Data Mining

Esteban Marquer
Maxime Guillaume

November 2019

1 Encoding and decoding SPMF files

To be able to use the SPMF software on our data, we need to firs encode said data in a format that is readable by SPMF, process it with the tool, and then decode the output to be able to read it and interpret it. In our case, we filter out part of the data while encoding it.

1.1 Selected attributes

We decided to center our analysis on the exploration of the impact of the socio-economical profile on the housing condition. Using the set of attributes proposed within our instructions as a basis, we kept the attributes we considered as defining for the housing and the people in the housing. We also removed most overlapping attributes, while preferring having multiple attributes than a single attribute aggregation all the information. This choice is intended to ease the itemset extraction and make it easier to interpret. Finally, we added the SURF attribute, which was not in the original set, as we consider it a ddefining attribute of the housing conditions.

The attributes we kept had a limited enough variety of values to limiting the computational cost and the difficulty of interpretation. In particular, the only continuous attributes were encoded as slices.

We selected the following list of attributes (or column names) for our data:

- 2 binary attributes (COUPLE and SEXE);
- 2 ordinal attributes (NBPI and NPERR) and 2 continuous attributes represented by slices (AGER20 and SURF);
- 12 nominal attributes (ASCEN, CATL, CMBL, CS1, DIPL_15, GARL, INAT, MODV, NA38, SANI, STOCD, TACT).

The final list of attributes we selected for socio-economical and housing-related attributes are presented respectively in Table 14 (page 10) and Table 15 (page 10).

1.2 Processes and files involved in the encoding and decoding

The encoding and decoding steps are handled within a single Python script (spmf_encoding.py), meant to be used in command line.

Column filtering We need to take a subset of attributes, according to the list we defined (see subsection 1.1, page 1). This list of attributes is stored in the columns.txt, with a line per attribute name. Using a separate file allows us to easily maintain the script and adapt the list of attribute to take into account if necessary. When we execute the script, the list is loaded in memory and the data to process is loaded with the Pandas library. We use this specific library for one of its ability to select only specific columns from the structured data it loads (CSV with ; separator in our case). Using this feature, the filtering of the attributes is performed early on and we obtain a filtered and easy to handle table containing the data.

Binarization of the attributes As said at the beginning of this section, SPMF needs a specific type of file to work. Indeed, each multi-valuated attribute has to be transformed into a binary attribute (either present or absent for a specific individual). To do this, we build new attributes by concatenating each attribute name with every possible value for the attribute. Each row of the table is then composed of the new binary attributes corresponding

to the values of the original attributes of the individual, and the label encoder (presented next paragraph) handles the representation of the binary attributes.

For example, in Table 1 (page 2), the binarization creates from the value 64 of the attribute AGER20 produced the binary attribute AGER20_64. Another value for this attribute on another line would create another binary attribute.

	AGER20	ANARR	ANEMR	BAIN
Before binarization	64	Z	2	Z
After binarization	AGER20_64	ANARR_Z	ANEMR_2	BAIN_Z

Table 1: Binarization of the attributes

Encoding the attributes using sklearn.LabelEncoder A particularity of the SPMF files is that for each line, the attributes must be encoded into a positive number and sorted in increasing order.

To do this, the basic intuition is to build a mapping between the attributes domain and the positive integer. We use for this a label encoder from the sklearn library which encodes categorical features into integers. As this tool provides both the encoding and decoding of the features, and does not require additional development, it fits our use in this context. Also, the provided encoded feature set is already sorted, which is even better.

Pickling the sklearn.LabelEncoder At some point, it will be necessary to decode what has been done, and for that we will need to reuse the label encoder. To store the label encoder we choose to serialize it using joblib [1]. While a pickled object is not human readable, it is fast to load and ready to use. In fact, once the encoder is saved, decoding only requires 3 steps:

- loading the label encoder pickle;
- opening the encoded data;
- decoding by aplying the encoder on the data.

1.3 Examples of data before and after encoding and decoding

Code 1: Sample of the data before encoding, containing all the attributes

```
44;1;3;2008;9;058;64;057;055;1959;Z;009;02;1;Z;1;Z;Z;1;0;Z;1;Z;Z;2;1;8;82;85;Z;99;
Z;A;Z;Z;Z;ZZ;999999999;2;2;1;Z;Z;Z;Z;2;1;11;1;2;2;3.161598156331 41;1;3;1;2;2;M;21;
40;ZZ;ZZ;ZZZZZ;00;00;997;0;01;997;1;05;2;1;00;01;ZZZZ;0;2;ZZ;1;30;ZZ;B;22;7;25;250;
Z;Z;3;2;2;500;44;1;1;Z
```

Code 2: Sample of the data after encoding, containing only the selected attributes

```
10 13 16 19 25 34 35 41 43 52 94 119 126 134 136 140 150 158
```

Code 3: Sample of the data before decoding

```
16 134 19 #SUP: 647846
```

Code 4: Sample of the data after decoding, containing the attribute labels

```
"CATL_1 CMBL_2 SANI_2" 647846
```

2 Itemset extraction

We extracted itemsets using FPGrown and we also extracted closed-itemsets using FPClose similarly to the practical session we did a few months back. The intent was to compare the kind of outputs of the two algorithms. We exclusively used the graphical interface of SPMF, without modifying the Java parameters, as we obtained decent computation times even without modifying them.

[1] The joblib package is an extension of the pickle package specific to sklearn.

The amount[2] of itemset extracted with the different support threshold we tested (from 90% to 40%) is presented Table 2 (page 3). We will perform a more in-depth analysis of a few samples from each slice of support (40 to 50%, *etc.*) in the next section.

FPGrowth is an algorithm extracting frequent itemsets. However, "FPClose is an algorithm of the FPGrowth family of algorithms, designed for mining frequent closed itemsets."[3] As FPClose is limited to only closed itemsets, the number of itemset extracted by this algorithm is as expected way smaller then those extracted by FPGrowth.

Threeshold	FPGrowth	FPClose
90%	3	2
80%	9	6
70%	15	9
60%	22	13
50%	47	29
40%	119	69

Table 2: Number of itemset depending on the minimal support threshold for FPGrowth and FPClose

3 Analysis of 10 chosen itemsets

While itemsets with a high support inform us on frequent and redundant properties of the items, rarer itemsets can inform us on specific itemsets. That's why we selected 10 itemsets from those extracted by the FPGrowth algorithm by selecting itemsets from different slices of support (above 90%, 90% to 80%, 80% to 70%, *etc.* until 40%). For reference, the total population (100% of support) is 1474560.

As written in subsection 1.1 (page 1), we choose to focus on the exploration of the impact of the socio-economical profile on the housing condition. The itemsets we choose are centered on this view, and we split the items in two groups depending on which aspect they refer to (according to Table 14, page 10 and Table 15, page 10).

In the following subsections we will present the 10 itemsets we selected. We selected:

- the 2 main itemsets with support above 90%;
- 2 itemsets with support between 90% and 80%;
- 1 itemset with support between 80% and 70%;
- 1 itemset with support between 70% and 60%;
- 2 itemsets with support between 60% and 50%;
- 3 itemsets with support between 50% and 40%.

The meaning of the attribute tags are presented Table 14 (page 10) and Table 15 (page 10). For each presented itemset the specific values of the attributes will be explained.

We did not find very interesting relations between the socio-economical profile on the housing condition by studiing the itemsets themselves.

3.1 Above 90% of support

The first two itemsets we selected are CATL_1 and CATL_1 SANI_2, with a support of 1444294 and 1400247 respectively (above 90% of the total number of samples). We also find SANI_2 with the same support as CATL_1 SANI_2, meaning that each time we have SANI_2 we also have CATL_1 (SANI_2 is a generator of CATL_1 SANI_2).

- CATL corresponds to the kind of housing, with CATL_1 being for main residence.
- SANI_2 means that there is a bathroom, in other words either a bath or a shower in a dedicated room (the other possible values for SANI are either a bath or a shower but without a dedicated room, or no bath nor shower).

[2] To reproduce the results: launch the script Reproduce/count_item.sh
[3] https://www.philippe-fournier-viger.com/spmf/FPClose.php

We can interpret this as the vast majority (98%) of the studied data corresponding to main residences, with in most cases (95% of the total) a bathroom. In other words, except for 30266 outliers, the studied population describes main residences, which is to be expected. There is a surprisingly high (in our opinion at least) proportion of lodging not containing a proper bathroom. Given the French legislation, those most likely reffer to student rooms with a shared bathroom.

3.2 From 90% to 80% of support

We selected the itemset INAT_11, which has a support of 1293417. There is no other itemset with exactly the same support (the closest being CATL_1 INAT_11 with a suport of 1267347). INAT is the value for the nationality, with INAT_11 corresponding to the resident being native French people. As we can see, around 88% of the data corresponds to native French residents.

We selected the itemset ASCEN_2 with a support of 1203841 which is a generator of ASCEN_2 CATL_1. ASCEN_2 means that there is no elevator in the housing (ASCEN_1 for when there is). We can conclude that knowing that there is no elevator allows us to determine that the housing is a main residence.

3.3 From 80% to 70% of support

We selected the itemset ASCEN_2 CATL_1 INAT_11 SANI_2 with a support between 80% and 70% (1049023). If we consider this itemset by itelf, finding it within the upper slices of support can be interpreted as the corresponding properties being frequently together. In other words, native French residents (INAT_11) having their main housing (CATL_1) with bathroom (SANI_2) and without no elevator in the housing (ASCEN_2) is a frequent occurrence in the data.

3.4 From 70% to 60% of support

We selected the itemset GARL_1 with a support of 985420, which is a generator of GARL_1 CATL_1. GARL_1 means the housing is equipped with parking lots (GARL_2 for when there is none). We can conclude that knowing that the housing is equipped with parking lots allows us to determine that the housing is a main residence.

3.5 From 60% to 50% of support

We selected the itemset STOCD_10 with a support of 798042, which is a generator of STOCD_10 CATL_1. STOCD_10 correspond to the resident being the owner of the residence. It is the first interesting resident-related attribute which appears in the itemsets. We can conclude that knowing that the resident is the landlord allows us to determine that the housing is a main residence. We can note that around 54% of the data correspond to landlords.

From the support of the itemset COUPLE_2, we can observe that 53% of the data (779870) corresponds to people not officially living in couple ("a déclaré ne pas vivre en couple").

3.6 From 50% to 40% of support

From the support of the itemset COUPLE_1, we can observe that 47% of the data (694689) corresponds to people officially living in couple ("a déclaré vivre en couple"). It corresponds to the counterpart of the itemset COUPLE_2, so this result is exactly as expected.

From the support of the itemset CMBL_2, we can observe that 45% of the data (667604) of the housing uses city gas for heating. As CMBL_2 is the generator of CATL_1 CMBL_2, so this statistic only describes only main housings.

Similarly, most pairs of generators and itemsets that we find only allow to deduce that we find CATL_1, which is also the most frequent itemset (and itemset of length 1), and most itemsets are presented in such pairs (one itemset with and one without CATL_1). For example:

- CATL_1 INAT_11 STOCD_10 and its generator INAT_11 STOCD_10;
- CATL_1 INAT_11 NA38_ZZ SANI_2 and its generator INAT_11 NA38_ZZ SANI_2;
- CATL_1 ASCEN_2 SANI_2 STOCD_10 and its generator ASCEN_2 SANI_2 STOCD_10;
- CATL_1 COUPLE_2 SANI_2 and its generator COUPLE_2 SANI_2.

4 Association rules extraction

We extracted the association rules with different support and confidence values (see table 3). We have chosen supports varying from 5 to 90. As we want high confidence rules, we choose relatively high confidence thresholds (between 30 and 100). As expected, the lower the parameters, the more rules we get.

Minsup	Minconf	Number of rules
90	90	2
90	80	2
90	70	2
80	90	8
80	80	14
80	70	14
70	90	17
70	80	45
70	70	50
50	95	49
50	50	220
30	100	110
30	95	288
30	80	773
30	50	1604
30	30	2208
15	85	8618
10	100	11235
10	95	26069
5	95	218314
5	85	490246

Table 3: Number of rules extracted by FPGrowth Association Rules

4.1 Selection criterion

As we choose to focus on the exploration of the impact of the socio-economical profile on the housing condition, the filtering process for a rule $r : \alpha \rightarrow \beta$ is based on the following criterion: if α contains at least one element from the people attributes set (see Table 14, page 10) and β at least one element from the housing attributes set (see Table 15, page 10) then it is an interesting rule for us. In addition we apply an other filter to obtain just the rules concerning our 10 selected attributes:

- COUPLE_2;
- SEXE_2;
- CATL_1;
- CATL_1 SANI_2;
- INAT_11;
- ASCEN_2;
- ASCEN_2 CATL_1 INAT_11 SANI_2;
- GARL_1;
- STOCD_10;
- CMBL_2.

4.2 Obtained rules

Each association rule $r : \alpha \rightarrow \beta$ can be intepreted as if α occurs then it's likely that β appears. For example in our data we have SEXE_2 −−> CATL_1 #CONF: 0.98. We can conclude that the fact of being a woman implies that the census took place in the main house of the woman with a confidence of 0.98. We used this prototypical interpretation process to analyze our data.

We decided to present in this section the rules extracted with a min_conf of 95% and a min_sup of 50 as it is pretty small. From our analysis, the rules extracted do not bring us much new information compared to the itemset extraction. Indeed, for example in this configuration, we have only rules of the following form (where X is an itemset):

- X $\rightarrow$ CATL_1;
- X $\rightarrow$ CATL_1 SANI_2;
- X $\rightarrow$ SANI_2.

However, association rules are easier to read and interpret in general.

We have tried a lot of other hyperparemeter combinations and we haven't made any discoveries so we have to drastically lowered the confidence and the support to try and get more interesting rules. Consequently, we have a lot of rules that are nor extremely trustworthy. Those rules are presented in Table 4 to Table 13 (page 6 to page 9). We also have tried to study the impact of housing condition on socio-economical profile (by inverting our condition on α and β) and the results are not more interesting.

α	β
COUPLE_2	CATL_1

Table 4: Rules extracted with COUPLE_2 in α

α	β
SEXE_2	CATL_1

Table 5: Rules extracted with SEXE_2 in α

α	β
ASCEN_2 CATL_1 GARL_1 INAT_11	SANI_2
ASCEN_2 CATL_1 INAT_11	SANI_2
ASCEN_2 GARL_1 INAT_11	CATL_1
ASCEN_2 GARL_1 INAT_11	CATL_1 SANI_2
ASCEN_2 GARL_1 INAT_11 SANI_2	CATL_1
ASCEN_2 INAT_11	CATL_1
ASCEN_2 INAT_11	CATL_1 SANI_2
ASCEN_2 INAT_11 SANI_2	CATL_1
ASCEN_2 STOCD_10	CATL_1
CATL_1 GARL_1 INAT_11	SANI_2
CATL_1 INAT_11	SANI_2
CATL_1 NA38_ZZ	SANI_2
CATL_1 STOCD_10	SANI_2
COUPLE_2	CATL_1
GARL_1 INAT_11	CATL_1
GARL_1 INAT_11	CATL_1 SANI_2
GARL_1 INAT_11 SANI_2	CATL_1
INAT_11	CATL_1
INAT_11	CATL_1 SANI_2
INAT_11 NA38_ZZ	CATL_1
INAT_11 SANI_2	CATL_1
NA38_ZZ	CATL_1
NA38_ZZ SANI_2	CATL_1
SANI_2 STOCD_10	CATL_1
SEXE_2	CATL_1
STOCD_10	CATL_1
STOCD_10	CATL_1 SANI_2

Table 6: Rules extracted with CATL_1 in α or β

α	β
ASCEN_2 GARL_1 INAT_11	CATL_1 SANI_2
ASCEN_2 INAT_11	CATL_1 SANI_2
GARL_1 INAT_11	CATL_1 SANI_2
INAT_11	CATL_1 SANI_2
STOCD_10	CATL_1 SANI_2

Table 7: Rules extracted with CATL_1 SANI_2 in β

α	β
ASCEN_2 CATL_1 GARL_1 INAT_11	SANI_2
ASCEN_2 CATL_1 INAT_11	SANI_2
ASCEN_2 GARL_1 INAT_11	CATL_1
ASCEN_2 GARL_1 INAT_11	CATL_1 SANI_2
ASCEN_2 GARL_1 INAT_11	SANI_2
ASCEN_2 GARL_1 INAT_11 SANI_2	CATL_1
ASCEN_2 INAT_11	CATL_1
ASCEN_2 INAT_11	CATL_1 SANI_2
ASCEN_2 INAT_11	SANI_2
ASCEN_2 INAT_11 SANI_2	CATL_1
CATL_1 GARL_1 INAT_11	SANI_2
CATL_1 INAT_11	SANI_2
GARL_1 INAT_11	CATL_1
GARL_1 INAT_11	CATL_1 SANI_2
GARL_1 INAT_11	SANI_2
GARL_1 INAT_11 SANI_2	CATL_1
INAT_11	CATL_1
INAT_11	CATL_1 SANI_2
INAT_11	SANI_2
INAT_11 NA38_ZZ	CATL_1
INAT_11 SANI_2	CATL_1

Table 8: Rules extracted with INAT_11 in α or β

α	β
ASCEN_2 CATL_1 GARL_1 INAT_11	SANI_2
ASCEN_2 CATL_1 INAT_11	SANI_2
ASCEN_2 GARL_1 INAT_11	CATL_1
ASCEN_2 GARL_1 INAT_11	CATL_1 SANI_2
ASCEN_2 GARL_1 INAT_11	SANI_2
ASCEN_2 GARL_1 INAT_11 SANI_2	CATL_1
ASCEN_2 INAT_11	CATL_1
ASCEN_2 INAT_11	CATL_1 SANI_2
ASCEN_2 INAT_11	SANI_2
ASCEN_2 INAT_11 SANI_2	CATL_1
ASCEN_2 STOCD_10	CATL_1

Table 9: Rules extracted with ASCEN_2 in α or β

α	β
ASCEN_2 CATL_1 INAT_11 SANI_2 STOCD_10	GARL_1
GARL_1 STOCD_10	ASCEN_2 CATL_1 INAT_11 SANI_2
STOCD_10	ASCEN_2 CATL_1 INAT_11 SANI_2

Table 10: Rules extracted with ASCEN_2 CATL_1 INAT_11 SANI_2 in α or β

α	β
ASCEN_2 CATL_1 GARL_1 INAT_11	SANI_2
ASCEN_2 GARL_1 INAT_11	CATL_1
ASCEN_2 GARL_1 INAT_11	CATL_1 SANI_2
ASCEN_2 GARL_1 INAT_11	SANI_2
ASCEN_2 GARL_1 INAT_11 SANI_2	CATL_1
CATL_1 GARL_1 INAT_11	SANI_2
GARL_1 INAT_11	CATL_1
GARL_1 INAT_11	CATL_1 SANI_2
GARL_1 INAT_11	SANI_2
GARL_1 INAT_11 SANI_2	CATL_1

Table 11: Rules extracted with GARL_1 in α or β

α	β
ASCEN_2 STOCD_10	CATL_1
CATL_1 STOCD_10	SANI_2
SANI_2 STOCD_10	CATL_1
STOCD_10	CATL_1
STOCD_10	CATL_1 SANI_2
STOCD_10	SANI_2

Table 12: Rules extracted with STOCD_10 in α or β

α	β
CATL_1 NA38_ZZ	SANI_2
INAT_11 NA38_ZZ	CATL_1
NA38_ZZ	CATL_1
NA38_ZZ SANI_2	CATL_1

Table 13: Rules extracted with NA38_ZZ in α or β

5 Appendix

Attribute	Brief description
AGER20	Age
COUPLE	Declaration of life as a couple
CS1	Socio-professional category
DIPL_15	Highest diploma
INAT	Nationality
MODV	Lifestyle/household organization
NA38	Economical activity
NPERR	Number of persons
SEXE	Sex
STOCD	Kind of occupation (Proprietary, *etc.*)
TACT	Kind of activity

Table 14: Person-related attributes

Attribute	Brief description
ASCEN	Presence of elevator
CATL	Kind of housing
CMBL	Main fuel
GARL	Parking slots
NBPI	Room number
SANI	Toilets
SURF	Surface

Table 15: Housing-related attributes