Applying the FAIR Principles to Accelerate Health Research in Europe in the Post COVID-19 Era
J. Delgado et al. (Eds.)

doi:10.3233/SHTI210799

Federated Mining of Interesting Association Rules Over EHRs

Carlos MOLINA[a,1], Belen PRADOS-SUAREZ[a] and Beatriz MARTINEZ-SANCHEZ[b]
[a] *Software Engineering Department, University of Granada, Spain*
[b] *Computer Science Department, San Cecilio Hospital, Granada, Spain*

Abstract. Federated learning has a great potential to create solutions working over different sources without data transfer. However current federated methods are not explainable nor auditable. In this paper we propose a Federated data mining method to discover association rules. More accurately, we define what we consider as interesting itemsets and propose an algorithm to obtain them. This approach facilitates the interoperability and reusability, and it is based on the accessibility to data. These properties are quite aligned with the FAIR principles.

Keywords. Electronic Health Records, Data Mining, Privacy, Federated Learning

1. Introduction

Nowadays one of the main issues to achieve a proper health data access are the security and the privacy protection. Federated learning [1] has arisen as a solution to deal with them. In this approach, the data sources involved collaborate to learn a model and share what has been learnt with no need for data transfer. To this purpose they use to distribute the calculation between the sources, working locally over the data, and sharing only the calculated values. With no data transfer, the security and privacy protection can be easily achieved.

Most these methods are based on the optimization of numerical values (e.g. neural networks weights by means of Federated Average [2], or Support Vector Machines planes [3]). In those approaches, normally a local model for each data source is learnt. These models are then combined into a global model. However, these approaches have problems when the data distribution is not uniform between the sources, or when it is necessary to adapt the data distributions (see [1] for more details). Moreover, although they obtain good results, is not possible for the user to understand the underlying mathematical model and why a concrete answer is given.

Explainable Artificial Intelligence [4] methods could solve it, and here is where our proposal lays. To our best knowledge there are no federated proposals of data mining techniques as widely used as the Association Rules [5]. With these techniques the answers can be understood by the user, and it is even possible to audit the results to knowthe reasoning inside the learnt models. Our aim here is to find out which rules have interest, as much if they correspond to frequent cases (that affect a great part of the population) as if they are related to infrequent ones (like rare diseases).

[1] Corresponding Author, Carlos Molina, University of Granada; E-mail: carlosmo@ugr.es. ORCID: Carlos Molina (0000-0002-7281-3065), Bele´n Prados-Suarez (0000-0002-3980-102X).

Having federated methods means a step forward in the interoperability since any of the sources can benefit from the results calculated collaboratively. Even more, having a scheme to apply a similar method over different data structures in different sources supports the reusability. Finally, the aggregated response of the coordinator method offers a homogeneous access to all the underlying data sources, which is an improvement in the accessibility to the information, preserving the privacy. It all makes our proposal quite aligned with the FAIR principles [6].

2. Methods

In this section we first present the parameters needed in the process. Next, the algorithm to extract the association rules to work with the EHR data is explained.

The concept of interesting itemset is different from frequent itemset [7]. The latter one represents those facts that occur together in a great number of the cases studied. However, it doesn't allow to discover itemsets that are highly related that but are not quite numerous, like what happens with rare diseases. The purpose of the interesting itemsets is to model not only what is relevant for being frequent, but also what is relevant for being highly related although its global frequency is low. To model this concept, we define the following measure:

Definition 1 *The* Interest *of the itemset* $\{i_1, \ldots, i_n\}$ *is the function* In *defined as follows:*

$$In([i_1, \ldots, i_n]) = \frac{\sup([i_1, \ldots, i_n])}{\frac{\min\{\sup(i_1), \ldots, \sup(i_n)\}}{\max\{\sup(i_1), \ldots, \sup(i_n)\}}} \in [0, +\infty]$$

The function *In* measures how the relative frequency of the items increases when they appear together. For example, a value $In(its) = 2$ means that the relative frequency of the items doubles when they appear together. In our approach, to consider an itemset *its* it has to be frequent ($sup(its) >= threshold_{sup}$) and interesting ($In(its) >= threshold_{In}$). Normally the *consistency* is used to measure the quality of the association rules [5]; but this measure has problems with very frequent items (see [8] for details). To avoid this issue, we follow the proposal of [8,9] to use the *Certainty factor CF* (see [10] for details). This measure was first proposed for an expert system in medicine. In the case of the association rules, the *CF* avoids the *consistency* problems with very frequent itemsets. We consider an association rule *r* when $CF(r) >= threshold_{CF}$.

Once we have presented the measure to be used in the association rules extraction, we present the federated algorithms to work with EHR data. In our approach, we have two different processes, one for the *coordinator* and another for each *node*.

The *Coordinator* algorithm controls the extraction process (Algorithm 1). First (lines 2-4), the *Coordinator* asks the *nodes* to extract the frequent itemset of size 1. When all the nodes have answered, the coordinator checks if there is an identified frequent itemset which does not appear in the answer of some of the nodes (lines 6-11). In that case, the node without that frequent itemset is asked to give its support, so the global support is correctly calculated (function *supportItemset* in Algorithm 2). Let us note that only the coordinator has the global support of the itemsets, so none of the nodes gets information

Table 1. Results of the experiments (Fi=number of interesting itemsets, AR=number of association rules)

Data dist.	Uniform								Random							
No. of nodes	1		2		3		4		1		2		3		4	
Config.	F_i	AR	F_i	AR	F_i	AR	F_i	AR	F_i	AR	F_i	AR	F_i	AR	F_i	AR
1	8	0	8	0	8	0	8	0	8	0	8	0	8	0	8	0
2	68	107	68	107	68	107	68	107	68	107	68	107	68	107	68	107
3	290	497	290	497	290	497	290	497	290	497	290	497	290	497	290	497
4	243	436	243	436	243	436	243	436	243	436	243	436	243	436	243	436

from the others nodes in the process. In this step, the *In* function is not applied (all the itemsets have only one item so *In* always values 1). This schema is repeated for itemsets of size 2 and more until for some size we get no interesting itemsets (lines 14-36). The main difference is this case is that we can identify the *Interesting itemsets* using the *In* function to reduce the number of itemsets. In lines 26-30 the coordinator calculates the *In* value of the identified frequent itemsets. An itemset *its* is valid if $In(its) >= threshold_{In}$. After this process, the coordinator sends to each node the interesting itemsets to be considered for next step (function *setInterestingItem- set* in Algorithm 2). The nodes, when generating candidates for frequent itemsets of size N, only consider the itemsets that include an interesting itemset of size $N-1$ (line 6 in Algorithm 2). To avoid information transfer, the coordinator only sends to each node the interesting itemsets that have been identified by that node as frequent. With the frequent itemsets calculated, the association rules are generated considering only the rules with a $CF >= threshold_{CF}$ (lines 37-42).

3. Results

To test the proposed algorithm, we used data from COVID-19 patients [11]. In this dataset we have information from 2547 patients. We have tested the methods splitting the data from 1 node (only one node) to 4 nodes, considering uniform distribution and random. In Table 1 we show the results (number of interesting itemsets and association rules) for each of the configurations considering three sets of parameters:

- $Conf_1$=$threshold_{sup} = 0.1, threshpold_{In} = 2, threshold_{CF} = 0.5$;
- $Conf_2$=$threshold_{sup} = 0.05, threshpold_{In} = 5, threshold_{CF} = 0.7$;
- $Conf_3$=$threshold_{sup} = 0.025, threshpold_{In} = 10, threshold_{CF} = 0.7$;
- $Conf_4$= $threshold_{sup} = 0.01, threshpold_{In} = 20, threshold_{CF} = 0.7$.

4. Discussion

The experiments show that the distribution of the data and the number of nodes has no influence on the results. This means that the federated learning process works well independently from the number of nodes and the data distribution.

The proposed method used a synchronous schema of communication. It means that the Coordinator waits for all the nodes to finish each operation. If data distribution is

very unbalanced (e.g. one of the data sources has a really greater amount of data than the others) then the *Coordinator* will wait for that *node* to finish meanwhile the other *nodes* and the *Coordinator* are idle. If one of the *nodes* is very slow, due to computational resources or high workload, the *Coordinator* and other *nodes* will have a similar behaviour (waiting for the slow node to finish its operations).

5. Conclusions

We have presented the need for explainable federated mining methods and we have proposed a federated association rule mining algorithm that works with EHR data. It is able to deal with different number of sources and data distributions without quality loose. We have also defined a measure of the interest of an itemset. These federated techniques require a framework that integrates them to take advantage of their potential. We plan to integrate the proposed methods with the EHRagg, [12]. As we have mentioned in the previous section, the synchronous schema has some problems, so an asynchronous proposal that can build an incremental solution would also be interesting.

Acknowledgements

This research is partially supported by PGC2018-096156-B-I00 Recuperacio´n y Descrip- cion de Ima´genes mediante Lenguaje Natural usando te´cnicas de Aprendizaje Profundo y Computacio´n Flexible of the Ministerio de Ciencia, Innovacio´n.

References

[1] Xu J, Glicksberg BS, Su C, Walker P, Bian J, Wang F. Federated learning for healthcare informatics. Journal of Healthcare Informatics Research. 2021 Mar;5(1):1-9.
[2] McMahan B, Moore E, Ramage D, Hampson S, y Arcas BA. Communication-efficient learning of deep networks from decentralized data. InArtificial intelligence and statistics 2017 Apr 10 (pp. 1273-1282). PMLR.
[3] Brisimi TS, Chen R, Mela T, Olshevsky A, Paschalidis IC, Shi W. Federated learning of predictive models from federated Electronic Health Records. Int J Med Inform. 2018 Apr;112:59-67.
[4] Gunning D, Stefik M, Choi J, Miller T, Stumpf S, Yang GZ. XAI—Explainable artificial intelligence. Science Robotics. 2019 Dec 18;4(37).
[5] Srikant R, Agrawal R. Mining generalized association rules.
[6] Wilkinson MD, Dumontier M, Aalbersberg IJ, Appleton G, Axton M, Baak A, Blomberg N, Boiten JW, da Silva Santos LB, Bourne PE, Bouwman J. The FAIR Guiding Principles for scientific data management and stewardship. Scientific data. 2016 Mar 15;3(1):1-9.
[7] Hu J, Li XY. Association rules mining including weak-support modes using novel measures. WSEAS Transactions on Computers. 2009 Mar 1;8(3):559-68.
[8] Delgado M, Marín N, Sánchez D, Vila MA. Fuzzy association rules: general model and applications. IEEE transactions on Fuzzy Systems. 2003 Apr 8;11(2):214-25.
[9] Marin N, Molina C, Serrano JM, Vila MA. A complexity guided algorithm for association rule extraction on fuzzy datacubes. IEEE Transactions on Fuzzy Systems. 2008 Jun 6;16(3):693-714.
[10] Shortliffe EH, Buchanan BG. A model of inexact reasoning in medicine. Mathematical biosciences. 1975 Apr 1;23(3-4):351-79.
[11] MH HOSPITALES. Covid Data Save Lives. https://www.hmhospitales.com/coronavirus/covid-data-save-lives/english-version, 2021.
[12] Prados-Suárez B, Fernández CM, Yañez CP. Electronic health records aggregators (EHRagg). Methods of Information in Medicine. 2020 May;59(02/03):096-103.

Appendix: Algorithms

Algorithm 1 Coordinator process

```
1: function RUN(nodes, threshold_sup, threshold_In, threshold_CF)
2:     for node_i ∈ nodes do
3:         freq_i^1 = node_i.frequentItemSet(1, threshold_sup)
4:     end for
5:     freq^1 = ∪_{i=1..n} freq_i^1
6:     for node_i ∈ nodes do
7:         its_update = freq^1 − freq_i^1
8:         for its ∈ its_update do
9:             Update support its with node_i.SupportItemset(its)
10:        end for
11:    end for
12:    N = 1
13:    freq_In^N = freq^N
14:    while freq_In^N ☐ = ∅ do
15:        N = N + 1
16:        for node_i ∈ nodes do
17:            freq_i^N = node_i.frequentItemSet(N, threshold_sup)
18:        end for
19:        freq^N = ∪_{i=1..n} freq_i^N
20:        for node_i ∈ nodes do
21:            its_update = freq^N − freq_i^N
22:            for its ∈ its_update do
23:                Update support its with node_i.SupportItemset(its)
24:            end for
25:        end for
26:        for its ∈ freq^N do
27:            if In(its) >= threshold_In then
28:                freq_In^N = freq_In^N ∪ its
29:            end if
30:        end for
31:        for node_i ∈ nodes do
32:            its_In = freq_In^N ∩ freq_i^N
33:            node_i.setInterestingItemset(its_In, N)
34:        end for
35:        freq = freq ∪ freq_In^N
36:    end while
37:    rules_Cand = Generate rules using itemsets in freq
38:    for rule ∈ rules_Cand do
39:        if then CF(rule) >= threshold_CF
40:            result = result ∪ rule
41:        end if
42:    end for
43:    return result
44: end function
```

Algorithm 2 Node process

```
1:  function FREQUENTITEMSET(N, theshold_sup)
2:      if N = 1 then
3:          return freq^1 = frequent local itemsets its of size 1 with sup(its) >= threshold_sup
4:          return freq^1
5:      else
6:          its_cand = Generate itemsets combining freq^{N-1} and freq_In^{N-1}
7:          for its ∈ its_cand do
8:              if Sup(its) >= threshold_sp then
9:                  freq^N = freq^N ∪ its
10:             end if
11:         end for
12:         return freq^N
13:     end if
14: end function
15:
16: function SUPPORTITEMSET(its)
17:     return Sup(its)
18: end function
19:
20: function SETINTERESTINGITEMSET({its_1, ..., its_n}, N)
21:     freq_In^N = {its_1, ..., its_n}
22: end function
```