2023 IEEE 33rd International Workshop on Machine Learning for Signal Processing (MLSP) | 979-8-3503-2411-2/23/$31.00 ©2023 IEEE | DOI: 10.1109/MLSP55844.2023.10285970

# INDEPENDENT VECTOR ANALYSIS WITH SPARSE INVERSE COVARIANCE ESTIMATION: AN APPLICATION TO MISINFORMATION DETECTION

*Lucas P. Damasceno*[⋆] *Egzona Rexhepi*[†] *Allison Shafer*[†] *Ian Whitehouse*[†]
*Charles C. Cavalcante*[⋆] *Roberto Corizzo*[†] *Zois Boukouvalas*[†]

[⋆] Federal University of Ceará, Fortaleza - CE, 60455-760, Brazil
[†] American University, Washington, DC 20016, USA

## ABSTRACT

The analysis of multiple sets of data, such as multi-subject or multi-modal data, is crucial for many computer science and engineering problems. The primary objective in such problems is to extract relevant features for machine learning tasks, commonly classification or detection tasks. Independent vector analysis (IVA) has emerged as a powerful technique for multi-modal learning and feature extraction due to its minimal model assumptions, i.e., statistical independence and ability to effectively maximize interactions within and across datasets. Despite the appeal of minimizing model assumptions, prior knowledge, such as sparse associations of features, is usually available. Therefore, incorporating sparsity constraints into IVA can mitigate the effects of confounding features promising to significantly improve the utility of the final extracted features. This paper discusses integrating sparsity constraints into the IVA model through the inverse covariance matrix and introduces a new algorithm, IVA with sparse inverse covariance estimation (IVA-SPICE). We present the parallel implementation of IVA-SPICE and demonstrate its efficacy through simulations involving diverse parameters. Finally, we demonstrate the usefulness of IVA-SPICE in multi-modal misinformation detection, highlighting its practical capabilities.

***Index Terms***— Independent Vector Analysis, Inverse Covariance Estimation, Graphical Lasso, Multi-modal Misinformation Detection.

## 1. INTRODUCTION

The analysis of multiple sets of data is a vital process commonly encountered in many problems in computer science and engineering. In such applications, data is collected from multiple sources, including various subjects or modalities. To gain insights from these multi-modal or multi-subject datasets, it is essential to identify the relevant features that can be used for machine learning tasks such as classification or detection. Identifying relevant features is crucial because it determines the accuracy and reliability of machine learning models. Furthermore, by analyzing data from multiple sources, one can identify patterns and correlations that might be missed when analyzing single datasets. As a result, this can enhance the accuracy and reliability of models, making them suitable for various applications including image and speech recognition, natural language processing tasks, and recommendation systems.

With its well-structured formulation, independent vector analysis (IVA) provides an ideal starting point for developing effective methods for the analysis of multiple sets of data [1]. By estimating joint features, IVA possesses the capability to effectively capture distinctive characteristics present in multi-modal data. These features can then be used to improve the performance of machine learning tasks [2, 3]. By establishing a source component vector (SCV), IVA facilitates a smart integration of various datasets, enabling one to take full statistical information across multiple data sets and promoting efficient multi-modal learning. The goal in multi-modal feature extraction using IVA is to estimate joint features such that each SCV is maximally independent of all other SCVs. Although this might be a natural assumption in many problems, it might be too strong of an assumption in certain applications. Incorporating reliable and meaningful prior information about the associations of joint features can result in better estimation of each SCV, and thus, its corresponding estimated covariance matrix or precision matrix will better reveal sparse associations between extracted low dimensional features across the modalities.

In this paper, we discuss a popular technique for effectively estimating the precision matrix — inverse covariance matrix — based on a lasso penalty applied to the inverse covariance matrix [4]. We present how to effectively use it to discover sparse interactions among the features within an SCV. This work makes several contributions. First, we demonstrate how to integrate graphical lasso into the IVA formulation under the assumption that each SCV follows a multivariate Gaussian distribution. Next, we present a novel IVA algorithm, IVA with sparse inverse covariance estimation (IVA-SPICE), which can be used for joint feature extraction when sparse interactions, i.e., *conditional statistical dependencies*, among the features within each SCV exist. In this study, we introduce a parallel implementation of IVA-SPICE and showcase its efficacy through simulations involving diverse parameters. Additionally, we demonstrate the usefulness of IVA-SPICE in multi-modal misinformation detection.

## 2. METHODS

### 2.1. Independent vector analysis (IVA)

Using random vector notation, let each dataset $\mathbf{x}^{[k]}$, $k = 1, ..., K$ be a linear mixture of $N$ statistically independent sources

$$\mathbf{x}^{[k]} = \mathbf{A}^{[k]}\mathbf{s}^{[k]}, \quad k = 1, ..., K, \tag{1}$$

where $\mathbf{A}^{[k]} \in \mathbb{R}^{K\times K}$, $k = 1, ..., K$ are invertible mixing matrices and $\mathbf{s}^{[k]} = [s_1^{[k]}, ..., s_N^{[k]}]^\top$ is the vector of latent sources for the $k$th dataset. The $n$th source component vector (SCV) $\mathbf{s}_n = [s_n^{[1]}, ..., s_n^{[k]}]^\top$, can be defined by concatenating the $n$th source from each of the $K$ datasets. The goal of IVA is to estimate $K$ demixing matrices to yield source estimates $\mathbf{y}^{[k]} = \mathbf{W}^{[k]}\mathbf{x}^{[k]}$, such that each SCV is maximally independent of all other SCVs.

The IVA optimization parameter is defined as a set of demixing matrices $\mathbf{W}^{[1]}, \ldots, \mathbf{W}^{[K]}$, which can be collected into a three-dimensional array $\mathcal{W} \in \mathbb{R}^{N\times N\times K}$ and can be estimated through the

minimization of the IVA objective function given by

$$J_{IVA}(\mathcal{W}) = \sum_{n=1}^{N} H(\mathbf{y}_n) - \sum_{k=1}^{K} \log \left| \det \left( \mathbf{W}^{[k]} \right) \right| + C. \quad (2)$$

Here $H(\mathbf{y}_n)$ denotes the differential entropy of the estimated $n$th SCV that serves as the term for modeling the sparse associations of the multi-modal joint features.

Optimizing the objective function (2) over the entire range of invertible matrices presents challenges in terms of convergence, incorporation of sparsity constraints, and efficiency. To address this issue, a widely used decoupling technique [5, 6] is employed, which divides the minimization of (2) into a series of sub-problems. This technique involves minimizing the objective function separately for each row vector $\mathbf{w}_1^{[k]}, \ldots, \mathbf{w}_N^{[k]}$ of the demixing matrix. By individually optimizing the cost function with respect to each row vector, we *simplify the integration of sparsity constraints into the IVA framework* and *facilitate the implementation of parallel IVA algorithms*. This work aims to provide solutions for these particular aspects.

### 2.2. Decoupling trick, gradient, and update rule

By following the work from [5], we mathematically demonstrate the decoupling trick. To simplify the notation, we start by considering the cost function (2) without the superscript $[k]$. Let $\mathbf{W}_n = [\mathbf{w}_1, \ldots, \mathbf{w}_{n-1}, \mathbf{w}_{n+1}, \ldots, \mathbf{w}_N]^\top \in \mathbb{R}^{(N-1)\times N}$ denote the matrix that contains all rows of $\mathbf{W}$ except the $n$th one. Since the determinant of a matrix is invariant under row permutation up to a sign ambiguity, the square of the $\det(\mathbf{W})$ term in (2) is written as

$$\begin{aligned} \det(\mathbf{W})^2 &= \det(\mathbf{W}\mathbf{W}^\top) = \det\left( \begin{bmatrix} \mathbf{W}_n \\ \mathbf{w}_n^\top \end{bmatrix} \begin{bmatrix} \mathbf{W}_n \mathbf{w}_n^\top \end{bmatrix} \right) \\ &= \det\left( \begin{bmatrix} \mathbf{W}_n \mathbf{W}_n^\top & \mathbf{W}_n \mathbf{w}_n \\ \mathbf{w}_n^\top \mathbf{W}_n^\top & \mathbf{w}_n^\top \mathbf{w}_n \end{bmatrix} \right) \\ &= \det(\mathbf{W}_n \mathbf{W}_n^\top) \mathbf{w}_n^\top (\mathbf{I} - \mathbf{W}_n^\top (\mathbf{W}_n \mathbf{W}_n^\top)^{-1} \mathbf{W}_n) \mathbf{w}_n, \end{aligned}$$

where the term $\mathbf{H}_n = \mathbf{I} - \mathbf{W}_n^\top (\mathbf{W}_n \mathbf{W}_n^\top)^{-1} \mathbf{W}_n$ is the orthogonal projection onto the null space of $\mathbf{W}_n$. By definition, the matrix $\mathbf{H}_n$ is rank one, and thus, $\mathbf{H}_n = \mathbf{h}_n \mathbf{h}_n^\top$, where $\mathbf{h}_n$ is perpendicular to all row vectors of $\mathbf{W}_n$. Thus,

$$\begin{aligned} |\det(\mathbf{W})| &= \sqrt{\det(\mathbf{W}_n \mathbf{W}_n^\top)^2 \mathbf{w}_n^\top \mathbf{h}_n \mathbf{h}_n^\top \mathbf{w}_n} \\ &= |\det(\mathbf{W}_n \mathbf{W}_n^\top)| |(\mathbf{h}_n^\top \mathbf{w}_n)|. \end{aligned} \quad (3)$$

Therefore, by reintroducing the superscript $[k]$, the cost function (2) can be written as

$$\begin{aligned} J_{IVA}(\mathbf{w}_n^{[k]}) = \sum_{n=1}^{N} H(\mathbf{y}_n) - \log |(\mathbf{h}_n^\top \mathbf{w}_n^{[k]})| \\ - \log |\det((\mathbf{W}_n^{[k]})(\mathbf{W}_n^{[k]})^\top)| - H(\mathbf{x}^{[k]}), \end{aligned} \quad (4)$$

where the terms $H(\mathbf{x}^{[k]})$ and $\log |\det((\mathbf{W}_n^{[k]})(\mathbf{W}_n^{[k]})^\top)|$ are independent of $\mathbf{w}_n^{[k]}$. The gradient of (4) w.r.t. $\mathbf{w}_n^{[k]}$ is given by

$$\frac{\partial J_{\text{IVA}}}{\partial \mathbf{w}_n^{[k]}} = E\left\{ \phi_n^{[k]}(\mathbf{y}_n)\, \mathbf{x}^{[k]} \right\} - \frac{\mathbf{h}_n^{[k]}}{\left(\mathbf{h}_n^{[k]}\right)^\top \mathbf{w}_n^{[k]}}. \quad (5)$$

Thus, the estimate of each $\mathbf{W}^{[k]}$ can be determined w.r.t. each row vector $\mathbf{w}_n^{[k]}$, $n = 1, \ldots, N$ independently by using the gradient update rule

$$(\mathbf{w}_n^{[k]})^{\text{new}} \leftarrow (\mathbf{w}_n^{[k]})^{\text{old}} - \gamma \frac{\partial J_{IVA}(\mathbf{w}_n^{[k]})}{\partial \mathbf{w}_n^{[k]}}. \quad (6)$$

As can be seen in (6), each gradient direction depends on the corresponding estimated probability density of each SCV, and if sparse associations within an SCV exist, the IVA algorithm may experience sub-optimal convergence or even divergence due to inaccurate estimation of the probability density for each SCV.

### 2.3. IVA-SPICE

Incorporating prior knowledge about the sparse associations of the underlying features within an SCV can result in a better model match. This will result in a better estimation of the SCVs, and thus, its corresponding estimated precision matrices will better reveal sparse associations between extracted low dimensional features across the multiple sets of data. To identify the sparse associations among the underlying joint features, a logical approach is to enforce a multivariate Gaussian model on each sample covariance matrix (SCV) by utilizing a sparse inverse covariance matrix as a parameterization. [4].

Therefore, under the assumption that each SCV follows a multivariate Gaussian distribution we have that

$$\phi_n^{[k]}(\mathbf{y}_n) = \mathbf{y}_n^\top \boldsymbol{\Sigma}_n^{-1} \mathbf{e}_k, \quad (7)$$

where $\mathbf{e}_k$ is the unit vector. Therefore, the gradient of (4) with respect to each row vector $\mathbf{w}_n^{[k]}$ of $\mathbf{W}^{[k]}$ is given by

$$\frac{\partial J_{\text{IVA}}}{\partial \mathbf{w}_n^{[k]}} = E\left\{ \mathbf{x}^{[k]} \mathbf{y}_n^\top \right\} \boldsymbol{\Sigma}_n^{-1} \mathbf{e}_k - \frac{\mathbf{h}_n^{[k]}}{\left(\mathbf{h}_n^{[k]}\right)^\top \mathbf{w}_n^{[k]}}. \quad (8)$$

Utilizing the inherent properties of the precision matrix presents valuable opportunities for the application of IVA. The sparsity of each precision matrix $\boldsymbol{\Sigma}_n^{-1}$ becomes an informative characteristic, as it captures the conditional dependencies among the underlying features within an SCV by representing them as zero values in the inverse covariance [7, 8]. Leveraging this sparsity reduces the impact of confounding joint features and facilitates the identification of essential relationships. Nevertheless, the precise estimation of each $\boldsymbol{\Sigma}_n^{-1}$ associated with the SCVs is of utmost importance. Accurate estimation is crucial not only for capturing sparse associations among features but also for ensuring optimal performance in IVA as a whole.

**Input:** $\mathbf{X} \in \mathbb{R}^{N \times V \times K}$
For each $k = 1, .., K$, initialize $\mathbf{W} \in \mathbb{R}^{N \times N}$
**for** $n$ = *1:N* **do**
  Estimate $\boldsymbol{\Sigma}_n^{-1}$ using graphical lasso to fully characterize $\phi_n^{[k]}(\mathbf{y}_n)$ in (7)
  Compute $\mathbf{h}_n^{[k]}$ for $k = 1, ..., K$
  **for** $k$ = *1:K* **do**
    Calculate the derivative $\frac{\partial J_{\text{IVA}}}{\partial \mathbf{w}_n^{[k]}}$ (8)
    $(\mathbf{w}_n^{[k]})^{\text{new}} \leftarrow (\mathbf{w}_n^{[k]})^{\text{old}} - \gamma \frac{\partial J_{\text{IVA}}}{\partial \mathbf{w}_n^{[k]}}$
  **end**
**end**
$J_{\text{IVA}} = \sum_{n=1}^{N} H(\mathbf{y}_n) - \sum_{k=1}^{K} \log \left| \det \left( \mathbf{W}^{[k]} \right) \right|$
Repeat steps 3 to 12 until convergence in $\mathcal{W}$
**return** $\mathcal{W}$

**Algorithm 1:** Pseudo-code of the IVA-SPICE algorithm

Graphical lasso is a regularized maximum likelihood estimation approach that uses $\ell_1$ penalty to encourage sparsity in the inverse covariance matrix [4]. It solves the optimization problem using convex optimization techniques, such as coordinate descent or proximal gradient descent, and has good optimality conditions. Given that graphical lasso estimates $\boldsymbol{\Sigma}_n^{-1}$ effectively and due to its efficiency compared to other approaches, we integrate this estimation technique into the IVA-Gaussian (IVA-G) [5] algorithm and develop a new *IVA sparse-based algorithm: IVA with sparse inverse covariance estimation, or IVA-SPICE.*

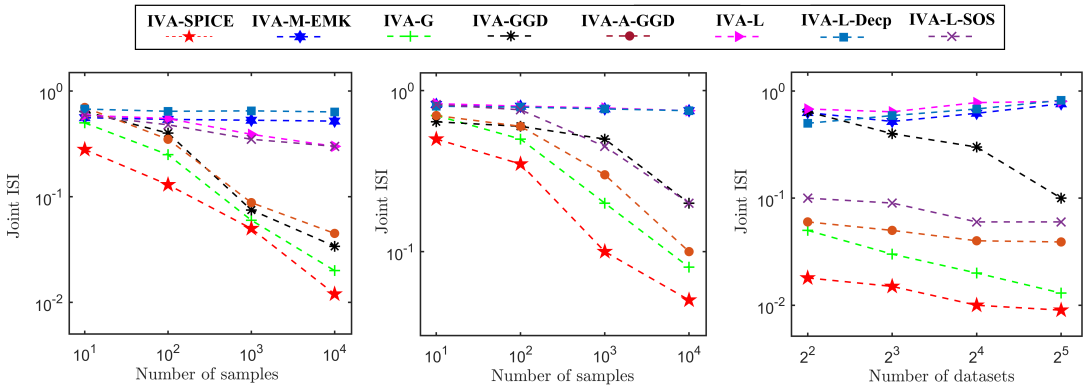


**Fig. 1**. The performance is compared in various scenarios based on the average Joint ISI. (Left) Joint ISI as a function of number of samples for $N = 10, K = 4$ and $d = 1$. (Center) Joint ISI as a function of number of samples for $N = 10, K = 32$ and $d = 1$. (Right) Joint ISI as a function of number of datasets for $N = 10, d = 1$ and $V = 10,000$.

The pseudo-code for the IVA-SPICE algorithm is provided in Algorithm 1. The core component of this algorithm is the loop outlined in lines 2-9, emphasizing the importance of accurately estimating each $\mathbf{\Sigma}_n^{-1}$ for capturing sparse associations among the underlying joint features. Moreover, it is notable that the computational complexity of IVA-SPICE primarily resides within lines 2-9 of Algorithm 1 and as we see in later section, distributing individual iterations of the main loop among separate computational resources is highly desirable to minimize the overall execution time.

## 3. NUMERICAL EXPERIMENTS

### 3.1. Results based on simulated data

In the initial set of our experiments, we assess the effectiveness of IVA-SPICE using simulated input data that is sparse. In all experiments, the $n$th sample covariance matrix is a $K$-dimensional, zero-mean multivariate Gaussian random vector that is characterized by a sparse inverse covariance matrix $\mathbf{\Sigma}_n^{-1}$. The sparsity of the matrix is determined by the density of non-zero elements. In our experiments, we investigate the impact of varying the density $d$ on the covariance matrix sparsity of the generated data. The density ranges from 0 to 1, where 0 denotes non-sparsity and 1 represents the highest level of sparsity. We also consider the number of components $N$, datasets $K$, and samples $V$ as variables. To evaluate the performance, we employ Joint ISI, a global metric that measures separation quality when the ground truth is available. A Joint ISI value of zero indicates perfect separation. We conduct ten trials and report the averaged results. The results are averaged over ten trials. For all of our experiments, we compare IVA-SPICE with several IVA algorithms. Different IVA algorithms *explicitly* model the underlying sources by assuming different probability densities for the underlying SCVs. Specifically, the IVA-L algorithm [9] incorporates higher order statistics (HOS) and assumes a Laplacian distribution for the underlying SCVs. On the other hand, IVA-G [5] leverages linear dependencies but does not consider HOS. Meanwhile, IVA-A-GGD [10] is a more comprehensive IVA implementation that takes both second and higher order statistics into account. This algorithm assumes a multivariate generalized Gaussian distribution (MGGD) for the underlying SCVs, where multivariate Gaussian and Laplacian distributions are considered as special cases by estimating the distribution's parameters. Finally, the most flexible IVA algorithm, IVA-M-EMK [11], is based on the multivariate maximum entropy principle, allowing it to model a wide range of PDFs.

Figure 1 presents the source separation performance varying the

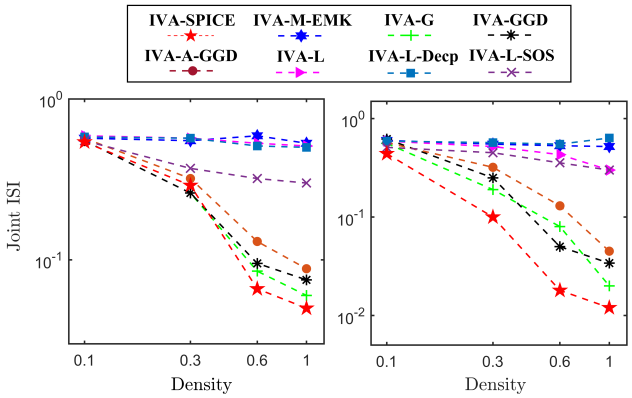


**Fig. 2**. The performance comparison is based on the average Joint ISI, varying with the density of non-zero values on the true precision matrices. (Left) $N = 10, K = 4, V = 1,000$. (Right) $N = 10, K = 4, V = 10,000$.

number of samples from 100 to 10,000 with $N = 10$, $K = 4$, $d = 1$ and $N = 10$, $K = 32$, $d = 1$, respectively. Moreover, in the right plot in Figure 1 with $N = 10$, $d = 1$, and $V = 10,000$, we can see that increasing the number of datasets is negligible. For this scenario, where the density of non-zero values is held at 1, we can see that IVA-SPICE attains superior separation results in the sparser setting. On the other hand, in Figure 2, with $K$ fixed at 4, we maintain $N = 10$ for the first column while using $V = 1,000$ and $V = 10,000$ for the second. From the results, we can observe that IVA-SPICE outperforms all other IVA algorithms in scenarios where there is inherent sparsity in the sample covariance matrices. In addition, an essential feature of IVA-SPICE is that it demonstrates a considerable improvement in its source separation performance with a notable decrease in error rates as more samples become available. In our experiments, enlarging the sample size is the optimal solution as IVA operates within a maximum likelihood framework [11].

### 3.2. Parallel implementation of IVA-SPICE

Machine learning and signal processing tools are expected to possess the capability to handle large-scale data commonly encountered in real-world applications. As previously discussed, the computational complexity of IVA-SPICE primarily lies within the lines 2-9 of Algorithm 1. To decrease the overall execution time, it is de-

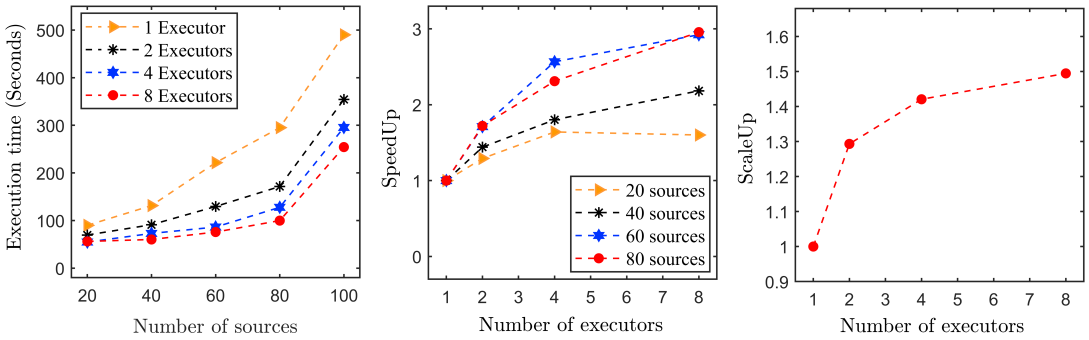


**Fig. 3**. Scalability experiments with distributed IVA-SPICE in terms of execution time (left), SpeedUp (center), and ScaleUp (right).

sirable to distribute separate iterations of the main loop to different computational resources.

Our optimized IVA implementation leverages Pandas User-Defined Function (Pandas UDFs) and the Apache Spark framework to enable distributed computation [12].Pandas UDFs, alternatively referred to as vectorized UDFs, are custom functions that leverage Apache Arrow for data transfer and enable vectorized operations. These UDFs have the potential to enhance performance by up to $100x$ when compared to Python UDFs that operate on a row-by-row basis. Specifically, in our work, the graphical lasso of the IVA loop is implemented as a Pandas UDF adopting iterators on Pandas series. Algorithmically, processing steps prior to the graphical lasso loop are executed in a sequential manner. After then, our strategy loads the entire dataset as a distributed Spark DataFrame via PySpark, i.e. the Python Apache Spark API. Embedding the graphical lasso loop in a Pandas API allows the driver node, which is in charge of orchestrating the computational workload, to divide the DataFrame into partitions and assign them to different executors in the computational cluster. A Python interpreter runs on each executor and trains the graphical lasso model on its partition, using all available local cores to run vectorized operations on the Pandas series in a parallel fashion. Once a partition has been processed by an executor, it is returned as a result to the driver, which collects all partial results and combines them to obtain the final result. Pseudo-code of our training strategy is described in Algorithm 2. Post-processing steps following the graphical lasso loop are executed in a sequential manner.

The driver initializes the number of partitions based on the cluster configuration $C$, which defines the number of executors, the number of cores for each executor, and the amount of RAM memory for each executor. To evaluate the scalability of our solution, the experiments feature a cluster configuration involving up to 8 executors nodes, each equipped with 8 cores and 16GB of RAM memory. Experiments in Figure 4 show the execution time (left), the SpeedUp (center), and the ScaleUp (right).

SpeedUp is computed as the ratio between the time execution with the single-executor implementation, and the corresponding execution time with multiple executors, with an increasing number of sources ($N$). ScaleUp is obtained simultaneously increasing the problem size ($N$: $10, 20, 30, 40$) and the number of executors: $(1, 2, 4, 8)$, and computing ratios between execution times with any configuration $vs.$ the initial one with $N = 10$ and 1 executor. For instance, the ScaleUp for $x = 2$ is computed as the ratio between the execution time with $N = 20$ – 2 executors, and the time with $N = 10$ – 1 executor. While the ideal curve for SpeedUp is a 45-degree line, the ideal curve for ScaleUp is a flat line ($y = 1$), which is rarely obtained in practice. In the results, we observe that the SpeedUp increases as the number of sources $N$ and the number of executors increases. The maximum SpeedUp achieved is $2.96x$, with 8 executors and $N = 80$.

**Input:** $\mathbf{D}$ – entire dataset, C – cluster configuration
$\mathbf{P}$ = Driver initializes partitions for $\mathbf{D}$ based on C
**for** $P_i \in \mathbf{P}$ **do**
  `// Executor computation (parallel):`
  $GL_i$ = Train graphical lasso model on $P_i$
  $R_i = [GL_i.cov,\ GL_i.inv]$
  **return** $R_i$
**end**
`// Results aggregation on driver:`
$R = R_1 \cup R_2 \cup \cdots \cup R_k$
**return** $R$

**Algorithm 2:** Distributed pseudo-code (graphical lasso loop)

Overall, the results highlight that a significant reduction of the execution time can be obtained with the distributed implementation of IVA-SPICE as more sources are added to the data and more executors are available. It is worth noting that, in our experiments, no differences in the ISI results were observed for the single-executor vs. multi-executor implementation. As a result, the reduction in the execution time achieved with the distributed implementation does not come at the cost of reduced accuracy for the IVA-SPICE converging solution.

## 4. APPLICATION TO MISINFORMATION DETECTION

We demonstrate the usefulness of IVA-SPICE in multi-modal misinformation detection. We formulate the problem of joint feature generation for multi-modal misinformation detection as an IVA problem in the following way. Using matrix notation, let $\mathbf{X}^{[k]} \in \mathbb{R}^{d \times V}$ is the $k^{\text{th}}$ observation matrix from $k$th modality, where $d$ denotes the number of initial high-level feature vectors in the $k^{\text{th}}$ modality and $V$ denotes the total number of samples. As we see in the next section, examples of high-level features include pre-trained word embeddings, pre-trained image embeddings, topics, and captions. The model is given by $\mathbf{X}^{[k]} = \mathbf{A}^{[k]}\mathbf{S}^{[k]}, \quad k = 1, ..., K$, where $\mathbf{A}^{[k]} \in \mathbb{R}^{d \times N}$ is the $k^{\text{th}}$ mixing matrix and $\mathbf{S}^{[k]} \in \mathbb{R}^{N \times V}$ are latent variable estimates which in our setting, correspond to the $k^{\text{th}}$ set of feature estimates and are estimated using IVA-SPICE. The sparsity of each $\mathbf{\Sigma}_n^{-1}$ becomes an informative characteristic, as it may capture the sparse associations among the underlying joint features. The machine learning algorithm used for detecting misinformation will be trained using the estimated joint features.

### 4.1. Dataset and high-level feature extraction

For this work, we utilize the final processed training and test datasets from our previous study [3], which were created using the MediaE-

val2016 Image Verification Corpus [13]. The datasets used in this study comprise distinct sets of labeled training and test tweet data, specifically from the 2016 vintage. The training dataset consists of tweets related to a specific set of events that are entirely different from those discussed in the tweets found in the test dataset. Each tweet record is assigned a label indicating whether it is categorized as "fake" or "real", with "fake" tweets including multimedia content that misrepresents the referred event [13].

The working datasets include records with a single corresponding image. The text data consists of pre-processed text without emoji characters, stop words, URLs, Twitter handles, time stamps, select punctuation, and words less than two characters. The text data was normalized by lowercasing each word, reducing multi-spaces to one space, and lemmatizing the words. The dataset includes only tweets identified as using English or a similar language using the Langid Python package and the International Organization for Standardization (ISO) code for languages. The text data does not include tweets that were denoted as being retweeted. Records that resulted in null values during prior feature extraction are not present. The image data was pre-processed by resizing the images to 224x224 pixels and normalizing them.

The training dataset comprises 9,140 tweet records, involving 352 distinct images, and representing 15 different events. Out of these training tweets, 5,127 are classified as fake, while 4,013 are classified as real. Among the 15 events in the training dataset, five events have a mix of both real and fake tweets, while the remaining ten events exclusively consist of fake tweets. On the other hand, the test dataset consists of 796 tweet records, associated with 92 unique images, and representing 23 distinct events. It's important to note that the events included in the training data and testing data do not overlap. In the test dataset, 467 tweets are categorized as fake, while 329 tweets are classified as real. Out of the 23 events, seven events have a combination of real and fake tweets, one event has only real tweets, and 15 events consist solely of fake tweets.

To represent the text in each tweet record, we utilize a 300-dimensional Word2Vec embedding vector [14] and extract features from the datasets. By averaging the word embedding vectors of the tweet's words, we effectively capture the essence of the tweet. This feature was created using a Word2Vec model trained on the Google News corpus. We include a 4,096-dimensional fully connected layer from a pre-trained VGG-16 model for each image associated with an individual tweet record. We also create a feature we call "Image2text" using an Image Captioning PyTorch model [15] pre-trained with ResNet101 features to generate captions for each image in our datasets. Then, we use the same Word2Vec model and averaging method previously used to create a 300-dimensional Word2Vec embedding for each generated caption. The final feature we create and utilize is a 200-dimension vector representing the top 200 topics assigned to a tweet using topic modeling. To conduct the topic modeling, we utilize the term frequency-inverse document frequency (TF-IDF) vectorizer to generate a TF-IDF matrix for the text from the tweets, using a vocabulary that consists of words that are in less than 95 percent and in more than one percent of the tweets in the datasets. We then apply non-negative matrix factorization (NMF) to assign 200 topics to each tweet. We use 200 topics to ensure that the feature's dimensions are compatible with our other matrices to conduct our analysis.

### 4.2. Classification procedure

The classification process comprises four stages. In the first stage, we construct our set of tweets as follows. We represent the training observation matrices for each modality with $\mathbf{X}_{\text{train}}^{[k]} \in \mathbb{R}^{d_k \times V_{\text{train}}}$ where $d_k$ denotes the number of initial high-level feature vectors in each modality and $V_{\text{train}}$ denotes the number of training tweets. Similarly, $\mathbf{X}_{\text{test}}^{[k]} \in \mathbb{R}^{d_k \times V_{\text{test}}}$ denotes the corresponding testing observation matrices. In the second stage, the mean from each dataset is removed so they are centered, and PCA is applied to each $\mathbf{X}_{\text{train}}^{[k]}$, for $k = 1, 2, \ldots, K$. For the PCA step, we use an order $N$, which, in our setting, denotes the number of features from each modality. Then, for each $k = 1, 2, \ldots, K$, we obtain $\hat{\mathbf{X}}_{\text{train}}^{[k]} \in \mathbb{R}^{N \times V_{\text{test}}}$ and vertically concatenate each $\hat{\mathbf{X}}_{\text{train}}^{[k]}$ to form a three dimensional array $\hat{\mathbf{X}}_{\text{train}} \in \mathbb{R}^{N \times V_{\text{train}} \times k}$. In the third stage, we perform IVA on $\hat{\mathbf{X}}_{\text{train}}$, and since we have $K$ modalities, IVA provides $K$ demixing matrices $\mathbf{W}^{[k]} \in \mathbb{R}^{N \times N}$, for $k = 1, 2, \ldots, K$. Then, using the estimated demixing matrices we generate $\mathbf{Y}_{\text{train}}^{[k]} = \mathbf{W}^{[k]} \left(\hat{\mathbf{X}}_{\text{train}}^{[k]}\right)^{\top}$, for $k = 1, 2, \ldots, K$. The training dataset $\mathbf{Y}_{\text{train}}$ is formed by averaging the estimated SCVs, which can be obtained by concatenating the estimated sources from $\mathbf{Y}_{\text{train}}^{[k]}$, for $k = 1, 2, \ldots, K$. It is important to note that $\mathbf{Y}_{\text{train}}$ includes all the extracted features from the multi-modal data, which is used to train the classification model. Conversely, the testing dataset is created by subtracting the training mean from each multi-modal testing dataset and applying the PCA transformations derived from the training phase. Additionally, the demixing matrices from the training phase are used to transform the testing datasets in the following manner, $\mathbf{Y}_{\text{test}}^{[k]} = \mathbf{W}^{[k]} \left(\hat{\mathbf{X}}_{\text{test}}^{[k]}\right)^{\top}$, for $k = 1, 2, \ldots, K$, where $\mathbf{Y}_{\text{test}}^{[k]} \in \mathbb{R}^{N \times V_{\text{test}}}$. Finally, the testing dataset $\mathbf{Y}_{\text{test}}$ is formed by averaging the estimated SCVs, which can be obtained by concatenating the estimated sources from $\mathbf{Y}_{\text{test}}^{[k]}$, for $k = 1, 2, \ldots, K$. In the fourth stage, we train the classification model using $(\mathbf{Y}_{\text{train}})^{\top}$. The specific form of the classification model is not critical as our IVA-based fusion approach is independent of the classification procedure. Nonetheless, we provide an illustrative example using support vector machines (SVMs), which have exhibited dependable results in various applications, particularly when dealing with smaller datasets [16]. Following the training of the classification model, we assess its performance utilizing an unseen dataset, $(\mathbf{Y}_{\text{test}})^{\top}$. In order to maintain consistency, a grid search cross-validation with a five-fold scheme is utilized for hyper-parameter optimization, model training, and testing in all experiments. To generate well-converged statistics, the entire process is repeated five times, with shuffling occurring prior to each iteration.

### 4.3. Classification performance

In the initial set of experiments, we evaluate IVA-SPICE against several other IVA algorithms, measuring the F1-score as the number of training samples increases. Specifically, we consider the scenario where $K = 4$ and $N = 100$. In addition, to generate the final joint features, we average all estimated SCVs to prevent possible overfitting. Based on the information depicted in the left plot of Figure 4, it can be deduced that IVA-SPICE surpasses all other IVA algorithms as the number of training samples rises. It is worth noting, that in the poor sample case, IVA-SPICE has the worst performance, and this is due to the sub-optimal estimation of the sparse inverse covariance matrix. IVA-M-EMK is invariant to the increase in the training samples. Both IVA-G and IVA-L do not yield good performances as they are not suitable to model the sparse underlying associations within each SCV. IVA-SPICE is an excellent fusion approach for misinformation detection, as demonstrated by its high F1-score with respect to the number of training tweets. Our approach's classification results are on par with those obtained in related studies, including [17].

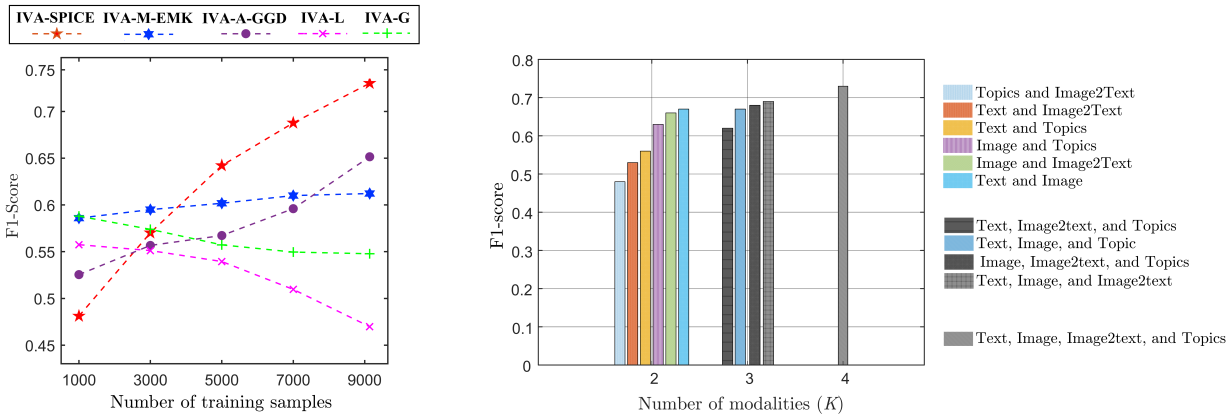


**Fig. 4**. (Left) F1-score is employed as a metric for performance comparison in both the IVA-based classification model, considering varying numbers of training samples. (Right) The performance is compared in terms of the F1-score, considering different numbers of modalities fused to train the classification model. When $K = 2$, we combine $\frac{4!}{2!2!}$ modalities, similar for $K = 3$.

Our misinformation data fusion approach enables the evaluation of each modality's contribution and the identification of the optimal combination of modalities for achieving the highest prediction score. To demonstrate this, we evaluate the performance of IVA-SPICE based on the number of fused modalities. The right plot in Figure (4) illustrates the F1-scores as a function of the number of multi-modal data utilized to train the IVA transformations and classification model. In this experiment, we maintain $N = 100$, and average all estimated SCVs. As $K$ increases, IVA-SPICE effectively employs sparsity through the inverse covariance matrix (precision matrix), reducing the impact of confounding joint features and providing superior classification performance in all fusion scenarios.

## 5. CONCLUSION

This study brings attention to various intriguing avenues for future exploration. As evidenced in our findings, the utilization of the IVA-SPICE multivariate data fusion model yields improved detection performance with an increasing number of modalities. Consequently, it would be beneficial to incorporate supplementary modalities like video-based or metadata-based modalities in our investigation. Finally, we suggest determining the optimal conditions of IVA-SPICE on the space of probability density functions.

## 6. REFERENCES


[1] T. Adalı, M. Anderson, and G.-S. Fu, "Diversity in independent component and vector analyses: Identifiability, algorithms, and applications in medical imaging," *IEEE Signal Processing Magazine*, vol. 31, no. 3, pp. 18–33, 2014.

[2] Z. Boukouvalas, M. Puerto, D. C. Elton, P. W. Chung, and M. D. Fuge, "Independent vector analysis for molecular data fusion: Application to property prediction and knowledge discovery of energetic materials," in *2020 28th European Signal Processing Conference (EUSIPCO)*. IEEE, 2021, pp. 1030–1034.

[3] L. P. Damasceno, A. Shafer, N. Japkowicz, C. C. Cavalcante, and Z. Boukouvalas, "Efficient multivariate data fusion for misinformation detection during high impact events," in *Discovery Science: 25th International Conference, DS 2022, Montpellier, France, October 10–12, 2022, Proceedings*. Springer, 2022, pp. 253–268.

[4] J. Friedman, T. Hastie, and R. Tibshirani, "Sparse inverse covariance estimation with the graphical LASSO," *Biostatistics*, vol. 9, no. 3, pp. 432–441, 2008.

[5] M. Anderson, T. Adalı, and X.-L. Li, "Joint blind source separation with multivariate gaussian model: Algorithms and performance analysis," *Signal Processing, IEEE Transactions on*, vol. 60, no. 4, pp. 1672–1683, April 2012.

[6] X.-L. Li and X.-D. Zhang, "Nonorthogonal joint diagonalization free of degenerate solution," *IEEE Transactions on Signal Processing*, vol. 55, no. 5, pp. 1803–1814, 2007.

[7] A. Dempster, "Covariance selection," *Biometrics*, vol. 28, no. 1, pp. 157–175, March 1972.

[8] J. Whittaker, *Graphical Models in Applied Multivariate Statistics*. Wiley Series in Probability and Mathematical Statistics, 1990.

[9] T. Kim, T. Eltoft, and T.-W. Lee, "Independent vector analysis: An extension of ica to multivariate components," in *Independent Component Analysis and Blind Signal Separation*. Springer, 2006, pp. 165–172.

[10] Z. Boukouvalas, G.-S. Fu, and T. Adalı, "An efficient multivariate generalized gaussian distribution estimator: Application to iva," in *2015 49th Annual Conference on Information Sciences and Systems (CISS)*. IEEE, 2015, pp. 1–4.

[11] L. P. Damasceno, C. C. Cavalcante, T. Adalı, and Z. Boukouvalas, "Independent vector analysis using semi-parametric density estimation via multivariate entropy maximization," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 3715–3719.

[12] R. Corizzo and T. Slenn, "Distributed node classification with graph attention networks," in *2022 IEEE International Conference on Big Data (Big Data)*. IEEE, 2022, pp. 3720–3725.

[13] C. Boididou, S. Papadopoulos, M. Zampoglou, L. Apostolidis, O. Papadopoulou, and Y. Kompatsiaris, "Detection and visualization of misleading content on twitter," *International Journal of Multimedia Information Retrieval*, vol. 7, no. 1, pp. 71–86, 2018.

[14] T. Mikolov, K. Chen, G. S. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," *CoRR*, vol. abs/1301.3781.

[15] R. Luo, B. Price, S. Cohen, and G. Shakhnarovich, "Discriminability objective for training descriptive captions," *arXiv preprint arXiv:1803.04376*, 2018.

[16] C. Moroney, E. Crothers, S. Mittal, A. Joshi, T. Adalı, C. Mallinson, N. Japkowicz, and Z. Boukouvalas, "The case for latent variable vs deep learning methods in misinformation detection: An application to covid-19," in *International Conference on Discovery Science*. Springer, 2021, pp. 422–432.

[17] C. Boididou, S. Papadopoulos, M. Zampoglou, L. Apostolidis, O. Papadopoulou, and I. Kompatsiaris, "Detection and visualization of misleading content on twitter," *International Journal of Multimedia Information Retrieval*, vol. 7, 03 2018.