Data Science
by demonstration

UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Lidia Contreras Ochando
Universitat Politècnica de València
Doctorado en Informática
liconoc@upv.es

Directores:

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

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

Colaboradores:

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

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

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

DSIC DEPARTAMENT DE SISTEMES INFORMÀTICS I COMPUTACIÓ

dmip

Introducción

Un proyecto de ciencia de datos sigue diferentes pasos:

Data wrangling

Este paso del proceso engloba:

- Es el proceso más tedioso, aburrido y repetitivo
- Consume hasta el **80%** del tiempo del proyecto[1]

Objetivo: **(Semi) Automatizar el proceso de data wrangling**

Metodología

Programación inductiva[2]

Relacionado con "máquinas que se programan solas":

El programa recibe:
- Unos pocos ejemplos
- Conocimiento previo

→ El resultado es una hipótesis sobre cómo obtener los nuevos ejemplos a partir del conocimiento previo.

MagicHaskeller[3]

MagicHaskeller es un sistema de programación funcional inductiva que ayuda con la programación en Haskell, automatizando la inducción de funciones a partir de ejemplos.

MagicHaskeller recibe un ejemplo de partida (a) y el resultado esperado (b), y devuelve la lista de funciones (f) que cumplen que:

$$f(a) \rightarrow b$$

```
f "3/29/86" == "03/29/86"
the predicate is f "3/29/86" == "03/29/86"


append zeroChar


\a -> concat (prepend_x (splitStringWithPunctuation a) zeroChar)
\a -> concat (words (append zeroChar a))
\a -> append zeroChar (changePunctuationString a slashChar)

\a -> takeOneOfArray (words (append zeroChar a)) a
\a -> takeOneOfArray (words (append zeroChar a)) zChar
\a -> takeOneOfArray (words (append zeroChar a)) xChar
\a -> takeOneOfArray (words (append zeroChar a)) yChar

\f -> ?
```

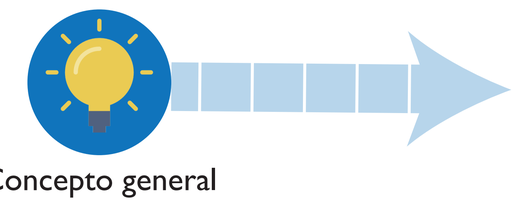
Captura de pantalla de una ejecución de MagicHaskeller

Utilizamos **MagicHaskeller** como herramienta inductiva:

- Ampliamos y especializamos su conocimiento previo de propósito general
- Añadimos nuevas funciones específicas para data wrangling

Trabajos relacionados:

- **FlashFill**[4]**:** Herramienta para automatizar transformaciones repetitivas sobre textos, en Excel.
- **FlashExtract**[5]**:** Extrae datos estructurados a partir de archivos de texto y páginas web.
- **FlashRelate**[6]**:** Extrae datos estructurados a partir de hojas de calculo.



Concepto general

Dates	Months
29/03/86	
12 May 2005	
1995-3-31	
Juny, 15th	
08/12/75	
...	...
30.06.1990	
September	

1. Se parte de un conjunto de datos a limpiar

→

Dates	Months
29/03/86	03
12 May 2005	05
1995-3-31	03
Juny, 15th	
08/12/75	
...	...
30.06.1990	
September	

2. Se completan uno o varios ejemplos

→

f "29/03/86" == "03"
&
f "12 May 20015" == 05
&
f "1995-3-31" == "03"

3. Se envían los ejemplos a MagicHaskeller

→

\a -> extractMonth a

4. MagicHaskeller devuelve la función común

→

Dates	Months
29/03/86	03
12 May 2005	05
1995-3-31	03
Juny, 15th	06
08/12/75	12
...	...
30.06.1990	06
September	09

5. Aplicamos la función al resto de datos

Primeros resultados

Ejemplos realizados con datos relacionados con fechas:

Problema	Ejemplo de partida (a)	Solución	Solución aplicada a nuevos ejemplos
Cambiar signo de puntuación	f "29/03/86" == "29-03-86"	changePunctuationString a dash	f ("25.6.93") = "25-6-93" f ("3/June/90") = "3-June-90"
Cambiar formato de fecha	f "03/29/86" == "29/03/86"	concat (exchange (splitStringWithPunctuation a) xy)	f ("25.6.93") = "6.25.93" f ("3/June/90") = "June/3/90"
Extraer día en formato ordinal	f "03/29/86" == "29th"	extractDayOrdinal a	f ("25.6.93") = "25th" f ("3/June/90") = "3rd"
Extraer día de la semana	f "Sunday, 9 November 2014" == "Sunday"	extractWeekDay a	f ("2 of September of 2010, Monday") = "Monday" f ("Tuesday, 3rd") = "Tuesday"
Convertir mes	f "March" == "03"	convertMonth a	f ("December") = "12" f ("11") = "November"
Extraer año corto y hacerlo largo	f "03/29/86" == "1986"	append nineteenChar (extractFromString a z)	f ("3-6-99") = "1999" f ("9.November.90") = "1990"

Trabajo Futuro:

- Ampliar los campos (datos personales, siglas, teléfonos, emails...)
- Añadir más interacción con el usuario (recomendaciones)
- Crear herramienta o servicio web que permita usar con datasets reales

Referencias

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. FlashFill: S. Gulwani. Automating string processing in spreadsheets using input-output examples
5. FlashExtract: V. Le and S. Gulwani. Flashextract: A framework for data extraction by examples
6. FlashRelate: D. W. Barowy, S. Gulwani, T. Hart, and B. G. Zorn. Flashrelate: extracting relational data from semistructured spreadsheets using examples

Descarga el póster: