Enhancing Inductive Programming by Function Ranking

A Machine Learning Application for Data Wrangling Automation

Lidia Contreras-Ochando, Cèsar Ferri, José Hernández-Orallo, Susumu Katayama, Fernando Martínez-Plumed, María José Ramírez-Quintana

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Introduction

Ways Artificial Intelligence systems learn (by *data-intensive* they are):

- **Data-driven bottom-up** → Appropriate from large volumes of data.:
- **Theory-driven top-down** → Better for learning from a few examples.

THEORY-DRIVEN TOP-DOWN SYSTEMS

- Specially useful if information about the domain can be incorporated as **background knowledge (BK)**.
- **Scalability issues** because of a combinatorial explosion of the hypothesis space.
- **Inductive Programming (IP)** [1] is a clear example of this family of techniques.

EXAMPLE

Id	Input	Output
1	25-03-74	25
2	03/29/86	29
3	1998/12/25	25
4	…	…

Table 1. Example of a dataset with an input column of dates in very different formats and the output where the day has been extracted.

We consider IP for the automation of ***data wrangling*** problems (Data preparation tasks for transforming their raw format to a structured and valuable form), as the example in Table 1. This problem can be solved by computers if they recognise (1) they are handling dates and (2) have a sufficiently rich set of functions to deal with dates. This size of the BK in terms of number of functions is known as ***breadth (b)***, the minimum number of functions that have to be combined in the solution is known as the ***depth (d)***. Hardness depends on *d* and *b*, in a way that is usually exponential, *O(b^d)* [2].

How can we keep both, and especially b, at very low levels?

Goal

(Semi) Automation of data wrangling tasks, controlling the depth and breadth of the inductive inference by choosing a domain-specific background knowledge (DSBK) for the problem and selecting the right primitives from it in theory-driven learning.

Id	Station	Date	Output
1	001	6-10-16 20:35	2016
...	...	...	...
69851	001	06/10/2016 00:25:45	2016
4	001	06/10/2016 00:18:36	2016

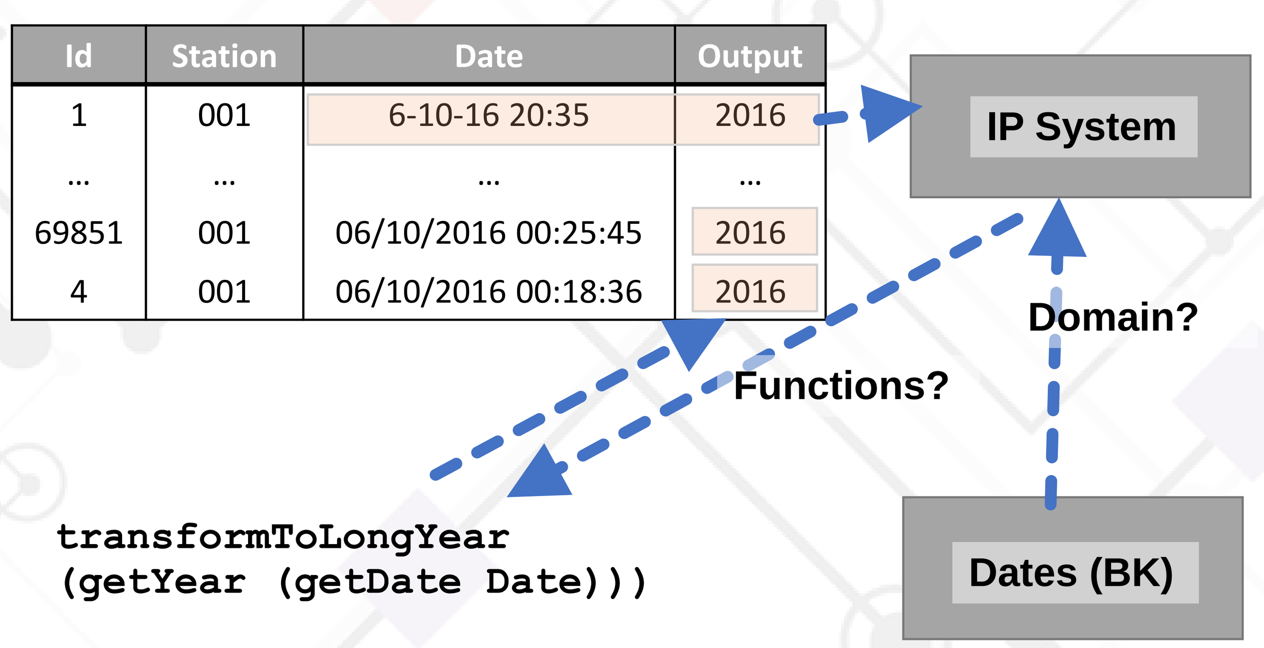

Figure 1. Overall idea for automating data wrangling with an IP system. The first row (Data and Output) is used as a input predicate for the IP system. The functions returned using the correct domain are applied to the rest of the instances (Date) to fill the rest of the outputs.

Experiments

DOMAINS

- **Dates** (222 functions)
- **Emails** (207 functions)
- **Names** (215 functions)
- **Phones** (227 functions)
- **Times** (239 functions)
- **Units** (213 functions)

DATA

Data (datasets of data wrangling problems):

- Training: 124 datasets.
- Test: 33 datasets.

Metafeatures: 54 descriptive characteristics of the problems.

IP Learning System: MagicHaskeller [3]

domain	input	output
dates-7	*10 12 69* 04/05/99 31/03/75	*10-12-69* 04-05-99 31-03-75
emails-4	*Nancy.FreeHafer@fourthcoffee.com* Andrew.Cenici@northwindtraders.com Laura.Giussani@adventure-works.com	*fourthcoffee.com* northwindtraders.com adventure-works.com
names-5	*Dr. B. Schdur* Prof. R. G. H Laabertink H. Huifen, PhD	*Dr.* Prof. PhD
phones-1	*3237087700* 1635879240 1854379620	*323-708-7700* 163-587-9240 185-437-9620
times-16	*1:34:00 PM CST* 01:55 08:40 UTC	*34* 55 40
units-5	*56.77cl* 84Kg 87 s	*Volume* Mass Time

Table 2. Some examples from the 33 datasets used for testing. The first row of each dataset Is the example given to the system to learn.

METHODOLOGY

Strategies:

1. ***Default (baseline)***: Using the default BK.
2. ***Global (baseline)***: BK composed by all the domains.
3. ***User Domain (reference): Using*** the correct DSBK.
4. ***Ranking (4)***: Ranking all the functions of the global BK.
5. ***Inferred Domain (3) + Ranking (4)***: Ranking functions of the DSBK predicted by the domain classifier.

(1)Take the first example from a dataset

Id	Input	Output
1	Damian Gobbee	D.Gobbee
2	Damancio Hivser-Kleiner	D.Hivser-Kleiner
3	Prof. Edward Davis	E.Davis
4	...	...

(2) Extract its metafeatures

start_upper	end_lower	has_blanks	has_numbers	has_at	Has_dot	…
1	0	1	0	0	1	…

(3) Detect the domain

Dates	Emails	Names	Phones	Times	Units
0	0,02	0,98	0	0	0

(4) Predict &Score functions

reduceName	reduceSpaces	getTitle	getAfterDot	…
0,99	0,95	0,036	0,024	…

RESULTS

Pred \ Actual	dates	emails	names	phones	times	units	text
dates	321	0	0	0	0	2	7
emails	0	155	1	0	0	0	0
names	0	1	234	0	0	0	0
phones	2	0	0	309	0	0	0
times	45	0	0	15	432	30	30
units	0	0	0	0	0	118	2
text	28	24	35	0	0	30	393
Error	0.19	0.14	0.09	0.13	0.05	0	0.34

Table 3. Confusion matrix of the domain classifier model with 10-fold cross-validation. "text" contains datasets of basic string manipulation problems (as baseline problems).

domain	strategy	avg_time	avg_acc
dates	default	77.85	0.2
	global	63.71	0.56
	user-domain	34.89	0.56
	ranking	1.84	**1**
	infer+rank	**1.78**	**1**
emails	default	88.42	0.16
	global	68.6	0
	user-domain	87,81	0.56
	ranking	**35,55**	0.74
	infer+rank	90,21	**0.8**
names	default	63.12	0.12
	global	2.27	0.92
	user-domain	1.84	**1**
	ranking	1.59	**1**
	infer+rank	**1.52**	**1**
phones	default	61.12	0
	global	79.46	0.32
	user-domain	39.06	0.4
	ranking	2.74	0.8
	infer+rank	**2.26**	**1**
times	default	36.46	0.2
	global	85.78	0.4
	user-domain	50.53	0.4
	ranking	3.96	0.8
	infer+rank	**2.56**	**1**
units	default	67.47	0.33
	global	61.36	0.66
	user-domain	8.16	**1**
	ranking	3.43	**1**
	infer+rank	**2.75**	**1**

Table 4. Average results for the 33 testing datasets by domain using the first example of each dataset as input for the system. "avg_time" is the average induction time (in seconds). "avg_acc" is the average accuracy transforming the rest of instances of the datsets. Best results in bold. Note:A penalisation (in seconds) is applied to the emails domain since the first predicted domain is incorrect in three cases.

Conclusions

We have a system that:

(1) uses off-the-self IP and ML techniques

(2) has short response time

(3) it can work on any device and architecture (as API)

(4) is fully automated

(5) covers a wide range of manipulation problems

(6) is replicable to other domains and systems.

All the datasets are published:
http://users.dsic.upv.es/~flip/datawrangling/

FUTURE WORK

- Study the strategies over other systems.
- Consider the relationships between functions.
- Exploring the use of a hierarchical classifier.

References

[1] Gulwani, S. et al. Inductive programming meets the real world. *Communications of the ACM 2015*.

[2] Henderson, R. Incremental learning in inductive programming. *Workshop on Approaches and Applications of Inductive Programming, 2009.*

[3] Katayama, S. An analytical inductive functional programming system that avoids unintended programs. *Workshop on Partial evaluation and program manipulation, 2012.*.

Acknowledgements

This work has been partially supported by the EU (FEDER) and Spanish MINECO grant TIN2015-69175-C4-1-R, by Generalitat Valenciana PROMETEOII/2015/013, and by INCIBE (Ayudas para la excelencia de los equipos de investigación avanzada en ciberseguridad). L. Contreras-Ochando is supported by FPU-MECD grant REF FPU15/03219.

宮崎大学 University of Miyazaki

DSIC DEPARTAMENT DE SISTEMES INFORMÀTICS I COMPUTACIÓ

ELP GROUP

dmip

DOWNLOAD THE POSTER