fcaR, Formal Concept Analysis with the R language

Motivation, success stories and future work with the **fcaR** library

Domingo López-Rodríguez

Department of Applied Mathematics
Universidad de Málaga

Personal homepage: `https://dominlopez.netlify.app`
fcaR homepage: `https://malaga-fca-group.github.io/fcaR`

Motivation

Why to develop an R package for FCA?

- R, together with Python, are the two most widely used programming languages in Machine Learning and Data Science.
- In R there are already libraries for association rule mining that have become standard: **arules**.
- There is no library in R that implements the basic ideas and functions of FCA and allows them to be used in other contexts.

Our purpose

- To help disseminate FCA as a knowledge discovery tool.
- To be able to perform rapid testing of new ideas, algorithms, etc., both from a theoretical and practical point of view.
- Rapid prototyping of new solutions that can be integrated into more complex computational systems.
- To enable the application of FCA to real problems: automatic reasoning and recommender systems.

Design principles

Usability

- Direct execution of most classical algorithms (even in the fuzzy setting).
- Provide methods to operate on contexts, concept lattice and implications.
- **Logic**: include the SL_{FD} logic to compute closure wrt implication sets.
- Interoperability:
  - Read/write datasets in various formats (CSV, CTX, . . .).
  - Import and export to **arules**.
- Allow reproducible research.
- Provide lots of documentation with examples.

Implementation

- Modern programming paradigms (object-oriented).
- Classes representing entities: contexts, lattices, implications. . .
- Allow for extensions: new algorithms, new ideas. . .
- Use base `R` for the interface, but bottlenecks implemented in `C`.

The **fcaR** library

Library availability

Contributed Packages

Available Packages

Currently, the CRAN package repository features 18994 available packages.

The package is in a stable phase in a repository on Github and on CRAN.

- Unit tests
- Vignettes with demos
- Status:
  - lifecycle: stable
  - CRAN version: 1.1.1
  - downloads: ~22K

Classes and methods

Classes

Class name	Use
"Set"	A basic class to store a fuzzy set using sparse matrices
"Concept"	A pair of sets (extent, intent) forming a concept for a given formal context
"ConceptLattice"	A set of concepts with their hierarchical relationship. It provides methods to compute notable elements, sublattices and plot the lattice graph
"ImplicationSet"	A set of implications, with functions to apply logic and compute closure of attribute sets
"FormalContext"	It stores a formal context, given by a table, and provides functions to use derivation operators, simplify the context, compute the concept lattice and the Duquenne-Guigues basis of implications

Table 1: Main classes found in **fcaR**.

Main methods

Formal Contexts
intent
extent
closure
clarify
reduce
standardize
find_concepts
find_implications

Concept Lattice
supremum
infimum
sublattice
meet_irreducibles
join_irreducibles
subconcepts
superconcepts
lower_neighbours
upper_neighbours

Implication Set
closure
recommend
apply_rules
to_basis

Fuzzy extension

Let $\mathbb{K} = (G, M, I)$ be a formal context and $\mathbb{L} = \langle L, \wedge, \vee, \otimes, \rightarrow, 0, 1\rangle$ a complete residuated lattice, and define the operators $^{\uparrow} : L^G \rightarrow L^M$, $^{\downarrow} : L^M \rightarrow L^G$ as:

$$A^{\uparrow}(m) := \bigwedge_{g \in G} (A(g) \rightarrow I(g, m))$$

$$B^{\downarrow}(g) := \bigwedge_{m \in M} (B(m) \rightarrow I(g, m))$$

This operators form a Galois connection, which allow us to study the associated closure system by means of the concept lattice and of the basis of implications.

Sample of use in fuzzy setting

	a1	a2	a3	a4	a5	a6
o1	0	1	½	½	1	0
o2	1	1	1	0	0	0
o3	½	½	0	0	0	1
o4	0	0	0	1	½	0
o5	0	0	1	½	0	0
o6	½	0	0	0	0	0

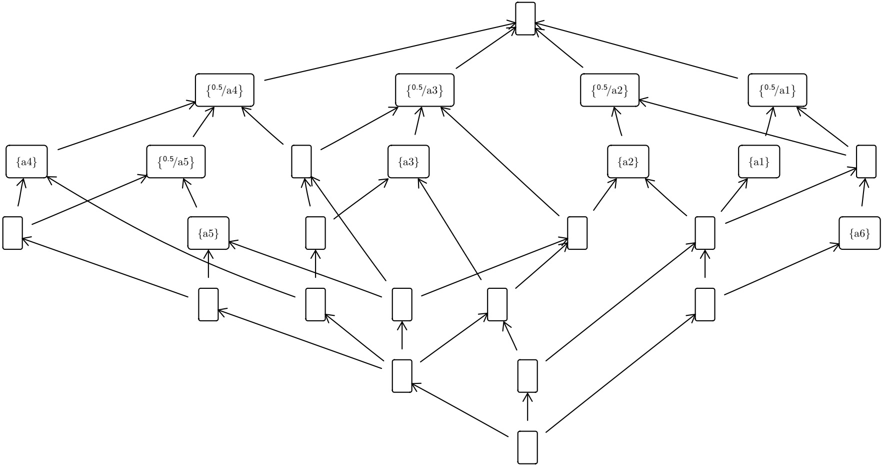
Table 2: Fuzzy (graded) formal context named “fc”.

With a fuzzy formal context we can perform the most common operations in FCA, as mentioned before.

```
# This finds the entire set of concepts
# and the canonical basis of implications.
fc$find_implications()
```

Concept lattice

```
fc$concepts$plot(to_latex = TRUE)
```



Basis of implications

```
fc$implications
```

$$
\begin{aligned}
1:&\quad \{^{0.5}/a6\} &\Rightarrow&\quad \{^{0.5}/a1, ^{0.5}/a2, a6\}\\
2:&\quad \{^{0.5}/a5\} &\Rightarrow&\quad \{^{0.5}/a4\}\\
3:&\quad \{^{0.5}/a3, ^{0.5}/a4, ^{0.5}/a5\} &\Rightarrow&\quad \{a2, a5\}\\
4:&\quad \{^{0.5}/a3, a4\} &\Rightarrow&\quad \{a3\}\\
5:&\quad \{^{0.5}/a2, ^{0.5}/a4\} &\Rightarrow&\quad \{a2, ^{0.5}/a3, a5\}\\
6:&\quad \{^{0.5}/a2, ^{0.5}/a3\} &\Rightarrow&\quad \{a2\}\\
7:&\quad \{a2, a3, ^{0.5}/a4, a5\} &\Rightarrow&\quad \{a4\}\\
8:&\quad \{^{0.5}/a1, ^{0.5}/a4\} &\Rightarrow&\quad \{a1, a2, a3, a4, a5, a6\}\\
9:&\quad \{^{0.5}/a1, ^{0.5}/a3\} &\Rightarrow&\quad \{a1, a2, a3\}\\
10:&\quad \{^{0.5}/a1, a2\} &\Rightarrow&\quad \{a1\}\\
11:&\quad \{a1, ^{0.5}/a2\} &\Rightarrow&\quad \{a2\}\\
12:&\quad \{a1, a2, a3, a6\} &\Rightarrow&\quad \{a4, a5\}
\end{aligned}
$$

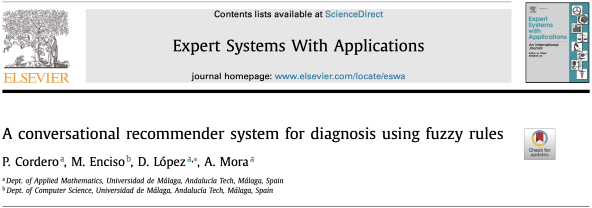
A remark on the Simplification Logic

SL_{FD}		Equivalence rules
[Ref]	$\dfrac{A \supseteq B}{A \Rightarrow B}$	
[Frag]	$\dfrac{A \Rightarrow B \cup C}{A \Rightarrow B}$	$\{A \Rightarrow B\} \equiv \{A \Rightarrow B \smallsetminus A\}$
[Comp]	$\dfrac{A \Rightarrow B,\ C \Rightarrow D}{A \cup C \Rightarrow B \cup D}$	$\{A \Rightarrow B, A \Rightarrow C\} \equiv \{A \Rightarrow BC\}$
[Simp]	$\dfrac{A \Rightarrow B,\ C \Rightarrow D}{A(C \smallsetminus B) \Rightarrow D \smallsetminus B}$	$A \subseteq C \Rightarrow \{A \Rightarrow B, C \Rightarrow D\} \equiv \{A \Rightarrow B, A(C \smallsetminus B) \Rightarrow D \smallsetminus B\}$
		$A \subseteq D \rightarrow \{A \Rightarrow B, C \Rightarrow BD\} \equiv \{A \Rightarrow B, C \Rightarrow D\}$

The SL_{FD} `closure` algorithm makes use of the above equivalence rules to compute the closure X^+ of a set X using a set of implications Σ, and return a simplified Σ' where the attributes in X^+ do not appear, and such that:

$$\{\varnothing \Rightarrow X\} \cup \Sigma \equiv \{\varnothing \Rightarrow X^+\} \cup \Sigma'$$

Use as a recommendation system


Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa

A conversational recommender system for diagnosis using fuzzy rules

P. Cordero[a], M. Enciso[b], D. López[a,*], A. Mora[a]

[a] Dept. of Applied Mathematics, Universidad de Málaga, Andalucía Tech, Málaga, Spain
[b] Dept. of Computer Science, Universidad de Málaga, Andalucía Tech, Málaga, Spain

The purpose is to use logic tools to build a conversational recommendation system from a fuzzy dataset such as this:

COSAS_1	COSAS_2	COSAS_3	FICAL_1	FICAL_2	FICAL_3	dx_ss	dx_other
extreme	moderate	absent	absent	absent	absent	extreme	absent
absent	absent	absent	extreme	m/ severe	m/ severe	absent	extreme
absent	absent	absent	absent	absent	absent	extreme	absent
moderate	extreme	moderate	m/ severe	mild	absent	extreme	absent
absent	absent	moderate	absent	absent	absent	absent	extreme
absent	absent	absent	absent	absent	absent	absent	extreme

Preprocessing steps

- Scale the context to a graded dataset:

	COSAS_1	COSAS_2	COSAS_3	FICAL_1	FICAL_2	FICAL_3	dx_ss	dx_other
O1	1	½	0	0	0	0	1	0
O2	0	0	0	1	⅔	⅔	0	1
O3	0	0	0	0	0	0	1	0
O4	½	1	½	⅔	⅓	0	1	0
O5	0	0	½	0	0	0	0	1
O6	0	0	0	0	0	0	0	1

Table 3: Scaled context.

- Use the tools in the package (the NextClosure algorithm, mainly) to build the Duquenne-Guigues basis of implications.

Conversation

1. The system asks the user to provide a symptom and a degree associated with it: $\{^{d_x}/\mathrm{x}\}$ where $x \in M$ and $d_x \in L$.
2. It computes the closure $\{^{d_x}/\mathrm{x}\}^+_\Sigma$ and its associated reduced set of implications Σ'.
3. If the closure contains an attribute identifying a disease, then a diagnosis has been produced. The system stops the process and provides the disease as the recommendation.
4. Optional step: the user has the possibility to give feedback about the closure, updating the degree of any symptom.
5. If the user declines to provide a feedback, agreeing with the information provided, then new symptoms (in the LHS of Σ') have to be introduced to continue with the conversation, going to Step 1.

Results

	Accuracy	Sensitivity	Specificity	Precision
ALS	0.360	0.333	0.380	0.290
IBCF (Cosine)	0.555	0.475	0.615	0.483
IBCF (Pearson)	0.770	0.466	1.000	1.000
LIBMF	0.491	0.901	0.181	0.455
SVD	0.376	0.515	0.271	0.349
SVDF	0.431	1.000	0.000	0.431
UBCF (Cosine)	0.608	0.967	0.335	0.524
UBCF (Pearson)	0.525	0.783	0.330	0.470
C5.0	0.674	0.636	1.000	1.000
PART	0.883	0.847	0.950	0.970
JRip	0.752	0.814	0.688	0.731
Random Forest	0.953	0.924	1.000	1.000
xgboost	0.818	0.963	0.713	0.706
k-nn	0.589	0.603	0.544	0.815
Proposal	0.982	0.996	0.948	0.955

Reproducible research with fcaR and interoperability

All classes have a `to_latex()` method to export in a suitable form to a LaTeX document. **fcaR** code can be embedded in RMD files (plain text + code + results) and produce a presentation (such as this one!) or a complete paper:

Implications and logic

The knowledge stored in a formal context can also be represented as a set of implications, which are expressions of the form $A \Rightarrow B$ where A and B are sets of attributes or items, indicating that, for every object in which the set of attributes A is present, also B is present. This interpretation is similar to the one defined in data mining/machine learning over the so-named association rules. The confidence (a well-known estimator of the rules' quality) has value 1 in all the implications.

For instance $\{^{0.5}/\text{P1}\} \Rightarrow \{\text{P4}\}$ is a valid implication in the previous example, having the following interpretation: when the attribute P1 has degree at least 0.5 then we have P4 with degree 1.

The *Duquenne-Guigues basis* of implications (Guigues and Duquenne, 1986) is a set of valid implications from which all other valid implications can be deduced. The Duquenne-Guigues basis in our example is given by:

$$
\begin{array}{lrcl}
1: & \varnothing & \Rightarrow & \{^{0.5}/\text{P2}, {}^{0.5}/\text{P4}\} \\
2: & \{^{0.5}/\text{P2}, \text{P4}\} & \Rightarrow & \{\text{P2}\} \\
3: & \{\text{P2}, {}^{0.5}/\text{P4}\} & \Rightarrow & \{\text{P4}\} \\
4: & \{\text{P2}, {}^{0.5}/\text{P3}, \text{P4}\} & \Rightarrow & \{\text{P3}\} \\
5: & \{^{0.5}/\text{P1}, {}^{0.5}/\text{P2}, {}^{0.5}/\text{P4}\} & \Rightarrow & \{\text{P2}, \text{P4}\} \\
6: & \{^{0.5}/\text{P1}, \text{P2}, \text{P3}, \text{P4}\} & \Rightarrow & \{\text{P1}\}
\end{array}
$$

In Cordero et al. (2002), the simplification logic, denoted as SL_{FD}, was introduced as a method to manipulate implications (functional dependencies or if-then rules), removing redundancies or computing closures of attributes. This logic is equivalent to Armstrong's Axioms (Armstrong, 1974), which are well known from the 80s in databases, artificial intelligence, formal concept analysis, and others. The axiomatic system of this logic considers reflexivity as the axiom scheme

$$\texttt{[Ref]}\ \frac{A \supseteq B}{A \Rightarrow B}$$

together with the following inference rules called fragmentation, composition and simplification, respectively, which are equivalent to the classical Armstrong's axioms of augmentation and, more importantly, transitivity.

$$\texttt{[Frag]}\ \frac{A \Rightarrow B \cup C}{A \Rightarrow B} \qquad \texttt{[Comp]}\ \frac{A \Rightarrow B,\ C \Rightarrow D}{A \cup C \Rightarrow B \cup D} \qquad \texttt{[Simp]}\ \frac{A \Rightarrow B,\ C \Rightarrow D}{A \cup (C \smallsetminus B) \Rightarrow (D \smallsetminus B)}$$

The main advantage of SL_{FD} with respect to Armstrong's Axioms is that the inference rules may be considered as equivalence rules, (see the work by Mora et al. (2012) for further details and proofs), that is, given a set of implications Σ, the application of the equivalences transforms it into an equivalent set. In the package presented in this paper, we develop the following equivalences:

1. Fragmentation Equivalency **[FrEq]**: $\{A \Rightarrow B\} \equiv \{A \Rightarrow B \smallsetminus A\}$.
2. Composition Equivalency **[CoEq]**: $\{A \Rightarrow B, A \Rightarrow C\} \equiv \{A \Rightarrow B \cup C\}$.
3. Simplification Equivalency **[SiEq]**: If $A \subseteq C$, then
$$\{A \Rightarrow B, C \Rightarrow D\} \equiv \{A \Rightarrow B, A \cup (C \smallsetminus B) \Rightarrow D \smallsetminus B\}$$
4. Right-Simplification Equivalency **[rSiEq]**: If $A \subseteq D$, then
$$\{A \Rightarrow B, C \Rightarrow B \cup D\} \equiv \{A \Rightarrow B, C \Rightarrow D\}$$

Usually, many areas, the implications have always atomic attributes on the right-hand side. We emphasize that this logic can manage *aggregated* implications, i.e. the implications' consequents do not have to be singletons. This represents an increase of the logic efficiency.

This logic removes attribute redundancies in some of the implications in the Duquenne-Guigues basis presented before. Particularly, the implications with numbers 2, 3, 4, 5 and 6 are simplified to:

$$
\begin{array}{lrcl}
2: & \{\text{P4}\} & \Rightarrow & \{\text{P2}\} \\
3: & \{\text{P2}\} & \Rightarrow & \{\text{P4}\} \\
4: & \{^{0.5}/\text{P3}, \text{P4}\} & \Rightarrow & \{\text{P3}\} \\
5: & \{^{0.5}/\text{P1}\} & \Rightarrow & \{\text{P4}\} \\
6: & \{^{0.5}/\text{P1}, \text{P3}\} & \Rightarrow & \{\text{P1}\}
\end{array}
$$

One of the primary uses of a set of implications is computing the closure of a set of attributes, the maximal fuzzy set that we can arrive at from these attributes using the given implications.

Derivation operators

The methods that implement the derivation operators are named after them: `intent()`, `extent()` and `closure()`. They can be applied on objects of type `"Set"`, representing fuzzy sets of objects or attributes:

```
> S <- Set$new(fc$objects, O1 = 1, O2 = 1)
> S
{O1, O2}
> fc$intent(S)
{P2 [0.5], P4 [0.5]}
> T <- Set$new(fc$attributes, P1 = 1, P3 = 1)
> T
{P1, P3}
> fc$extent(T)
{}
> fc$closure(T)
{P1, P2, P3, P4}
```

In addition, we can perform *clarification* on the formal context, by using `fc$clarify()`, giving:

```
FormalContext with 3 objects and 3 attributes.
          P1   P3  [P2, P4]
      O1  0   0.5     0.5
      O4  0    1      0.5
 [O2, O3] 0.5  0       1
```

The duplicated rows and columns in the formal context have been collapsed, and the corresponding attributes and objects' names are grouped together between brackets, e.g., [P2, P4].

Concept lattice

The command to compute the concept lattice for a `"FormalContext"` `fc` is `fc$find_concepts()`. The lattice is stored in `fc$concepts`, which is of the `"ConceptLattice"` class.

```
> fc$concepts
A set of 8 concepts:
1: ({O1, O2, O3, O4}, {P2 [0.5], P4 [0.5]})
2: ({O1, O4}, {P2 [0.5], P3 [0.5], P4 [0.5]})
3: ({O1 [0.5], O4}, {P2 [0.5], P3, P4 [0.5]})
4: ({O1 [0.5], O2, O3, O4 [0.5]}, {P2, P4})
5: ({O1 [0.5], O4 [0.5]}, {P2, P3, P4})
6: ({O2, O3}, {P1 [0.5], P2, P4})
7: ({O2 [0.5], O3 [0.5]}, {P1, P2, P4})
8: ({}, {P1, P2, P3, P4})
```

In order to know the *cardinality* of the set of concepts (that is, the number of concepts), we can use `fc$concepts$size()`, which gives 8 in this case. The complete list of concepts can be printed with `fc$concepts$print()`, or simply `fc$concepts`. Also, they can be translated to LaTeX using the `to_latex()` method, as mentioned before.

The typical subsetting operation in R with brackets is implemented to select specific concepts from the lattice, giving their indexes or a boolean vector indicating which concepts to keep. The same rules for subsetting as in R base apply:

```
> fc$concepts[c(1:3, 5, 8)]
A set of 5 concepts:
1: ({O1, O2, O3, O4}, {P2 [0.5], P4 [0.5]})
2: ({O1, O4}, {P2 [0.5], P3 [0.5], P4 [0.5]})
3: ({O1 [0.5], O4}, {P2 [0.5], P3, P4 [0.5]})
4: ({O1 [0.5], O4 [0.5]}, {P2, P3, P4})
5: ({}, {P1, P2, P3, P4})
```

In addition, the user can compute concepts' *support* (the proportion of objects whose set of attributes contains the intent of a given concept) by means of `fc$concepts$support()`.

```
> fc$concepts$support()
[1] 1.00 0.50 0.25 0.50 0.00 0.50 0.00 0.00
```

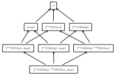



Figure 4: Hasse diagram for a sublattice of the `cobre32` formal context.

an individual) and returns the degree of the diagnosis attributes using the implications extracted from the formal context as an inference engine.

Next, we use the NextClosure algorithm to extract implications and compute the set of concepts, using `fc$find_implications()`.

The concept lattice is quite big (14706 concepts); therefore, it cannot be plotted here for space and readability reasons. For this reason, we only plot a sublattice of small size in Figure 4.

There is an aggregate of 985 implications extracted. Let us compute the average cardinality of the LHS and the RHS of the extracted rules:

```
> colMeans(fc$implications$size())
     LHS      RHS
2.417597 1.954146
```

Note that our paradigm can deal with non-unit implications, that is, where there is more than one attribute in the RHS of the implication. This feature is an extension of what is usual in other paradigms, for example, in transactional databases.

We can use the *simplification logic* to remove redundancies and reduce the LHS and RHS size of the implications. The reason to do this is to decrease the computational cost of computing closures:

```
> fc$implications$apply_rules(rules = c('simplification', 'rsimplification'))
> colMeans(fc$implications$size())
     LHS      RHS
1.998308 1.557191
```

We can see that the average cardinality of the LHS has been reduced from 2.418 to 1.998 and that the one of the RHS, from 1.954 to 1.557.

With the simplified implication set, we can build a recommender system by simply wrapping the `recommend()` method inside a function:

```
> diagnose <- function(S) {
+
+   fc$implications$recommend(S = S,
+                             attribute_filter =
+                               c('dx_ss', 'dx_other'))
+
+ }
```

This function can be applied to `"Set"`s that have the same attributes as those of the formal context. The `attribute_filter` argument specifies which attributes are of interest, in our case, the diagnosis attributes.

Let us generate some sets of attributes and get the recommendation (diagnosis) for each one:

```
> S1 <- Set$new(attributes = fc$attributes,
+               COSAS_1 = 1/2, COSAS_2 = 1, COSAS_3 = 1/2,
+               COSAS_4 = 1/6, COSAS_5 = 1/2, COSAS_6 = 1)
> diagnose(S1)
  dx_ss dx_other
      1        0
```

Where to find help

https://malaga-fca-group.github.io/fcaR/

fcaR 1.1.1 Reference Articles Changelog

fcaR: Tools for Formal Concept Analysis

The aim of this package is to provide tools to perform fuzzy formal concept analysis (FCA) from within R. It provides functions to load and save a Formal Context, extract its concept lattice and implications. In addition, one can use the implications to compute semantic closures of fuzzy sets and, thus, build recommendation systems.

Stories of success

Where *else* have we used **fcaR**?

Minimal generators from positive and negative attributes: analysing the knowledge space of a maths course

mathematics

MDPI

Article

Simplifying Implications with Positive and Negative Attributes: A Logic-Based Approach

Francisco Pérez-Gámez, Domingo López-Rodríguez, Pablo Cordero, Ángel Mora and Manuel Ojeda-Aciego *

Departamento Matemática Aplicada, Universidad de Málaga, 29071 Málaga, Spain;
franciscoperezgamez@uma.es (F.P.-G.); dominlopez@uma.es (D.-L.R.); pcordero@uma.es (P.C.);
amora@uma.es (Á.M.)
* Correspondence: aciego@uma.es

Future works

Future developments

Web application

- Web app for **fcaR** to improve the usability by non-experts.

Some extensions

- Integrate association rules in the library (Luxenburger's basis).
- Logic for mixed attributes: new algorithms to compute bases of mixed implications, iterative closure algorithm...
- Other extensions: $\{\circ, +, -, \imath\}$.

Other algorithms

- Concept lattice (InClose, FastCbO, NextNeighbour)
- Canonical basis of implications
- Direct bases and minimal generators.
- Parallelization of the above.

Ad hoc algorithms for the computation of the fuzzy concept lattice

Preliminary results

	Concepts	Algorithm	Tests	PartialTests	Intents	Time
	<int>	<chr>	<dbl>	<dbl>	<dbl>	<dbl>
1	131260	NextClosure	17641654	0	317549772	10.8
2	131260	Fuzzy_FCbO	1791025	1791025	53730780	0.940
3	131260	Fuzzy_InClose2	1958250	2019504	12691526	0.299
4	131260	Fuzzy_InClose4	1865932	1927163	12496417	0.269
5	131260	Fuzzy_InClose5	895211	956442	7587263	0.203
6	131260	Fuzzy_InClose7	303520	364751	4713802	0.163

THANK YOU VERY MUCH