Artificial Intelligence Research and Development
A. Cortés et al. (Eds.)

doi:10.3233/FAIA220334

Towards and Efficient Algorithm for Computing the Reduced Mutual Information

Martí RENEDO-MIRAMBELL [a] and Argimiro ARRATIA [a,1]
[a] *Soft Computing Research Group (SOCO)*
at Intelligent Data Science and Artificial Intelligence Research Center
Department of Computer Sciences,
Polytechnical University of Catalonia, Barcelona, Spain.
marti.renedo@gmail.com, argimiro@cs.upc.edu

Abstract. In [1], Newman et al. introduced the Reduced Mutual Information (RMI), a measure of the similarity between two partitions of a set useful in clustering and community detection. The computation of RMI requires counting the amount of contingency tables with fixed row and column sums, a #P-complete problem, for which the authors suggest to use analytical approximations that work in general, but for other not so pathological cases these give highly inaccurate approximations. We propose a hybrid scheme based on combining existing Markov chain Monte Carlo methods with analytical approximations to make more accurate estimates of the number of contingency tables in all cases.

Keywords. mutual information, contingency tables, clustering, Markov chain Monte Carlo

1. Introduction

The computation or approximation of the number of contingency tables with fixed row and column sums is a necessary step for the computation of Reduced Mutual Information (RMI). This is a #*P*-complete problem (see e.g. [2]), so we don't have any algorithm to perform the exact computation efficiently, which rules it out for even moderately sized networks. When introducing the RMI [1], Newman et al. suggest using analytical approximations, but they have important limitations. Particularly, they don't give accurate results when row and column sums contain numerous small elements (instead, the approximation is accurate when the contingency tables are very dense). On the other hand, it is possible to use a Markov chain Monte Carlo method, as described in section 2 , but it is much slower to compute. The idea behind our approach is to separate the part of the table for which the analytical formula is accurate, and use that to then obtain the result with fewer steps of the Monte Carlo method.

Reduced Mutual Information. Given r and s two labelings of a set of n elements, the Reduced Mutual Information is defined as:

[1] Corresponding Author: Argimiro Arratia, e-mail: argimiro@cs.upc.edu

$$\mathrm{RMI}(r;s) = I(r;s) - \frac{1}{n}\log\Omega(a,b). \tag{1}$$

where $\Omega(a,b)$ is a integer equal to the number $R \times S$ of non-negative integer matrices with row sums $a = \{a_r\}$ and column sums $b = \{b_s\}$ (i.e., contingency tables). In practice, computing or at least approximating $\Omega(a,b)$ with enough accuracy is the main challenge in obtaining the Reduced Mutual Information of two partitions.

2. Analytical approximation

The following approximation works in cases where the numbers of clusters R and S are relatively small relative to the total number of elements, resulting in very populated clusters. Let a and b vectors of lengths R and S respectively be the margins of the contingency table, and $\Omega(a,b)$ the corresponding number of contingency tables. Also, define: $w = \dfrac{n}{n+\frac{1}{2}RS}$, $x_r = \dfrac{1-w}{R} + \dfrac{wa_r}{n}$, $y_s = \dfrac{1-w}{S} + \dfrac{wb_s}{n}$, $\mu = \dfrac{R+1}{R\sum_s y_s^2} - \dfrac{1}{R}$, and $\upsilon = \dfrac{S+1}{S\sum_r x_r^2} - \dfrac{1}{S}$. Then:

$$\begin{aligned}\log\Omega(a,b) \simeq &(R-1)(S-1)\log(n+\frac{1}{2}RS) + \frac{1}{2}(R+\upsilon-2)\sum_s \log y_s \\ &+ \frac{1}{2}(S+\mu-2)\sum_r \log x_r + \frac{1}{2}\log\frac{\Gamma(\mu R)\Gamma(\upsilon S)}{[\Gamma(\upsilon)\Gamma(R)]^S[\Gamma(\mu)\Gamma(S)]^R}.\end{aligned} \tag{2}$$

However, this approximation can become highly inaccurate when the conditions aren't met. This can easily happen, for example, when a relatively high number of vertices are left isolated forming their own clusters, even if the rest of the clusters are large.

3. Monte Carlo approximation

An alternative approach is to use a Monte Carlo method to estimate the number of contingency tables by successively iterating over the set of solutions using an appropriately defined Markov chain. The method, introduced in [3], uses a nested chain of subsets $\Sigma_{ab} = H_1 \supset H_2 \supset \ldots \supset H_t$. Then, Monte Carlo sampling is used to estimate each ratio $|H_i|/|H_{i+1}|$, which will allow the estimation of the whole set by just being able to enumerate H_t, which will be small (more specifically, it will contain a single element).

Random walk. First let's define a random walk on the set Σ_{ab} of matrices with row sums a and column sums' b. Let $M \in \Sigma_{ab}$. A pair of rows i_1, i_2 and columns j_1, j_2 is selected randomly. Then, $M' \in \Sigma_{ab}$ is obtained by adding 1 to the (i_1,j_1), (i_2,j_2) elements and substracting 1 to the (i_1,j_2), (i_2,j_1) elements, or viceversa, each of the two possibilities with probability $\frac{1}{2}$. This gives a connected, symmetric, aperiodic Markov chain on Σ_{ab}.

Subset chain. Let $M \in \Sigma_{ab}$. Then, define $[\Sigma_{ab}|M;(k,l)]$ the subset of Σ_{ab} containing only tables that match M in all positions strictly preceding (k,l) in the lexocographic order. Then, if (k',l') succeeds (k,l), then $[\Sigma_{ab}|M;(k',l')] \subseteq [\Sigma_{ab}|M;(k,l)]$. This gives

a chain of subsets $\Sigma_{ab} = [\Sigma_{ab}|M;(1,1)] \subseteq ... \subseteq [\Sigma_{ab}|M;(r,s)]$. The following result is proved in [3].

Theorem 3.1 *The random walk on $[\Sigma_{ab}|M;(k,l)]$ is ergodic and has uniform stationary distribution for all $M \in \Sigma_{ab}$.*

4. Hybrid analytical Monte Carlo approximation

We redefine the subset chain of the Markov Monte Carlo method to reduce its length by estimating the size of the biggest subset we can have analytically. We want to concentrate all the denser communities on one corner of the matrix, so a and b are sorted in ascending order. Then, divide the matrix into four blocks Q_1, Q_2, Q_3, Q_4 such that $|(\Sigma_{Q_4})_{ab}|$ can be estimated analytically. Of course, it is not possible to extend this estimation directly using the method described in section 2 because not all elements of Q_1, Q_2 and Q_3 precede those of Q_4 unless Q_4 has only one row.

Order relation. Here we will define an order in which to traverse the matrix M of $R \times S$ elements, or equivalently, a total order relation on the set $[R] \times [S]$. Let $\prec$, and $\preceq$ denote the lexicographical order (the strict and non-strict versions respectively), and $p \in [R] \times [S]$ the element at the lower right corner of Q_1. Then, we define the strict order relation $\sqsubset$ as follows:

$$
\begin{aligned}
&x \sqsubset y \iff x \prec y && \text{if } x,y \in Q_1 \cup Q_2 \\
&x \sqsubset y && \text{if } x \in (Q_1 \cup Q_2), y \in (Q_3 \cup Q_4) \\
&(x_1,x_2) \sqsubset (y_1,y_2) \iff x_2 < y_2 \text{ or} (x_2 = y_2 \text{ and } x_1 < y_1) && \text{if } x_1,y_1 > p_1
\end{aligned}
$$

In other words, $\sqsubset$ puts the elements of Q_1 and Q_2 first in lexicographical order, and then those of Q_3 and Q_4 in a variation of the lexicographical order that goes from left to right and top to bottom in that order. That puts all elements of Q_4 after any element of Q_1, Q_2, and Q_3. We will denote $\sqsubseteq$ the non-strict version of the strict order relation $\sqsubset$.

Hybrid algorithm. With the order relation $\sqsubset$, we can define $[\Sigma_{ab}|M;(k,l)]_{\sqsubset}$ as the subset of Σ_{ab} containing tables that match M in all positions strictly preceding (k,l) in the $\sqsubset$ order. To obtain a random walk on $[\Sigma_{ab}|M;(k,l)]_{\sqsubset}$, we just need to uniformly select a pair of rows $i_1 < i_2 \leq R$ and columns $j_1 < j_2 < S$ such that $(k,l) \sqsubseteq (i_1,j_1)$. Only elements that succeed (k,l) in the $\sqsubseteq$ order will be modified by the random walk. We have

Corollary 4.1 *of theorem 3.1. The random walk on $[\Sigma_{ab}|M;(k,l)]_{\sqsubset}$ is ergodic and has uniform stationary distribution for all $M \in \Sigma_{ab}$.* □

Then, the resulting algorithm can be described as follows:

- Rearrange the rows and columns of M so that their sums are in ascending order.
- Determine $p = (p_1,p_2)$ the position of the upper left corner of Q_4. This is the cutoff point between the small and large communities, here we are using the first row and column with size > 1.
- Estimate the values $|H_1|/|H_2|$, $|H_2|/|H_3|$, ..., $|H_{q-1}|/|H_q|$, where $H_q = [\Sigma_{ab}|M;(p_1,p_2)]_{\sqsubset}$, with the Markov chain Monte Carlo method.
- Approximate H_q with the analytical formula described in section 2.
- Multiply the chain of fractions from the previous steps to obtain $H_1 = \Sigma_{ab}$.

5. Experiments and discussion

To test the standard Markov chain Monte Carlo and the hybrid algorithms, we use two vectors to set the margins of the tables, and execute both. The chosen vectors are: $a = (20,10,10,5,5,1,1,1,1,1,1,1,1,1,1)$ and $b = (10,10,5,5,5,5,5,1,1,1,1,1)$, which correspond to a network of 50 nodes. Even in such a relatively small network, exact counting algorithms are not practical. Using the analytical approximation alone, the results are not meaningful because of the presence of a few isolated vertices, which makes the contingency tables less dense.

For the hybrid method, the matrix is split such that Q_4 sub-matrix is formed by the rows and columns with sum greater than 1. The computation took 24.2 seconds, almost twice as fast as the standard Markov chain Monte Carlo method (41.32 seconds). The term $\frac{1}{n}\log\Omega(a,b)$ estimated with each method differs by less than 0.01, so there is not a significant loss of accuracy when the method is used for the computation of the Reduced Mutual Information. In comparison, using only the analytical formula on the whole matrix produces an estimation that is off by over 0.3, which is clearly too inaccurate to obtain any meaningful estimation of the Reduced mutual Information.

If we instead study a case with fewer single element labels: $a = (25,25,15,10,4,1)$ and $b = (25,20,15,9,8,8,1,1,1)$, the difference is much more apparent with the hybrid method taking 2.98 seconds compared to 37.41 of the standard Monte Carlo.

It is worth noting that the implementation of the Markov chain uses a naive sampling method that doesn't take advantage of the sparsity of the matrix in some areas. When the chosen elements that have to be decreased by one are already 0, the matrix remains invariant for that step of the chain. Then, when the matrix is very sparse and most of the steps are going to be invariant, it is possible to optimize the process by simply simulating the number of invariant steps before the matrix changes with a geometric distribution, and then sampling only from the rows and columns which will result in a step that modifies the matrix. This optimization would be a lot more beneficial on the sparser parts of the matrix (Q_1, Q_2, Q_3) and much less on the Q_4 sub-matrix, which would benefit the hybrid method more than the standard Monte Carlo method.

The implementation of the RMI measure presented here will be released as part of the `clustAnalytics` R package [4], with the goal to provide a readily available tool for cluster analysis on networks.

References

[1] Newman MEJ, Cantwell GT, Young JG. Improved mutual information measure for clustering, classification, and community detection. Phys Rev E. 2020 Apr;101:042304. Available from: https://link.aps.org/doi/10.1103/PhysRevE.101.042304.

[2] Dyer M, Kannan R, Mount J. Sampling contingency tables. Random Structures & Algorithms. 1997;10(4):487-506.

[3] Diaconis P, Gangolli A. Rectangular Arrays with Fixed Margins. In: Aldous D, Diaconis P, Spencer J, Steele JM, editors. Discrete Probability and Algorithms. Springer New York; 1995. p. 15-41.

[4] Renedo-Mirambell M. clustAnalytics: Cluster Evaluation on Graphs; 2022. R package version 0.3.1. Available from: https://CRAN.R-project.org/package=clustAnalytics.