SATRDAY

Find your dream job

- A Shiny app for job matching

Paris
February 23, 2019

Highlighting HR data

Resumes
Job descriptions

administrative data
HR IS

career interview data
soft skills

Highlighting HR data

Application : « Système d'information des cadres supérieurs – SICS »

- Search : Resumes & job descriptions
- Visualize agents' progress
- Recommendation « basket competencies »
- Career management
- …
- POC TINDER

POC TINDER

a fun game to find a dream job

input

PDF

450 job descriptions

output

Fun app to find 6 dream job descriptions

POC TINDER

a fun game to find a dream job

input

PDF

450 job descriptions

Model & process a user experience

Design & code app

output

Fun app to find 6 dream job descriptions

MODEL & ALGO

TEXT PREPROCESSING

STEP 1 : define terms

input

450 job descriptions

- Tokenize sentences into words
- Stem words : e. g. studied --> study
- Remove stopwords : the, and, if, in…
- Use n-grams
- Remove infrequent words
- Tf-idf

- text2vec
- stringi
- data.table

output

	t_1	t_2	…	t_m
f_1	p_{11}	p_{12}		p_{1m}
f_2	p_{21}	p_{22}		p_{2m}
…				
f_n	p_{n1}	p_{n2}		p_{nm}

Matrix
n job descriptions x m terms

CLUSTERING

STEP 2 : define clusters

input

	t_1	t_2	...	t_m
f_1	p_{11}	p_{12}		p_{1m}
f_2	p_{21}	p_{22}		p_{2m}
...				
f_n	p_{n1}	p_{n2}		p_{nm}

Matrix
n job descriptions x m terms

- K-means

or

- LDA : Latent Dirichlet Allocation

N.B :

- lda
- LDAvis
- stats

output

	C_1	C_2	...	C_k
t_1	P_{11}	P_{12}		P_{1k}
t_2	P_{21}	P_{22}		P_{2k}
...				
t_m	P_{m1}	P_{m2}		P_{mk}

Matrix
m terms x k clusters

CLUSTERING

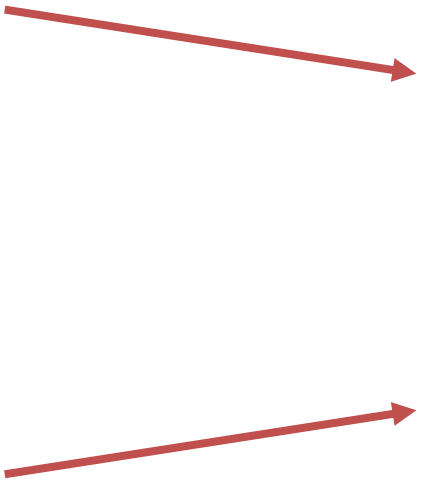
STEP 2 : define clusters

Matrix
n job descriptions x m terms

	t_1	t_2	...	t_m
f_1	p_{11}	p_{12}		p_{1m}
f_2	p_{21}	p_{22}		p_{2m}
...				
f_n	p_{n1}	p_{n2}		p_{nm}

Matrix
m terms x k clusters

	C_1	C_2	...	C_k
t_1	P_{11}	P_{12}		P_{1k}
t_2	P_{21}	P_{22}		P_{2k}
...				
t_m	P_{m1}	P_{m2}		P_{mk}



Matrix
n job descriptions x k clusters

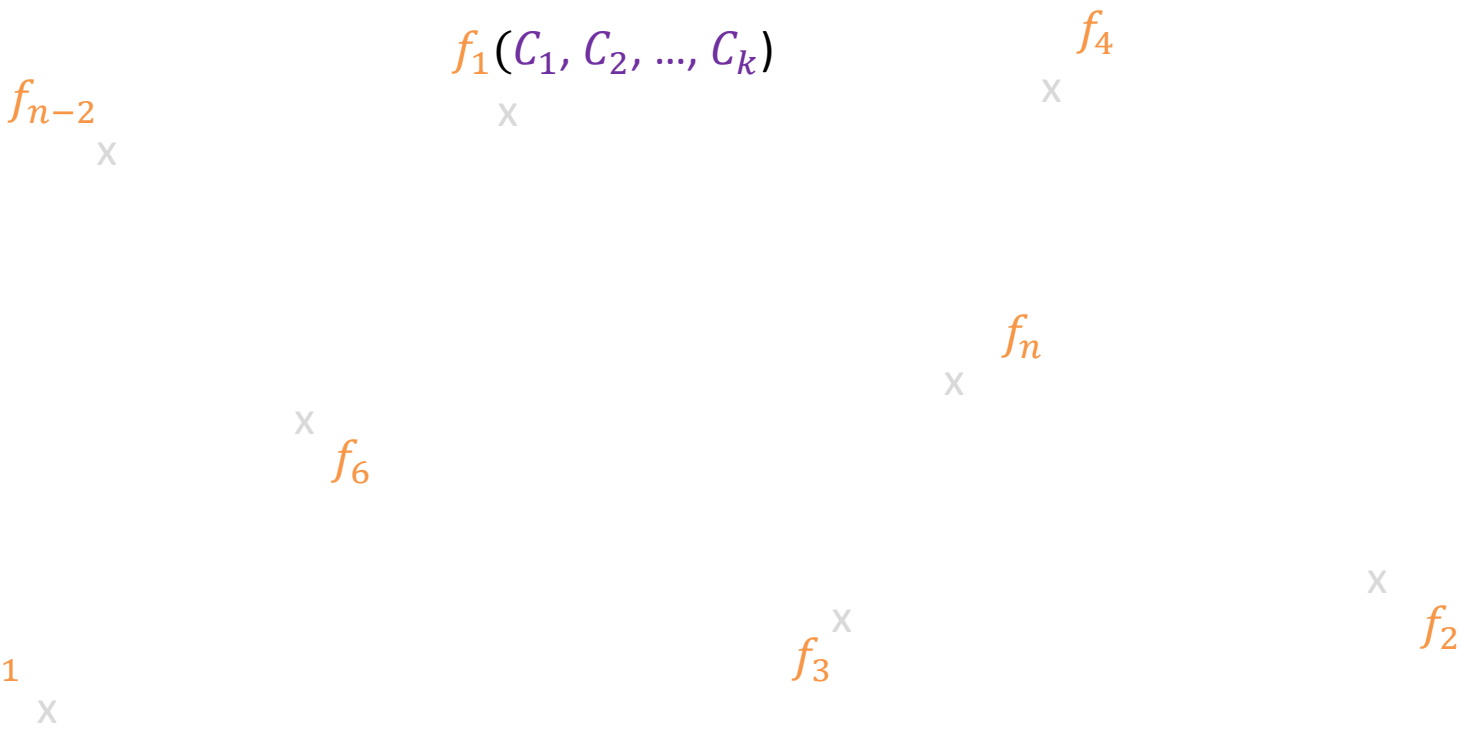
	C_1	C_2	...	C_k
f_1	W_{11}	W_{12}		W_{1k}
f_2	W_{21}	W_{22}		W_{2k}
...				
f_n	W_{n1}	W_{n2}		W_{nk}

Each job description is defined by a combination of clusters

USER PROCESS

Moving in a space of job descriptions

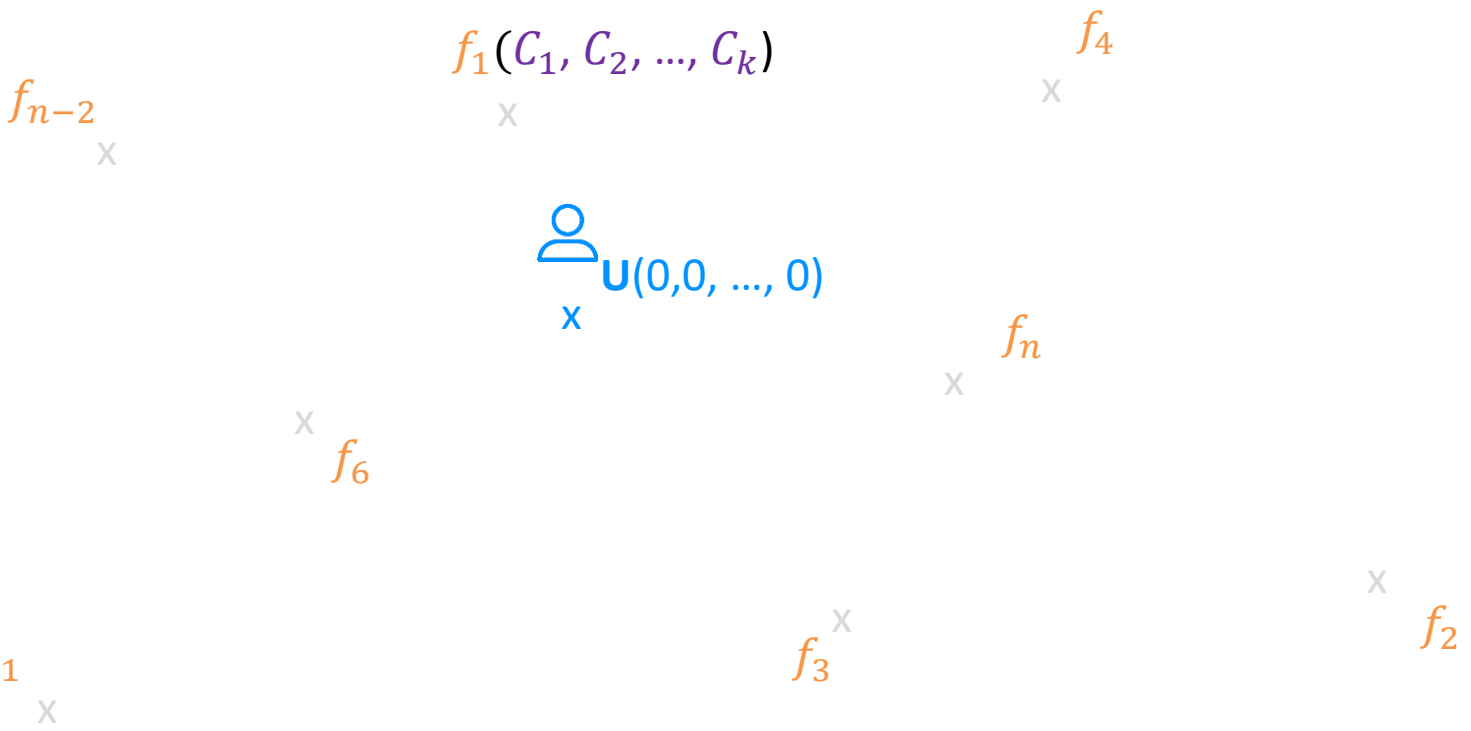
Job descriptions are part of a k-dimensional space:



USER PROCESS

Moving in a space of job descriptions

The user U is seen as another point in this space:



USER PROCESS

Moving in a space of job descriptions

Module 1 :
Tags selection

Module 2 :
Job card selection

Module 3 :
Job card validation

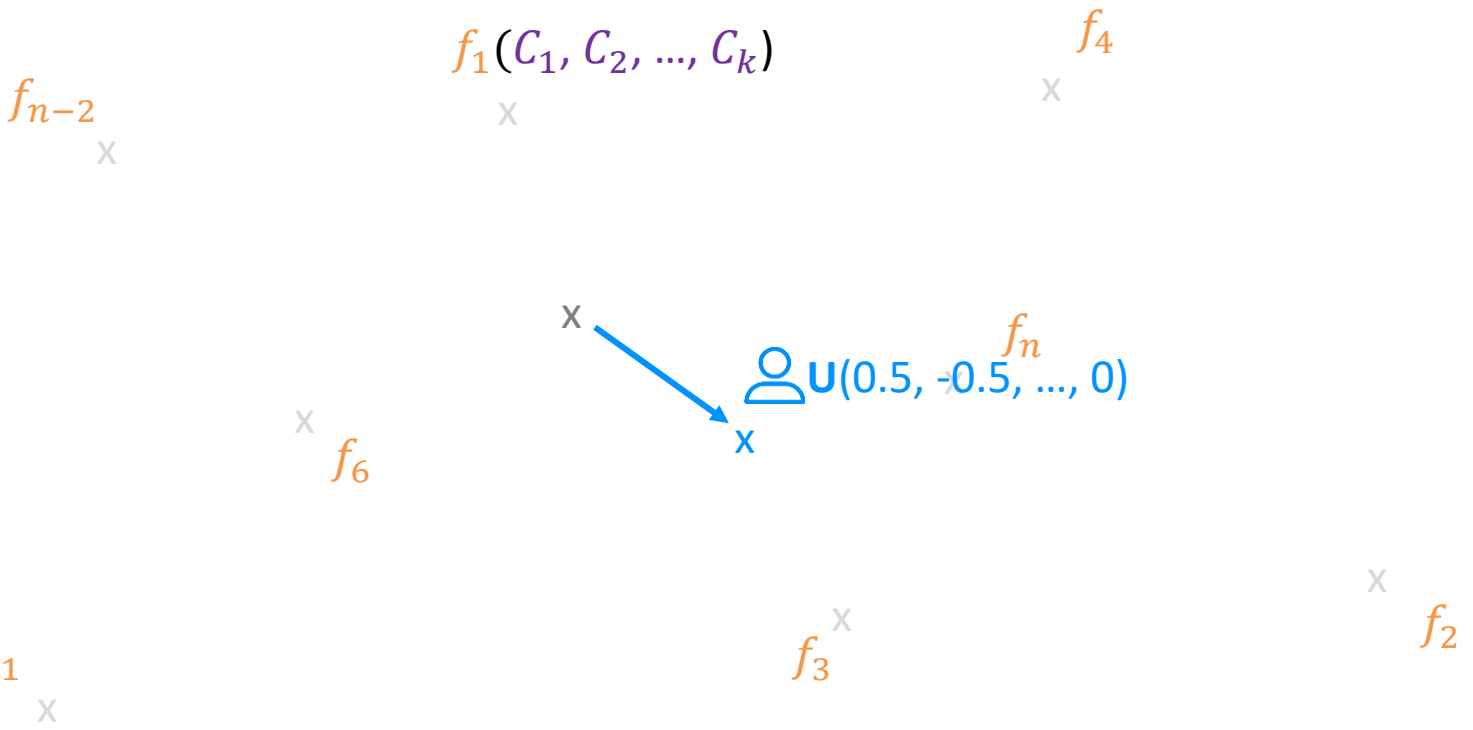
Module 4 :
Job card restitution

Step 1 :
the **user** is in the centre of the space

USER PROCESS

Moving in a space of job descriptions

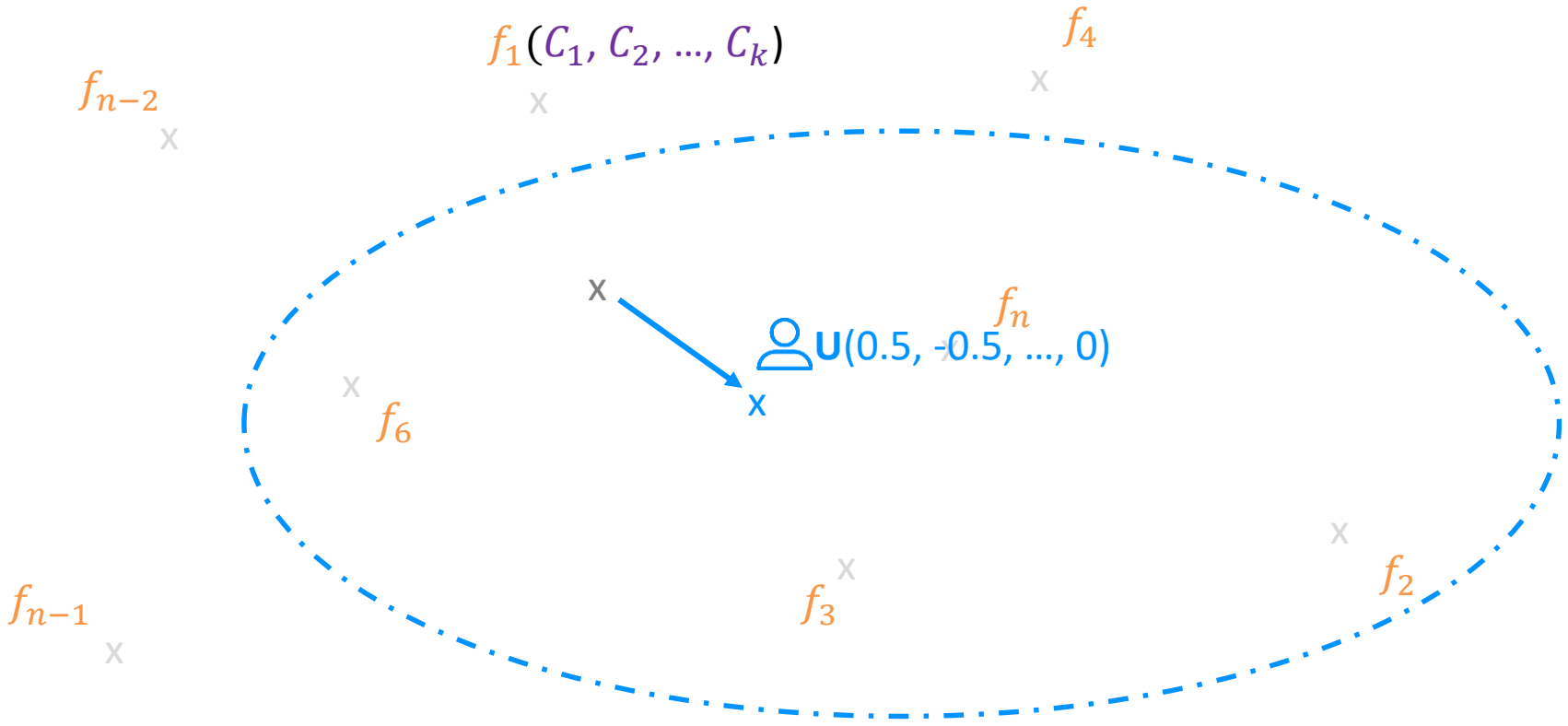
New position of the user after his choices in the first module:



USER PROCESS

Moving in a space of job descriptions

New position of the user after his choices in the first module:



USER PROCESS

Moving in a space of job descriptions

Module 1 :
Tags selection

Module 2 :
Job card selection

Module 3 :
Job card validation

Module 4 :
Job card restitution

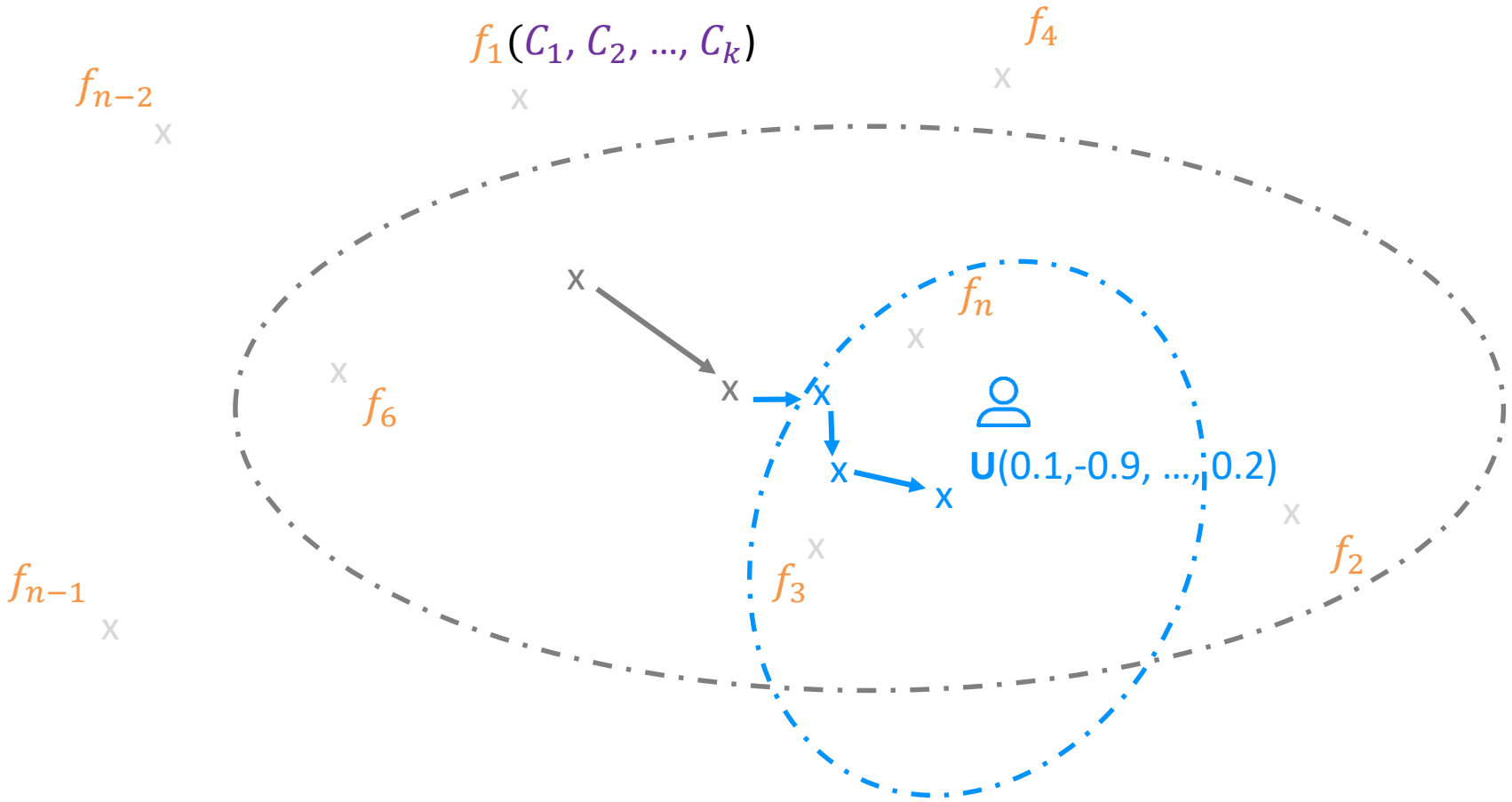
Step 1 :
the **user** is in the centre of the space

Step 2 :
the **user** moves in the space
the space is reduced

USER PROCESS

Moving in a space of job descriptions

At each step, the user's profile is recomputed through a weighted mean:



USER PROCESS

Moving in a space of job descriptions

Module 1 :
Tags selection

Module 2 :
Job card selection

Module 3 :
Job card validation

Module 4 :
Job card restitution

Step 1 :
the **user** is in the centre of the space

Step 2 :
the **user** moves in the space
the space is reduced

Step 3 :
the **user** moves in the space
the space is reduced

USER PROCESS

Moving in a space of job descriptions

Module 1 :
Tags selection

Module 2 :
Job card selection

Module 3 :
Job card validation

Module 4 :
Job card restitution

Step 1 :
the **user** is in the centre of the space

Step 2 :
the **user** moves in the space
the space is reduced

Step 3 :
the **user** moves in the space
the space is reduced

Step 4 :
the **user** moves in the space
6 dream job descriptions

SHINY APP

SHINY APP

main features

Shiny modules

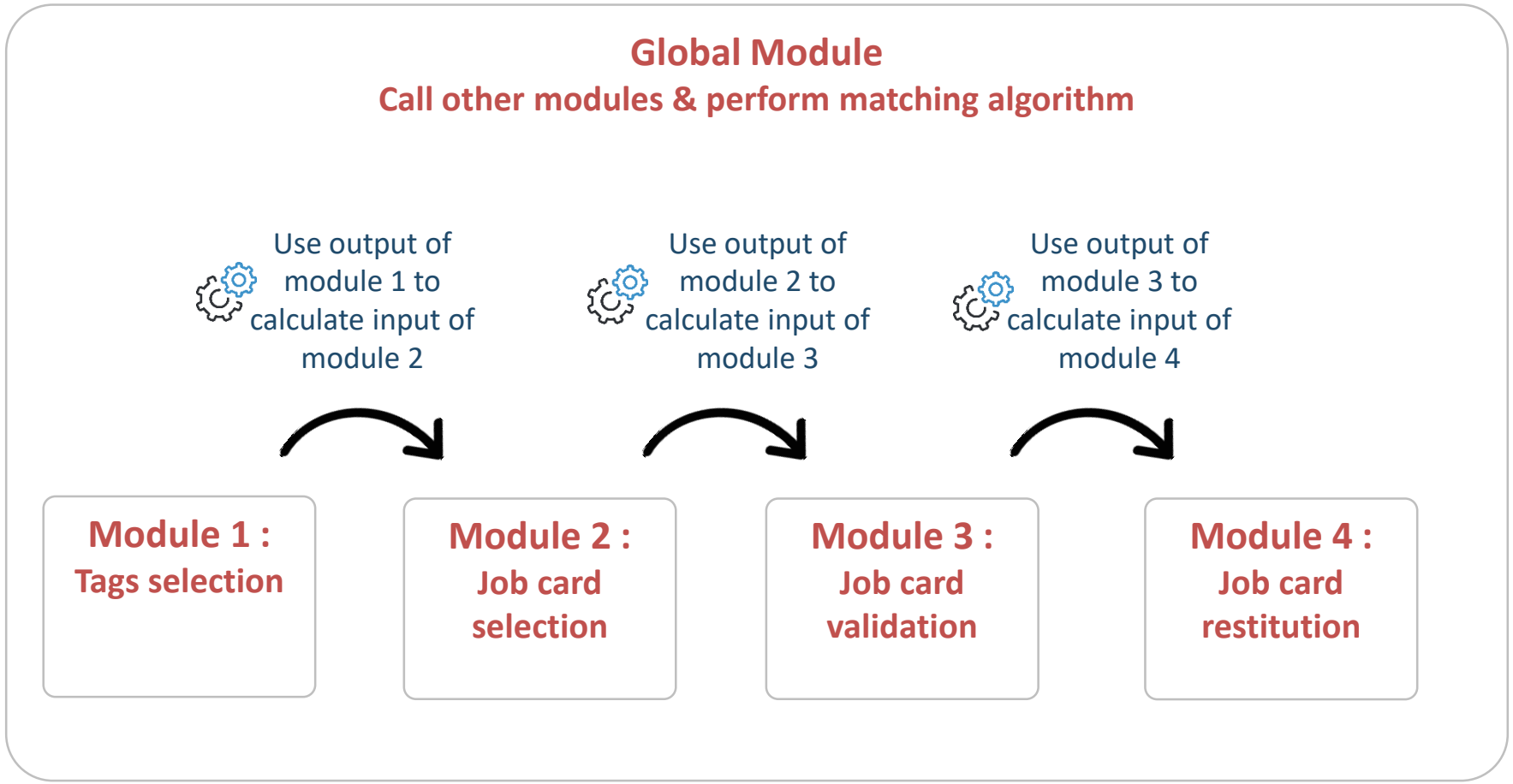
Integrated in an R package

Persistent data storage

User tracking

SHINY APP

architecture



MODULES

Why use Shiny modules ?

Breaking code in smaller blocks

Avoid big UI & Server with thousands of lines of code

Function & documentation

Modules are R functions, so can be documented in an R package

Re-usable

Modules can be reused within the same application or in a new one

Readability

Modules simplify the architecture of applications by using them as independent components

MODULES

Design

input

What does the module need to work?

Module

UI

Server

output

What is the output of the module?

ReactiveValues

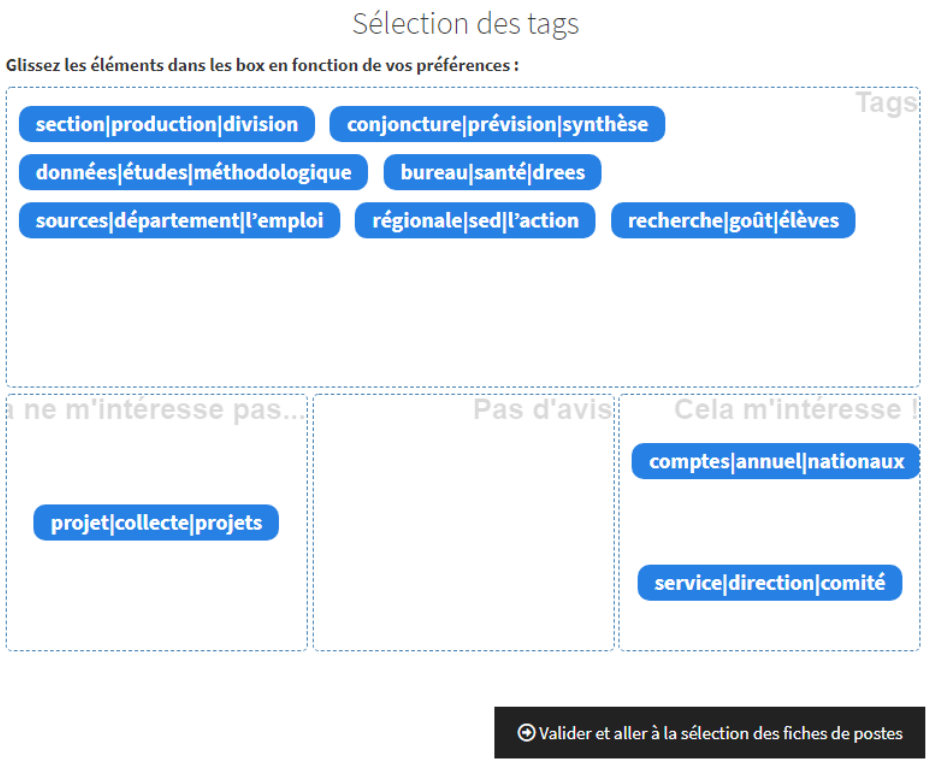
MODULES

Our four modules

input

A vector of tags to be displayed

Module Sélection des Tags



output

A reactiveValues with user's choices

MODULES

Our four modules

input

A reactive function returning a table with job descriptions

Module Sélection des Fiches

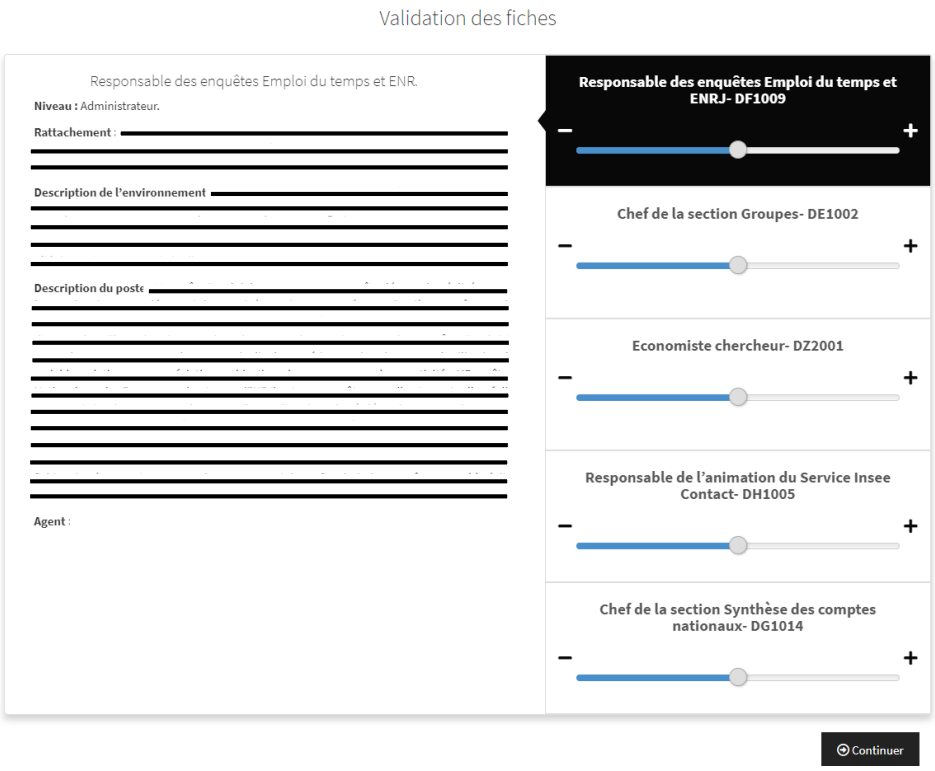
output

A reactiveValues with jobs selected / rejected

MODULES

Our four modules

Module Validation des Fiches



input

A reactive function returning a table with job descriptions

output

A reactiveValues with score given by user to each job

MODULES

Our four modules

Module Restitution des Fiches

input

A reactive function returning a table with job descriptions

output

No Shiny ouput

MODULES

Developing using modules

Independent
(it can be run without the others)

- we were able to develop them in parallel
- at the end, all that remains is to assemble the components
- Easier to debug, if the app crashes, we can just look at the incriminated module
- Easier to test

MODULES

Developing using modules

Better, Smaller, Easier, Stronger

filename	language	loc	blank_lines	comment_lines
R\module-selectionFiche.R	R	204	23	100
R\module-matchingFichesPostes.R	R	203	21	42
R\module-gestionFichesListe.R	R	190	22	22
R\module-editMacro.R	R	169	19	44
R\module-restitutionFiches.R	R	120	21	70
R\module-validationFiches.R	R	84	21	67
R\module-selectionTags.R	R	46	9	55

- Smaller scripts (max 200 lines of code)
- Better organization in a package (clearly identify modules)
- Easier to modify

MODULES

Developing using modules

Documentation

(each module is documented like an R function and exported)

Module Sélection Fiche

Description

Présentation des fiches deux à deux pour que l'utilisateur puisse donner sa fiche de préférence parmi celles proposées.

Usage

```
selectionFicheUI(id)

selectionFicheServer(input, output, session, data_fiches = NULL,
  n_fiches_max = 5)
```

Arguments

`id`	ID du module.
`input, output, session`	Arguments standards de Shiny.
`data_fiches`	Une fonction `reactive` contenant les données des fiches de poste.
`n_fiches_max`	Nombre de couples de fiches de postes à présenter à l'utilisateur.

Value

Une `reactiveValues` avec les slots suivants :

- **selected** : ID des fiches de poste sélectionnées par l'utilisateur.
- **rejected** : ID des fiches de poste rejetées par l'utilisateur.

Examples

Explicitly specify input value

Explicitly specify output value

MODULES

Developing using modules

Package integration

Properly populate your NAMESPACE

`@import shiny` may be sufficient

But `@importFrom` can avoid some problems of deployment

Correctly manage your JavaScript/CSS dependencies in inst/

Don't use `include*` function, make the effort to make proper dependencies management with `htmltools`

MODULES

Ressources

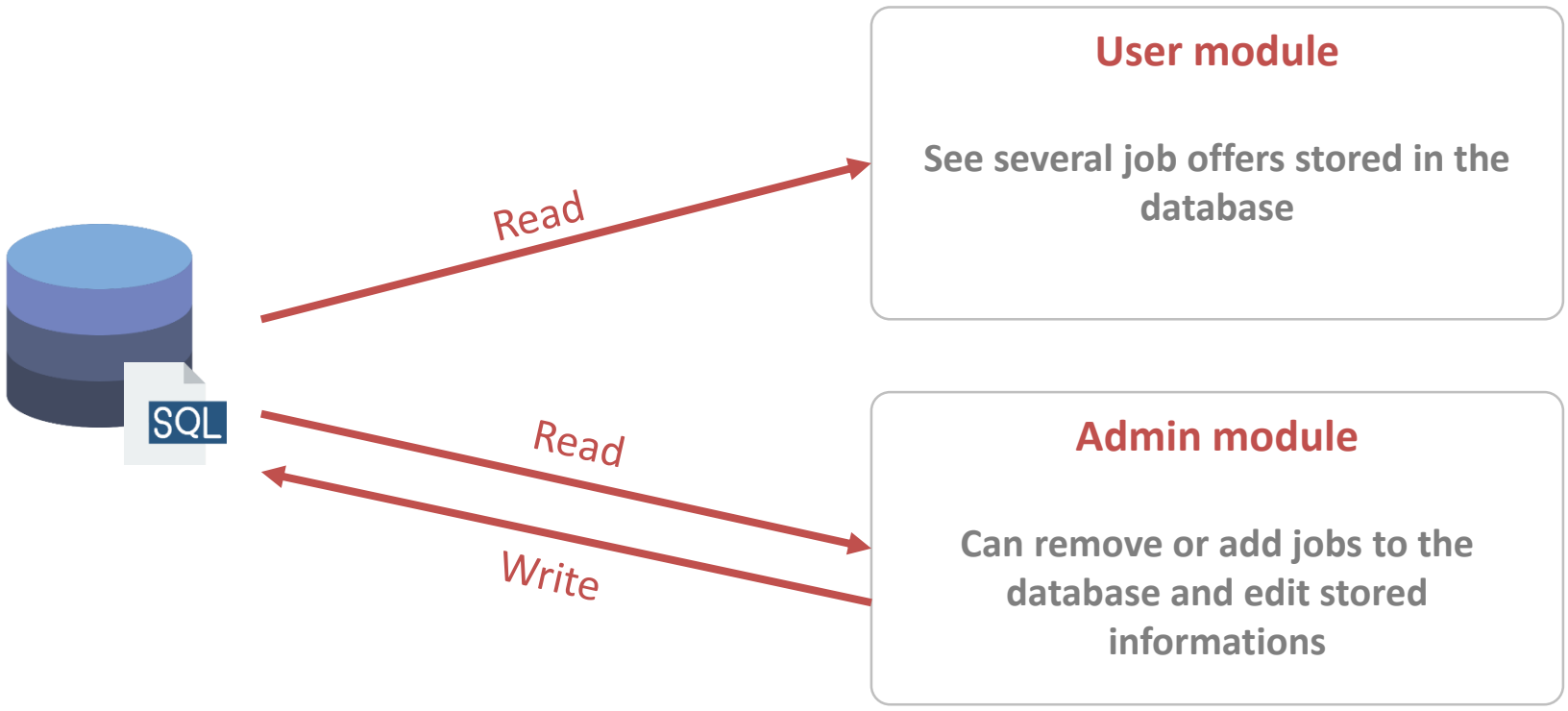
Modularizing Shiny app code, Joe Cheng : http://shiny.rstudio.com/articles/modules.html

Effective use of Shiny modules, Eric Nantz : https://rpodcast.github.io/rsconf-2019/#1

Modules Shiny : Pourquoi les utiliser ? Quentin Fazilleau : https://www.ardata.fr/post/2019/02/11/why-using-modules/

DATABASE

Backend for the application



DATABASE

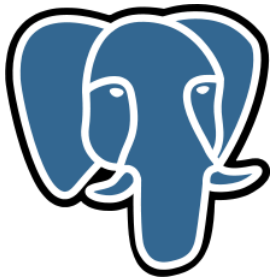
SQL with DBI

Prototype : SQLite

- Local database
- Each function deals with connection/disconnection
- Can be used (and tested) outside application

Deployment : PostgreSQL

- Remote server
- Only call to dbConnect to modify



TRACKING

Records outputs / inputs

Validate what has been done

What job descriptions have been proposed to the user?

Are those jobs pertinent?

Does our algorithm converge?

How many users go through the entire process?

Go further

Supervised learning: use results to see what users prefer

Cross results with internal HR data

APPS

Our Shiny gallery

http://shinyapps.dreamrs.fr/

dreamRs

La data science avec un grand R

Conseil et expertise
en datascience

Accompagnement
et formation R

Développement
d'outils

https://www.dreamrs.fr/

Merci ☺

Des questions ?



victor.perrier@dreamrs.fr fanny.meyer@dreamrs.fr etienne.sanchez@dreamrs.fr

https://www.dreamrs.fr/
@dreamrs_fr