

Um estudo comparativo de algoritmos de ICA em alfabetos finitos sob modelos lineares e não-lineares

Mateus Marcuzzo da Rosa e Daniel G. Silva

Resumo—Analisamos algoritmos de análise de componentes independentes em sinais de alfabeto finito recentemente propostos, seja no paradigma linear i.e. corpo de Galois, seja no paradigma não-linear, em que se admite mapeamentos arbitrários. Experimentos são conduzidos na aplicação de separação cega de fontes, mas também em um cenário onde os dados são gerados sem o tradicional modelo de mistura linear. Embora o método não-linear seja potencialmente mais robusto que técnicas lineares, os experimentos conduzidos indicaram um desempenho superior somente em alguns cenários, além de apresentar o maior tempo de execução médio. Na separação cega de fontes o algoritmo AMERICA confirmou sua eficácia, tanto quanto à taxa de separação, quanto em tempo de execução.

Palavras-Chave—Separação cega de fontes, Análise de Componentes Independentes, Informação Mútua, Corpos finitos.

Abstract—Recently proposed Independent Component Analysis over finite alphabets algorithms are analysed in this work, whether they are under the linear paradigm i.e. Galois fields, whether they comprise a permutation based model. Numerical experiments are performed for the classical application of Blind Source Separation, and for a non-linearly-generated dataset. Despite the non-linear method can potentially provide more robust solutions, the results indicated a superior performance on few scenarios, with a higher average execution time. The blind source separation experiment confirmed AMERICA algorithm as the most efficient in terms of separation rate, with the lowest absolute values of execution time.

Keywords—Blind source separation, Independent Component Analysis, Mutual information, Finite fields.

I. INTRODUÇÃO

A análise de componentes independentes (ICA, do inglês *independent component analysis*) é a tarefa que consiste em, dado um vetor aleatório, encontrar um mapeamento que produza um novo vetor cujas variáveis sejam estatisticamente independentes entre si (ou o mais próximo disso) [1]. Como tarefa de processamento de sinais, ICA encontra como mais relevante aplicação, por vezes confundida com a própria técnica em si, o problema de separação cega de fontes (BSS, do inglês *blind source separation*) [2].

Há uma vasta literatura sobre os aspectos teóricos e sobre algoritmos de ICA [3]. Contudo, isto vale na maioria dos casos para o contexto de valores reais ou complexos. Recentemente, surgiram técnicas que consideram o domínio de sinais pertencentes a um corpo finito, mais especificamente um corpo de Galois em $\mathbb{GF}(P)$, P primo ou potência de um primo.

Mateus Marcuzzo da Rosa, Departamento de Engenharia Elétrica - Faculdade de Tecnologia - Universidade de Brasília Brasília, DF, 70910-900, e-mail: mateus.marcuzzo@aluno.unb.br; Daniel G. Silva, Departamento de Engenharia Elétrica - Faculdade de Tecnologia - Universidade de Brasília Brasília, DF, 70910-900, e-mail: danielgs@ene.unb.br.

Apesar do pouco tempo, já há vários algoritmos de ICA em $\mathbb{GF}(P)$ bem estabelecidos, os quais endereçam o problema por meio da busca, primordialmente, de um mapeamento *linear* do vetor de observações, e assim se estabeleceram como ferramenta direta na solução do problema de BSS em corpos finitos [4]. Neste sentido, trabalhos anteriores estabelecem como métrica de qualidade dos algoritmos a capacidade de extração das fontes originais do problema de BSS, sem dar devida ênfase à capacidade de maximização da independência i.e. a avaliação da realização da tarefa de ICA *per se*. Ademais, duas propostas de 2018, os algoritmos SA4ICA [5] e GLICA [6], passaram a agregar novas abordagens para o problema, mas ainda não foram comparadas com a técnica reconhecidamente mais robusta, o algoritmo AMERICA [4].

Em paralelo, Painsky *et al.* [7] estendem o conceito de ICA do domínio de corpos finitos para sinais discretos em alfabetos finitos, de uma cardinalidade Q arbitrária. Procura-se, nesta nova formulação, uma *permutação* do vetor de observações, o que se pode compreender como um mapeamento desta *vez não-linear* dos dados originais, que busca, assim como o cenário linear, a maximização da independência estatística entre as componentes resultantes. É importante ressaltar daí que, em teoria, a solução de ICA em $\mathbb{GF}(P)$ torna-se um caso particular desta formulação generalizada.

Este trabalho se propõe, dentro deste panorama atual, realizar um inédito estudo comparativo entre os algoritmos recentemente apresentados não somente sob a perspectiva da capacidade de separação na aplicação de BSS, mas também da capacidade de maximização da independência, agregando aí também a comparação com o algoritmo de ICA não-linear QICA [7]. O propósito final é verificar se, ainda que empregando um modelo de mapeamento restrito como o linear, os algoritmos são capazes de promover projeções tão independentes quanto o algoritmo que, em tese, pode encontrar transformações que levariam a soluções mais robustas.

O resto do trabalho adota a seguinte organização: na Seção II apresentamos o problema de ICA e a sua aplicação em BSS, na Seção III discutimos brevemente os algoritmos estudados neste artigo, na Seção IV discutimos os experimentos de BSS e ICA bem como seus resultados, e por fim terminamos o trabalho na Seção V com a conclusão e sugestões para trabalhos futuros.

II. ICA E SUA APLICAÇÃO EM BSS

Considere uma matriz \mathbf{X} de dimensões $K \times T$, que contém T amostras de um vetor aleatório K -dimensional \mathbf{x} , cujas componentes são variáveis aleatórias discretas e de cardinalidade Q . A tarefa de ICA consiste de encontrar um

mapeamento invertível $\mathcal{F}(\cdot)$

$$\mathbf{Y} = \mathcal{F}(\mathbf{X}) \quad (1)$$

tal que as amostras \mathbf{y} em \mathbf{Y} tenham suas componentes estatisticamente independentes entre si, ou o mais próximo disto.

Se $\mathcal{F}(\cdot)$ é uma transformação linear, estamos portanto definindo o problema no domínio de sinais em um corpo finito de ordem $Q = P$, onde P é um primo ou potência de um primo. Neste caso, o modelo de projeção é representado por uma matriz \mathbf{W} invertível

$$\mathbf{Y} = \mathbf{W}\mathbf{X}. \quad (2)$$

Se $\mathcal{F}(\cdot)$ não cumpre as premissas previamente citadas, temos o caso mais geral de ICA não-linear. Neste cenário, o mapeamento compreende de uma função de permutação das Q^K possíveis observações do vetor \mathbf{x} .

Ainda que de maneira indireta, o critério estabelecido entre as diferentes técnicas de ICA para se determinar a independência das componentes é a minimização da informação mútua ou *total correlation* [8]

$$\mathcal{F}_*(\cdot) = \arg \min_{\mathcal{F}} C(\mathbf{y}) \quad (3)$$

onde $\mathbf{y} = [y_1, \dots, y_K]^T$, $C(\mathbf{y}) = \sum_{i=1}^K H(y_i) - H(\mathbf{y})$ e $H(\cdot)$ é a entropia de Shannon. Se a informação mútua entre um conjunto de variáveis aleatórias é zero, estas são independentes entre si [9].

Observe que até então não há nenhuma hipótese se \mathbf{X} é expresso através de algum modelo generativo. Caso se assuma que a matriz de observações \mathbf{X} é oriunda de um processo de mistura linear de outro conjunto de sinais \mathbf{S} , denominados fontes

$$\mathbf{X} = \mathbf{A}\mathbf{S}, \quad (4)$$

pode-se daí formular o problema de BSS em corpos finitos, em que se procura a matriz \mathbf{W} de separação tal que

$$\mathbf{Y} = \mathbf{W}(\mathbf{A}\mathbf{S}) = (\mathbf{P}\mathbf{D})\mathbf{S} \quad (5)$$

onde \mathbf{P} e \mathbf{D} são matrizes que representam ambiguidades de permutação e escala, respectivamente. Se as fontes forem independentes entre si, não degeneradas e não possuírem distribuição uniforme, pode-se encontrar \mathbf{W} por meio da aplicação de ICA [10].

III. ALGORITMOS

São quatro os algoritmos de ICA em alfabetos finitos sob análise neste trabalho: AMERICA, SA4ICA e GLICA, que empregam o modelo de projeção linear, e por fim o algoritmo QICA, que adota a abordagem não-linear.

O algoritmo AMERICA [4], [10] baseia-se no conceito de extração sequencial de componentes pelo critério de mínima entropia. A técnica recebe como entrada um tensor que descreve a probabilidade conjunta (estimada) de cada realização \mathbf{x} do vetor de observações. Esta estrutura permite calcular a entropia associada a cada possibilidade de sinal extraído, que é fruto da combinação linear das componentes de \mathbf{x} . Após o cálculo das entropias de todas as possibilidades de combinação, o algoritmo constrói uma matriz de separação

\mathbf{W} , invertível, onde cada uma de suas linhas representa o vetor de extração de uma componente distinta. Tais vetores são escolhidos da seguinte forma: primeiro escolhe-se a combinação de menor entropia associada e a colocamos como vetor-linha da matriz, em seguida escolhe-se a próxima linha tal que não seja combinação linear daquelas já presentes na matriz de separação. Este processo ocorre até que a matriz tenha *rank* completo, ou seja, todas as linhas escolhidas, por minimização da entropia associada a elas, sejam linearmente independentes entre si. Vale ressaltar que o algoritmo recorre a transformadas rápidas de Fourier para, com base no referido tensor de probabilidades, calcular rapidamente a entropia de um possível sinal a extrair.

O método GLICA foi recentemente proposto [6] e parte do mesmo princípio de operação do AMERICA, incluindo seu critério de otimização. Porém, o método recebe a matriz de observações \mathbf{X} no lugar do tensor de probabilidades e não recorre a estratégias de cálculo rápido de entropia. Seu funcionamento se dá pelo ordenamento das entropias associadas às combinações lineares das componentes de \mathbf{x} e, ao varrer um vetor que guarda os índices dessas entropias, escolhe-se aqueles vetores de extração que não sejam combinações lineares dos anteriores, para montar a matriz de separação.

O algoritmo SA4ICA [5] utiliza a meta-heurística *Simulated Annealing* [11] para minimizar a entropia dos sinais extraídos, assim como as técnicas anteriores, no entanto, o método testa somente as combinações de duas em duas componentes do vetor de observações. O algoritmo começa com uma candidata a matriz de separação e, com o decorrer das iterações, ela se modifica por meio de operações lineares tais que levem a uma melhoria na função custo. O SA4ICA conta com alguns hiper-parâmetros que devem ser definidos: a temperatura inicial T_0 ; um fator de decaimento β , que define como a temperatura inicial vai decair, em alusão ao processo de arrefecimento que inspira a meta-heurística; além disso, testa-se a solução a cada nova temperatura num total de k vezes.

Por fim, o algoritmo QICA [7] realiza ICA não-linear em sinais de alfabeto de tamanho Q . Para isto, o algoritmo busca uma permutação das observações a fim de minimizar a informação mútua entre as componentes deste novo mapeamento. Tenta-se encontrar esta permutação pela resolução de problemas lineares menores, por esta razão o algoritmo também se denomina *Piecewise Linear Relaxation Algorithm* [7]. Este algoritmo conta com duas versões, uma que explora de maneira exaustiva o espaço de permutações e outra, que é considerada neste trabalho, que emprega a técnica de gradiente descendente no processo de busca, visando um custo computacional factível. Neste sentido, os hiper-parâmetros da técnica são: as quantidades mínima e máxima de problemas lineares a serem resolvidos, min_k e max_k , respectivamente; além da quantidade de iterações I da técnica de gradiente.

IV. ANÁLISE EXPERIMENTAL

Dois experimentos são conduzidos neste trabalho. O primeiro envolve a aplicação de algoritmos de ICA na solução de diferentes instâncias do problema de BSS, isto é, para 2 fontes ou mais e corpos primos de diferentes ordens. Dado que,

neste contexto, os dados observados são fruto de um processo de mistura linear, consideramos somente os três primeiros algoritmos de ICA descritos na Seção III: os métodos GLICA e SA4ICA, estes dois recentemente propostos, e o algoritmo-referência de desempenho na tarefa, AMERICA.

O segundo experimento envolve a tarefa de decomposição de um vetor \mathbf{x} em componentes maximamente independentes entre si, com a importante ressalva de que \mathbf{x} não é derivado de nenhum modelo generativo linear, especificamente, não é definido pela combinação linear de fontes independentes. Neste caso, além dos algoritmos já citados, adiciona-se ao comparativo o algoritmo QICA [7], de ICA não-linear.

Em termos de hiper-parâmetros dos algoritmos, eles foram fixados em: $\min_k = 2$, $\max_k = 10$ e $I = 1000$ para o algoritmo QICA, e $T_0 = 1$, $\beta = 0.995$, $k = 5$ para o algoritmo SA4ICA. Os ensaios foram executados em um computador com processador Intel(R) Core(TM) i7-4810MQ CPU @ 2.80GHz e 8GB de memória instalada.

A. Experimento de BSS

Como métrica de desempenho, neste experimento conta-se o percentual médio de vezes em que o algoritmo resolve completamente o problema de separação, isto é, todas as fontes são recuperadas. Também medimos o tempo médio de execução dos algoritmos. As distribuições das fontes são geradas aleatoriamente e independentes tais que suas divergências de Kullback-Leibler [9] excedam 0.2 em relação à distribuição uniforme, a fim de garantir a característica de não-uniformidade. Cada cenário de simulação é composto por uma possibilidade dos parâmetros $P \in \{2, 3, 5, 7\}$; $K = 2, 3, \dots, 7$; $T \in \{2^8, 2^9, 2^{10}, 2^{11}, 2^{12}\}$; que são repetidamente executados em 40 rodadas de forma independente. Além disso, as matrizes de mistura são geradas aleatoriamente com o esquema de [12], de forma a garantir invertibilidade.

Como há 330 formulações possíveis de cenários, não podemos expor todas as visualizações. Portanto, apresentamos aqui os gráficos e as tendências mais interessantes, enquanto os demais estão disponibilizados em [13].

Os algoritmos convergem, ao adotarmos valores cada vez maiores de T , para 100% de separação. Contudo, para $P = 5$ e $K = 7$, o algoritmo SA4ICA sofre uma queda brusca e tem seu desempenho inferior a 50%, como mostra a Figura 1. Isto possivelmente se deve à dificuldade de varrer um espaço de busca cada vez maior, para a mesma configuração de iterações. Vale notar que esta queda é também acompanhada por um valor de informação mútua superior em relação aos demais algoritmos, ilustrado na Figura 2. O tempo total destas simulações foi em torno de 43 horas.

Dada a grande similaridade entre as técnicas quanto à capacidade de separação, parte-se para uma análise comparativa em termos de tempo de execução médio¹. Primeiramente, podemos observar pela Figura 3 que, em relação à quantidade de amostras, o algoritmo SA4ICA apresenta um comportamento constante, mas no entanto mais custoso que os demais, enquanto o método GLICA, ainda que muito próximo do AMERICA, tem um tempo de execução que indica um

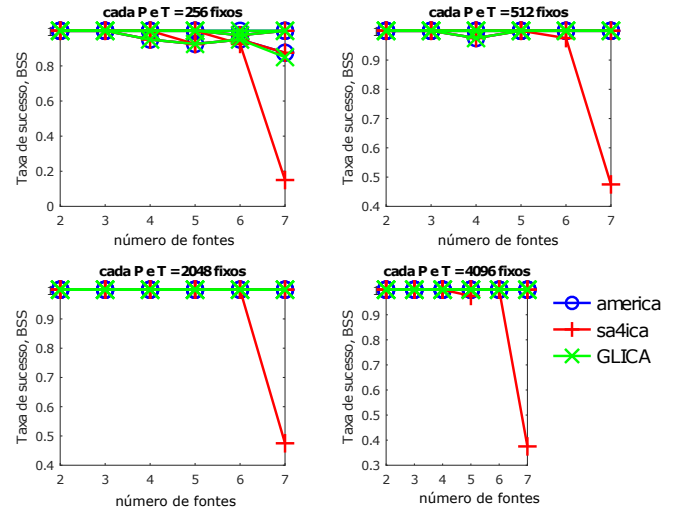


Fig. 1. Taxa média de separação total (sucesso) dos algoritmos. Linhas representando diferentes valores de P estão sobrepostas, sob os mesmos algoritmos em cores iguais. Com exceção do caso anômalo do SA4ICA, os algoritmos convergem para 100% de separação ao aumentarmos T .

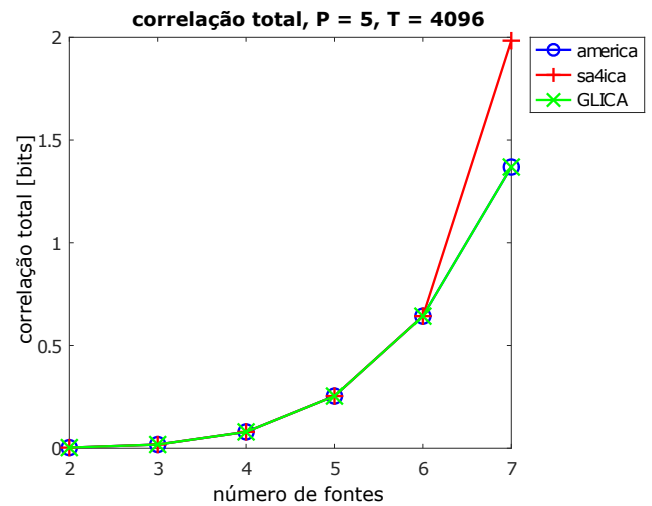


Fig. 2. Correlação Total média ao final de cada técnica, para $P = 5$, $T = 4096$, em função de K .

comportamento proporcional a $O(T)$, tendo sido mais lento que o AMERICA. Este aspecto foi observado em diversos cenários, mas pode ser visto de forma mais clara no caso em que $P = 5$ e $K = 7$. Este comportamento observado no gráfico confirma o fato dos métodos AMERICA e SA4ICA não dependerem da estimação frequencial das probabilidades a cada iteração, pois ambos se baseiam na estimativa da entropia com base na técnica de tensor de probabilidade e transformada rápida de Fourier, ao contrário da implementação do GLICA.

Com respeito à quantidade de fontes, podemos observar que todos os métodos, em maior ou menor grau, parecem seguir uma tendência de crescimento exponencial, tal como ilustra a Figura 4. Enquanto o algoritmo GLICA demonstra um crescimento de menor inclinação, os algoritmos SA4ICA e AMERICA apresentam curvas mais acentuadas, com os mai-

¹Os gráficos estão em escala logarítmica, para melhor visualização.

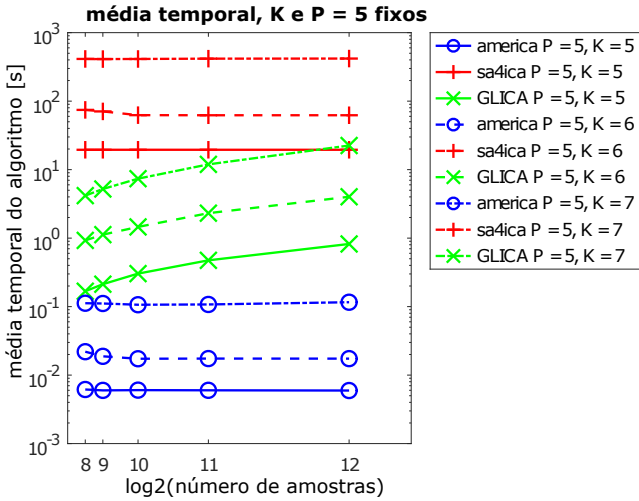


Fig. 3. Variando a quantidade de amostras, tempo médio de execução [segundos] dos algoritmos para P e K fixos.

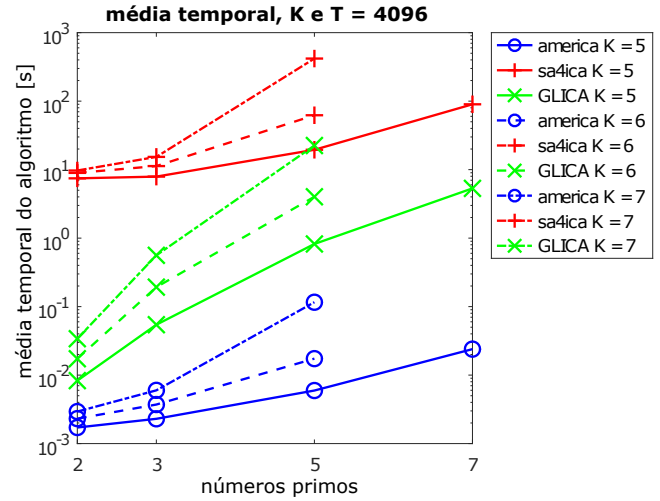


Fig. 5. Variando os primos, tempo médio de execução [segundos] dos algoritmos para K e T fixos.

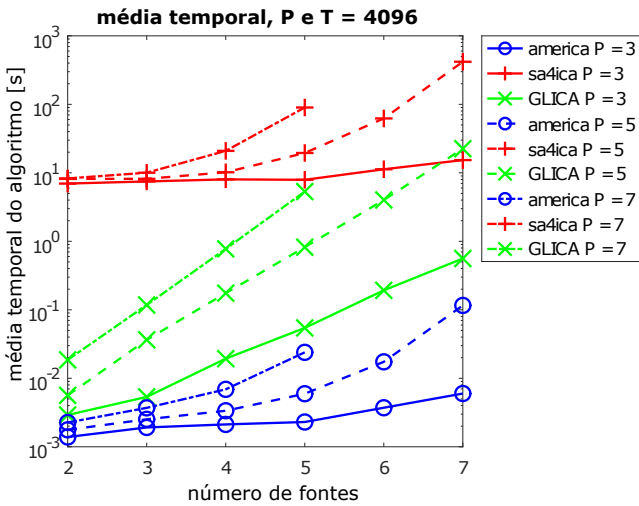


Fig. 4. Variando a quantidade de fontes, tempo médio de execução [segundos] dos algoritmos para P e T fixos.

ores valores absolutos associados, mais uma vez, ao SA4ICA.

Um comportamento similar dos métodos também pode ser observado com relação à ordem do corpo finito, como ilustra a Figura 5. Vale observar que o cenário para $P = 7$ com $K = 6$ ou 7 não foi possível de ser executado por conta do tempo de simulação ter se tornado demasiadamente elevado.

B. Experimento de ICA

Nesta análise, mediremos a informação mútua entre as componentes do vetor de saída dos algoritmos de ICA, após processar como entrada amostras de um vetor aleatório geradas a partir de uma função massa de probabilidade conjunta dada pela distribuição Zipf

$$P(X = \mathbf{x}_i | s, P^K) = \frac{(i+1)^{-s}}{\sum_{l=1}^{P^K} l^{-s}}, \quad (6)$$

com parâmetro $s = 1.05$ [14], onde K agora representa a dimensão do vetor de símbolos em um alfabeto P -ário.

Observe, portanto, que estamos diante de uma tarefa em que \mathbf{x} tem componentes dependentes entre si e não necessariamente geradas por meio de um processo linear de mistura. Os parâmetros de simulação agora são $P = 2, 3$; $K = 2, 3, 4, 5$; $T \in \{2^{13}, 2^{14}\}$. Este experimento é realizado com mais amostras por conta da alta demanda de amostras para boa estimação do algoritmo QICA.

O tempo total das simulações foi de aproximadamente 10h. Espera-se que os melhores algoritmos de ICA levem a menores valores de informação mútua. Nos resultados levantados, percebeu-se que o algoritmo QICA só obteve esta métrica inferior ou igual aos demais algoritmos no cenário de $P = 2$ com dimensão até 3. A partir de dimensões maiores, houve um crescimento proporcionalmente maior às outras técnicas, o que pode ser observado na Figura 6.

Quando adotamos $P = 3$, a informação mútua resultante nos dados gerados pelo algoritmo QICA foi maior ou igual às

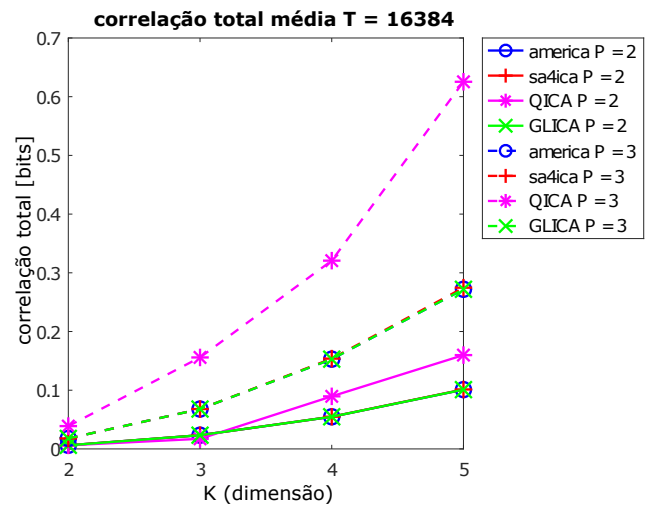


Fig. 6. Variando o número de fontes, informação mútua média [bits] dos algoritmos para T e P fixos

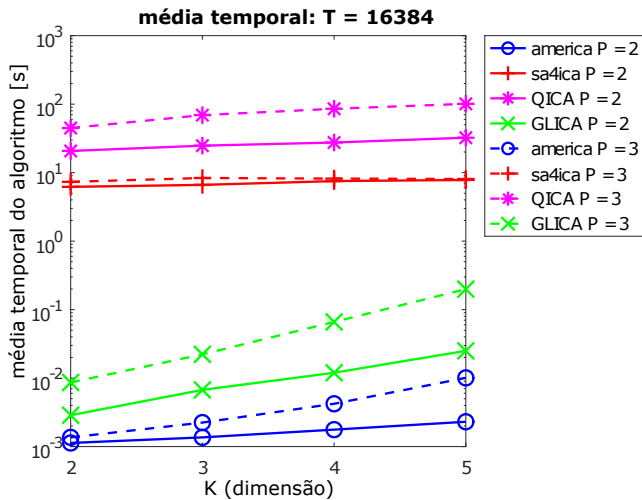


Fig. 7. Variando a dimensão, tempo médio de execução dos algoritmos [segundos] para P e T fixos.

dos outros métodos, como ilustrado na Figura 6. Além disso, os outros algoritmos praticamente coincidem nas medidas, enquanto o método QICA se distancia dos demais, independentemente de $P = 2$ ou $P = 3$.

O tempo médio dos algoritmos, ao observarmos a variação da dimensão do problema, é dado pela Figura 7, onde não foi possível adotar valores maiores de dimensão porque as referidas técnicas passavam a apresentar tempos de execução inviáveis (cerca de 15 a 20 minutos para executar apenas uma rodada do experimento). O algoritmo QICA mostrou-se o mais lento de todos, para os cenários avaliados. Para $P = 3$, como já observado no experimento de BSS, o SA4ICA possui tempo médio praticamente constante com a dimensão do problema.

V. CONCLUSÃO

Este trabalho iniciou uma investigação experimental dos algoritmos mais recentes de ICA em corpos finitos dentro da clássica aplicação de BSS, seguido da sua comparação com a técnica de ICA generalizado para sinais discretos, que explora um modelo não-linear de mapeamento. Além da capacidade de separação de fontes no caso de BSS, investigou-se a capacidade de minimização da medida de informação mútua nos dados produzidos pelas diferentes técnicas.

Quanto ao experimento de BSS, os algoritmos, diante dos cenários apresentados, executam efetivamente a sua tarefa quando crescem a quantidade de amostras, demonstrando convergência. A única exceção foi algoritmo SA4ICA apresentando abrupta queda quando $P = 5$ e $K = 7$, muito possivelmente por não conseguir varrer o espaço de estados de maneira efetiva. O GLICA e AMERICA são algoritmos mais rápidos e o SA4ICA como o algoritmo mais lento destes três.

O algoritmo QICA, ainda que execute ICA não-linear, somente obtém o melhor desempenho, quanto à minimização da informação mútua, em alguns poucos cenários. Diante da maioria dos casos, o QICA apresentou desempenho inferior e com tendência de piora mais acentuada para instâncias maiores do problema. Além disso, o algoritmo obteve os maiores

tempos médios de execução, o que limita seu uso prático. É importante ressaltar, todavia, que os ensaios deste trabalho trabalham com uma quantidade de amostras que se aproxima de cenários mais realistas, em termos práticos, mas que é bem inferior à quantidade dos experimentos realizados em [7].

Percebe-se com este trabalho que, embora hipoteticamente mais limitados em termos de modelo de mapeamento, os algoritmos de ICA em corpos finitos podem levar a soluções de boa qualidade, mesmo em comparação a técnicas de maior flexibilidade na modelagem, como os algoritmos generalizados em alfabetos finitos.

Em trabalhos futuros é desejável aumentar a quantidade de rodadas, para fins de significância estatística, bem como testar, para o caso de ICA pura, alfabetos maiores e outros tamanhos para a dimensão do problema, K . Ademais, distribuições conjuntas além da Zipf forneceriam um panorama mais acurado do desempenho dos algoritmos não-lineares em relação aos lineares. Por fim, em vista do extenso tempo total de simulações, é interessante buscar uma maneira de implementar os próprios algoritmos e os experimentos em caráter multi-processos, a fim de viabilizar a aplicação e experimentação das técnicas em tempo factível para instâncias maiores, ainda não analisadas.

AGRADECIMENTOS

Este trabalho foi parcialmente financiado pelo CNPq (processo 433887/2018-4).

REFERÊNCIAS

- [1] P. Comon, "Independent component analysis, a new concept?" *Signal processing*, vol. 36, no. 3, pp. 287–314, 1994.
- [2] J. M. T. Romano, R. Attux, C. C. Cavalcante, and R. Suyama, *Unsupervised signal processing: channel equalization and source separation*. CRC Press, 2016.
- [3] A. Hyvärinen, "Independent component analysis: recent advances," *Phil. Trans. R. Soc. A*, vol. 371, no. 1984, p. 20110534, 2013.
- [4] A. Yeredor, "Independent component analysis over galois fields of prime order," *IEEE Transactions on Information Theory*, vol. 57, no. 8, pp. 5342–5359, 2011.
- [5] D. G. Silva and R. Attux, "Simulated Annealing for Independent Component Analysis Over Galois Fields," *IEEE Signal Processing Letters*, vol. 25, no. 4, pp. 516–520, 2018.
- [6] A. Painsky, S. Rosset, and M. Feder, "Linear independent component analysis over finite fields: Algorithms and bounds," *IEEE Transactions on Signal Processing*, vol. 66, no. 22, pp. 5875–5886, 2018.
- [7] —, "Generalized independent component analysis over finite alphabets," *IEEE Transactions on Information Theory*, vol. 62, no. 2, pp. 1038–1053, 2015.
- [8] S. Watanabe, "Information theoretical analysis of multivariate correlation," *IBM Journal of research and development*, vol. 4, no. 1, pp. 66–82, 1960.
- [9] T. M. Cover and J. A. Thomas, *Elements of information theory*. John Wiley & Sons, 2012.
- [10] H. W. Gutch, P. Gruber, A. Yeredor, and F. J. Theis, "ICA over finite fields—Separability and algorithms," *Signal Processing*, vol. 92, no. 8, pp. 1796–1808, 2012.
- [11] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, no. 4598, pp. 671–680, 1983.
- [12] A. Yeredor, "Matlab® Code for $\mathbb{GF}(P)$, AMERICA and MEXICO." [Online]. Available: <http://www.eng.tau.ac.il/arie/ICA4GFP.rar>
- [13] M. Marcuzzo, "Comparação de algoritmos de ICA para o caso de BSS e ICA pura, 2019." [Online]. Available: https://github.com/MateusMarcuzzo/sbrt2019_ICA_comparative
- [14] A. Painsky, "Matlab® implementation of A. Painsky, S. Rosset and M. Feder, "generalized independent component analysis over finite alphabets", Transactions on Information Theory, 2015." [Online]. Available: <https://sites.google.com/site/amichaipainsky/software>