FastAMI — a Monte Carlo Approach to the Adjustment for Chance in Clustering Comparison Metrics

Kai Klede, Leo Schwinn, Dario Zanca, Björn Eskofier

Machine Learning and Data Analytics Lab,
Friedrich-Alexander Universität Erlangen-Nürnberg (FAU)
{kai.klede, leo.schwinn, dario.zanca, bjoern.eskofier}@fau.de

Abstract

Clustering is at the very core of machine learning, and its applications proliferate with the increasing availability of data. However, as datasets grow, comparing clusterings with an adjustment for chance becomes computationally difficult, preventing unbiased ground-truth comparisons and solution selection. We propose FastAMI, a Monte Carlo-based method to efficiently approximate the Adjusted Mutual Information (AMI) and extend it to the Standardized Mutual Information (SMI). The approach is compared with the exact calculation and a recently developed variant of the AMI based on pairwise permutations, using both synthetic and real data. In contrast to the exact calculation our method is fast enough to enable these adjusted information-theoretic comparisons for large datasets while maintaining considerably more accurate results than the pairwise approach.

Introduction

Clustering comparison measures such as the *mutual information* or the *Rand index* are most commonly used to validate clusterings when ground truth information is available (Aggarwal and Reddy 2014). In the context of graphs, the mutual information of the neighborhoods of two nodes indicates their similarity and can predict missing links (Hoffman, Steinley, and Brusco 2015; Shakibian and Moghadam Charkari 2017). Another application is multiple-clustering algorithms, which measure the mutual information or one of its variants to identify multiple qualitatively different clustering solutions for a single dataset (Müller et al. 2010; Wei et al. 2021). Other uses include categorical feature selection, where each feature is understood as a cluster or for the solution selection in consensus clustering (Lancichinetti and Fortunato 2012).

A well-known problem with these clustering comparison measures is that they do not assume a constant baseline value when comparing two random clustering partitions. Instead, they tend to be larger when the number of clusters approaches the number of data points (Nguyen, Epps, and Bailey 2009), leading to a bias towards smaller clusters when used as an external validation criterion. To obtain a constant baseline, these measures have been adjusted by their expectation value under random permutations of the cluster labels in the *Adjusted Rand index* (ARI) (Hubert and Arabie 1985) and the *Adjusted Mutual information* (AMI) (Nguyen, Epps, and Bailey 2010).

Romano et al. (2016) created an overview of those clustering comparison measures mentioned above and others. They analyzed the relationships between these measures and formulated a guideline for which one to use in which situation. Quoting directly from the abstract, the authors conclude that

> ARI should be used when the reference clustering has large equal-sized clusters; AMI should be used when the reference clustering is unbalanced and there exist small clusters.

However, it has been shown that the computational complexity of the adjusted mutual information is $\mathcal{O}(\max(R, C)N)$, where R, C are the numbers of clusters in the clusterings to be compared and N is the number of data points (Romano et al. 2014). For large datasets with many small clusters and few bigger clusters, which is exactly the domain where the AMI should be used, its computation becomes difficult and often impractical. Many datasets originating from different domains exhibit these characteristics, such as social networks, collaboration networks, or web graphs (Leskovec and Krevl 2014). In practice, many authors resort to the non-adjusted normalized mutual information (Tian et al. 2019; Chunaev 2020; Karim et al. 2021), which is faster to compute, but does not account for random coincidences.

As a workaround, Lazarenko and Bonald (2021) proposed to consider only pairwise permutations in the adjustment for chance, i.e., to only adjust for clusterings where two individual samples exchanged their cluster labels. While this approach allows for faster computations ($\mathcal{O}(RC)$), these pairwise permutations lead to a higher amount of shared information. In Figure 1 we show that the actual *expected mutual information* (EMI) is overestimated by the pairwise EMI. Consequently, the pairwise AMI systematically underestimates the AMI, and the results cannot be compared with existing values. Even the relative order of cluster similarity is affected, as shown in Table 1.

With FastAMI, we propose a Monte Carlo-based approach to enable AMI comparisons for large datasets with small clusters[1]. The presented method allows for fine-grained control over its accuracy and addresses the short-

[1]Code at https://github.com/mad-lab-fau/fastami-benchmark

comings of the pairwise approach. We prove a pessimistic upper bound to the expected runtime that is on par with the pairwise approach. We demonstrate, on synthetic and real data, that FastAMI is considerably faster in practice while producing more accurate results. FastAMI also works when both the exact and the pairwise implementation fail due to memory requirements or impractical runtimes. In the process, we identify and fix a common flaw in synthetic clustering comparison benchmarks.

While the AMI is unbiased when comparing two random clusterings, it has been shown that it still exhibits a bias towards selecting larger clusters when comparing multiple random clusterings against a fixed ground truth (Romano et al. 2014). As a solution, it was suggested to account also for the variance in the *Standardized Mutual Information* (SMI), which, according to the authors, should be employed when $N/RC < 5$ and more than three clusterings are compared. However, adoption of the SMI is lagging behind the AMI, and a potential reason might be its steep computational cost $\mathcal{O}(RCN^3)$. We extend our Monte Carlo-based approach to the SMI, making it a computationally feasible alternative for small to medium-sized datasets.

Background and Related Work

Let S be a set of N data points $\{s_1, s_2, \dots s_N\}$, and two clusterings the surjections $u\colon S \to \{1, 2, \dots, R\}$ and $v\colon S \to \{1, 2, \dots, C\}$ that partition the dataset S into R and C clusters respectively. We denote with $a_i \coloneqq |u^{-1}(i)|$, $b_j \coloneqq |v^{-1}(j)|$ the cluster sizes and $n_{ij} \coloneqq |u^{-1}(i) \cap v^{-1}(j)|$ the number of shared data points between two clusters.

The *mutual information* (Edwards 2008) of the two clusterings is given by

$$I(u, v) \coloneqq \sum_{i=1}^{R} \sum_{j=1}^{C} \frac{n_{ij}}{N} \log \left(\frac{N \cdot n_{ij}}{a_i \cdot b_j} \right), \tag{1}$$

where $n_{ij}/N \log\left(N n_{ij}/(a_i b_j)\right)$ is defined to be zero for $n_{ij} = 0$. For example, the mutual information of two clusterings that assign every point in the dataset to a single cluster is zero. In the other extreme, when every point is an individual cluster in both clusterings, the mutual information is maximal $\log N$. These simple examples already illustrate a problem for the use as a cluster evaluation metric: The mutual information is expected to increase with the number of clusters (Figure 1), which induces a bias towards more granular clusterings.

Therefore, the mutual information is adjusted with its expected value under random permutations of the cluster assignments, while the number and size of the clusters $A = \{a_1, \dots a_R\}$, $B = \{b_1, \dots b_C\}$ is kept constant (*random permutation model*) (Nguyen, Epps, and Bailey 2009)

$$\begin{aligned} \mathrm{E}\{I \mid A, B\} = &\sum_{i=1}^{R} \sum_{j=1}^{C} \sum_{\substack{n_{ij}=\max(0,\\ a_i+b_j-N)}}^{\min(a_i,b_j)} \\ &\frac{n_{ij}}{N} \log \left(\frac{N n_{ij}}{a_i b_j} \right) P\{n_{ij} | a_i, b_j, N\}. \end{aligned} \tag{2}$$

Here $P\{n_{ij} | a_i, b_j, N\}$ denotes the probability that n_{ij} of the a_i data points in the cluster $u^{-1}(i)$ also lie in the cluster $v^{-1}(j)$ of size b_j after random permutation and is given by the hypergeometric distribution.

The *adjusted mutual information* can subsequently be defined as

$$\mathrm{AMI}(u, v) \coloneqq \frac{I(u, v) - \mathrm{E}\{I | A, B\}}{\mathrm{avg}(H(u), H(v)) - \mathrm{E}\{I | A, B\}}, \tag{3}$$

where $\mathrm{avg}(H(u), H(v))$ is an upper bound to the mutual information given A, B that serves as a normalizer, with the entropy

$$H(u) \coloneqq \sum_{i=1}^{R} \frac{a_i}{N} \log \frac{a_i}{N}. \tag{4}$$

The arithmetic and geometric mean, the minimum, and maximum can be used for avg (Nguyen, Epps, and Bailey 2010). In this work, we choose the arithmetic mean following the default choice of `sklearn` (Pedregosa et al. 2011).

To reduce the computational complexity of the expected mutual information, Lazarenko and Bonald (2021) restricted its computation to pairwise permutations only

$$\mathrm{EMI}_{\text{pair}}(u, v) \coloneqq \mathrm{E}\{I(u, \sigma \circ v) \mid \sigma \in S_N : \sigma \text{ pairwise}\}. \tag{5}$$

A permutation $\sigma \in S_N$ is pairwise when there exist $i, j \in \{1, \dots, N\}$ such that $\sigma(i) = j$, $\sigma(j) = i$, and $\sigma(t) = t\ \forall t \neq i, j$. While the original authors defined the pairwise AMI without normalization, we normalize it as in equation 3 to enable a direct comparison with the exact results.

The AMI has a constant baseline when comparing two clusterings directly. However, when comparing clusterings via a constant ground truth reference, the AMI is biased towards larger numbers of clusters (Romano et al. 2014). The definition of the *standardized mutual information* as

$$\mathrm{SMI}(u, v) \coloneqq \frac{I(u, v) - \mathrm{E}\{I \mid A, B\}}{\sqrt{\mathrm{Var}\{I \mid A, B\}}} \tag{6}$$

accounts for that bias. The SMI indicates by how many standard deviations two clusterings deviate from each other under the hypothesis of random and independent clusterings.

Method

The AMI exhibits poor runtimes, mainly because the EMI is computationally expensive. The calculation of $\mathrm{Var}\{I | A, B\}$ for the SMI is even more demanding. Therefore we approximate the EMI and the variance of the mutual information via Monte Carlo sampling.

Monte Carlo Estimate of the EMI

We observe that the shared information of two clusters, i.e., the summands in Equation 2, is fully determined by the cluster sizes a, b, their overlap n and the total number of samples N. Hence, the expected mutual information can be reformulated in terms of the probability $P\{a|A\}$ of a cluster size a in the marginals A

$$\begin{aligned} E\{I|A, B\} = &RC \sum_{a} \sum_{b} P\{a|A\} P\{b|B\} \\ &\sum_{n} \frac{n}{N} \log \left(\frac{Nn}{ab} \right) P\{n|a, b, N\}. \end{aligned} \tag{7}$$

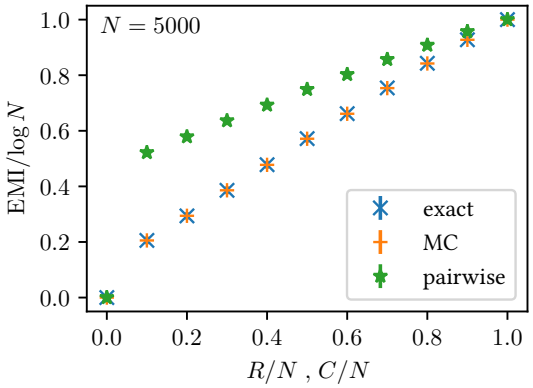

Figure 1: The mutual information is expected to increase as the number of clusters $R = C$ in the clusterings to be compared approaches the number of data points N. The expected value under pairwise permutations overestimates the expected mutual information, while our Monte Carlo (MC) approach reproduces the exact results. The figure shows the average values for 200 pairs of clustering marginals A, B, where each marginal is chosen uniformly at random from all partitions of $N = 5000$ into $R = C$ parts.

We can now proceed to draw samples for a, b and n and approximate the sum through the sample mean. We can readily generate variates for a and b in constant time via Walker's method of alias (Walker 1974, 1977) after an initial setup cost of $\mathcal{O}(R \log R + C \log C)$ for determining the cluster size frequencies of A and B.

Directly drawing the overlap of two clusters n from the hypergeometric distribution would lead to a high occurrence of no overlap $n = 0$ in the case of small cluster sizes. However, these terms do not contribute to the mutual information. To exploit that sparsity, we absorb the linear term of the mutual information into the probability density function

$$nP\{n|a, b, N\} = n\frac{\binom{b}{n}\binom{N-b}{a-n}}{\binom{N}{a}}. \tag{8}$$

The binomial coefficients are rewritten as

$$n\binom{b}{n} = \frac{b!}{(n-1)!\,(b-n)!} = b\binom{b-1}{n-1}, \tag{9}$$

allowing us to draw variates $n-1$ according to the hypergeometric distribution with $a-1, b-1$ and $N-1$

$$\begin{aligned} nP\{n|a, b, N\} =& \frac{a \cdot b}{N}\frac{\binom{b-1}{n-1}\binom{N-b}{a-n}}{\binom{N-1}{a-1}} \\ =& \frac{a \cdot b}{N}P\{n-1|a-1, b-1, N-1\}. \end{aligned} \tag{10}$$

Several methods to draw hypergeometric variates in on average constant time exist (Kachitvichyanukul and Schmeiser 1985; Hörmann 1994), here we chose a ratio of uniforms approach introduced by Stadlober (1990). The approximation of Equation 7 terminates, when the relative error estimate of the EMI reaches a desired precision $s_{\text{EMI}}/\text{EMI} \leq p$. To ensure termination when the EMI approaches zero, the algorithm switches to an absolute error criterion $s_{\text{EMI}} \leq p$ when $\text{EMI} < 1$.

Summarizing all the steps above, we formulate Algorithm 1, a Monte Carlo estimate for the EMI. Samples can be drawn in constant time after an initial setup cost of $\mathcal{O}(R \log R + C \log C)$, such that in practice, the number of Monte Carlo samples governs the runtime.

Algorithm 1 The core of FastAMI is a Monte Carlo approximation to the expected mutual information. We use a numerically stable variant of Welford's online algorithm for calculating the mean and sample variance and define a stopping criterion when the desired precision is reached.

Require: Cluster sizes A, B, EMI precision p, minimum number of samples $i_{\text{min}} > 1$
Ensure: $\text{EMI} \sim \mathcal{N}\left(E\{I|A, B\}, s^2_{\text{EMI}}\right)$ where $\min\{s_{\text{EMI}}/\,\text{EMI}, s_{\text{EMI}}\} \leq p$
$N \leftarrow \sum_{a \in A} a$
$i \leftarrow 0,\ \text{EMI} \leftarrow 0,\ M_2 \leftarrow 0$
Setup Walker random sampling for A, B
while $i < i_{\text{min}}$ or $M_2 > (p \max\{1, \text{EMI}\})^2 \cdot i \cdot (i-1)$ **do**
  $i \leftarrow i + 1$
  Draw a, b via Walker's Alias method
  Draw $m \sim P\{m|a-1, b-1, N-1\}$ via Stadlober's method
  $x \leftarrow ab \log\left[(m+1)N/(ab)\right]$
  $\Delta_0 \leftarrow x - \text{EMI}$
  $\text{EMI} \leftarrow \text{EMI} + \Delta_0/i$
  $\Delta_1 \leftarrow x - \text{EMI}$
  $M_2 \leftarrow M_2 + \Delta_0\Delta_1$
end while
$s_{\text{EMI}} = \sqrt{M_2/(i \cdot (i-1))}$

Upper Bound to the Expected EMI Runtime

Given that samples can be drawn on average in constant time, we derive an asymptotic upper bound to the expected number of samples required for convergence of the EMI.

Theorem 1. *The expected number of samples required for Algorithm 1 with precision p to terminate on two clusterings with R and C clusters and $N \geq 3$ data points is asymptotically bounded by $\mathcal{O}\left(RC/p^2\right)$.*

Proof. The absolute error of the Monte Carlo approximation s_{EMI} for a given number of samples N_{samples} is

$$s^2_{\text{EMI}} = \frac{\text{Var}\left\{\frac{RC}{N^2} ab \log\left(\frac{Nn}{ab}\right)|A, B\right\}}{N_{\text{samples}}}, \tag{11}$$

in the limit of many samples, according to the central limit theorem. Conversely, the expected number of samples to reach a particular error is

$$N_{\text{samples}} = \frac{\text{Var}\left\{\frac{RC}{N^2} ab \log\left(\frac{Nn}{ab}\right)|A, B\right\}}{s^2_{\text{EMI}}}. \tag{12}$$

The Bhatia-Davis inequality (Bhatia and Davis 2000) provides an upper bound to the variance when the variates are bounded $m \leq \frac{RC}{N^2} ab \log\left(\frac{N \cdot n}{ab}\right) \leq M$

$$\mathrm{Var}\left\{\frac{RC}{N^2} ab \log\left(\frac{Nn}{ab}\right) | A, B\right\} \leq (M - \mathrm{EMI})\,(\mathrm{EMI} - m)\,. \tag{13}$$

Equality holds when the probability mass function is concentrated on the extremes of the interval. The information contained in the overlap of two clusters is larger or equal than zero ($m = 0$), where equality is reached when one of the clusterings has only a single cluster $a = N$ or $b = N$. For the upper bound, note that the overlap of two clusters is maximally the size of the smaller cluster $n \leq \min\{a, b\}$. Without loss of generality, assume $a \leq b$

$$ab \log \frac{nN}{ab} \leq Nb \log \frac{N}{b} \leq \frac{N^2}{e} = M \quad \text{for } N \geq e, \tag{14}$$

With Equation 13 this gives the following upper bound to the expected number of required Monte Carlo samples

$$N_{\text{samples}} \leq \frac{(RC - \mathrm{EMI})\,\mathrm{EMI}}{s^2_{\mathrm{EMI}}} \tag{15}$$

The convergence criterion in Algorithm 1 requires $s_{\mathrm{EMI}} \leq p$ for $\mathrm{EMI} \leq 1$ and $s_{\mathrm{EMI}} \leq p\,\mathrm{EMI}$ for $\mathrm{EMI} > 1$, such that

$$N_{\text{samples}} \leq \mathcal{O}\left(\frac{RC}{p^2}\right). \tag{16}$$

□

This amounts to the same asymptotic runtime as the pairwise adjusted EMI for a fixed precision p. However, the Bhatia-Davis inequality gives only a crude theoretical limit and the experiments will shed more light on the algorithm's runtime.

Monte Carlo Estimation of the Variance

We compare two approaches for approximating the variance of the mutual information:

- *Separate Monte Carlo* - We split the variance into two terms $\mathrm{Var}\{I \mid A, B\} = \mathrm{E}\{I \mid A, B\}^2 - \mathrm{E}\{I^2 \mid A, B\}$, where the first term is approximated as in Algorithm 1. We apply Equation 10 and related identities to the explicit formula for $\mathrm{E}\{I^2 \mid A, B\}$ as given in theorem 1 in (Romano et al. 2014), and approximate it via an analogous algorithm.
- *Direct Monte Carlo* - Instead of generating individual values for n_{ij}, we randomly sample the space of all contingency matrices $(n_{ij})_{i \in \{1,\dots,R\}}^{j \in \{1,\dots,C\}}$ at once with a method developed by Patefield (1981) and observe the mutual information. The estimators for the EMI and the variance are then simply the sample mean and variance.

The direct Monte Carlo approach can also be applied to the EMI. However, the whole contingency matrix would be kept in memory. This is not practical for the approximation of the AMI, since memory footprint was one of the limiting factors we optimized (see Table 2).

Experiments

In the previous section, we introduced a method to approximate the EMI and variance via Monte Carlo sampling. We now proceed to evaluate how quickly these methods converge. Different parameter regimes for R, C, and N are explored using synthetic data, and we demonstrate the practical use of the method on real datasets from a wide range of fields.

A Laptop with an Intel i7-10750H with 32 GB RAM was used for the synthetic experiments and the experiments on the *Benchmark Suite for Clustering Algorithms*. The experiments with the larger datasets from the *Stanford Large Network Dataset Collection* were performed on an Intel Xeon E5-2680 v4 system with 512 GB RAM.

Synthetic Data

Previous works usually generated random clusterings by assuming a fixed number of clusters and then assigning each data point to a random cluster uniformly (Nguyen, Epps, and Bailey 2009; Romano et al. 2014). This method overemphasizes balanced clusterings, where all clusters have similar sizes. Choosing a random probability distribution instead of the uniform assignment allows more variance in the cluster sizes (Lazarenko and Bonald 2021). However, both methods do not guarantee the fixed number of clusters they were intended to generate. In the first case, exactly one cluster gets assigned zero data points with a probability of $C\left((C-1)/C\right)^N$ and using the inclusion-exclusion principle, the probability of at least one empty cluster is

$$P\{\exists i : i \notin \operatorname{Im} v\} = \sum_{i=1}^{C} (-1)^{i-1} \binom{C}{i} \left(\frac{C-i}{C}\right)^N. \tag{17}$$

$\operatorname{Im} v$ denotes the image of clustering v, i.e. the set of all labels. For a fixed number of data points N, the chance of an empty cluster grows with the number of clusters, such that for example for $N = 100$ and $C = 30$ this probability is already $\approx 66.5\,\%$. This chance is even amplified when selecting the probabilities at random since cluster labels can have arbitrarily low probability.

Instead, we propose a maximum entropy approach: Given a fixed number of data points N and fixed numbers of clusters R, C, we randomly sample the space of all possible cluster size distributions A and B that fulfill these constraints, i.e., we uniformly sample the integer partitions of R and C. We use a rejection sampling method that was outlined by Arratia and Desalvo (2016) and originally developed by Fristedt (1993). Sampling cluster size distributions is enough for the EMI, which depends only on A, B, whereas the SMI depends on two concrete clusterings. In this case, we create an ordered cluster from the marginals and shuffle it randomly, guaranteeing fixed R, C.

EMI The computationally demanding part of computing the adjusted mutual information is obtaining the expected value of the mutual information (EMI). Therefore, we compare the methods to compute the EMI for $R = C$ and a fixed $N = 5000$. We focus on $R = C$ since the asymptotic

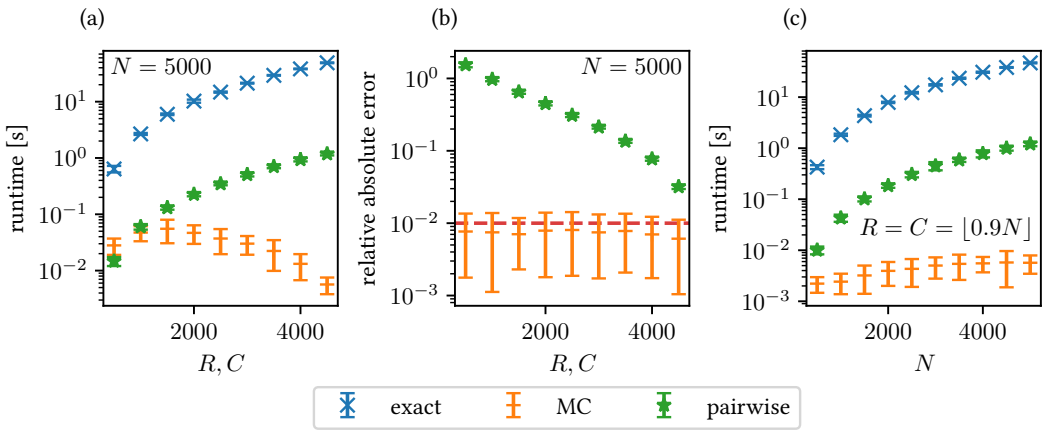

Figure 2: We compute the EMI of 200 pairs of random cluster size distributions with $R = C$ clusters and N data points, comparing three different methods. (a) The exact EMI as implemented in `sklearn` (Pedregosa et al. 2011) and the pairwise EMI take increasingly longer to run as the number of clusters $R = C$ approaches the number of data points N. Our Monte Carlo method not only outperforms the latter two, but it performs best for larger numbers of clusters. (b) The Monte Carlo implementation reaches the prescribed precision of 0.01 (dashed, red line) on average, while the pairwise adjustment introduces model-dependent errors. In (c) we repeat the experiment but now varying N whilst fixing $R = C = \lfloor 0.9N \rfloor$. The performance gap grows with increasing dataset size.

runtime for the pairwise and our Monte Carlo algorithm increases the most for that configuration. Both the exact calculation and the pairwise EMI exhibit a nearly 100-fold increase in runtime, as the number of clusters increases from 500 to 4500 (Figure 2a). This slowdown is problematic since it is the regime where the AMI is preferable to the Rand Index, as stated by Romano et al. (2016). On the other hand, our Monte Carlo method performs best in the case of many clusters. The reason is that when smaller clusters dominate as $R = C$ increases, the variance of the number of shared data points n between clusters decreases, and fewer Monte Carlo samples are required for convergence. In the case of only a few larger clusters, there are fewer cluster size values the variates a and b can assume, and hence the runtime peaks somewhere between those extremes. This empirical result confirms that the upper bound in theorem 1 overestimates the runtime.

In the regime of many singleton clusters, i.e., high R, C, many permutations do not affect the contingency matrix (n_{ij}) and hence the mutual information. These permutations contribute to the slowdown of the exact and the pairwise approach, but it also means that leaving them out only slightly affects the pairwise EMI and the relative error is low in that regime (Figure 2b). On the other hand, the pairwise approach has high model-dependent errors, where the method is fast. Our approach allows for tunable relative errors across the parameter range (dashed red line in Figure 2b).

In a second experiment, we show that the runtime of the exact and pairwise method gets even worse as the number of data points N increases (Figure 2c).

SMI We compare the exact SMI implemented by Romano et al. (2014) with the separate and direct Monte Carlo approach for estimating the variance. We terminate the calculation when the estimated absolute or relative error estimate

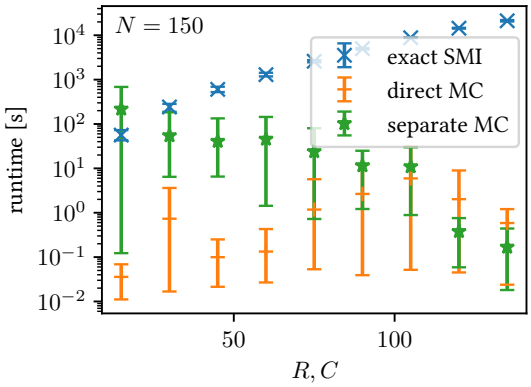

Figure 3: We compute the SMI of 10 pairs of clusterings with a given number of clusters $R = C$ to a precision of 0.1. Both Monte Carlo approaches outperform the exact calculation, but the direct approach is multiple orders of magnitude faster in the regime of many smaller clusters.

of the SMI is below the precision $p = 0.1$. Both approximate algorithms outperform the exact SMI, but the separate approach slows down as the number of clusters increases. In contrast, the direct approach provides a speed-up of multiple orders of magnitude across the whole parameter range (Figure 3). A possible explanation could be that the direct approach naturally incorporates the constraint that the variance is larger or equal than zero. The sample variance of the mutual information is always positive, but the difference of two separate estimators for $\mathrm{E}\{I \mid A, B\}^2$ and $\mathrm{E}\{I^2 \mid A, B\}$ can also be negative. In the following we chose the direct approach for FastSMI.

Variant	Result returned	Mean runtime [s]	Spearman (incomplete)
sklearn AMI	$100\,\%$	$3.6 \cdot 10^{-3}$	-
FastAMI	$100\,\%$	$1.3 \cdot 10^{-2}$	1.000
pairwise AMI	$100\,\%$	$1.6 \cdot 10^{-3}$	0.577
exact SMI	$38.3\,\%$	4.8	-
FastSMI	$99.2\,\%$	$2.4 \cdot 10^{-2}$	(0.997)

Table 1: Benchmark results on the *Benchmark Suite for Clustering Algorithms* (Gagolewski et al. 2020) for FastAMI with precision $p = 0.01$ and FastSMI with precision $p = 0.1$. Both pairwise AMI and FastAMI improve the runtime of the exact version, and FastAMI nearly perfectly matches the relative ranking of the exact result. However, with a maximum R/N of 0.03, the Benchmark suite is not in the imbalanced domain where the exact AMI is impractical (Figure 2a). The exact SMI timed out after $20\,\mathrm{s}$ on $61.7\,\%$ of the $66\,004$ comparisons in the Benchmark (column *result returned*). FastSMI returned more results before the timeout in less time per comparison and achieved a high Spearman correlation with the exact results where they were available.

Real Data

The previous section demonstrated how Monte Carlo approximation speeds up the EMI and SMI for random clusterings. This section shows how that speed-up translates to the practically more relevant AMI, on clusterings that arise from real datasets of various domains.

Lazarenko and Bonald (2021) used 79 datasets from the *Benchmark Suite for Clustering Algorithms* (Gagolewski et al. 2020) to compare the solutions of several clustering algorithms via the AMI and pairwise AMI. They used the Spearman rank correlation to measure the quality of the results. As a first benchmark, we reproduce these results in Table 1 and extend them to evaluate FastAMI. While our method has a significantly higher Spearman correlation with the exact solution than the pairwise AMI, it was slower compared to the pairwise approach. For the SMI on the other hand, $61.7\,\%$ of the comparisons timed out after $20\,\mathrm{s}$. FastSMI ($p = 0.1$) completed $99.2\,\%$ of the benchmark with a high Spearman correlation with the exact results, enabling SMI comparisons for medium sized datasets. However, with an average runtime of $0.013\,\mathrm{s}$ per comparison, the benchmark suite is not in the domain where an approximation of the AMI is necessary due to runtime constraints.

Instead, we propose a benchmark based on a collection of large real-world datasets, taken from the *Stanford Large Network Dataset Collection* (Leskovec and Krevl 2014). We select the datasets designated to community detection from that collection (See Table 2) and apply the following six methods to find clusterings (Staudt, Sazonovs, and Meyerhenke 2016): 1) Connected Components, 2) Degree Ordered Label Propagation, 3) Label Propagation, 4) Leiden, 5) Louvain, and 6) Louvain Map Equation.

The benchmark then consists of a pairwise comparison of the six clusterings, using AMI, pairwise AMI, and FastAMI. We measure the total runtime and peak memory consumption for the 15 comparisons in each dataset and limit every individual comparison to a maximum of $2000\,\mathrm{s}$ and $503.6\,\mathrm{GiB}$ of memory. If any of these limits is exceeded, we report the respective lower bound in Table 2 and omit the result from further calculations. Following Lazarenko and Bonald (2021), we report the Spearman rank correlation between the exact results and the respective approximations, excluding the cases where the exact calculation was aborted. Additionally, we report the mean absolute error where exact values are available and substitute it with the according Monte Carlo estimate otherwise.

The AMI is preferable to the ARI when the clusterings are imbalanced (Romano et al. 2016). In the synthetic experiments, we only fixed the number of clusters without explicitly specifying the imbalance. For the real datasets, we measure the balance using the mean normalized entropy

$$\mathrm{Balance}(U) = \sum_{u_k \in U} \frac{H(u_k)}{|U| \log |\operatorname{Im} u_k|}, \tag{18}$$

where U denotes the set of clustering solutions, obtained via the six different methods for the same dataset. A value of one indicates that all the clusterings are perfectly balanced, i.e., every cluster i within a clustering u has the same size. A value of zero on the other hand means that all clusterings consist of a single large cluster and are thus maximally imbalanced.

As expected from the synthetic experiments, our method outperforms the others in terms of runtime. FastAMI also has a much lower memory footprint than the `sklearn` implementation and the pairwise AMI, since it does not keep the (sparse) contingency table in memory. While FastAMI does slow down for datasets with more nodes N, it does not struggle with imbalanced datasets in contrast to `sklearn` and the pairwise AMI. FastAMI performs comparably to the pairwise adjustment regarding the Spearman correlation. The real benefit in terms of quality is that the FastAMI results can directly be compared with the traditional AMI as demonstrated by the mean absolute error, whereas the pairwise AMI is systematically lower and somewhat harder to interpret.

Conclusion

This paper presents an effective and practical method for approximating the Adjusted Mutual Information. FastAMI takes advantage of the sparsity and low variance of contingency tables of clusterings, enabling AMI comparisons on datasets that were computationally inaccessible before. We analyzed the behavior of our approximation scheme based on synthetic data and demonstrated state-of-the-art performance on a suite of real datasets. In future work, one could readily parallelize Algorithm 1 to improve its performance further. We extended our algorithm to the standardized mutual information, rendering variance-adjusted clustering comparisons computationally feasible for moderately sized datasets.

Dataset	Nodes	Balance	Method	Result returned	Total time [s]	Peak memory	MAE (est.)	Spearman (incompl.)
Email	$1.0 \cdot 10^3$	$3.5 \cdot 10^{-2}$	sklearn	15/15	0.6	208 KiB	-	-
			pairwise	15/15	**0.4**	184 KiB	0.357	0.943
			FastAMI	15/15	0.6	**164** KiB	**0.001**	**0.989**
DBLP	$3.2 \cdot 10^5$	$4.3 \cdot 10^{-3}$	sklearn	13/15	> 7539.9	-	-	-
			pairwise	15/15	361.0	89.6 GiB	(0.310)	(0.972)
			FastAMI	15/15	**4.5**	**106.4** MiB	(**0.003**)	(**0.986**)
Amazon	$3.3 \cdot 10^5$	$3.8 \cdot 10^{-3}$	sklearn	13/15	> 8988.4	-	-	-
			pairwise	15/15	329.9	57.6 GiB	(0.421)	(0.966)
			FastAMI	15/15	**4.5**	**26.8** MiB	(**0.005**)	(**0.972**)
Youtube	$1.1 \cdot 10^6$	$2.4 \cdot 10^{-4}$	sklearn	9/15	$> 15\,967.7$	-	-	-
			pairwise	13/15	-	> 503.6 GiB	(0.260)	(0.913)
			FastAMI	**15/15**	**7.4**	**119.4** MiB	(**0.005**)	(**0.957**)
Wikipedia	$1.8 \cdot 10^6$	$2.6 \cdot 10^{-2}$	sklearn	15/15	714.3	12.0 GiB	-	-
			pairwise	15/15	32.7	8.4 GiB	0.210	**0.975**
			FastAMI	15/15	**7.8**	**140.7** MiB	**0.001**	0.973
Orkut	$3.1 \cdot 10^6$	$2.7 \cdot 10^{-2}$	sklearn	15/15	2072.0	27.3 GiB	-	-
			pairwise	15/15	69.9	19.2 GiB	0.276	0.967
			FastAMI	15/15	**15.4**	**341.0** MiB	**0.001**	**0.991**
LiveJournal	$4.0 \cdot 10^6$	$2.7 \cdot 10^{-4}$	sklearn	10/15	$> 14\,496.1$	-	-	-
			pairwise	13/15	-	> 503.6 GiB	(0.149)	(0.937)
			FastAMI	**15/15**	**23.9**	**340.8** MiB	(**0.003**)	(0.937)
Friendster	$6.6 \cdot 10^7$	$1.3 \cdot 10^{-5}$	sklearn	0/15	-	> 503.6 GiB	-	-
			pairwise	6/15	-	> 503.6 GiB	(0.031)	-
			FastAMI	**15/15**	**439.3**	**4.6** GiB	(**0.003**)	-

Table 2: Benchmark of the AMI (`sklearn`), the pairwise AMI and FastAMI, regarding time and memory complexity. We evaluated these methods on the results of six different community detection algorithms for large graph datasets, taken from the Stanford Large Network Dataset Collection (Leskovec and Krevl 2014). In the column *result returned*, we report how often an algorithm successfully terminated and how often it exceeded the time or memory limit of 2000 s and 503.6 GiB. Our method outperforms `sklearn` and Lazarenko's pairwise algorithm both in the runtime and the memory footprint, enabling AMI comparisons for dataset sizes that were previously inaccessible. While the pairwise AMI is comparable in terms of Spearman correlation, our method has the additional benefit of tunable absolute errors, allowing for a direct comparison with the exact metric. The mean absolute error is given as Monte Carlo estimate and the Spearman correlation was calculated on a subset when the exact solution was not available.

Acknowledgments

This work was funded by the *Bayerischen Verbundförderprogramm (BayVFP) – Förderlinie Digitalisierung – Förderbereich Informations- und Kommunikationstechnik* of the Bavarian Ministry of Economic Affairs, Regional Development and Energy and supported by *Bayern Innovativ – Bayerische Gesellschaft für Innovation und Wissenstransfer mbH*.

References

Aggarwal, C. C.; and Reddy, C. K., eds. 2014. *Data Clustering: Algorithms and Applications*. CRC Press. ISBN 978-1-46-655821-2.

Arratia, R.; and Desalvo, S. 2016. Probabilistic Divide-and-Conquer: A New Exact Simulation Method, With Integer Partitions as an Example. *Comb. Probab. Comput.*, 25(3): 324–351.

Bhatia, R.; and Davis, C. 2000. A Better Bound on the Variance. *Am. Math. Mon.*, 107(4): 353–357.

Chunaev, P. 2020. Community detection in node-attributed social networks: A survey. *Comput. Sci. Rev.*, 37: 100286.

Edwards, S. 2008. Thomas M. Cover and Joy A. Thomas, Elements of Information Theory (2nd ed.), John Wiley & Sons, Inc. (2006). *Inf. Process. Manag.*, 44(1): 400–401.

Fristedt, B. 1993. The Structure of Random Partitions of Large Integers. *Transactions of the American Mathematical Society*, 337(2): 703–735.

Gagolewski, M.; et al. 2020. Benchmark Suite for Clustering Algorithms – Version 1. https://github.com/gagolews/clustering_benchmarks_v1. Accessed: 2021-10-15.

Hoffman, M.; Steinley, D. L.; and Brusco, M. J. 2015. A

note on using the adjusted Rand index for link prediction in networks. *Soc. Networks*, 42: 72–79.

Hörmann, W. 1994. A universal generator for discrete log-concave distributions. *Computing*, 52(1): 89–96.

Hubert, L.; and Arabie, P. 1985. Comparing Partitions. *Journal of Classification*, 2(1): 193–218.

Kachitvichyanukul, V.; and Schmeiser, B. 1985. Computer Generation of Hypergeometric Random Variates. *Journal of Statistical Computation and Simulation*, 22(2): 127–145.

Karim, M. R.; Beyan, O.; Zappa, A.; Costa, I. G.; Rebholz-Schuhmann, D.; Cochez, M.; and Decker, S. 2021. Deep learning-based clustering approaches for bioinformatics. *Briefings Bioinform.*, 22(1): 393–415.

Lancichinetti, A.; and Fortunato, S. 2012. Consensus Clustering in Complex Networks. *Scientific Reports*, 2(1): 336.

Lazarenko, D.; and Bonald, T. 2021. Pairwise Adjusted Mutual Information. arXiv:2103.12641.

Leskovec, J.; and Krevl, A. 2014. SNAP Datasets: Stanford Large Network Dataset Collection. https://snap.stanford.edu/data/index.html. Accessed: 2022-03-22.

Müller, E.; Günnemann, S.; Färber, I.; and Seidl, T. 2010. Discovering Multiple Clustering Solutions: Grouping Objects in Different Views of the Data. In Webb, G. I.; Liu, B.; Zhang, C.; Gunopulos, D.; and Wu, X., eds., *ICDM 2010, The 10th IEEE International Conference on Data Mining, Sydney, Australia, 14-17 December 2010*, 1220. IEEE Computer Society.

Nguyen, X. V.; Epps, J.; and Bailey, J. 2009. Information theoretic measures for clusterings comparison: is a correction for chance necessary? In Danyluk, A. P.; Bottou, L.; and Littman, M. L., eds., *Proceedings of the 26th Annual International Conference on Machine Learning, ICML 2009, Montreal, Quebec, Canada, June 14-18, 2009*, volume 382 of *ACM International Conference Proceeding Series*, 1073–1080. ACM.

Nguyen, X. V.; Epps, J.; and Bailey, J. 2010. Information Theoretic Measures for Clusterings Comparison: Variants, Properties, Normalization and Correction for Chance. *J. Mach. Learn. Res.*, 11: 2837–2854.

Patefield, W. M. 1981. Algorithm AS 159: An Efficient Method of Generating Random R × C Tables with Given Row and Column Totals. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 30(1): 91–97.

Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; Vanderplas, J.; Passos, A.; Cournapeau, D.; Brucher, M.; Perrot, M.; and Duchesnay, E. 2011. Scikit-Learn: Machine Learning in Python. *Journal of Machine Learning Research*, 12: 2825–2830.

Romano, S.; Bailey, J.; Nguyen, X. V.; and Verspoor, K. 2014. Standardized Mutual Information for Clustering Comparisons: One Step Further in Adjustment for Chance. In *Proceedings of the 31th International Conference on Machine Learning, ICML 2014, Beijing, China, 21-26 June 2014*, volume 32 of *JMLR Workshop and Conference Proceedings*, 1143–1151.

Romano, S.; Nguyen, X. V.; Bailey, J.; and Verspoor, K. 2016. Adjusting for Chance Clustering Comparison Measures. *J. Mach. Learn. Res.*, 17: 134:1–134:32.

Shakibian, H.; and Moghadam Charkari, N. 2017. Mutual Information Model for Link Prediction in Heterogeneous Complex Networks. *Scientific Reports*, 7(1): 44981.

Stadlober, E. 1990. The ratio of uniforms approach for generating discrete random variates. *Journal of computational and applied mathematics*, 31(1): 181–189.

Staudt, C. L.; Sazonovs, A.; and Meyerhenke, H. 2016. NetworKit: A tool suite for large-scale complex network analysis. *Netw. Sci.*, 4(4): 508–530.

Tian, T.; Wan, J.; Song, Q.; and Wei, Z. 2019. Clustering single-cell RNA-seq data with a model-based deep learning approach. *Nat. Mach. Intell.*, 1(4): 191–198.

Walker, A. J. 1974. New fast method for generating discrete random numbers with arbitrary frequency distributions. *Electronics Letters*, 10(8): 127–128.

Walker, A. J. 1977. An Efficient Method for Generating Discrete Random Variables with General Distributions. *ACM Trans. Math. Softw.*, 3(3): 253–256.

Wei, S.; Han, G.; Wang, R.; Yang, Y.; Zhang, H.; and Li, S. 2021. Inductive Multi-view Multiple Clusterings. In *2021 7th International Conference on Big Data and Information Analytics (BigDIA)*, 308–315.