable!
- Algorithm adaptation methods
  - Applicability depends on ability to handle regression data
    - Easy for decision-tree-based methods (e.g. PCT[9] framework)

Single-target Decomposition Techniques

- The simplest one is Binary Relevance: $h(\boldsymbol{x}) \rightarrow \boldsymbol{y} \overset{BR}{\Rightarrow} h_i(\boldsymbol{x}) \rightarrow y_i, i = 1, \ldots, m$

X_1	X_2	…	X_n	Y_1	Y_2	…	Y_m
0.12	1	…	12	0	1	…	1
2.34	9	…	-5	0	1	…	0
1.22	3	…	40	1	0	…	0
2.18	2	…	8	?	?	…	?
1.76	7	…	23	?	?	…	?

Single-target Decomposition Techniques

- The simplest one is Binary Relevance: $h(\boldsymbol{x}) \rightarrow \boldsymbol{y} \overset{BR}{\Rightarrow} h_i(\boldsymbol{x}) \rightarrow y_i, i = 1, \ldots, m$

h_1

X_1	X_2	…	X_n	Y_1	Y_2	…	Y_m
0.12	1	…	12	0	1	…	1
2.34	9	…	-5	0	1	…	0
1.22	3	…	40	1	0	…	0
2.18	2	…	8	?	?	…	?
1.76	7	…	23	?	?	…	?

Single-target Decomposition Techniques

- The simplest one is Binary Relevance: $h(\boldsymbol{x}) \rightarrow \boldsymbol{y} \overset{BR}{\Rightarrow} h_i(\boldsymbol{x}) \rightarrow y_i, i = 1, \ldots, m$

h_2

X_1	X_2	…	X_n	Y_1	Y_2	…	Y_m
0.12	1	…	12	0	1	…	1
2.34	9	…	-5	0	1	…	0
1.22	3	…	40	1	0	…	0
2.18	2	…	8	?	?	…	?
1.76	7	…	23	?	?	…	?

Single-target Decomposition Techniques

- The simplest one is Binary Relevance: $h(\boldsymbol{x}) \rightarrow \boldsymbol{y} \overset{BR}{\Rightarrow} h_i(\boldsymbol{x}) \rightarrow y_i, i = 1, \ldots, m$

h_m

X_1	X_2	…	X_n	Y_1	Y_2	…	Y_m
0.12	1	…	12	0	1	…	1
2.34	9	…	-5	0	1	…	0
1.22	3	…	40	1	0	…	0
2.18	2	…	8	?	?	…	?
1.76	7	…	23	?	?	…	?

Single-target Decomposition Techniques

- The simplest one is Binary Relevance: $h(\boldsymbol{x}) \rightarrow \boldsymbol{y} \overset{BR}{\Rightarrow} h_i(\boldsymbol{x}) \rightarrow y_i, i = 1, \ldots, m$
- Better ones (considering label dependencies):
  - Classifier Chains
  - Stacking

X_1	X_2	…	X_n	Y_1	Y_2	…	Y_m
0.12	1	…	12	0	1	…	1
2.34	9	…	-5	0	1	…	0
1.22	3	…	40	1	0	…	0
2.18	2	…	8	?	?	…	?
1.76	7	…	23	?	?	…	?

Single-target Decomposition Techniques

- The simplest one is Binary Relevance: $h(\boldsymbol{x}) \rightarrow \boldsymbol{y} \overset{BR}{\Rightarrow} h_i(\boldsymbol{x}) \rightarrow y_i, i = 1, \ldots, m$
- Better ones (considering label dependencies):
  - Classifier Chains: $h(\boldsymbol{x}) \rightarrow \boldsymbol{y} \overset{CC}{\Rightarrow} h_1(\boldsymbol{x}) \rightarrow y_1, h_2(\boldsymbol{x}y_1) \rightarrow y_2, \ldots, h_m(\boldsymbol{x}y_1 \ldots y_{m-1}) \rightarrow y_m$
  - Stacking

h_1

X_1	X_2	…	X_n	Y_1	Y_2	…	Y_m
0.12	1	…	12	0	1	…	1
2.34	9	…	-5	0	1	…	0
1.22	3	…	40	1	0	…	0
2.18	2	…	8	?	?	…	?
1.76	7	…	23	?	?	…	?

Single-target Decomposition Techniques

- The simplest one is Binary Relevance: $h(\boldsymbol{x}) \rightarrow \boldsymbol{y} \overset{BR}{\Rightarrow} h_i(\boldsymbol{x}) \rightarrow y_i, i = 1, \ldots, m$
- Better ones (considering label dependencies):
  - Classifier Chains: $h(\boldsymbol{x}) \rightarrow \boldsymbol{y} \overset{CC}{\Rightarrow} h_1(\boldsymbol{x}) \rightarrow y_1, h_2(\boldsymbol{x}y_1) \rightarrow y_2, \ldots, h_m(\boldsymbol{x}y_1 \ldots y_{m-1}) \rightarrow y_m$
  - Stacking

h_2

X_1	X_2	…	X_n	Y_1	Y_2	…	Y_m
0.12	1	…	12	0	1	…	1
2.34	9	…	-5	0	1	…	0
1.22	3	…	40	1	0	…	0
2.18	2	…	8	?	?	…	?
1.76	7	…	23	?	?	…	?

Single-target Decomposition Techniques

- The simplest one is Binary Relevance: $h(\boldsymbol{x}) \rightarrow \boldsymbol{y} \overset{BR}{\Rightarrow} h_i(\boldsymbol{x}) \rightarrow y_i, i = 1, \ldots, m$
- Better ones (considering label dependencies):
  - Classifier Chains: $h(\boldsymbol{x}) \rightarrow \boldsymbol{y} \overset{CC}{\Rightarrow} h_1(\boldsymbol{x}) \rightarrow y_1, h_2(\boldsymbol{x}y_1) \rightarrow y_2, \ldots, h_m(\boldsymbol{x}y_1 \ldots y_{m-1}) \rightarrow y_m$
  - Stacking

h_m

X_1	X_2	…	X_n	Y_1	Y_2	…	Y_m
0.12	1	…	12	0	1	…	1
2.34	9	…	-5	0	1	…	0
1.22	3	…	40	1	0	…	0
2.18	2	…	8	?	?	…	?
1.76	7	…	23	?	?	…	?

Single-target Decomposition Techniques

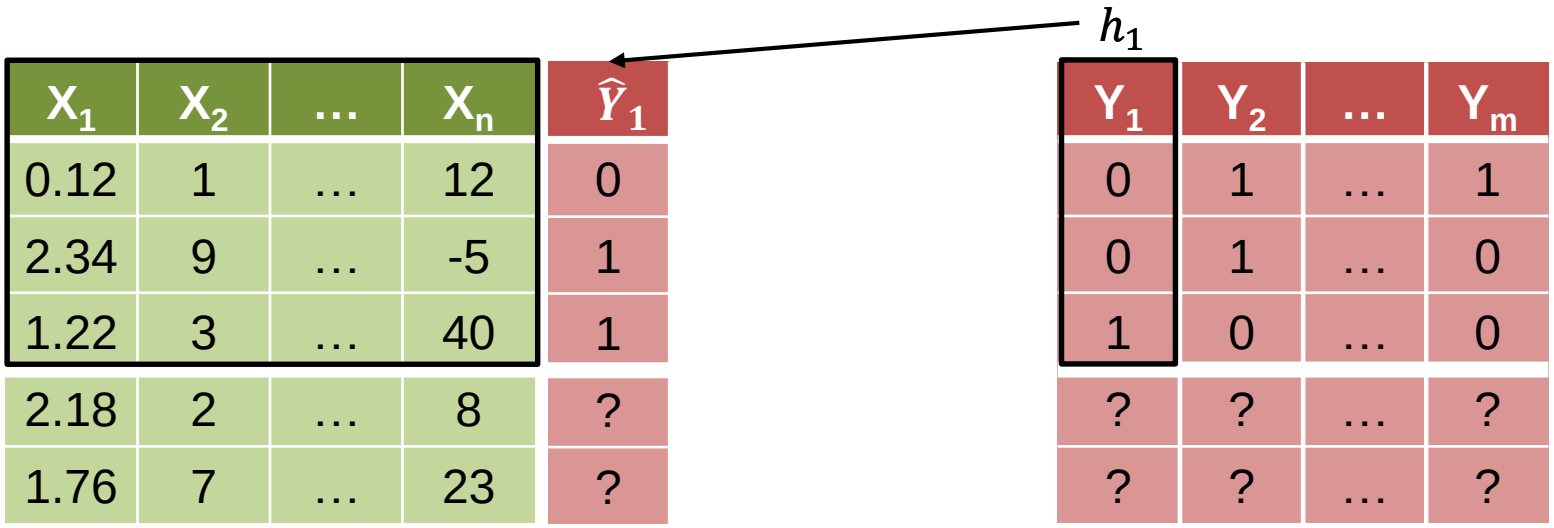
- The simplest one is Binary Relevance: $h(\boldsymbol{x}) \rightarrow \boldsymbol{y} \overset{BR}{\Rightarrow} h_i(\boldsymbol{x}) \rightarrow y_i, i = 1, \ldots, m$
- Better ones (considering label dependencies):
  - Classifier Chains: $h(\boldsymbol{x}) \rightarrow \boldsymbol{y} \overset{CC}{\Rightarrow} h_1(\boldsymbol{x}) \rightarrow y_1, h_2(\boldsymbol{x}y_1) \rightarrow y_2, \ldots, h_m(\boldsymbol{x}y_1 \ldots y_{m-1}) \rightarrow y_m$
  - Stacking: $h(\boldsymbol{x}) \rightarrow \boldsymbol{y} \overset{Stacking}{\Longrightarrow} h'_i(\boldsymbol{x}\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_m) \rightarrow y_i, i = 1, \ldots, m$ where $\hat{y}_1$'s are obtained by applying BR on the training examples

X_1	X_2	…	X_n
0.12	1	…	12
2.34	9	…	-5
1.22	3	…	40
2.18	2	…	8
1.76	7	…	23

Y_1	Y_2	…	Y_m
0	1	…	1
0	1	…	0
1	0	…	0
?	?	…	?
?	?	…	?

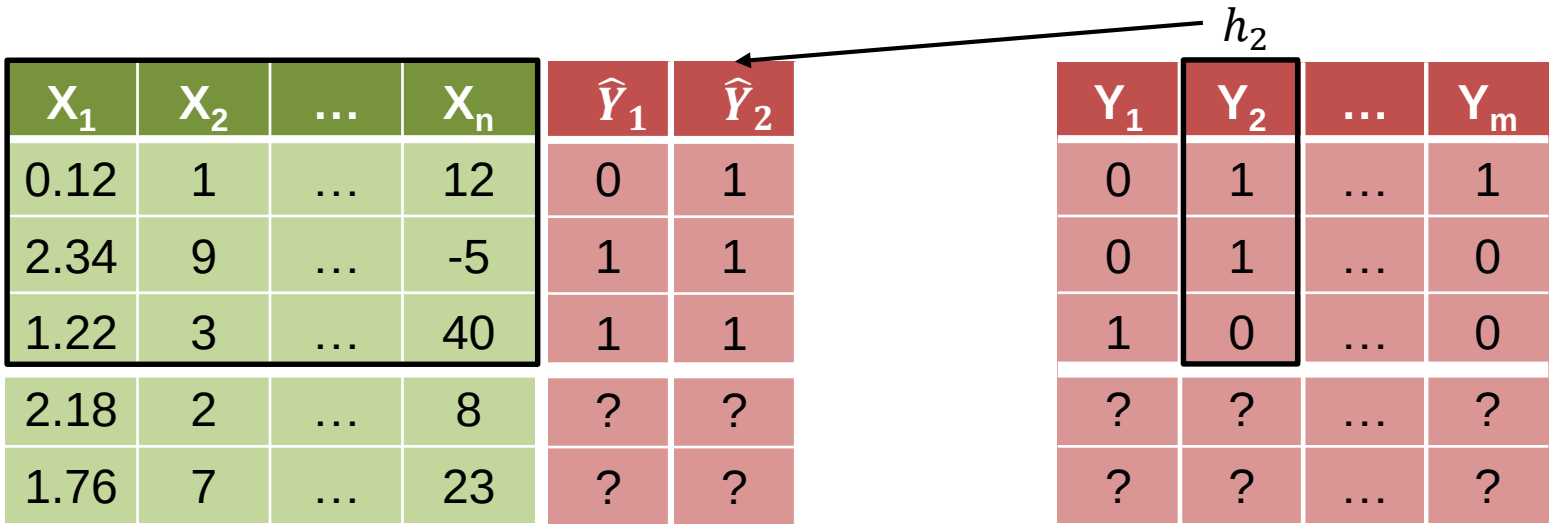
Single-target Decomposition Techniques

- The simplest one is Binary Relevance: $h(\boldsymbol{x}) \rightarrow \boldsymbol{y} \overset{BR}{\Rightarrow} h_i(\boldsymbol{x}) \rightarrow y_i, i = 1, \ldots, m$
- Better ones (considering label dependencies):
  - Classifier Chains: $h(\boldsymbol{x}) \rightarrow \boldsymbol{y} \overset{CC}{\Rightarrow} h_1(\boldsymbol{x}) \rightarrow y_1, h_2(\boldsymbol{x}y_1) \rightarrow y_2, \ldots, h_m(\boldsymbol{x}y_1 \ldots y_{m-1}) \rightarrow y_m$
  - Stacking: $h(\boldsymbol{x}) \rightarrow \boldsymbol{y} \overset{Stacking}{\Longrightarrow} h'_i(\boldsymbol{x}\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_m) \rightarrow y_i, i = 1, \ldots, m$ where $\hat{y}_1$'s are obtained by applying BR on the training examples



Single-target Decomposition Techniques

- The simplest one is Binary Relevance: $h(\boldsymbol{x}) \rightarrow \boldsymbol{y} \overset{BR}{\Rightarrow} h_i(\boldsymbol{x}) \rightarrow y_i, i = 1, \ldots, m$
- Better ones (considering label dependencies):
  - Classifier Chains: $h(\boldsymbol{x}) \rightarrow \boldsymbol{y} \overset{CC}{\Rightarrow} h_1(\boldsymbol{x}) \rightarrow y_1, h_2(\boldsymbol{x}y_1) \rightarrow y_2, \ldots, h_m(\boldsymbol{x}y_1 \ldots y_{m-1}) \rightarrow y_m$
  - Stacking: $h(\boldsymbol{x}) \rightarrow \boldsymbol{y} \overset{Stacking}{\Longrightarrow} h'_i(\boldsymbol{x}\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_m) \rightarrow y_i, i = 1, \ldots, m$ where $\hat{y}_1$'s are obtained by applying BR on the training examples



Single-target Decomposition Techniques

- The simplest one is Binary Relevance: $h(\boldsymbol{x}) \rightarrow \boldsymbol{y} \overset{BR}{\Rightarrow} h_i(\boldsymbol{x}) \rightarrow y_i, i = 1, \ldots, m$
- Better ones (considering label dependencies):
  - Classifier Chains: $h(\boldsymbol{x}) \rightarrow \boldsymbol{y} \overset{CC}{\Rightarrow} h_1(\boldsymbol{x}) \rightarrow y_1, h_2(\boldsymbol{x}y_1) \rightarrow y_2, \ldots, h_m(\boldsymbol{x}y_1 \ldots y_{m-1}) \rightarrow y_m$
  - Stacking: $h(\boldsymbol{x}) \rightarrow \boldsymbol{y} \overset{Stacking}{\Longrightarrow} h'_i(\boldsymbol{x}\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_m) \rightarrow y_i, i = 1, \ldots, m$ where $\hat{y}_1$'s are obtained by applying BR on the training examples

h_m

X_1	X_2	…	X_n	$\hat{Y}_1$	$\hat{Y}_2$	…	$\hat{Y}_m$	Y_1	Y_2	…	Y_m
0.12	1	…	12	0	1	…	1	0	1	…	1
2.34	9	…	-5	1	1	…	0	0	1	…	0
1.22	3	…	40	1	1	…	0	1	0	…	0
2.18	2	…	8	?	?	…	?	?	?	…	?
1.76	7	…	23	?	?	…	?	?	?	…	?

Single-target Decomposition Techniques

- The simplest one is Binary Relevance: $h(\boldsymbol{x}) \rightarrow \boldsymbol{y} \overset{BR}{\Rightarrow} h_i(\boldsymbol{x}) \rightarrow y_i, i = 1, \ldots, m$
- Better ones (considering label dependencies):
  - Classifier Chains: $h(\boldsymbol{x}) \rightarrow \boldsymbol{y} \overset{CC}{\Rightarrow} h_1(\boldsymbol{x}) \rightarrow y_1, h_2(\boldsymbol{x}y_1) \rightarrow y_2, \ldots, h_m(\boldsymbol{x}y_1 \ldots y_{m-1}) \rightarrow y_m$
  - Stacking: $h(\boldsymbol{x}) \rightarrow \boldsymbol{y} \overset{Stacking}{\Longrightarrow} h'_i(\boldsymbol{x}\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_m) \rightarrow y_i, i = 1, \ldots, m$ where $\hat{y}_1$'s are obtained by applying BR on the training examples

optional

								h'_1			
X_1	X_2	…	X_n	$\hat{Y}_1$	$\hat{Y}_2$	…	$\hat{Y}_m$	Y_1	Y_2	…	Y_m
0.12	1	…	12	0	1	…	1	0	1	…	1
2.34	9	…	-5	1	1	…	0	0	1	…	0
1.22	3	…	40	1	1	…	0	1	0	…	0
2.18	2	…	8	?	?	…	?	?	?	…	?
1.76	7	…	23	?	?	…	?	?	?	…	?

Single-target Decomposition Techniques

- The simplest one is Binary Relevance: $h(\boldsymbol{x}) \rightarrow \boldsymbol{y} \overset{BR}{\Rightarrow} h_i(\boldsymbol{x}) \rightarrow y_i, i = 1, \ldots, m$
- Better ones (considering label dependencies):
  - Classifier Chains: $h(\boldsymbol{x}) \rightarrow \boldsymbol{y} \overset{CC}{\Rightarrow} h_1(\boldsymbol{x}) \rightarrow y_1, h_2(\boldsymbol{x}y_1) \rightarrow y_2, \ldots, h_m(\boldsymbol{x}y_1 \ldots y_{m-1}) \rightarrow y_m$
  - Stacking: $h(\boldsymbol{x}) \rightarrow \boldsymbol{y} \overset{Stacking}{\Longrightarrow} h'_i(\boldsymbol{x}\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_m) \rightarrow y_i, i = 1, \ldots, m$ where $\hat{y}_1$'s are obtained by applying BR on the training examples

optional

X_1	X_2	…	X_n	$\hat{Y}_1$	$\hat{Y}_2$	…	$\hat{Y}_m$	Y_1	Y_2	…	Y_m
0.12	1	…	12	0	1	…	1	0	1	…	1
2.34	9	…	-5	1	1	…	0	0	1	…	0
1.22	3	…	40	1	1	…	0	1	0	…	0
2.18	2	…	8	?	?	…	?	?	?	…	?
1.76	7	…	23	?	?	…	?	?	?	…	?

h'_2

Single-target Decomposition Techniques

- The simplest one is Binary Relevance: $h(\boldsymbol{x}) \rightarrow \boldsymbol{y} \overset{BR}{\Rightarrow} h_i(\boldsymbol{x}) \rightarrow y_i, i = 1, \ldots, m$
- Better ones (considering label dependencies):
  - Classifier Chains: $h(\boldsymbol{x}) \rightarrow \boldsymbol{y} \overset{CC}{\Rightarrow} h_1(\boldsymbol{x}) \rightarrow y_1, h_2(\boldsymbol{x}y_1) \rightarrow y_2, \ldots, h_m(\boldsymbol{x}y_1 \ldots y_{m-1}) \rightarrow y_m$
  - Stacking: $h(\boldsymbol{x}) \rightarrow \boldsymbol{y} \overset{Stacking}{\Longrightarrow} h'_i(\boldsymbol{x}\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_m) \rightarrow y_i, i = 1, \ldots, m$ where $\hat{y}_1$'s are obtained by applying BR on the training examples

												h'_m
optional	X_1	X_2	…	X_n	$\hat{Y}_1$	$\hat{Y}_2$	…	$\hat{Y}_m$	Y_1	Y_2	…	Y_m
	0.12	1	…	12	0	1	…	1	0	1	…	1
	2.34	9	…	-5	1	1	…	0	0	1	…	0
	1.22	3	…	40	1	1	…	0	1	0	…	0
	2.18	2	…	8	?	?	…	?	?	?	…	?
	1.76	7	…	23	?	?	…	?	?	?	…	?

Regressor Chains and Stacking

- Similarly to BR, Stacking and CC are directly applicable in MTR simply using a regressor instead of a binary classifier
- The resulting MTR methods:
  - Multi-target Stacking (MTS)
  - Regressor Chains (RC)
- Both Stacking and CC are considered better than BR in MLC, especially for multivariate losses
- Are the MTR equivalents better than doing independent regressions?
- Let's test it…

Experimental Setup

- What about benchmark MTR datasets?
  - Generally scarce (we found only 4 publicly available)
  - We composed 8 new datasets from real-world data (next slide)
- Performance measure
  - A commonly used measure is relative root mean squared error:

$$\text{rrmse} = \frac{\sqrt{\sum_{(x,y)\in D_{test}} \left(\hat{y}_j - y_j\right)^2}}{\sqrt{\sum_{(x,y)\in D_{test}} \left(\bar{Y}_j - y_j\right)^2}}$$

  - If we average over m targets we get: $\text{arrmse} = \frac{1}{\text{m}} \sum_{j=1}^{m} rrmse_j$
- Base regressor
  - There are many options, we picked a strong one: Bagging of 100 regression trees (BRT100)

MTR Benchmark Datasets

	Domain	Name	Examples	Features	Targets
existing	manufacture	EDM	154	16	2
	environment	SF1/SF2	323/1066	10 (d)	3
	environment	WQ	1060	16	14
new[1]	environment	RF1/RF2	9125	64/576	8
	price prediction	ATP1d/ATP7d	337/296	411	6
	price prediction	SCM1d/SCM20d	9803/8966	280/61	16
	artificial	OES97/OES10	334/403	263/298	16

All datasets are available at http://mulan.sourceforge.net

[1]Many thanks to Will Groves from the University of Minnesota for the new datasets!

Empirical Results

- ST (independent regressions) is better in half of the datasets but improvements were possible in the other half
- If we look at individual targets, ST is better only in 46/114 targets
- Not a clear winner between MTS and ERC

Dataset	ST	MTS	ERC
EDM	**74,21**	74,30	74,35
SF1	113,54	112,70	**105,01**
SF2	114,94	**94,48**	105,32
WQ	**90,83**	91,10	90,97
OES97	**52,48**	52,59	52,54
OES10	**42,00**	42,01	42,02
ATP1d	37,35	37,16	**37,10**
ATP7d	52,48	**51,43**	53,43
SCM1d	47,75	47,41	**47,09**
SCM20d	77,68	78,62	**77,55**
RF1	**69,63**	82,37	79,47
RF2	**69,64**	81,75	79,61

Discussion

- Despite improvements ST still seems too strong…
- The addition of meta-variables seems to hurt performance in some cases!
- Explanation
  - Not all targets mutually dependent → irrelevant features are added
- Questions
  - Shouldn't trees do better at ignoring irrelevant attributes?
  - Are there other factors that degrade performance compared to ST?

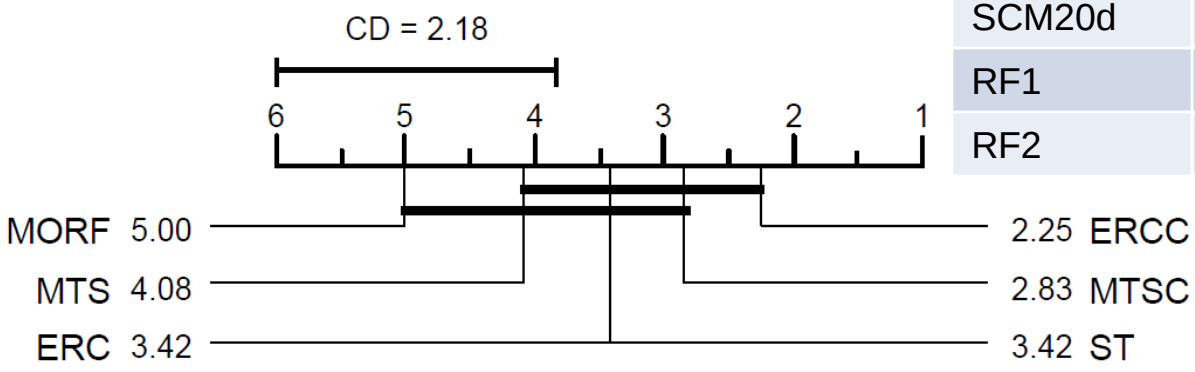
ERC and MTS reconsidered

- "Target- or meta-variables" different from ordinary input variables
  - Values known during training – unknown during prediction
- At prediction time, both methods have to rely on estimates!
- But what values to use at training time?
  - ERC uses the available true values
  - MTS uses in-sample estimates obtained by ST models
- A core assumption of supervised learning is violated in both cases: Train and test data should be IID!
- Consequence: True dependency of "target-variables" with the prediction target can be falsely estimated!
- Proposed solution: Use CV estimates during training
- Assumption: Distribution of CV estimates closer to the distribution observed at prediction time

Do CV Estimates Work?

- ST is better only in 2 datasets
- If we look at individual targets, ST is better in 33/114 targets

Dataset	ST	MTS	ERC
EDM	74,21	**73,96**	74,07
SF1	113,54	**106,80**	108,87
SF2	114,94	**105,53**	108,79
WQ	90,83	90,95	**90,59**
OES97	52,48	52,43	**52,39**
OES10	42,00	42,05	**41,99**
ATP1d	37,35	**37,17**	37,24
ATP7d	52,48	**50,74**	51,24
SCM1d	47,75	47,01	**46,63**
SCM20d	77,68	78,54	**75,97**
RF1	**69,63**	69,82	69,89
RF2	**69,64**	69,86	69,82

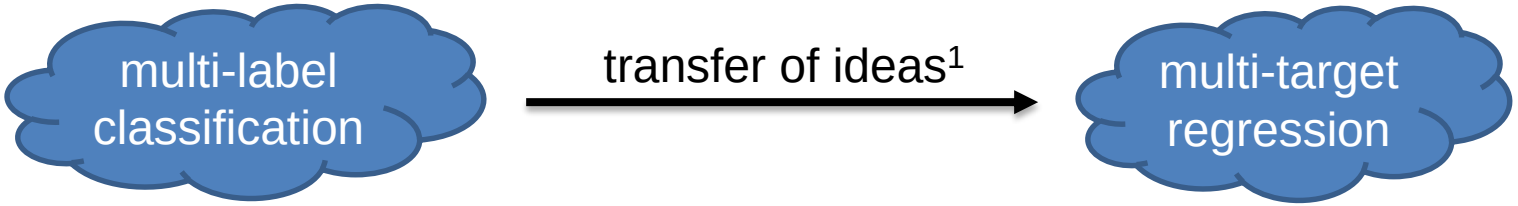


(a) Per dataset analysis.

Some Considerations on Losses

- In our evaluations we followed an individual target view
  - The goal was to improve the performance on each Y_i using $\boldsymbol{X}$ and information about other targets Y_j
  - A univariate loss (arrmse is decomposable)
- What about multivariate losses?
  - Theoretically, methods such us MTS and ERC that try to model label dependencies would perform even better compared to ST!
- What is the equivalent of multivariate MLC losses in MTR?
  - e.g. What is an analogous to subset 0/1 loss $l_{0/1}(\boldsymbol{y}, \hat{\boldsymbol{y}}) = [\![\boldsymbol{y} \neq \hat{\boldsymbol{y}}]\!]$?
- Motivating example:
  - Predict sales for products (e.g. pastries) with short expiration dates
  - Perhaps minimizing $\mathrm{rmse}_{max}(\boldsymbol{y}, \hat{\boldsymbol{y}}) = \max_{i=1\ldots m} \sqrt{(\hat{y}_i - y_i)^2}$ is more appropriate than minimizing $\mathrm{armse}(\boldsymbol{y}, \hat{\boldsymbol{y}}) = \frac{1}{m} \sum_{i=1}^{m} \sqrt{(\hat{y}_i - y_i)^2}$ in order to avoid an early run-out of any product

Modelling sets of labels in MTR?



RAkEL

random subset of labels

all combinations of binary label values

RLC[10]

random subset of targets

a random linear combination of targets

For the details of RLC please wait until Greg's talk on Thursday!

Multi-target Extension of Mulan

- MTR methods
  - Problem transformation
    - ST, MTS, RC, ERC, RLC
  - Algorithm adaptation
    - A wrapper of the CLUS library (e.g Multi-objective Bagging, Multi-objective Random Forest, FIRE, etc.)
- Evaluation framework
  - Supports cv and train/test evaluation
  - Several evaluation measures:
    - armse, arrmse, amae, armae,…easy to add more
- A multitude of base regressors from Weka!
- Available at http://mulan.sourceforge.net

Conclusions and Future Work

- Take-away messages
  - The knowledge transfer was successful!
  - The performance of ST can be improved by carefully exploiting information from other targets even if the case of univariate losses!
    - Explanation: Other targets act as extra features whose values are missing at prediction time!
- Future work
  - Comparison of the proposed methods with ST under non-decomposable loss functions
  - Which method/variant to prefer given dataset characteristics?
  - Test our cv extension on CC (and PCC!) and Multi-label Stacking

THANK YOU!
QUESTIONS?

If you are interested contact us at:
espyromi@csd.auth.gr and greg@csd.auth.gr

References

1. E. Spyromitros-Xioufis, G. Tsoumakas, W. Groves, I. Vlahavas. Multi-Label Classification Methods for Multi-Target Regression. ArXiv. 2014.
2. S. Godbole, S. Sarawagi. Discriminative methods for multi-labeled classification. Proc. PAKDD. 2004.
3. W. Cheng, E. Hüllermeier. Combining instance-based learning and logistic regression for multilabel classification. Machine Learning. 2009
4. J. Read, B. Pfahringer, G. Holmes, and E. Frank. Classifier chains for multi-label classification. Proc. ECML/PKDD. 2009.
5. W. Cheng, E. Hüllermeier, K. Dembczynski. Bayes optimal multilabel classification via probabilistic classifier chains. Proc. ICML. 2010
6. J. Fürnkranz, E. Hüllermeier, E. L. Mencía, & K. Brinker. Multilabel classification via calibrated label ranking. Machine Learning, 2008.
7. G. Tsoumakas, I. Katakis, I. Vlahavas. Random k-labelsets for multilabel classification. Knowledge and Data Engineering. 2011
8. J. Read, B. Pfahringer, G. Holmes. Multi-label classification using ensembles of pruned sets. Proc. ICDM. 2008.
9. H. Blockeel, L. De Raedt, & J. Ramon. Top-down induction of clustering trees. Proc. ICML. 1998.
10. G. Tsoumakas, E. Spyromitros-Xioufis, A. Vrekou, I. Vlahavas. Multi-Target Regression via Random Linear Target Combinations. Proc. ECML/PKDD. 2014.