Domain specific induction for data wrangling automation

Lidia Contreras Ochando
Universitat Politècnica de València
DSIC
liconoc@upv.es

Susumu Katayama
University of Miyazaki
skata@cs.miyazaki-u.ac.jp

Cèsar Ferri Ramírez
Universitat Politècnica de València
DSIC
cferri@dsic.upv.es

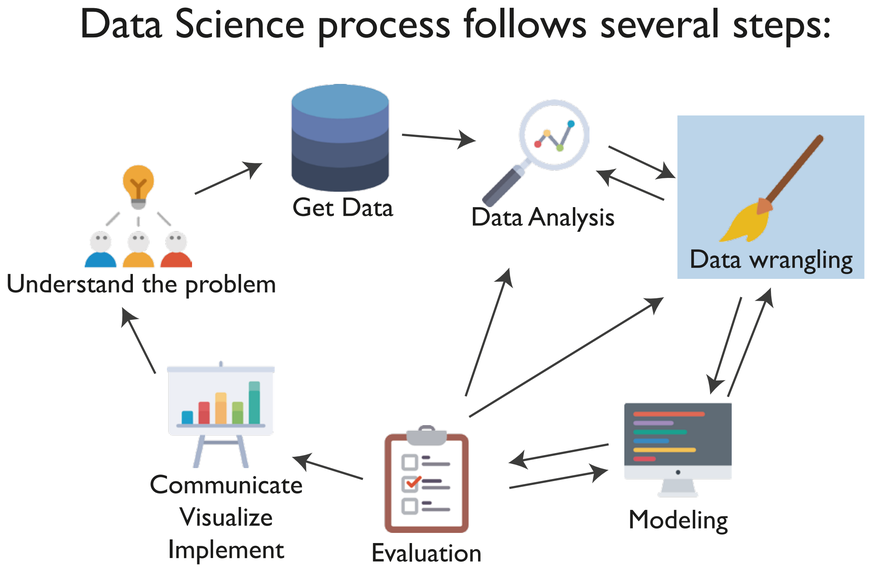
Fernando Martínez Plumed
Universitat Politècnica de València
DSIC
fmartinez@dsic.upv.es

José Hernández Orallo
Universitat Politècnica de València
DSIC
jorallo@dsic.upv.es

María José Ramírez Quintana
Universitat Politècnica de València
DSIC
mramirez@dsic.upv.es

Introduction

Data Science process follows several steps:



Data wrangling

This step includes:

- It is the most tedious, boring and repetitive step
- Spends up to **80%** of the project time[1]

Goal

(Semi-)Automate data wrangling process

Methodology

Inductive Programming[2]

The program receives:
- Some examples
- Background Knowledge

→ The result is a hypothesis on how to obtain new examples by using the knowledge.

MagicHaskeller[3]

MagicHaskeller is an inductive functional programming system that learns Haskell programs from pairs of input-output examples.

MagicHaskeller receives an input example (x) and the expected result (y), and returns a list of functions (f) that makes the values of the expressions fx and y be equal ($f\ x == y$).

We use **MagicHaskeller** as a inductive funcional programming tool:
- We extend its general background knowledge with specific domain functions.
- We have collected or created a set of datasets for data wrangling tests.

Overall idea

input	output
"03/29/86"	
"74-03-31"	
"99/12/13"	
"11.02.96"	
"31/05/17"	
"25/08/85"	
"05 30 85"	

We take a dataset of input-output pairs

input	output
"03/29/86"	"29"
"74-03-31"	"31"
"99/12/13"	
"11.02.96"	
"31/05/17"	
"25/08/85"	
"05 30 85"	

We complete a few examples (n)

f "03/29/86" == "29"
&&
f "74-03-31" == "31"

These examples are used as input predicates for MagicHaskeller

dates.dom → MagicHaskeller

MagicHaskeller uses the specific background knowledge for the domain (that contains b functions) with a maximun of functions concatenated d_{max}

\a -> extractDayCardinal a

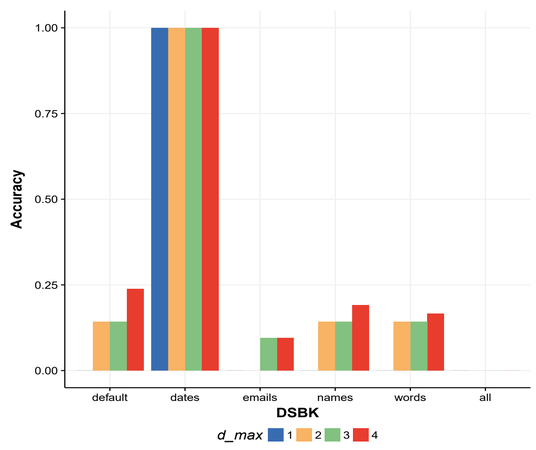
MagicHaskeller returns the result (f) with d functions concatenated

input	output
"03/29/86"	"29"
"74-03-31"	"31"
"99/12/13"	"13"
"11.02.96"	"11"
"31/05/17"	"31"
"25/08/85"	"25"
"05 30 85"	"30"

The function is applied to the rest of the inputs

Results

We have performed a set of experiments to analyse the performance of our approach:
- We analyse the capabilities of MagicHaskeller as a data wrangler by using the right DSBK.
- We compare our results with other data wrangling systems on a range of data wrangling problems.



Results for a dataset of dates when d_{max} goes from 1 to 4. The "right" domain for this dataset is dates.dom.

id dataset	input	output	FlashFill	Trifacta Wrangler	Our Approach
3	*03/29/86*	*29*			
	74-03-31	31	03	03	31
	05 30 85	30	30	30	30
		Accuracy:	**0.16**	**0.16**	**1**
6	*Nancy.FreeHafer@fourthcoffee.com*	*fourthcoffee*			
	Andrew.Cencini@northwindtraders.com	northwindtraders	northwindtraders	northwindtraders	northwindtraders
	Laura.Giussani@adventure-works.com	adventure-works	adventure-works	adventure	adventure-works
		Accuracy:	**1**	**0.93**	**1**
9	*Mr. Roger*	*Male*			
	Mrs. Simona	Female			Female
	Mr. John	Male			Male
		Accuracy:	**0**	**0**	**1**
16	*CAMP DRY DBL NDL 3.6 OZ*	*3.6 OZ*			
	DRY NDL 0.23 KG	0.23 KG	0 KG	0.23 KG	0.23 KG
		Accuracy:	**0**	**1**	**1**

Some illustrative outcomes and accuracy obtained for four datasets with our approach compared with other two tools: Microsoft FlashFill and Trifacta Wrangler. The first instance (in italic) for each dataset (input/output columns) is the one used for inducing the solution in the diferent tools.

Related Work

- **FlashFill**[4]: Tool for automate repeititve string transforms in Excel.
- **Trifacta Wrangler**[5]: Generates suggestions inferred automatically from user input.
- **OpenRefine**[6]: Provides a ser of built-in operators to specify data transformations.

References

1. D. Steinberg. How much time needs to be spent preparing data for analysis?

2. S. Gulwani, J. Hernandez-Orallo, E. Kitzelmann, S. H. Muggleton, U. Schmid, and B. Zorn. Inductive programming meets the real world.

3. S. Katayama. An analytical inductive functional programming system that avoids unintended programs.

4. S. Gulwani. Automating string processing in spreadsheets using input-output examples.

5. S. Kandel, A. Paepcke, J. Hellerstein, and J. Heer. Wrangler: Interactive visual specification of data transformation scripts.

6. K. Ham. Openrefine (version 2.5). http://openrefine. org. free, open-source tool for cleaning and transforming data.

Future Work

- Automate the detection of the domain by using machine learning techniques.
- Integrate in a more standalone tool or web-service in a more usable, standard and accesible format.
- Develop an API allowing its use with any language or tool.

Download the poster: